Package ‘RLumCarlo’

October 10, 2019

Type Package

Title Monte-Carlo Methods for Simulating Luminescence Phenomena

Version 0.1.0.9000-84

Date 2019-10-10

Author Johannes Friedrich [aut, trl] (<https://orcid.org/0000-0002-0805-9547>),
Sebastian Kreutzer [aut, trl, cre] (<https://orcid.org/0000-0002-0734-2199>),
Vasilis Pagonis [aut] (<https://orcid.org/0000-0002-4852-9312>),
Christoph Schmidt [aut] (<https://orcid.org/0000-0002-2309-3209>),
Ena Rajovic [ctb],
Alex Roy Duncan [ctb],
Christian Laag [ctb]

Maintainer Sebastian Kreutzer <sebastian.kreutzer@u-bordeaux-montaigne.fr>

Description A Collection of Functions to Simulate Luminescence Production in Minerals using Monte-Carlo methods.

Contact Package Developer Team <sebastian.kreutzer@u-bordeaux-montaigne.fr>

License GPL-3

BugReports https://github.com/R-Lum/RLumCarlo/issues

Depends R (>= 3.3.0),
utils,
magrittr

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

LinkingTo Rcpp (>= 1.0.2),
RcppArmadillo (>= 0.9.700.2.0)

Imports abind (>= 1.4-5),
doParallel (>= 1.0.15),
foreach (>= 1.4.7),
khroma (>= 1.2.0),
parallel,
methods,
Rcpp (>= 1.0.2)

Suggests R.rsp (>= 0.43.1),
testthat (>= 2.0.0)

Encoding UTF-8

VignetteBuilder R.rsp

RoxygenNote 6.1.1

R topics documented:

`plot_RLumCarlo` *Plot RLumCarlo Monte-Carlo Simulations Results*

Description

The function allows to visualise 'RLumCarlo' modelling results without extracting the values manually. Typically visualised values are the signal or the number of remaining electrons as averaged values with a polygon indicating modelling uncertainties.

Usage

```
plot_RLumCarlo(object, plot_uncertainty = "range", norm = FALSE,
  add = FALSE, ...)
```

Arguments

`object` list of class `RLumCarlo_Model_Output` (**required**): input object to be plotted, usually the required input object is generated by one for the functions preceeding with `run`. Alternatively a list of such objects can be provided.

`plot_uncertainty` logical (*with default*): sets the nature of the show uncertainty, allowed values are `range`, `sd` (standard deviation), `var` (variance) and `NULL` disables the uncertaintiy visualisation

`norm` logical (*with default*): normalise curve to the highest intensity

`add` logical (*with default*): allow overplotting of results by adding this curve to an existing plot

`...` further arguments that can be passed to control the plot output. Currently supported are: `xlab`, `xlim`, `ylim`, `main`, `lwd`, `type`, `pch`, `lty`,`col`, `grid`, `legend`. The arguments `lwd`, `type`, `pch`, `lty`, `col` can be provided as vector if `object` is a list

Details

The colour curves, the package khroma::khroma-package is used to provide colours that can be best distinguished, in particular by colour-blind users.

Value

This function returns a graphical output

Function version

0.1.0

How to cite

Friedrich, J., Kreutzer, S., 2019. plot_RLumCarlo(): Plot RLumCarlo Monte-Carlo Simulations Results. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany), Sebastian Kreutzer, IRAMAT-CRP2A, Université Bordeaux Montaigne (France)

`run_MC_CW_IRSL_LOC` *Run Monte-Carlo simulation for CW-IRSL for localised transition*

Description

Runs a Monte-Carlo (MC) simulation of constant wave infrared stimulated luminesence (CW-IRSL) using the generalized one trap (GOT) model.

