

# RLumModel - Using own parameter sets

*Johannes Friedrich*

*2016-07-01*

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Examples</b>	<b>1</b>
2.1	Pagonis 2009 . . . . .	2
2.1.1	Set parameters . . . . .	2
2.1.2	Running the simulation with RLumModel . . . . .	3
2.1.3	Re-calculate the original results . . . . .	3
2.2	Lawless 2009 . . . . .	8
2.2.1	Set parameters and recalculate the results . . . . .	8
2.3	Chen 2013 . . . . .	9
2.3.1	Set parameters . . . . .	10
2.3.2	Re-calculate the original results . . . . .	10
2.3.3	Sensitivity analysis with own parameters . . . . .	10
<b>3</b>	<b>Summary</b>	<b>13</b>
	<b>References</b>	<b>14</b>

## 1 Introduction

With ‘RLumModel’  $\geq 0.2.0$  is it possible to simulate quartz luminescence behaviour of own parameter sets or parameter sets, which are not included in the package but also known in literature. Wide spreaded OTOR (One-Trap-One-Recombination-center) models can be included, too. This vignette gives three comprehensive examples how to implement parameter sets, and proves the results recalculating the original simulations.

‘RLumModel’ offers maximum flexibility and fast calculation of ordinary first-order differential equations (ODEs) describing luminescence behaviour, because of:

- flexible handling of different numbers of electron traps and hole centres in generating the model equations. The user has not to care about coding ODEs in R.
- Solving these equations is performed with C++ (Eddelbuettel, 2013) and offers a faster calculation.
- Sensitivity analysis and inverse modelling is also possible with own parameter sets (see chapter 2.3.3)

## 2 Examples

This chapter shows the handling of own parameter sets with ‘RLumModel’. For this purpose three model parameters known from literature were taken: Pagonis et al. (2009), Lawless et al. (2009) and Chen and Pagonis (2013).

## 2.1 Pagonis 2009

Pagonis et al. (2009) presented parameters for their luminescence modelling of radioluminescence. This model was built for  $Al_2O_3 : C$ , but the rate equations are identical with describing electron movements in quartz. Below is a step-by-step manual for involving these parameters in 'RLumModel' and re-calculating the simulation made by Pagonis et al. (2009). Note that in the original publication Figure 3 and Figure 6 are inconsistent with each other. For a dose rate of 0.1 Gy/s an initial RL intensity of ca.  $1.5 \times 10^{14}$  is obtained (see Figure 6 in original publication and simulations below).

### 2.1.1 Set parameters

First of all the model parameters had to be set. In 'RLumModel' this can be done via `list()`. The list has to contain the following items:

- N: Concentration of electron- and hole traps [ $cm^{-3}$ ]
- E: Electron/Hole trap depth [eV]
- s: Frequency factor [ $s^{-1}$ ]
- A: Conduction band to electron trap and valence band to hole trap transition probability [ $s^{-1} \cdot cm^3$ ].
- B: Conduction band to hole centre transition probability [ $s^{-1} \cdot cm^3$ ].
- Th: Photo-eviction constant or photoionisation cross section, respectively. If not set: by default 0.
- E\_th: Thermal assistance energy [eV]. If not set: by default 0.
- k\_B: Boltzman constant  $8.617 \times 10^{-5}$  [eV/K]. If not set: by default  $8.617 \times 10^{-5}$ .
- W: activation energy 0.64 [eV] (for UV). If not set: by default 0.64.
- K:  $2.8 \times 10^7$  (dimensionless constant). If not set: by default  $2.8 \times 10^7$ .
- model: "customized"
- R (optional): Ionisation rate (pair production rate) equivalent to 1 Gy/s [ $s^{-1} \cdot cm^{-3}$ ]

