Package 'RLumModel'

January 8, 2016

Type Package

Title Modelling Ordinary Differential Equations Leading to Luminescence

Version 0.1.0

Date 2016-XX-XX

Author Johannes Friedrich [aut, trl, cre],
Sebastian Kreutzer [aut, ths],
Christoph Schmidt [aut, ths]

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

Description A collection of function to simulate luminescence signals in the mineral quartz based on published models.

Contact Package Developer Team <developer@model.r-luminescence.de>

License GPL-3

Depends R (>= 3.2.3), utils, Luminescence (>= 0.5.1)

Imports deSolve (>= 1.12)

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

Collate RLumModel-package.R calc_Signal.R create_DRT.sequence.R create_SAR.sequence.R model_LuminescenceSignals.R plot_concentrations.R read_SEQ2R.R set_ODE.R set_ODE_LM_OSL.R set_Pars.R simulate_CW_OSL.R simulate_LM_OSL.R simulate_RF.R simulate_TL.R simulate_heating.R simulate_illumination.R simulate_irradiation.R simulate_pause.R translate_Sequence.R

RoxygenNote 5.0.1

Suggests testthat

NeedsCompilation no

R topics documented:

RLumModel-package *Modelling Ordinary Differential Equations Leading to Luminescence*

Description

A collection of function to simulate luminescence signals in the mineral quartz based on published models.

Details

Package:	RLumModel
Type:	Package
Version:	0.1.0
Date:	2016-XX-XX
License:	GPL-3

Author(s)

Authors

Johannes Friedrich	University of Bayreuth, Germany
Sebastian Kreutzer	IRAMAT-CRP2A, Universite Bordeaux Montaigne, France
Christoph Schmidt	University of Bayreuth, Germany

Supervisor

Christoph Schmidt, University of Bayreuth, Germany
Sebastian Kreutzer, IRAMAT-CRP2A, Universite Bordeaux Montaigne, France

Support contact

<developers@model.r-luminescence.de>

Project source code repository

https://github.com/R-Lum/RLumModel

Related projects

http://www.r-luminescence.de
http://cran.r-project.org/package=Luminescence
http://shiny.r-luminescence.de
http://cran.r-project.org/package=RLumShiny

Package maintainer

Johannes Friedrich, University of Bayreuth, Germany
<johannes.friedrich@uni-bayreuth.de>

Acknowledgement

The work of Johannes Friedrich is gratefully supported by the DFG in framework of the project 'Modelling quartz luminescence signal dynamics relevant for dating and dosimetry' (SCHM 305114-1)

`ExampleData.ModelOutput`

Example data (TL curve) simulated from Bailey (2001 ,fig. 1)

Description

Example data (TL curve) simulated from Bailey (2001 ,fig. 1)

Format

A RLum.Analysis object containing one TL curve as RLum.Data.Curve.

Note

This example has only one record (TL). The used sequence was sequence <- list(IRR = c(temp = 20, dose = 10, DoseRate = 1), TL = c(temp_begin = 20, temp_end = 400, heating_rate = 5))

Source

model_LuminescenceSignals()

References

Bailey, R.M., 2001. Towards a general kinetic model for optically and thermally stimulated luminescence of quartz. Radiation Measurements 33, 17-45.

Examples

```
data(ExampleData.ModelOutput,envir = environment())
plot_RLum.Analysis(model.output)
```

`model_LuminescenceSignals`

Model Luminescence Signals

Description

This function models luminescence signals for quartz based on published physical models. It is possible to simulate TL, (CW-) OSL, RF measurements in a arbitrary sequence. This sequence is definded as a `list` of certain abrivations. Furthermore it is possible to load a sequence direct from the Riso Sequence Editor. The output is an `RLum.Analysis`object and so the plots are done by the `plot_RLum.Analysis` function. If a SAR sequence is simulated the plot output can be disabled and SAR analyse functions can be used.

Usage

```
model_LuminescenceSignals(model, sequence, lab.dose_rate = 1,
  simulate_sample_history = FALSE, plot = TRUE, verbose = TRUE,
  show.structure = FALSE, ...)
```

Arguments

model	character (**required**): set model to be used. Available models are: "Bailey2001", "Bailey2002", "Bailey2004", "Pagonis2007", "Pagonis2008"
sequence	list (**required**): set sequence to model as list or as *.seq file from the Riso sequence editor. To simulate SAR measurements there is an extra option to set the sequence list (cf. details).
lab.dose_rate	numeric (with default): laboratory dose rate in XXX Gy/s for calculating seconds into Gray in the *.seq file.
simulate_sample_history	logical (with default): FALSE (with default): simulation begins at laboratory conditions, TRUE: simulations begins at crystallization (all levels 0) process
plot	logical (with default): Enables or disables plot output
verbose	logical (with default): Verbose mode on/off
show.structure	logical (with default): Shows the structure of the result. Recommended to show record.id to analyse with plot_concentrations.
...	further arguments and graphical parameters passed to plot.default. See details for further information.