Usage

```
run_MC_CW_IRSL_LOC(A, times, clusters = 10, n_filled = 100, r,
  method = "par", output = "signal", ...)
```

Arguments

`A`	numeric (**required**): The transition probability (cm^3/s).
`times`	numeric (*with default*): The number of MC runs.
`clusters`	numeric (*with default*): The number of clusters.
`n_filled`	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
`r`	numeric (*with default*): The retrapping ratio.
`method`	character (*with default*): sequential `'seq'` or parallel processing `'par'`
`output`	character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)
`...`	further arguments

Details

$$I_{LOC}(t) = -dn/dt = A * (n^2/(r + n))$$

Value

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

Function version

0.1.0

How to cite

Kreutzer, S., 2019. run_MC_CW_IRSL_LOC(): Run Monte-Carlo simulation for CW-IRSL for localised transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C., 2017. Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars. Journal of Luminescence 181, 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##===========================================================================##
## Example 1: Simulate CW-IRSL
##===========================================================================##
## Not run:
run_MC_CW_IRSL_LOC(
 A = 0.12,
 r = 1,
 times = 0:100) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

---

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

Description

Runs a Monte-Carlo (MC) simulation of constant wave infrared stimulated luminesence (CW-IRSL) using the model.

Usage

```
run_MC_CW_IRSL_TUN(A, rho, times, clusters = 10, r_c = 0,
  delta.r = 0.1, N_e = 200, method = "seq", output = "signal", ...)
```

Arguments

`A`	numeric (**required**): The transition probability (cm^3/s).
`rho`	numeric (**required**): The calculated dimesionless Charge density (normally written Rho').
`times`	numeric (*with default*): The number of MC runs.
`clusters`	numeric (*with default*): The number of clusters.
`r_c`	numeric (*with default*): The retrapping ratio.
`delta.r`	numeric (*with default*):
`N_e`	numeric (*with default*): The number of electrons
`method`	character (*with default*): sequential `'seq'` or parallel processing `'par'`
`output`	character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)
`...`	further arguments

Details

####equation here please####

Value

This function returns a list.

Function version

0.2.0

How to cite

Friedrich, J., Kreutzer, S., 2019. run_MC_CW_IRSL_TUN(): Run Monte-Carlo simulation for CW-IRSL. Function version 0.2.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany), Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) 'Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars', Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
## Not run:

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

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

## End(Not run)
```

run_MC_CW_OSL_DELOC *Run Monte-Carlo simulation for CW-OSL for delocalized transition*

Description

Runs a Monte-Carlo (MC) simulation of constant wave infrared stimulated luminesence (CW-OSL) using the one trap one recombination center (OTOR) model.

Usage

```
run_MC_CW_OSL_DELOC(A, times, clusters = 10, N_e = 200,
  n_filled = N_e, R, method = "par", output = "signal", ...)
```

Arguments

A	numeric (**required**): The transition probability (cm^3/s).
times	numeric (*with default*): The number of MC runs.
clusters	numeric (*with default*): The number of clusters.
N_e	integer (*with default*): The number of electrons.
n_filled	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
R	numeric (*with default*): The retrapping ratio.
method	character (*with default*): sequential 'seq' or parallel processing 'par'
output	character (*with default*): output is either the 'signal' (the default) or 'remaining_e' (the remaining charges, electrons, in the trap)
...	further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1-R)))$$

Value

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

Function version

0.1.0

How to cite

Kreutzer, S., 2019. run_MC_CW_OSL_DELOC(): Run Monte-Carlo simulation for CW-OSL for delocalized transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C., 2017. Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars, Journal of Luminescence 181, 114–120. doi: 10.1016/j.jlumin.2016.09.014

Examples

```
##============================================================================##
## Example 1: Simulate CW-OSL
##============================================================================##
## Not run:
run_MC_CW_OSL_DELOC(
 A = 0.12,
 R = 1,
 times = 0:100) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)

#' @examples
##============================================================================##
## Example 2: Simulate CW-OSL DELOC with several parameter changes
##============================================================================##
## Not run:

