

Package ‘RLumCarlo’

October 8, 2019

Type Package

Title Monte-Carlo Methods for Simulating Luminescence Phenomena

Version 0.1.0.9000-51

Date 2019-10-08

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>>)

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

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 <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=RLumModel>

LinkingTo Rcpp,
RcppProgress,
RcppArmadillo

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

Suggests R.rsp

Encoding UTF-8

VignetteBuilder R.rsp

RoxygenNote 6.1.1

R topics documented:

methods	2
plot_RLumCarlo	3
run_MC_CW_IRSL_DELOC	4
run_MC_CW_IRSL_LOC	5
run_MC_CW_IRSL_TUN	6
run_MC_ISO_DELOC	8
run_MC_ISO_LOC	9
run_MC_ISO_TUN	10
run_MC_LM_OSL_DELOC	12
run_MC_LM_OSL_LOC	13
run_MC_LM_OSL_TUN	14
run_MC_TL_DELOC	16
run_MC_TL_LOC	17
run_MC_TL_TUN	18
Index	20

methods	<i>methods</i>
---------	----------------

Description

Summarize RLumCarlo Modelling results, so that they can be plotted easily

Usage

```
## S3 method for class 'RLumCarlo_Model_Output'
summary(object, ...)
```

Arguments

- object [list](#) of class RLumCarlo_Model_Output: RLumCarlo simulation output object produced by all
- ... further arguments passed to the method

Value

This function returns a [data.frame](#)

Function version

0.1.0

How to cite

Friedrich, J., Kreutzer, S., 2019. methods(): methods. 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 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

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, plot_uncertainty = TRUE,
               norm = FALSE, add = FALSE, ...)
```

Arguments

results	data.frame (required)
times	numeric (<i>optinal</i>): Optional vector for the x-axis
plot_uncertainty	logical (<i>with default</i>): Enable/disable uncertainty polygon plot
norm	logical (<i>with default</i>): Normalise curve to the highest intensity
add	logical (<i>with default</i>): allow overplotting of results
...	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

Value

This function returns a graphical output

Function version

0.1.0

How to cite

Friedrich, J., Kreutzer, S., 2019. plot_RLumCarlo(): Plot results from Monte-Carlo simulations with RLumCarlo. 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 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

run_MC_CW_IRSL_DELOC *Run Monte-Carlo simulation for CW-IRSL for GOT model*

Description

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

Usage

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

Arguments

A	numeric (required) : The transition probability in 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):
output	character (with default):
...	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_IRSL_DELOC(): Run Monte-Carlo simulation for CW-IRSL for GOT model. 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 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

##TODO

Examples

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

## End(Not run)
```

run_MC_CW_IRSL_LOC	<i>Run Monte-Carlo simulation for CW-IRSL for localised transition</i>
--------------------	--

Description

Runs a Monte Carlo (MC) simulation of constant wave infrared stimulated luminescence (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 in 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) :
output	character (with default) :
...	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

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_CW_IRSL_LOC(): Run Monte-Carlo simulation for CW-IRSL 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 Phenomena R package version 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

##TODO

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	<i>Run Monte-Carlo simulation for CW-IRSL</i>
--------------------	---

Description

Run Monte-Carlo simulation for CW-IRSL

Usage

```
run_MC_CW_IRSL_TUN(A, rho, times, clusters = 10, r = NULL, N_e = 200,
  method = "seq", 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

0.2.0

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 Phenomena R package version 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

Pagonis 2017

Examples

```
## Not run:

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

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

## End(Not run)
```

run_MC_ISO_DELOC

*Run Monte-Carlo simulation for ISO for GOT model***Description**

##TODO

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)
E	numeric (required)
T	numeric (with default)
times	numeric (with default)
clusters	numeric (with default):
N_e	integer (with default)
n_filled	integer (with default)
R	numeric (with default):
method	character (with default):
output	character (with default):
...	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_ISO_DELOC(): Run Monte-Carlo simulation for ISO for GOT model. 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 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

##TODO

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

##TODO

Usage

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

Arguments

s	numeric (required)
E	numeric (required)
T	numeric (with default)
times	numeric (with default):
clusters	numeric (with default):
n_filled	integer (with default):
r	numeric (with default):
method	character (with default):
output	character (with default):
...	further arguments

Details

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

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 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

##TODO

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

Description

Run Monte-Carlo simulation for isothermal measurements

Usage