Details

Defining a **sequence**

Arguments	Description	Sub-arguments
TL	thermally stimulated luminescence	'temp begin', 'temp end', 'heating rate'
OSL	optically stimulated luminescence	'temp', 'duration','optical_power'
ILL	illumination	'temp', 'duration','optical_power'
LM_OSL	linear modulated OSL	'temp', 'duration', optional: 'start_power', 'end_power'
RL/RF	radioluminescence	'temp','dose', 'dose_rate'
IRR	irradiation	'temp','dose', 'dose_rate'
CH	cutheat	'temp', optional: 'duration', 'heating_rate'
PH	preheat	'temp', 'duration' optional: 'heating_rate'
PAUSE	pause	'temp', 'duration'

Defining a **SAR-sequence**

Abrivation	Description	examples
RegDose	Dose points of the regenerative cycles	c(0, 80, 140, 260, 320, 0, 80)
TestDose	Test dose for the SAR cycles	50
PH	Temperature of the preheat	240
CH	Temperature of the cutheat	200
OSL_temp	Temperature of OSL read out	125
OSL_duration	Duration of OSL read out	default: 40

Irr_temp	Temperature of irradiation	default: 20
PH_duration	Duration of the preheat	default: 10
dose_rate	Dose rate of the laboratory irradiation source	default: 1
optical_power	Percentage of the full illumination power	default: 90
Irr_2recover	Dose to be recovered in a dose-recovery-test	20

Value

This function returns an RLum.Analysis object with all TL, (LM-) OSL and RF/RL steps in the sequence. Every entry is an RLum.Data.Curve object and can be plotted, analysed etc. with further RLum-functions.

Function version

0.1.0

Author(s)

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

References

Bailey, R.M., 2001. Towards a general kinetic model for optically and thermally stimulated luminescence of quartz. Radiation Measurements 33, 17-45.

Bailey, R.M., 2002. Simulations of variability in the luminescence characteristics of natural quartz and its implications for estimates of absorbed dose. Radiation Protection Dosimetry 100, 33-38.

Bailey, R.M., 2004. Paper I-simulation of dose absorption in quartz over geological timescales and it simplications for the precision and accuracy of optical dating. Radiation Measurements 38, 299-310.

Pagonis, V., Chen, R., Wintle, A.G., 2007: Modelling thermal transfer in optically stimulated luminescence of quartz. Journal of Physics D: Applied Physics 40, 998-1006.

Pagonis, V., Wintle, A.G., Chen, R., Wang, X.L., 2008. A theoretical model for a new dating protocol for quartz based on thermally transferred OSL (TT-OSL). Radiation Measurements 43, 704-708.

Soetaert, K., Cash, J., Mazzia, F., 2012. Solving differential equations in R. Springer Science & Business Media.