# define your parameters
A=c(0.1,0.3,0.5,1)
times=seq(0,60,1)
s=1e12
E=1
R<-c(1e-7,1e-6,0.01,0.1) # sequence of different R values
clusters=1000 # number of Monte Carlo simulations
N_e =c(200, 500, 700, 400) # number of free electrons
n_filled =c(200, 500, 100, 70) # number of filled traps
method="par"
output ="signal"
col=c(1,2,3,4) # ifferent colours for the individual curves
plot_uncertainty <- c(T,F,T,F)  # do you want to see the uncertainty?
add_TF <- c(F,rep(T, (length(R)-1)))
for (u in 1:length(R)){
 results <-run_MC_CW_OSL_DELOC(A=A[u], times, clusters =clusters, N_e = N_e[u],
                         n_filled = n_filled[u], R=R[u], method = method, output = output)
 plot_RLumCarlo(results,add=add_TF[u],legend = F, col=col[u], main=" your plot")
}
```

```
legend("topright",ncol=4,cex=0.55,title = "parameters" ,legend=c(paste0("A = ", A),
                                                         paste0("n_filled = ", n_filled),
                                                                 paste0("N_e = ", N_e),
                                                       paste0("R = ", R)),  text.col=col)


## End(Not run)
```

run_MC_ISO_DELOC *Run Monte-Carlo simulation for ISO for delocalized transition*

Description

Runs a Monte-Carlo (MC) simulation of isothermally stimulated luminesence (ISO-TL or ITL) using the one trap one recombination center (OTOR) model. Delocalized refers to involvement of the conduction band.

Usage

```
run_MC_ISO_DELOC(s, E, T = 20, times, clusters = 10, N_e = 200,
  n_filled = N_e, R, method = "par", output = "signal", ...)
```

Arguments

`s`	numeric (**required**): Escape frequency of the trap (s^-1).
`E`	numeric (**required**): Thermal activation energy of the trap (eV).
`T`	numeric (*with default*): Temperature (deg. C).
`times`	numeric (*with default*): The specified time within the simulation with the same syntax as the function seq().
`clusters`	numeric (*with default*): The number of MC runs.
`N_e`	integer (*with default*): The number of electrons.
`n_filled`	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
`R`	numeric (*with default*): The retrapping ratio.
`method`	character (*with default*): sequential `'seq'` or parallel processing `'par'`
`output`	character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)
`...`	further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1-R)))$$

Where in the function `n` := `n_filled` := `N` := `N_e`

Value

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

Function version

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_ISO_DELOC(): Run Monte-Carlo simulation for ISO for delocalized transition. Function version 0.0.1. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: 10.1016/j.jlumin.2018.11.024

Reuven, C. and S. Mckeever, 1997. Theory of thermoluminescence and related phenomena.

Examples

```
##===============================================================================##
## Example 1: Simulate ITL
##===============================================================================##



## Not run:
run_MC_ISO_DELOC(
 s = 3.5e12,
 E = 1.45,
 T = 200,
 R = 1,
 times = 0:10000) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

run_MC_ISO_LOC *Run Monte-Carlo simulation for ITL for localised transition*

Description

Runs a Monte-Carlo (MC) simulation of isothermally stimulated luminesence (ISO-TL or ITL) using the genralized one trap (GOT) model. Localized refers to excitation of an electron before it recombines, but without the involvement of the conduction band.

Usage

```
run_MC_ISO_LOC(s, E, T = 20, times, clusters = 10, n_filled = 100, r,
  method = "par", output = "signal", ...)
```

Arguments

`s`	numeric (**required**): Escape frequency of the trap (s^-1).
`E`	numeric (**required**): Thermal activation energy of the trap (eV).
`T`	numeric (*with default*): Temperature (deg. C).
`times`	numeric (*with default*): The specified time within the simulation with the same syntax as the function seq().
`clusters`	numeric (*with default*): The number of MC runs.
`n_filled`	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
`r`	numeric (*with default*): The retrapping ratio for localized models (dimensionless).
`method`	character (*with default*): sequential `'seq'` or parallel processing `'par'`
`output`	character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)
`...`	further arguments

