PROTOCOL

RadEro package
Version: 1.0.8

25/04/2025
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

Fallout radionuclides (FRNs), particularly ^{137}Cs, serve as reliable tracers of soil movement by sheet and rill erosion allowing to estimate an average rate value between the year 1963 and the sampling time. Numerous models have emerged to estimate both erosion or deposition rates using FRNs, though a versatile, physically-based ^{137}Cs model remains absent. RadEro, is a straightforward model to estimate soil redistribution 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.

Contents

PROTOCOL *updated in 25/04/2025*

1. PREPARATION STEPS

1.1. 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 four example files in the specified directory:

2 files with data of a **reference profile (**see Section 1.4. of this protocol):

- "input-data_example_reference.csv"
- "input-config_example_reference.js"

2 files with data of a **site profiles** where estimate redistribution rates:

- "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 reliable redistribution rate estimations, it is recommended to increase the "k-samples" and "e-samples" values to higher numbers. In the Section 2. “EXAMPLE OF USE” of this protocol we use "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 ^{137}Cs inventory data from soil profiles, using the configuration defined in the working directory.

The function requires the following:

- **data:** A data frame containing the ^{137}Cs inventory data from soil profiles. This file should be located in the working directory. **We recommend creating one for the reference profiles and another for the sampling sites.**
- **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.

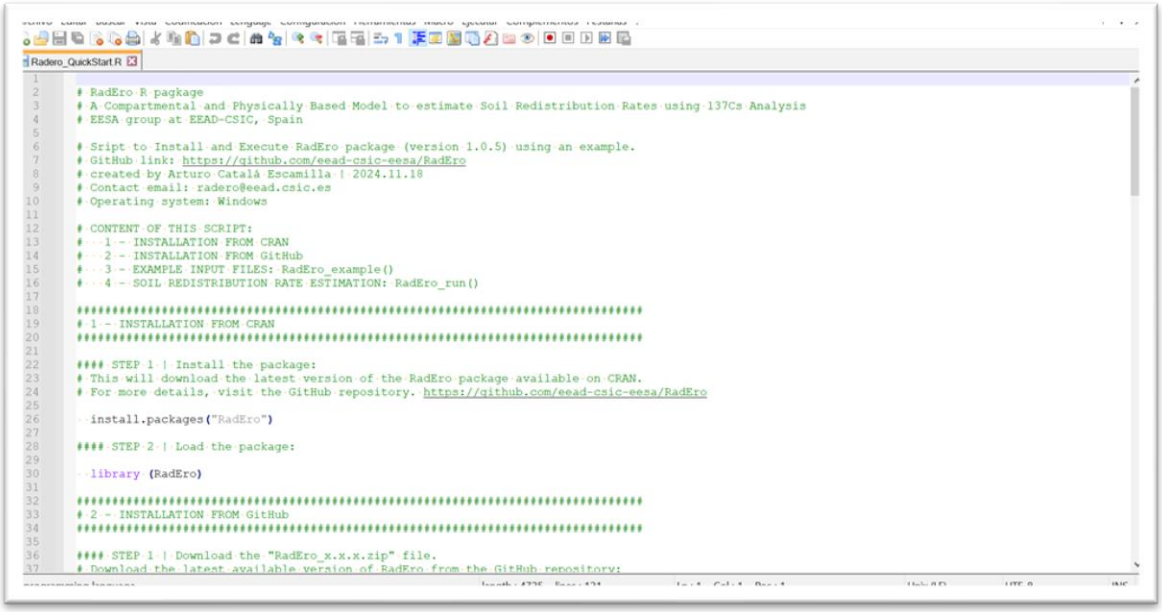


Fig 1. Screenshot of the *Radero_QuickStart.R* content.