**Note:**

- Not every publication uses the same definition of parameter A and B.
- When no thermal quenching is expected, set  $K = 0$ . A numerical value of W is necessary, if  $K \neq 0$ .
- The parameter `model = "customized"` is necessary to not load a stored model within the RLumModel package.
- The luminescence center, which is responsible for the luminescence production (TL-, OSL-, RF-signal), has to be the last entry in the model parameters, see examples below.
- For further details of the parameter see Bailey (2001) and Wintle (1975).
- The first two entries in N, A and B belong to the electron traps and the last two entries to the hole centres. This order is necessary.
- The first entry in N, E, s, A, B, Th and E\_th belong to the first energy level (e.g.  $110^\circ C$ ), the second, third, ... entries to second, third, ... energy levels.
- The entries Th and E\_th are not necessary, because when they miss in the definition of the own parameters they will automatically be set to 0.

```
own_parameters <- list(  
  N = c(2e15, 2e15, 2.4e16, 1e17),  
  E = c(0, 0, 0, 0),  
  s = c(0, 0, 0, 0),  
  A = c(2e-8, 2e-9, 4e-9, 1e-8),  
  B = c(0, 0, 5e-11, 4e-8),  
  Th = c(0, 0),  
  E_th = c(0, 0),  
  K = 0,
```

```
model = "customized",
R = 1.7e15)
```

It is important to notice, that in Pagonis et al. (2009)  $B$  is the valence band to hole centre probability, but in Bailey (2001) this is  $A_j$ . The default setting of `RLumModel` is the definition by Bailey (2001) and so the values of  $B$  (in Pagonis et al. (2009)) are  $A$  in the notation above.

As a next step it is possible to set own starting-parameters, also called state parameters. In the case of Pagonis et al. (2009) they submitted initial concentrations of electrons and holes. This can be done via:

```
own_state_parameters <- c(0, 0, 0, 9.4e15)
```

Here the first entry is the first electron trap, the second entry the second electron trap, the third entry the hole center and the fourth entry the luminescence center responsible for the RF signal. The vector `own_state_parameters` needs as much entries as energy levels used in the model.

### 2.1.2 Running the simulation with `RLumModel`

When all parameters are set, the simulation can be started. The main function in `RLumModel` is `model_LuminescenceSignals()` and the usage with own parameter sets is described below. For a general overview for creating a sequence, running `RLumModel` with stored models etc. the user is referred to Friedrich et al. (2016) and to the vignette ‘`RLumModel` - Getting started with `RLumModel`’.

For simulating the results of Pagonis et al. (2009) the following sequence is needed.

```
sequence <- list(RF = c(20, 0.1, 0.1))
```

This sequence describes a radiofluorescence simulation at 20 °C with a dose of 0.1 Gy and a dose rate of 0.1 Gy/s, so the stimulation time is 1s.

The parameters `own_parameters` and `own_state_parameters` in `model_LuminescenceSignals()` are prepared for using own created parameter sets. Parameter `model = "customized"` is necessary to not load stored parameters.

```
RF_Pagonis2009 <- model_LuminescenceSignals(
  model = "customized",
  sequence = sequence,
  own_parameters = own_parameters,
  own_state_parameters = own_state_parameters,
  verbose = FALSE)
```

As in the original publication, initially the RF signal increases and is followed by an approximately linear region until the stimulation ends. Figure 5 in Pagonis et al. (2009) shows the concentration of the luminescence center  $m_1$  for the stimulation time of 1s. With `RLumModel` this can be plotted very fast with the following command (for a detailed description see vignette ‘`RLumModel` - Getting started with `RLumModel`’).

### 2.1.3 Re-calculate the original results

Reproducing Figure 3 and Figure 6 in Pagonis et al. (2009) a loop over different dose rates is necessary. The following code lines are able to run the model for five different dose rates from 0.1 to 0.5 Gy/s and plot all contained RF curves and the initial RF signal. For a more detailed description of the loop and the single commands therein the user is referred to Friedrich et al. (2016) and the vignette ‘`RLumModel` - Getting started with `RLumModel`’.