Details

$$I_{LOC}(t) = -dn/dt = p(t) * (n^2/(r + n))$$

Where in the function `n` := `n_filled` := `N` := `N_e`

Value

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

Function version

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_ISO_LOC(): Run Monte-Carlo simulation for ITL for localised transition. Function version 0.0.1. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: 10.1016/j.jlumin.2018.11.024

Examples

```
##============================================================================##
## Example 1: Simulate ITL
##============================================================================##



    ## Not run:
run_MC_ISO_LOC(
 s = 3.5e12,
 E = 1.45,
 T = 200,
 r = 1,
 times = 0:10000) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

---

run_MC_ISO_TUN *Run Monte-Carlo Simulation for Isothermal Measurements for Tunneling Transition*

Description

Runs a Monte-Carlo (MC) simulation of isothermally stimulated luminesence (ISO-TL or ITL) using the tunneling (TUN) model. Tunneling refers to the movement of electrons from a trap directly to the recombination center.

Usage

```
run_MC_ISO_TUN(E, s, T = 200, rho, times, clusters = 10, r_c = 0,
  delta.r = 0.1, N_e = 200L, method = "par", output = "signal",
  ...)
```

Arguments

`E` numeric (**required**): Thermal activation energy of the trap (eV).

`s` numeric (**required**): Escape frequency of the trap (s^-1).

`T` numeric (**required**): Temperature (deg. C).

`rho` numeric (**required**): The calculated dimesionless charge density (normally written Rho') (also defined as (4 * pi rho/3)^1/3*r) where `rho` := the density of recombination centers within the material given in m^3 and `r` is _).

`times` numeric (*with default*): The specified time within the simulation with the same syntax as the function seq().

`clusters` numeric (*with default*): The number of MC runs.

`r_c` numeric (*with default*): The radius of tunneling (dimensionless)

`delta.r` numeric (*with default*):

`N_e` numeric (*with default*): The number of electrons.

`method` character (*with default*): sequential `'seq'` or parallel processing `'par'`

`output` character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)

`...` further arguments

Details

$$I_{TUN}(t) = -dn/dt = A * (n^2/(r + n))$$

Where in the function `n` := `n_filled` := `N` := `N_e` := `rho` := `rho'} := \code{r_c} := \code{rho'_c`

Value

This function returns a list.

Function version

0.1.0

How to cite

Friedrich, J., Kreutzer, S., 2019. run_MC_ISO_TUN(): Run Monte-Carlo Simulation for Isothermal Measurements for Tunneling Transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany), Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Univerité Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C., 2017. Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars. Journal of Luminescence 181, 114–120. doi: 10.1016/j.jlumin.2016.09.014

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: 10.1016/j.jlumin.2018.11.024

For a discussion of tunneling see: Aitken, M.J., 1985. Thermoluminescence dating. 276-280. doi: 10.1002/gea.3340020110

Examples

```
## Not run:
##============================================================================##
## Example 1: Simulate isothermal measurement
##============================================================================##
run_MC_ISO_TUN(
 E = 1.2,
 s = 1e10,
 T = 200,
 rho = 0.007,
 times = 0:5000) %>%
  plot_RLumCarlo(legend = TRUE)

## End(Not run)
```

---

run_MC_LM_OSL_DELOC *Run Monte-Carlo simulation for LM-OSL for delocalized transition*

Description

Runs a Monte-Carlo (MC) simulation of linearly modulated optically stimulated luminesence (LM-OSL) using the one trap one recombination center (OTOR) model.

Usage