1.2. 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

$$\text{Veff (cm}^3\text{/cm}^3\text{)} = \text{Volume of Fines (cm}^3\text{)} / \text{Total Volume (cm}^3\text{)}$$

$$\text{Volume of Fines (cm}^3\text{)} = \text{Total Volume (cm}^3\text{)} - \text{Volume of Stones (cm}^3\text{)}$$

$$\text{Total Volume (cm}^3\text{)} = \text{Sample Area (cm}^2\text{)} \times \text{Interval Depth (cm)}$$

$$\text{Volume of Stones* (cm}^3\text{)} = \text{Weight of Stones (g)} / \text{Density of Stones (2.65 g/cm}^3\text{)}$$

**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 k (cm²/month) quantifies the rate of vertical mixing of ^{137}Cs within the soil profile. The reference inventory is experimentally determined at a point assumed to have negligible erosion or deposition, and this value establishes the effective ^{137}Cs fallout deposition at the problem point (reference sectioned profiles). To estimate this value you need input data from a sectioned references profiles. This optimal k value (k_{ref}) estimated with the sectioned reference profile is then applied to estimate the corresponding soil redistribution rate (Mg ha^{-1} yr^{-1}) in eroded and depositional profiles.

- **e**: Redistribution rate. 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_reference.csv* template contains one example:

- **RefProfile**: Sectioned undisturbed soil profile with no erosion or deposition.

The *input-data_example.csv* template contains six example data types:

- **Profile1**: Sectioned unploughed soil profile. Erosive site.
- **Profile2**: Bulk unploughed soil profile. Erosive site.
- **Profile3**: Sectioned unploughed soil profile. Deposition site.
- **Profile4**: Bulk unploughed soil profile. Deposition site.
- **Profile 5:** Sectioned ploughed soil profile. Erosive site.
- **Profile 6:** Bulk ploughed soil profile. Erosive site.

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.

1.3. 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 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.*
}
{

1.4. Optimization parameters: k and e

Soil redistribution rates are determined through an inverse modelling approach; specifically an optimisation process that compares simulated and measured ^{137}Cs inventories and vertical profiles. Optimal model performance is achieved when the simulated ^{137}Cs inventory and vertical profile closely match the measured data. Each simulation within the optimisation process is defined by three model parameters: the migration rate e, the diffusion coefficient k, and the ^{137}Cs reference inventory (Fig.1). The migration rate e quantifies the net vertical soil movement per month (positive for gain, negative for loss). The diffusion coefficient k quantifies the rate of vertical mixing of ^{137}Cs within the soil profile. The reference inventory is experimentally determined at a point assumed to have negligible erosion or deposition, and this value establishes the effective ^{137}Cs fallout deposition at the problem point. The e parameter is not solely optimised based on the reference ^{137}Cs profile. Instead, e and k are optimised simultaneously to best fit the measured ^{137}Cs profile and total inventory at the sampling point itself.

When only the total ^{137}Cs inventory is available (i.e. bulk profiles, no sectioned soil profiles), a simplified analysis is performed. To avoid equifinality, where multiple migration rates can produce similar total inventories, the diffusion coefficient k is fixed to that of the reference site for the study area, and optimisation focuses solely on the migration rate e. In ploughed soils, tillage homogenises the ^{137}Cs inventory profile. Within the ploughed layer, there is no gradient in the ^{137}Cs inventory, so the diffusion coefficient (k) has a negligible effect on the model calculations. However, the deepest sections of the sampled ploughed soil profiles may contain soil layers that have not been affected by tillage, where the k value does have an impact.

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.2). The model then selects the value closest to the measured inventory, enabling the calculation of the soil migration rate (Mg ha^{-1} yr^{-1}).