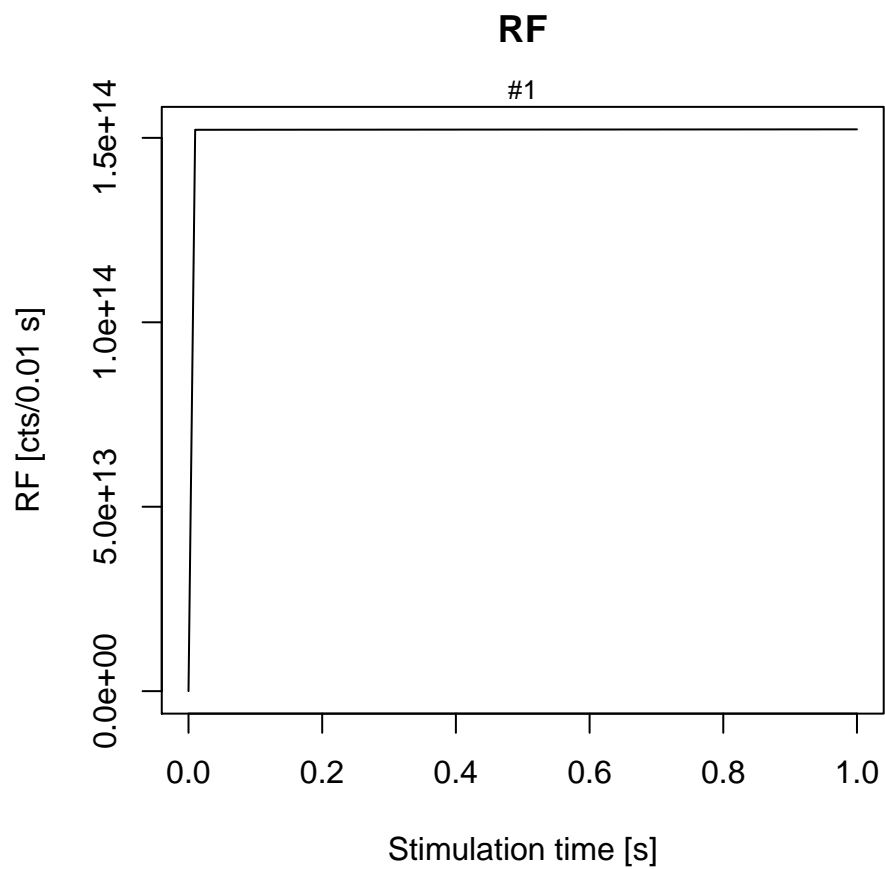


Figure 1: RF signal for 0.1 Gy/s

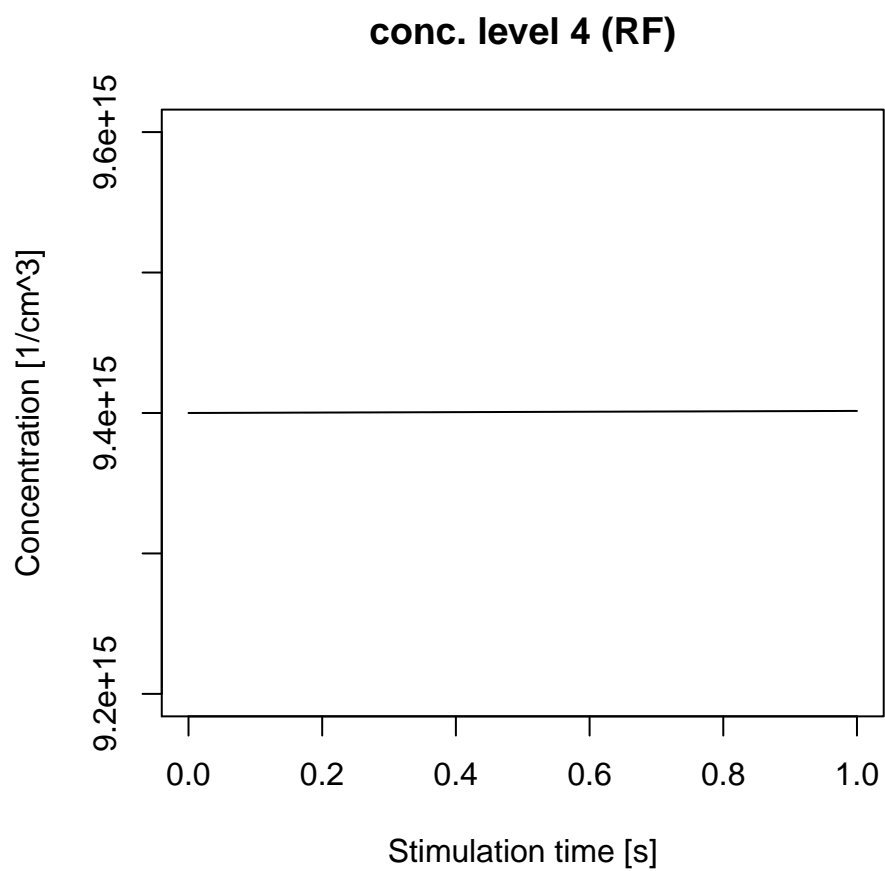


Figure 2: Concentration of m1 during RF

```

dose.rate <- seq(from = 0.1, to = 0.5, by = 0.1)

model.output <- lapply(dose.rate, function(x) {

  sequence <- list(RF = c(20, x, x))

  RF_data <- model_LuminescenceSignals(
    model = "customized",
    sequence = sequence,
    own_parameters = own_parameters,
    own_state_parameters = own_state_parameters,
    verbose = FALSE,
    plot = FALSE
  )

  ## "RF$" for exact matching RF and not (RF)
  return(get_RLum(RF_data, recordType = "RF$", drop = FALSE))

})

model.output.merged <- merge_RLum(model.output)

plot_RLum(
  object = model.output.merged,
  xlab = "Stimulation time [s]",
  ylab = "RF signal [a.u.]",
  legend.text = paste(dose.rate, "Gy/s"),
  legend.pos = "outside",
  combine = TRUE)

```

The following code calculates the initial RF signal for the five different dose rates.

```

dose.rate <- seq(from = 0.1, to = 0.5, by = 0.1)

model.output <- vapply(X = dose.rate, FUN = function(x) {

  sequence <- list(RF = c(20, x, x))

  temp <- model_LuminescenceSignals(
    model = "customized",
    sequence = sequence,
    own_parameters = own_parameters,
    own_state_parameters = own_state_parameters,
    verbose = FALSE,
    plot = FALSE
  )

  ## "RF$" for exact matching RF and not (RF)
  RF_curve <- get_RLum(temp, recordType = "RF$")

  return(max(get_RLum(RF_curve)[2,2]))

}, FUN.VALUE = 1)