```
run_MC_LM_OSL_DELOC(A, times, clusters = 10, N_e = 200,
  n_filled = N_e, R, method = "par", output = "signal", ...)
```

Arguments

A	numeric (**required**): The transition probability (cm^3/s).
times	numeric (*with default*): The number of MC runs.
clusters	numeric (*with default*): The number of clusters.
N_e	integer (*with default*): The number of electrons.
n_filled	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
R	numeric (*with default*): The retrapping ratio.

`method` character (*with default*): sequential `'seq'` or parallel processing `'par'`

`output` character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)

`...` further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1 - R)))$$

Value

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

Function version

0.1.0

How to cite

Kreutzer, S., 2019. run_MC_LM_OSL_DELOC(): Run Monte-Carlo simulation for LM-OSL for delocalized transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) 'Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars', Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##============================================================================##
## Example 1: Simulate LM-OSL
##============================================================================##
## Not run:
run_MC_LM_OSL_DELOC(
 A = 0.12,
 R = 1,
 times = 0:100) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

run_MC_LM_OSL_LOC *Run Monte-Carlo simulation for LM-OSL for localized transition*

Description

Runs a Monte-Carlo (MC) simulation of linearly modulated optically stimulated luminesence (LM-OSL) using the generalized one trap (GOT) model.

Usage

```
run_MC_LM_OSL_LOC(A, times, clusters = 10, n_filled = 100, r,
  method = "par", output = "signal", ...)
```

Arguments

`A`	numeric (**required**): The transition probability (cm^3/s).
`times`	numeric (*with default*): The number of MC runs.
`clusters`	numeric (*with default*): The number of clusters.
`n_filled`	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
`r`	numeric (*with default*):
`method`	character (*with default*): sequential `'seq'` or parallel processing `'par'`
`output`	character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)
`...`	further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1 - R)))$$

Value

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

Function version

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_LM_OSL_LOC(): Run Monte-Carlo simulation for LM-OSL for localized transition. Function version 0.0.1. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C. (2017) ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
##============================================================================##
## Example 1: Simulate LM-OSL
##============================================================================##
## Not run:
run_MC_LM_OSL_LOC(
 A = 0.12,
 r = 1,
 times = 0:100) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

---

run_MC_LM_OSL_TUN *Run Monte-Carlo simulation for LM-OSL*

Description

Run Monte-Carlo simulation for LM-OSL

Usage

```
run_MC_LM_OSL_TUN(A, rho, times, clusters = 10, r_c = 0,
  delta.r = 0.1, N_e = 200, method = "par", output = "signal", ...)
```

Arguments

`A`	numeric (**required**): The transition probaility (cm^3/s).
`rho`	numeric (**required**): The calculated dimesionless Charge density (normally written Rho’).
`times`	vector (*with default*): The number of MC runs.
`clusters`	numeric (*with default*): The number of clusters.
`r_c`	numeric (*with default*): The retrapping ratio.
`delta.r`	numeric (*with default*):
`N_e`	numeric (*with default*): The number of electrons.
`method`	character (*with default*): sequential `'seq'` or parallel processing `'par'`
`output`	character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)
`...`	further arguments

Details

ADD EQUATION

Value

This function returns a list.

Function version

0.1.0

How to cite

Friedrich, J., 2019. run_MC_LM_OSL_TUN(): Run Monte-Carlo simulation for LM-OSL. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

References

Pagonis, V. and Kulp, C. (2017) ‘Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars’, Journal of Luminescence. Elsevier, 181, pp. 114–120. doi: 10.1016/j.jlumin.2016.09.014.

Examples

```
## Not run:

##TODO: Primary example, should be verified
run_MC_LM_OSL_TUN(A = 10000, rho = 0.0001, times = 1:100, clusters = 10, r = NULL,
 delta.r = 0.1,
 N_e = 200, method = "par", output = "signal") %>%
 plot_RLumCarlo(norm = T)


## End(Not run)
```

run_MC_TL_DELOC *Run Monte-Carlo simulation for TL for delocalized transition*

Description

Runs a Monte-Carlo (MC) simulation of thermo-luminesence (TL) using the one trap one recombination center (OTOR) model. Delocalized refers to involvement of the conduction band.

Usage