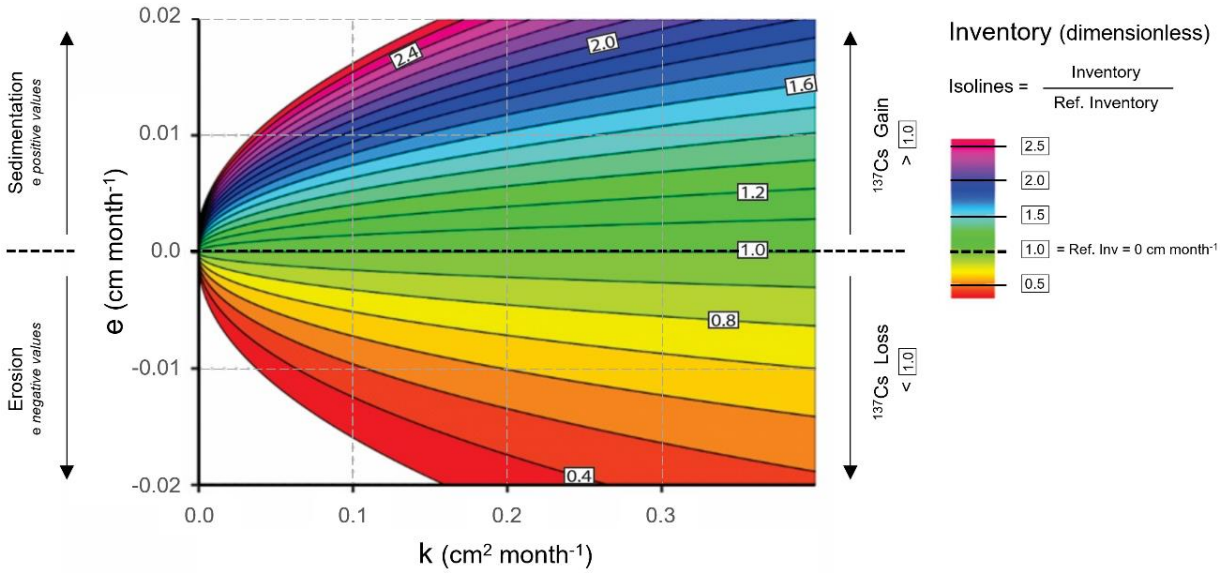


Fig.2: Graphical representation of a sampling domain defined by the range of values for diffusion coefficient k (cm^2 $month^{-1}$), and migration rate e (cm $month^{-1}$) for a reference undisturbed sectioned profile. The model simulates a finite number of potential inventory values within this domain based on the specified number of iterations. Isolines and color-coded areas represent dimensionless inventory values (ratios between the inventory at the points and the reference inventory) across the workspace. The isoline with a value of 1.0 is a horizontal line that represents the reference inventory, i.e. the baseline where no erosion or deposition occurs.

Figure 2 illustrates the model's iterative process using user-specified ranges for diffusion coefficient (k) and redistribution rates (e). Within this sampling domain, the simulation iterates to find the combination of k and e values that best matches the measured ^{137}Cs inventory and the vertical distribution of a stable reference profile. This optimal k value is then applied to estimate the corresponding soil redistribution rate m (Mg ha^{-1} yr^{-1}) in eroded and depositional bulk profiles.

As shown in Figure 2, inventory isolines for negative e values (erosion) are more widely spaced than those for positive values (deposition). This suggests greater precision in determining erosion rates. This discrepancy is attributed to the assumption that deposited soil originates from nearby soil.

2. EXAMPLE OF USE

2.1. Step 1: Create a New Working Directory

For each project, we recommend creating a separate folder. Each project requires at least an input data file and an input configuration file located in the same directory. Define your working directory for easier file manipulation:

direct <- R"(C:\your\own\working\directory)"

2.2. Step 2: Create and complete the input files

RadEro_example(target_dir = direct)

Four files have been created:

Two input files that define a reference site:

- input-data_example_reference.csv
- input-config_example_reference.js

Two input files that define four study sites:

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

In any ^{137}Cs redistribution rate estimation, it is necessary to have a reference site in order to determine the diffusion coefficient (k) that will be used for the study sites.

The RadEro model is capable of estimating the diffusion coefficient (k) directly from the reference site. For this reason, the "k" field in input-data_example_reference.csv is left empty, whereas in input-data_example.csv it contains a specified value.