```

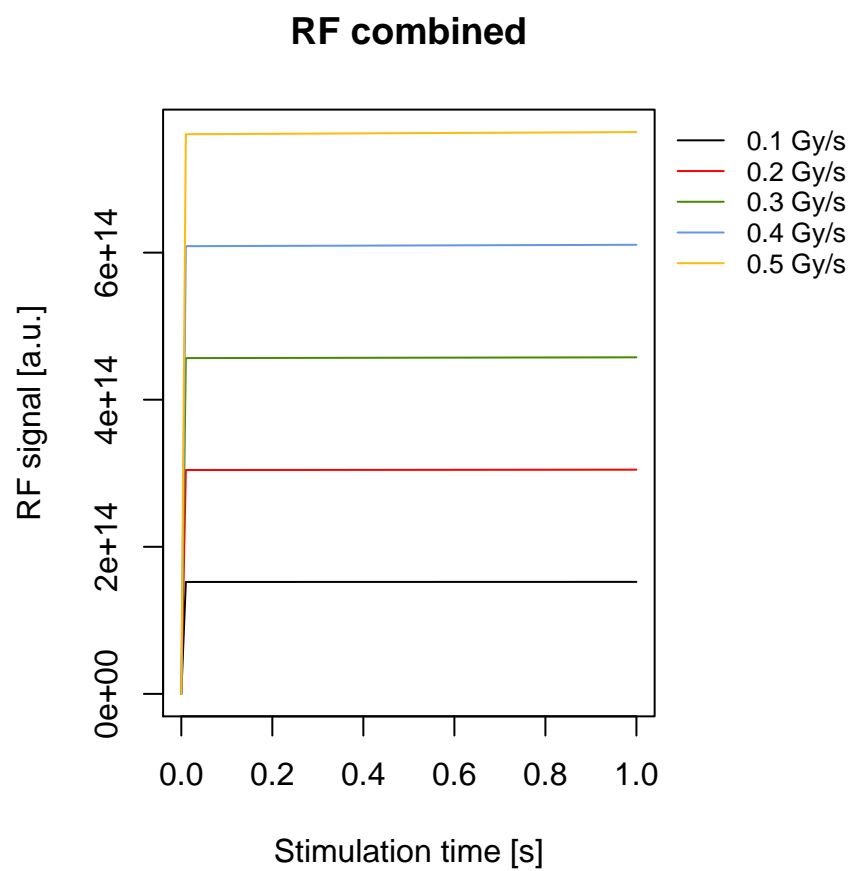


Figure 3: RF signals for different dose rates

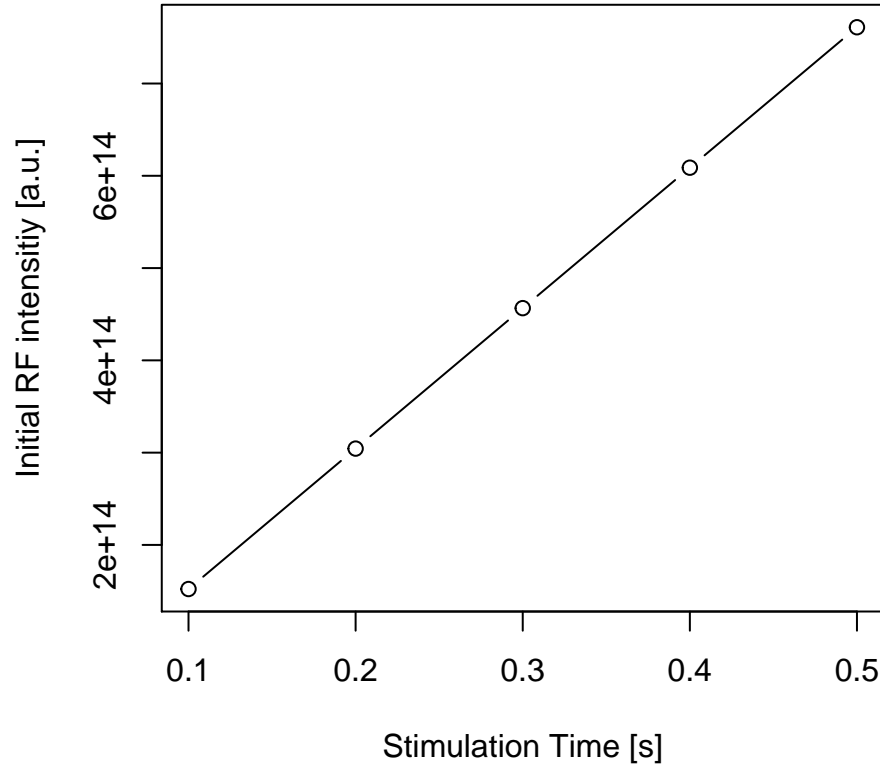


Figure 4: Initial RF signal for different dose rates with parameters of Lawless 2009

The results show that ‘RLumModel’ is able to simulate the same results as published in Pagonis et al. (2009) with only little effort. All these examples can be modified to own needs, e.g. own sequences or own parameters.

## 2.2 Lawless 2009

Lawless et al. (2009) investigated the sublinear dose dependence of TL and OSL. They published a set of model parameters to simulate the behaviour of the quartz luminescence system during different dose rates. In contrast to the example above, this simulation has no state parameters and so they were not defined.