```
run_MC_TL_DELOC(s, E, times, clusters = 10, N_e = 200,
  n_filled = N_e, R, method = "par", output = "signal", ...)
```

Arguments

`s` numeric (**required**): Escape frequency of the trap (s^-1).

`E` numeric (**required**): Thermal activation energy of the trap (eV).

`times` numeric (*with default*): The specified time within the simulation with the same syntax as the function seq().

`clusters` numeric (*with default*): The number of MC runs.

`N_e` integer (*with default*): The number of electrons.

`n_filled` integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.

`R` numeric (*with default*): The retrapping ratio.

`method` character (*with default*): sequential `'seq'` or parallel processing `'par'`

`output` character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)

`...` further arguments

Details

$$I_{DELOC}(t) = -dn/dt = p(t) * (n^2/(NR + n(1-R)))$$

where in the function `N` := `N_e`:= `n` :=`n_filled`

Value

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

Function version

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_TL_DELOC(): Run Monte-Carlo simulation for TL for delocalized transition. Function version 0.0.1. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: 10.1016/j.jlumin.2018.11.024

Reuven, C. and S. Mckeever, 1997. Theory of thermoluminescence and related phenomena.

Examples

```
##============================================================================##
## Example 1: Simulate TL
##============================================================================##
## Not run:
run_MC_TL_DELOC(
 s = 3.5e12,
 E = 1.45,
 R = 1,
 times = 100:450) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)

#' @examples
##============================================================================##
## Example 2: Plot multiple TL stimulation TL curves in R with varying params
##============================================================================##



      ## Not run:
# define your parameters
times=seq(100,450,1)
s=rep(3.5e12,4)
E=rep(1.45,4)
R<-c(0.7e-6,1e-6,0.01,0.1)
clusters=1000
N_e =c(400, 500, 700, 400)
n_filled =c(400, 500, 300, 70)
method="par"
output ="signal"
col=c(1,2,3,4) # different colours for the individual curves
plot_uncertainty <- c(TRUE,TRUE,TRUE,TRUE)  # do you want to see the uncertainty?
add_TF <- c(FALSE,rep(TRUE, (length(R)-1)))
for (u in 1:length(R)){
 results <-run_MC_TL_DELOC(times=times, s=s[u],E=E[u], clusters =clusters, N_e = N_e[u],
                        n_filled = n_filled[u], R=R[u], method = method, output = output)
 plot_RLumCarlo(results,add=add_TF[u],legend = FALSE, col=col[u], main=" your plot", ylim=c(0,20))
}
legend("topright",ncol=5,cex=0.55,title = "parameters" ,legend=c(paste0("E = ", E),
                                                                paste0("s = ", s),
                                                        paste0("n_filled = ", n_filled),
                                                                paste0("N_e = ", N_e),
                                                       paste0("R = ", R)),  text.col=col)

## End(Not run)
```

---

run_MC_TL_LOC *Run Monte-Carlo simulation for TL for localised transition*

Description

Runs a Monte-Carlo (MC) simulation of thermo-luminesence (TL) using the generalized one trap (GOT) model. Localized refers to excitation of an electron before it recombines, but without the involvement of the conduction band.

Usage

```
run_MC_TL_LOC(s, E, times, clusters = 10, n_filled = 100, r,
  method = "par", output = "signal", ...)
```

Arguments

`s`	numeric (**required**): Escape frequency of the trap (s^-1).
`E`	numeric (**required**): Thermal activation energy of the trap (eV).
`times`	numeric (*with default*): The specified time within the simulation with the same syntax as the function seq().
`clusters`	numeric (*with default*): The number of MC runs.
`n_filled`	integer (*with default*): The number of electron traps that are filled at the beginning of the simulation.
`r`	numeric (*with default*): The retrapping ratio.
`method`	character (*with default*): sequential `'seq'` or parallel processing `'par'`
`output`	character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)
`...`	further arguments

Details

$$I_{LOC}(t) = -dn/dt = p(t) * (n^2/(r + n))$$

where in the function `n` :=`n_filled`

Value

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

Function version

0.1.0

How to cite

Kreutzer, S., 2019. run_MC_TL_LOC(): Run Monte-Carlo simulation for TL for localised transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, CNRS - Université Bordeaux Montaigne (France)

References

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: 10.1016/j.jlumin.2018.11.024

Examples