Estimating the diffusion coefficient will generally take more time than estimating the soil redistribution rates for the study sites. However, both processes are triggered using the same function, RadEro_run().

2.3. Step 3: Estimate diffusion coefficient for the reference site

To execute RadEro_run(), you need both the input data file (where the "k" value is not defined) and the configuration file located in the same directory.

Using the provided example files and the previously defined directory:

RadEro_run("input-data_example_reference.csv", "input-config_example_reference.js", output_dir = direct)

Depending on the resolution parameters defined in the configuration file (specifically the "k-samples" and "e-samples" values), this process may take several minutes to complete.

Note that input files used int Section 2.4 already include the estimated "k_{ref}" value.

A folder containing the generated plots (Figure 5) and summary results data (saved in "results.txt") will be created within the "direct" directory. See Section 3 RESULTS FILES to better understand all the information calculated and results obtained.

The estimated "k" value can be retrieved from the "results.txt" file.

To visualise the results file in R:

results <- read.table(R"(results\results.txt)", header = FALSE, sep = ",")

View(results)

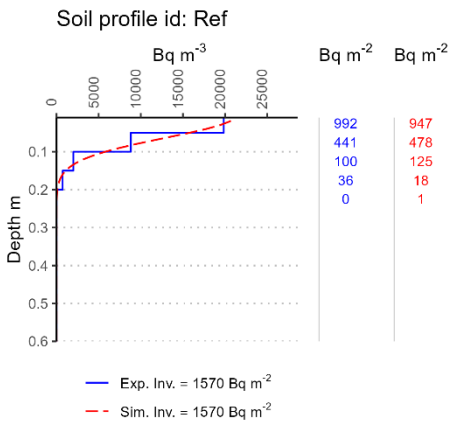


Fig.3 Reference profile plot from the RadEro_example function with model resolution (editable on the config file) "k-samples" = 4**00** and "e-samples"= 4**00**.

Data from the Results.txt file:

- *id = Ref*
- *e = 0 cm/month*
- *m = 0 Mg/ha/year*
- *k = 0.026 cm2/month*
- *kv = 0.0182 cm2/month*
- *Experimental-inventory = 1570 Bq/m2*
- *Simulated-inventory = 1570 Bq/m2*

2.4. Step 4: Estimate soil redistribution rates for the study sites

To execute **RadEro_run()** for a given study site, you must ensure that both the input data file and the configuration file are located in the same directory.
Using the example files and the previously defined directory:

RadEro_run("prueba.csv", "MS_config.js", output_dir = direct)

RadEro_run("input-data_example.csv", "input-config_example.js", output_dir = direct)

