Package 'RLumCarlo'

February 16, 2017

Type Package

Title Monte-Carlo Methods for Simulating Luminescence Phenomena

Version 0.0.1

Date 2017-01-27

Author Johannes Friedrich [aut, trl, cre]

Maintainer Johannes Friedrich <johannes.friedrich@uni-bayreuth.de>

Description
A collection of functions to simulate luminescence signals with Monte-Carlo methods in the mineral feldspar based on published models.

Contact Package Developer Team <johannes.friedrich@uni-bayreuth.de>

License GPL-3

Depends R (>= 3.3.0), utils

URL https://CRAN.R-project.org/package=RLumModel

Collate 'calc_RLumCarlo.R' 'plot_RLumCarlo.R' 'RcppExports.R' 'RLumCarlo-package.R' 'run_MC_ISO.R' 'run_MC_CW_IRSL.R' 'run_MC_TL.R' 'run_MC_LM_OSL.R' 'utils.R'

Imports abind, doParallel, foreach, parallel, methods, magrittr, Rcpp

LinkingTo Rcpp, RcppProgress, RcppArmadillo

Suggests R.rsp

VignetteBuilder R.rsp

RoxygenNote 6.0.1

NeedsCompilation yes

R topics documented:

`RLumCarlo-package`

Package: & RLumCarlo
Type: & Package
Version: & 0.0.1
Date: & 2017-01-31
License: & GPL-3

Description

Package:	RLumCarlo
Type:	Package
Version:	0.0.1
Date:	2017-01-31
License:	GPL-3

Author(s)

Authors

`calc_RLumCarlo` *Plot results from Monte-Carlo simulations with RLumCarlo*

Description

Plot results from Monte-Carlo simulations with RLumCarlo

Usage

```
calc_RLumCarlo(results)
```

Arguments

`results` `array`:

Value

This function returns a `data.frame`

Function version

0.0.1 [2017-01-27]

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

`plot_RLumCarlo` *Plot results from Monte-Carlo simulations with RLumCarlo*

Description

Plot results from Monte-Carlo simulations with RLumCarlo

Usage

```
plot_RLumCarlo(results, times = NULL, norm = FALSE, legend = FALSE,
  add = FALSE, ...)
```

Arguments

`results`	`data.frame`
`times`	`vector` (with default):
`norm`	`character` (with default):
`legend`	`logical` (with default):
`add`	`logical` (with default):
`...`	further arguments

Value

This function returns a graphical output

Function version

0.0.1 [2017-01-27]

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

run_MC_CW_IRSL *Run Monte-Carlo simulation for CW-IRSL*

Description

Run Monte-Carlo simulation for CW-IRSL

Usage

```
run_MC_CW_IRSL(A, rho, times, clusters = 10, r = NULL, N_e = 200,
  method = "par", output = "signal", ...)
```

Arguments

A	numeric
rho	numeric
times	vector (with default)
clusters	numeric (with default):
r	numeric (with default)
N_e	numeric (with default):
method	character (with default):
output	character (with default):
...	further arguments

Value

This function returns a list.

Function version

0.0.1 [2017-01-31]

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

References

Pagonis 2017

Examples

```
## Not run:

##============================================================================##
## Example 1: Simulate CW-IRSL measurement
##============================================================================##

run_MC_CW_IRSL(A = 0.12, rho = 0.003, times = 0:1000) %>%
  calc_RLumCarlo() %>%
    plot_RLumCarlo(norm = T, legend = T)

## End(Not run)
```

`run_MC_ISO` *Run Monte-Carlo simulation for isothermal measurements*

Description

Run Monte-Carlo simulation for isothermal measurements

Usage

```
run_MC_ISO(A, rho, times, clusters = 10, r = NULL, N_e = 200,
  method = "par", output = "signal", ...)
```

Arguments

`A`	`numeric`
`rho`	`numeric`
`times`	`vector` (with default)
`clusters`	`numeric` (with default):
`r`	`numeric` (with default)
`N_e`	`numeric` (with default):
`method`	`character` (with default):
`output`	`character` (with default):
`...`	further arguments

Value

This function returns a list.

Function version

0.0.1 [2017-01-27]

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

References

Pagonis 2017

Examples

```
## Not run:
##============================================================================##
## Example 1: Simulate isothermal measurement
##============================================================================##

times <- seq(0, 500)
run_MC_ISO(A = 0.20,
           rho = 0.007,
           times = times) %>%
  calc_RLumCarlo() %>%
    plot_RLumCarlo(legend = T)

## End(Not run)
```

`run_MC_LM_OSL` *Run Monte-Carlo simulation for LM-OSL*

Description

Run Monte-Carlo simulation for LM-OSL

Usage

```
run_MC_LM_OSL(A, rho, times, clusters = 10, r = NULL, delta.r = 0.1,
  N_e = 200, method = "par", output = "signal", ...)
```

Arguments

`A`	`numeric`
`rho`	`numeric`
`times`	`vector` (with default)
`clusters`	`numeric` (with default):
`r`	`numeric` (with default):
`delta.r`	`numeric` (with default):
`N_e`	`numeric` (with default):
`method`	`character` (with default):
`output`	`character` (with default):
`...`	further arguments

Value

This function returns a list.

Function version

0.0.1 [2017-01-27]

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

References

Pagonis 2017

`run_MC_TL` *Run Monte-Carlo simulation for TL*

Description

Run Monte-Carlo simulation for TL

Usage

```
run_MC_TL(s, E, rho, r_c, times, clusters = 10, N_e = 200, delta.r = 0.1,
  method = "par", output = "signal", ...)
```

Arguments

`s`	`list`
`E`	`numeric`
`rho`	`numeric`
`r_c`	`numeric` (with default)
`times`	`vector` (with default)
`clusters`	`numeric` (with default):
`N_e`	`numeric` (with default):
`delta.r`	`numeric` (with default):
`method`	`character` (with default):
`output`	`character` (with default):
`...`	further arguments

Value

This function returns an `array` with dimension length(times) x length(r) x clusters

Function version

0.0.1 [2017-01-27]

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

References

Pagonis 2017

Examples

```
## Not run:
##============================================================================##
## Example 1: Simulate TL measurement
##============================================================================##

times <- seq(200, 500) # time = temperature

run_MC_TL(s = 3.5e12,
          E = 1.45,
          rho = 0.015,
          r_c = 0.85,
          times = times) %>%
  calc_RLumCarlo() %>%
    plot_RLumCarlo(legend = T)

## End(Not run)
```

Index