### 2.2.1 Set parameters and recalculate the results

All used parameters are defined in the named list `own_parameters`.  $K=0$  was chosen, because no thermal quenching was simulated. Note: In the “Bailey 2001” notation  $B$  has the same meaning as  $Am$  in Lawless et al. (2009) (for details see example in chapter 2.1.1).

```
own_parameters <- list(
  N = c(1e14, 1e15),
  E = c(0, 0),
  s = c(0, 0),
  A = c(1e-13, 1e-14),
  B = c(0, 1e-7),
  K = 0,
  model = "customized",
  R = 1e8)
```



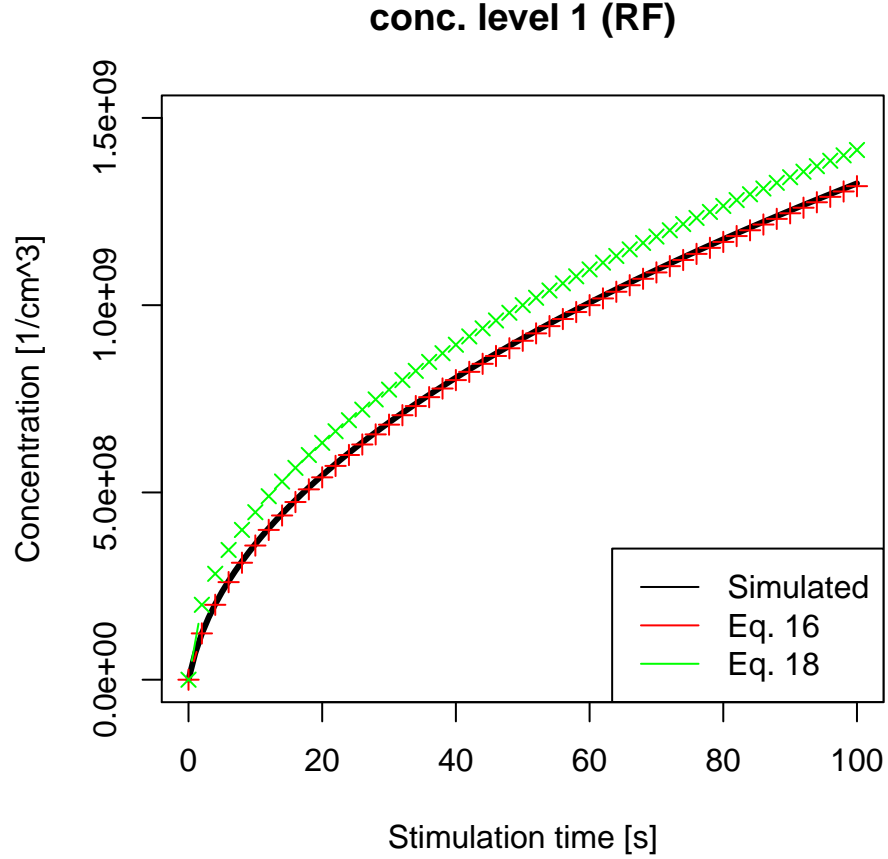


Figure 5: Concentration of Level 1 with numerical and analytical solutions

```
sequence <- list(RF = c(20, 100, 1))

RF_Lawless_2009 <- model_LuminescenceSignals(
  model = "customized",
  sequence = sequence,
  own_parameters = own_parameters,
  verbose = FALSE,
  plot = FALSE)

concentration_n <- Luminescence::get_RLum(
  RF_Lawless_2009,
  recordType = c("conc. level 1"))
```

This code leads to the following results and shows the same as plotted in Lawless et al. (2009), Fig. 2 (plot commands not shown here). More details to the equations mentioned in the legend are available in the original publication.

## 2.3 Chen 2013

Chen and Pagonis (2013) published a numerical model to investigate the quasi-equilibrium assumptions in TL. For the description of the system a OTOR model was used.