```
##============================================================================##
## Example 1: Simulate TL
##============================================================================##


## Not run:
run_MC_TL_LOC(
 s = 3.5e12,
 E = 1.45,
 r = 1,
 times = 100:450) %>%
   plot_RLumCarlo(legend = T)


## End(Not run)
```

run_MC_TL_TUN *Run Monte-Carlo Simulation for TL using Tunnelling Transition*

Description

Runs a Monte-Carlo (MC) simulation of thermo-luminesence (TL) using the tunneling (TUN) model. Tunneling transitions refers to the direct movement of electrons from a trap directly to the recombination centre.

Usage

```
run_MC_TL_TUN(s, E, rho, r_c = 0, times, clusters = 10, N_e = 200,
  delta.r = 0.1, method = "par", output = "signal", ...)
```

Arguments

s	list (**required**): Escape frequency of the trap (s^-1).
E	numeric (**required**): Thermal activation energy of the trap (eV).
rho	numeric (**required**): The calculated dimesionless Charge density.
r_c	numeric (*with default*): The dimensionless minimal critical radius.
times	vector (*with default*): The number of MC runs.
clusters	numeric (*with default*): The number of clusters.
N_e	numeric (*with default*): The number of electrons

`delta.r` numeric (*with default*): The approriate distance interval along the r axis (dimensionless).

`method` character (*with default*): sequential `'seq'` or parallel processing `'par'`

`output` character (*with default*): output is either the `'signal'` (the default) or `'remaining_e'` (the remaining charges, electrons, in the trap)

`...` further arguments

Details

$$I_{TUN}(t) = -dn/dt = A * (n^2/(r+n))$$

where in the function `N` := `N_e` := `rho` := `rho'}` := `\code{r_c}` := `\code{rho'_c`

Value

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

Function version

0.1.0

How to cite

Friedrich, J., Kreutzer, S., 2019. run_MC_TL_TUN(): Run Monte-Carlo Simulation for TL using Tunnelling Transition. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. RLumCarlo: Monte-Carlo Methods for Simulating Luminescence PhenomenaR package version 0.1.0.9000-84.

Author(s)

Johannes Friedrich, University of Bayreuth (Germany), Sebastian Kreutzer, IRAMAT-CRP2A, UMR 5060, Université Bordeaux Montaigne (France)

References

Pagonis, V. and Kulp, C., 2017. Monte Carlo simulations of tunneling phenomena and nearest neighbor hopping mechanism in feldspars. Journal of Luminescence 181, 114–120. doi: 10.1016/j.jlumin.2016.09.014

Pagonis, V., Friedrich, J., Discher, M., Müller-Kirschbaum, A., Schlosser, V., Kreutzer, S., Chen, R. and Schmidt, C., 2019. Excited state luminescence signals from a random distribution of defects: A new Monte Carlo simulation approach for feldspar. Journal of Luminescence 207, 266–272. doi: 10.1016/j.jlumin.2018.11.024

Further reading Aitken, M.J., 1985. Thermoluminescence dating. 276-280. doi: 10.1002/gea.3340020110

Examples

```
## Not run:
##============================================================================##
## Example 1: Simulate TL measurement
##============================================================================##
run_MC_TL_TUN(s = 3.5e12,
          E = 1.45,
          rho = 0.015,
          r_c = 0.85,
          times = 200:500) %>%
    plot_RLumCarlo(legend = T)

## End(Not run)
```

Index