See Also

plot, RLum, plot_concentrations, read_SEQ2R

Examples

```
##=================================================================##
## Example 1: Simulate sample history of Bailey2001
## (cf. Bailey, 2001, Fig. 1)
##=================================================================##

##set sequence with the following steps
## (1) Irradiation at 20 deg. C with a dose of 10 Gy and a dose rate of 1 Gy/s
```

```
## (2) TL from 20-400 deg. C with a rate of 5 K/s

sequence <-
  list(
    IRR = c(20, 10, 1),
    TL = c(20, 400, 5)
  )

##model sequence
model.output <- model_LuminescenceSignals(
  sequence = sequence,
  model = "Bailey2001",
  show.structure = TRUE
)

## Not run:
##============================================================================##
## Example 2: Simulate sequence at labour without sample history
##============================================================================##

##set sequence with the following steps
## (1) Irraditation at 20 deg. C with a dose of 100 Gy and a dose rate of 1 Gy/s
## (2) Preheat to 200 deg. C and hold for 10 s
## (3) LM-OSL at 125 deg. C. for 100 s
## (4) Cutheat at 200 dec. C.
## (5) Irraditation at 20 deg. C with a dose of 10 Gy and a dose rate of 1 Gy/s
## (6) Pause at 200 de. C. for 100 s
## (7) OSL at 125 deg. C for 100 s with 90 % optical power
## (8) Pause at 200 deg. C for 100 s
## (9) TL from 20-400 deg. C with a heat rate of 5 K/s
## (10) Radiolumiescence at 20 deg. C with a dose of 20 Gy and a dose rate of 0.01 Gy/s

sequence <-
 list(
   IRR = c(20, 100, 1),
   PH = c(200, 10),
   LM_OSL = c(125, 100),
   CH = c(200),
   IRR = c(20, 10, 1),
   PAUSE = c(200, 100),
   OSL = c(125, 100, 90),
   PAUSE = c(200, 100),
   TL = c(20, 400, 5),
   RF = c(20, 200, 0.01)
)

# call function "model_LuminescenceSignals", set sequence = sequence,
# model = "Pagonis2008" (palaeodose = 200 Gy) and simulate_sample_history = FALSE (default),
# because the sample history is not part of the sequence

model.output <- model_LuminescenceSignals(
   sequence = sequence,
   model = "Pagonis2008"
   )
```

```
##============================================================================##
## Example 3: Simulate SAR sequence
##============================================================================##

##set SAR sequence with the following steps
## (1) RegDose: set regenerative dose [Gy] as vector
## (2) TestDose: set test dose [Gy]
## (3) PH: set preheat temperature in deg. C
## (4) CH: Set cutheat temperature in deg. C
## (5) OSL_temp: set OSL reading temperature in deg. C
## (6) OSL_duration: set OSL reading duration in s

sequence <- list(
 RegDose = c(0,10,20,50,90,0,10),
 TestDose = 5,
 PH = 240,
 CH = 200,
 OSL_temp = 125,
 OSL_duration = 70)

# call function "model_LuminescenceSignals", set sequence = sequence,
# model = "Pagonis2007" (palaeodose = 20 Gy) and simulate_sample_history = FALSE (default),
# because the sample history is not part of the sequence

 model.output <- model_LuminescenceSignals(

 sequence = sequence,
 model = "Pagonis2007",
 plot = FALSE
 )

# in environment is a new object "model.output" with the results of
# every step of the given sequence.
# Plots are done at OSL and TL steps and the growth curve

# call "analyse_SAR.CWOSL" from RLum package
 results <- analyse_SAR.CWOSL(model.output,
                              signal.integral.min = 1,
                              signal.integral.max = 15,
                              background.integral.min = 601,
                              background.integral.max = 701,
                              fit.method = "EXP",
                              dose.points = c(0,10,20,50,90,0,10))


##============================================================================##
## Example 4: generate sequence from *.seq file and run SAR simulation
##============================================================================##

# call function "model_LuminescenceSignals", load *.seq file for sequence,
# set model = "Bailey2002" (palaeodose = 10 Gy)
# and simulate_sample_history = FALSE (default),
# because the sample history is not part of the sequence

model.output <- model_LuminescenceSignals(
  sequence = "inst/extdata/sample_SAR_cycle.SEQ",
  model = "Bailey2002",
```

```
  plot = FALSE
)


## call RLum package function "analyse_SAR.CWOSL" to analyse the simulated SAR cycle

results <- analyse_SAR.CWOSL(model.output,
                             signal.integral.min = 1,
                             signal.integral.max = 10,
                             background.integral.min = 301,
                             background.integral.max = 401,
                             dose.points = c(0,5,10,20,50,5,0),
                             fit.method = "EXP")

print(get_RLum(results))


##============================================================================##
## Example 5: compare different optical powers of stimulation light
##============================================================================##

# call function "model_LuminescenceSignals", model = "Bailey2004"
# and simulate_sample_history = FALSE (default),
# because the sample history is not part of the sequence
# the optical_power of the LED is varied and then compared.

optical_power <- seq(from = 0,to = 100,by = 20)

model.output <- lapply(1:length(optical_power), function(x){

sequence <- list(IRR = c(20, 50, 1),
                 PH = c(220, 10, 5),
                 OSL = c(125, 50, optical_power[x])
                 )

return(model_LuminescenceSignals(
       sequence = sequence,
       model = "Bailey2004",
       plot = FALSE
       ))
})

##combine output curves
model.output.merged <- merge_RLum(model.output)

##plot
plot_RLum(
 object = model.output.merged,
 xlab = "Illumination time [s]",
 ylab = "OSL signal [a.u.]",
 main = "OSL signal dependency on optical power of stimulation light",
 legend.text = paste("Optical power density", 20*optical_power/100, "mW/cm^2"),
 combine = TRUE)


## End(Not run)
```

plot_concentrations *Plot electron/hole concentrations of a specific record.id*

Description

The functions provides a plot of all changes in time of the electron respectively hole concentration in electron traps, hole centres, in the condunction and valence band.