### 2.3.1 Set parameters

This model is the first in this vignette which did not start its simulation at 20 °C. For this cases, `model_LuminescenceSignals()` offers a parameter called `own_start_temperature`. This parameter offers maximal flexibility for the user to set the initial temperature of the simulation. The parameter takes effect when `model = "customized"` is used, see example below.

```
own_parameters <- list(  
  N = c(1e9, 0),  
  E = c(0.4, 0),  
  s = c(1e11, 0),  
  A = c(1e-9, 0),  
  B = c(0, 1e-10),  
  Th = c(0, 0),  
  E_th = c(0, 0),  
  K = 0,  
  model = "customized")  
  
own_state_parameters <- c(1e8, 1e8)  
  
own_start_temperature <- -220  
  
sequence <- list(TL = c(-220, 130, 1))
```

### 2.3.2 Re-calculate the original results

Here the parameter `own_start_temperature` from the function `model_LuminescenceSignals()` is used to set the beginning of the measurement to -220°C. It is important, that 'RLumModel' always uses temperatures in °C.

```
TL_Chen2013 <- model_LuminescenceSignals(  
  model = "customized",  
  sequence = sequence,  
  own_parameters = own_parameters,  
  own_state_parameters = own_state_parameters,  
  own_start_temperature = own_start_temperature,  
  verbose = FALSE,  
  plot = TRUE)
```

With this result it is possible to plot the concentration of every single energy level, leading to the following plot (see also Fig. 6 in Chen and Pagonis (2013))

### 2.3.3 Sensitivity analysis with own parameters

Sensitivity analysis and inverse modelling with model parameters is one of the big advantages of 'RLumModel'. A manual how to perform a sensitivity analysis with 'RLumModel' can be found in the vignette 'RLumModel - Fitting model parameters to experimental data'.

The main function to use this feature is `fit_RLumModel2data()`. As in `model_LuminescenceSignals()`, `sequence`, `model` and `own` parameters had to be the parameters. Additional to that, the (required) parameter `seq.step2fit()` is used to choose the sequence step in the defined sequence, which will be analysed. In the example from Chen and Pagonis (2013) only the sequence step 1 is available and a analysis from this step is