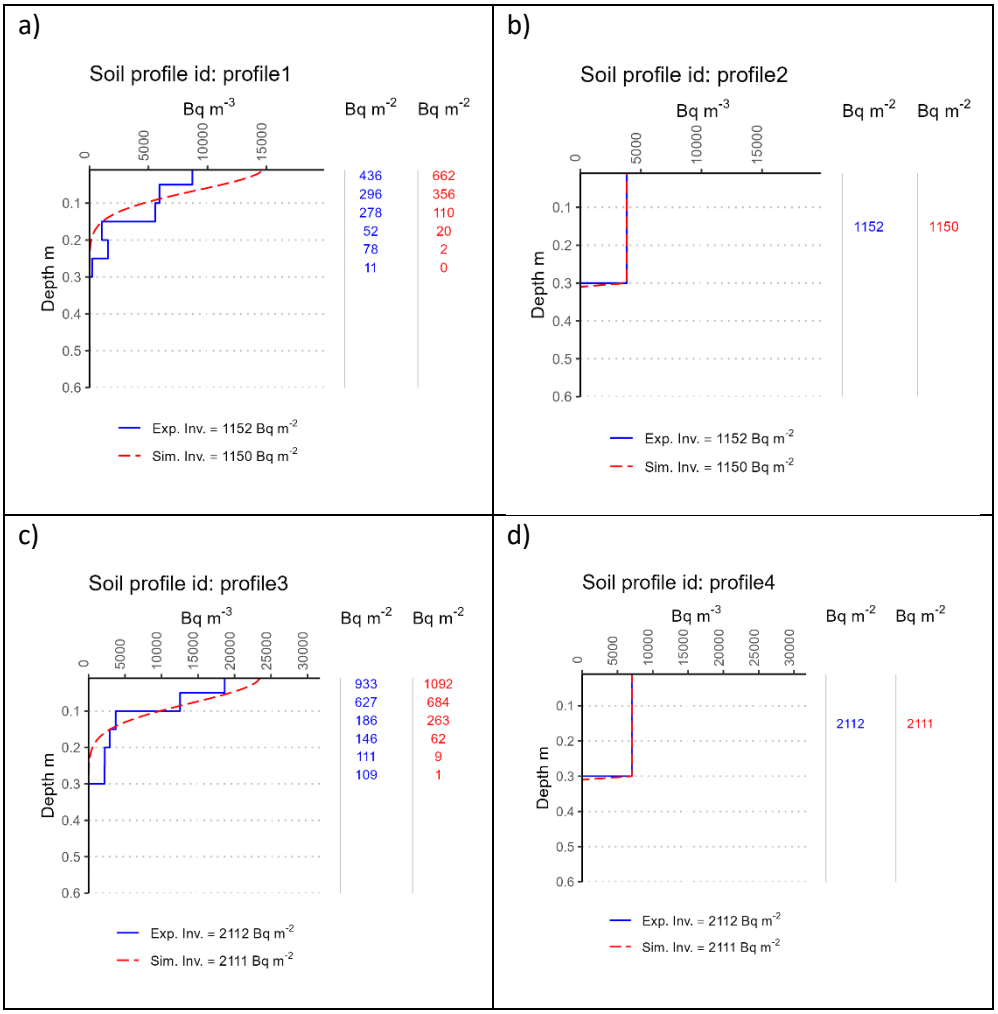
A folder containing the generated plots (Figure 6) and the estimated soil redistribution rates ("results.txt") will be created within the "direct" directory.

To visualise the results file in R:

results <- read.table(R"(results\results.txt)", header = FALSE, sep = ",")

View(results)

a. **Profile1**: Sectioned unploughed soil profile. Erosive site.
b. **Profile2**: Bulk unploughed soil profile. Erosive site.
c. **Profile3**: Sectioned unploughed soil profile. Deposition site.
d. **Profile4**: Bulk unploughed soil profile. Deposition site.
e. **Profile 5:** Sectioned ploughed soil profile. Erosive site.
f. **Profile 6:** Bulk ploughed soil profile. Erosive site.