Usage

```
plot_concentrations(object, record.id, plot.saturation = FALSE, ...)
```

Arguments

object	RLum.Analysis (**required**): S4 object of class RLum.Analysis, e.g. the values of model_LuminescenceSignals.
record.id	numeric (**required**): id of the simulated record, which is to plot. To see all record.ids use structure_RLum, see examples.
plot.saturation	logical (with default): plots the saturation of every level from a specific model.
...	further arguments and graphical parameters passed to plot.default and plot_RLum.Analysis.

Details

The function produces a multiple plot output and uses in main parts the Luminescence function plot_RLum.Analysis. A file output is recommended (e.g., pdf).

Value

Returns multiple plots.

Function version

0.1.0

Author(s)

Johannes Friedrich, University of Bayreuth (Germany),

References

Bailey, R.M., 2001. Towards a general kinetic model for optically and thermally stimulated luminescence of quartz. Radiation Measurements 33, 17-45.

Bailey, R.M., 2002. Simulations of variability in the luminescence characteristics of natural quartz and its implications for estimates of absorbed dose. Radiation Protection Dosimetry 100, 33-38.

Bailey, R.M., 2004. Paper I-simulation of dose absorption in quartz over geological timescales and it simplications for the precision and accuracy of optical dating. Radiation Measurements 38, 299-310.

Pagonis, V., Chen, R., Wintle, A.G., 2007: Modelling thermal transfer in optically stimulated luminescence of quartz. Journal of Physics D: Applied Physics 40, 998-1006.

Pagonis, V., Wintle, A.G., Chen, R., Wang, X.L., 2008. A theoretical model for a new dating protocol for quartz based on thermally transferred OSL (TT-OSL). Radiation Measurements 43, 704-708.

See Also

plot, plot_RLum.Analysis, model_LuminescenceSignals

Examples

```
##load data
data(ExampleData.ModelOutput, envir = environment())

##show structure
Luminescence::structure_RLum(model.output)

##plot all concentrations
plot_concentrations(object = model.output,
                    record.id = 1)

##plot only specific energy-band-level (e.g. 110 degree celsius trap, "concentration level 1")
plot_concentrations(object = model.output,
                    record.id = 1,
                    subset = list(recordType = "concentration level 1"))

##plot every level on a single plot
plot_concentrations(object = model.output,
                    record.id = 1,
                    plot.single = TRUE)
```

---

read_SEQ2R *Parse a Risoe SEQ-file to a sequence neccessary for simulating quartz luminescence*

Description

A SEQ-file created by the Risoe Sequence Editor can be imported to simulate the sequence written in the sequence editor.

Usage

```
read_SEQ2R(file, lab.dose_rate = 1, txtProgressBar = TRUE)
```

Arguments

file	character (**required**): a *.seq file created by the Risoe Sequence Editor
lab.dose_rate	character (with default): set the dose rate of the radiation source in the laboratory [Gy/s]. Default: 1 Gy/s
txtProgressBar	logical (with default): enables or disables the txtProgressBar for a visuell control of the progress. Default: txtProgressBar = TRUE

Details

Supported versions

Supppored and tested: version 4.36.

Value

This function returns a list with the parsed *.seq file and the required steps for model_LuminescenceSignals.

Function version

0.1.0

Author(s)

Johannes Friedrich, University of Bayreuth (Germany),

References

Riso: Sequence Editor User Manual. Available at: http://www.nutech.dtu.dk/english/-/media/Andre_Universitetsenheder/Nutech/Produkter%20og%20services/Dosimetri/radiation_measurement_instruments/tl_osl_reader/Manuals/SequenceEditor.ashx?la=da

See Also

model_LuminescenceSignals, readLines

Examples

```
##search "example_SAR_cycle.SEQ" in "extdata" in package "RLumModel"
path <- system.file("extdata", "example_SAR_cycle.SEQ", package="RLumModel")

sequence <- read_SEQ2R(file = path)
```

Index