```
run_MC_ISO_TUN(E, s, T = 200, rho, times, clusters = 10, r = NULL,
  N_e = 200, method = "par", output = "signal", ...)
```

Arguments

E	numeric (required)
s	numeric (required)
T	numeric (required)
rho	numeric (required)
times	numeric (<i>with default</i>)
clusters	numeric (<i>with default</i>):
r	numeric (<i>with default</i>)
N_e	numeric (<i>with default</i>)
method	character (<i>with default</i>)
output	character (<i>with default</i>)
...	further arguments

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. 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 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

Pagonis 2017

Examples

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

times <- seq(0, 5000)
run_MC_ISO_TUN(
  E = 1.2,
  s = 1e10,
  T = 200,
  rho = 0.007,
  times = times) %>%
```

```
plot_RLumCarlo(legend = T)

## End(Not run)
```

```
run_MC_LM_OSL_DELOC      Run Monte-Carlo simulation for LM-OSL for GOT model
```

Description

```
##TODO
```

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)
times	numeric (with default)
clusters	numeric (with default):
N_e	integer (with default)
n_filled	integer (with default)
R	numeric (with default):
method	character (with default):
output	character (with default):
...	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 GOT model. 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 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

##TODO

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 localised transition

Description

##TODO

Usage

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

Arguments

A	numeric (required)
times	numeric (with default):
clusters	numeric (with default):
n_filled	integer (with default):
r	numeric (with default):
method	character (with default):
output	character (with default):
...	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

0.0.1

How to cite

Kreutzer, S., 2019. run_MC_LM_OSL_LOC(): Run Monte-Carlo simulation for LM-OSL 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 Phenomena R package version 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

##TODO

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	<i>Run Monte-Carlo simulation for LM-OSL</i>
-------------------	--

Description

Run Monte-Carlo simulation for LM-OSL

Usage

```
run_MC_LM_OSL_TUN(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

0.1.0

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 Phenomena R package version 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

Johannes Friedrich, University of Bayreuth (Germany)

References

Pagonis 2017

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 GOT model***Description**

##TODO

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)
E	numeric (required)
times	numeric (with default)
clusters	numeric (with default):
N_e	integer (with default)
n_filled	integer (with default)
R	numeric (with default):
method	character (with default):
output	character (with default):
...	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_TL_DELOC(): Run Monte-Carlo simulation for TL for GOT model. 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 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

##TODO

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)
```

run_MC_TL_LOC

Run Monte-Carlo simulation for TL for localised transition

Description

##TODO

Usage

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

Arguments

s	numeric (required)
E	numeric (required)
times	numeric (with default)
clusters	numeric (with default):
n_filled	integer (with default)
r	numeric (with default):
method	character (with default):
output	character (with default):
...	further arguments

Details

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

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_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 Phenomena R package version 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

Author(s)

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

References

##TODO

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

Description

Run Monte-Carlo simulation for TL

Usage

```
run_MC_TL_TUN(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

0.1.0

How to cite

Friedrich, J., 2019. `run_MC_TL_TUN()`: Run Monte-Carlo simulation for TL. Function version 0.1.0. In: Friedrich, J., Kreutzer, S., Pagonis, V., Schmidt, C., 2019. `RLumCarlo`: Monte-Carlo Methods for Simulating Luminescence Phenomena R package version 0.1.0.9000-51. <https://CRAN.R-project.org/package=RLumModel>

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

## End(Not run)
```

Index

array, [4](#), [6](#), [8](#), [10](#), [12](#), [13](#), [16](#), [18](#), [19](#)

character, [4](#), [5](#), [7–9](#), [11–13](#), [15–17](#), [19](#)

data.frame, [2](#), [3](#)

integer, [4](#), [5](#), [8](#), [9](#), [12](#), [13](#), [16](#), [17](#)

list, [2](#), [19](#)

logical, [3](#)

methods, [2](#)

numeric, [3–5](#), [7–9](#), [11–13](#), [15–17](#), [19](#)

plot_RLumCarlo, [3](#)

run_MC_CW_IRSL_DELOC, [4](#)

run_MC_CW_IRSL_LOC, [5](#)

run_MC_CW_IRSL_TUN, [6](#)

run_MC_ISO_DELOC, [8](#)

run_MC_ISO_LOC, [9](#)

run_MC_ISO_TUN, [10](#)

run_MC_LM_OSL_DELOC, [12](#)

run_MC_LM_OSL_LOC, [13](#)

run_MC_LM_OSL_TUN, [14](#)

run_MC_TL_DELOC, [16](#)

run_MC_TL_LOC, [17](#)

run_MC_TL_TUN, [18](#)

summary.RLumCarlo_Model_Output
 (methods), [2](#)

vector, [7](#), [15](#), [19](#)