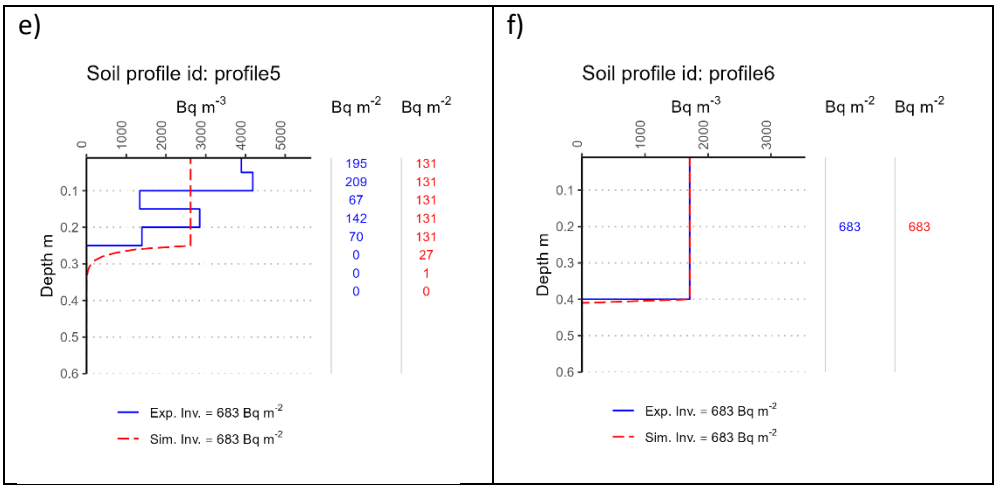


Fig.4 Example plots from RadEro_example function with model resolution (editable on the config file) "k-samples" = **1000** and "e-samples"= **1000**.

Data of the Results.txt file obtained from the RadEro_example function with model resolution (editable on the config file) "k-samples" = 1000 and "e-samples"= 1000:

id	e (cm/month)	m (Mg/ha/year)	k (cm^2/month)	kv (cm^2/month)	Experimental inventory (Bq/m^2)	Simulated inventory (Bq/m^2)
profile1	-0.00288	-3.3867	0.03	0.0189	1152	1150
profile2	-0.00288	-5.08	0.03	0.0189	1152	1150
profile3	0.00296	4.7244	0.03	0.0189	2112	2111
profile4	0.00296	3.3746	0.03	0.0189	2112	2111
profile5	-0.03496	-65.0255	0.03	0.0165	683	683
profile6	-0.03496	-60.4108	0.03	0.0165	683	683

3. 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 (Figure 7) : 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 horizontal axis represents the ^{137}Cs inventory in Bq m^{-3}, and the vertical axis shows the soil depth in meters. The experimental (blue) and simulated (red) inventories for each interval depth are listed at the right-hand side

of the plot in Bq m^{-2} and the total inventory for each soil profile are shown below each plot in Bq m^{-2}.

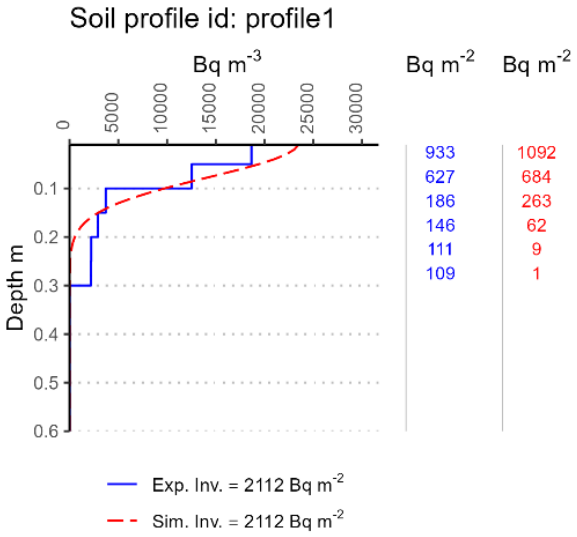


Fig.5 Example plot

- ID_tempfile : Experimental inventory (Bq kg^{-1}) per cell unit in the profile depth defined.
- results.txt (figure 6):
    - 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.
-

4. RadEro Model: Versión Updates

v1.0.0 - 2023-11-21

Creation of the Radimod package.
Creation of the "radi_start.R" function: initial base and setup.
Creation of the "build_config.R" and "resample_data.R" functions: internal reading functions not visible to users.
In the data/ folder, both data and config files are grouped; the goal is to eliminate the need for a separate config file — all will be managed through variables within "radi_start.R".
All input values are given in SI units.

v1.0.1 - 2023-12-19

Borja modifies the C++ code to allow opening and compiling it from within R. Package structure is now based on Rcpp.

v1.0.2 - 2024-01-05

Functional R version in package format. It is called from the directory containing the data and config files.
Currently tested only for a single survey dataset. Additional datasets need to be added to observe behaviour.
Conducted tests with Radiero data using different values for k-samples and e-samples to observe result variability at different resolutions (.../Cs_model/resultados/).

v1.0.2 - 2024-01-10

Activation of "Plot.R".
Analysis of transects and reference points using this version.

v1.0.3 - 2024-07-24

Package renamed from "Radimod" to "RadEro".
Function "radi_start" renamed to "RadEro_run".
Column names of the profile input data file changed from:

ID;PROFA;PROFB;CS;REFINV;VOLEF;DENS;K;E;MIX_ZI;MIX_ZF;MIX_YEARI;MIX_YEARF; MIX_FREC;PROF
to:
id;depth_i;depth_f;Cs137_invt;Ref_Cs137_invt;effVol;density;k;e;mix_depth_i;mix_depth_f;mix_year_i;mix_year_f;mix_frequency;lower_boundary

Option to scale all x-axes in plots using the AxisMaxValue parameter. Output messages in the console have been revised to be in perfect English.
New function "RadEro_example.R" created to generate example files.

v1.0.4 - 2024-07-24 — CRAN VERSION

First version accepted by CRAN.
DESCRIPTION file updated: definitions, references, emails... Dave Gamble added as author for
the cJSON.h code.
C code adjustments: removed fprintf and similar text output commands.
Improved redundant C code variables — all changes were CRAN-check recommendations.
C Code Changes:

- **model.c:**
  - Added C packages:
    - #include <unistd.h>
    - #include <limits.h>
  - Added file path variables:
    - #define CONFIGJS "_config.js"
    - #define TXTEXP "_exp.txt"
    - #define TXTNUM1 "_num.txt"
    - #define TXTNUM2 "_num2.txt"
    - #define TXTNUM3 "_num3.txt"
- **cJSON.c:**
  - Added "void" in function definitions.
  - Changed sprintf to snprintf for safer string handling.
  - Added (void)bytesRead; to avoid unused variable warnings.
  - Reorganised the code for clarity in functions like cJSON_DetachItemFromArray.
- **cJSON.h:**
  - Added "void" in function definitions.
- **data.io.h:**
  - Changed sprintf to snprintf, setting buffer size to 1024.
  - Added "void" to function declarations.
  - Added (void) to unused variables like x2, x3, trash, and j to avoid warnings.
  - Improved output of all numerical results generated by the model (changes initiated in v1.0.3).
- **defs.h:**
  - Added int cs_model(void);
  - Commented out fprintf, printf, etc.
  - Updated void write_activity to output all numerical results.
- **simulation.h and optimisation.h:**
  - Replaced sprintf with snprintf.
  - Assigned default value x1 = 0.0; to avoid errors if initial conditions are unmet.

(**Note:** Some changes were accidentally saved over the v1.0.3 code, but these only involve minor refactoring and do not affect functionality. To track changes more accurately, compare against
v1.0.2.)

R Code Changes:

All code commented, with variable and result explanations.

"RadEro_example.R" function provides example "input-", "config", and "data" files with very few iterations to meet CRAN testing time requirements.

Written examples are not *read* by CRAN to avoid confusion from temporary file generation.

Important: No function operates in the working directory; all must define a directory explicitly.

v1.0.5 - 2024-11-14 — CRAN UPDATE

Minor updates to v1.0.4 to meet CRAN requirements and maintain repository inclusion.

C Code: In rcpp_hello_world.cpp, added extern "C" int cs_model();.

Minor compilation issues corrected automatically by CRAN.

DESCRIPTION: Updated joint email to "radero@eead.csic.es".

v1.0.6 - 2025-03-10

Added two columns to graphical results showing estimated and measured inventory values.

v1.0.7 - 2025-03-28

Modified the plotting of BULK profiles: the simulated profile now adjusts to BULK geometry and thickness.

v1.0.8 - 2025-03-28

Adjusted default input parameters in example CONFIG files, extending the range for estimating redistribution rates (e).

Added new example files for estimating reference inventory: input-data_example_reference.csv and input-config_example_reference.js.

These new examples function similarly to the existing input-config_example.js and input-data_example.csv.