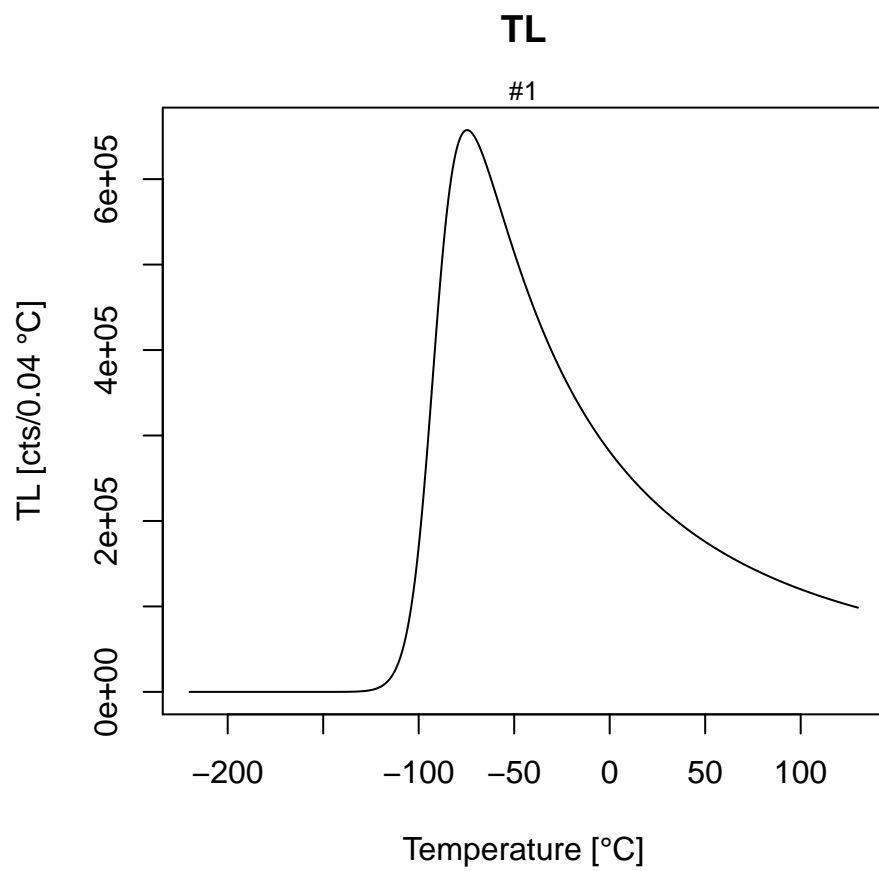


Figure 6: TL with parameter sets of Chen 2013

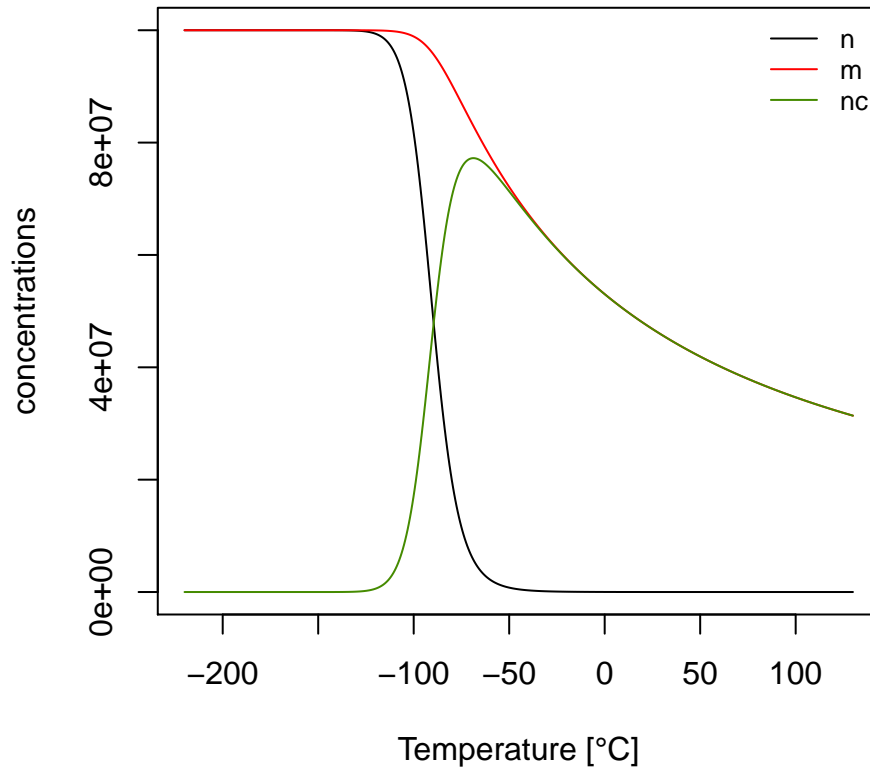


Figure 7: Concentrations of different energy levels

expected. Another (optional) parameter is ‘norm’: If true all results will be normalized to the highest value. This is recommended in fitting experimental data to model parameters, but not when a sensitivity analysis is performed.

To execute a sensitivity analysis with your own parameters, the function `extract_pars2FME()` is necessary to prepare the parameters for the usage with ‘RLumModel’.

The function `extract_pars2FME()` has two possible arguments: `parms` and `model`. If a own parameter set is used, the parameter to chose is `parms`. This is necessary to bring the parameters in the correct form for using the **R**-package FME.

```
func_FME <- fit_RLumModel2data(
  sequence = sequence,
  model = "customized",
  seq.step2fit = 1,
  own_parameters = own_parameters,
  own_state_parameters = own_state_parameters,
  own_start_temperature = own_start_temperature,
  norm = FALSE)

parms <- extract_pars2FME(parms = own_parameters)
```

Then, as described in the vignette ‘RLumModel - Fitting model parameters to experimental data’, different analysis and fitting methods can be performend on the parameter set.

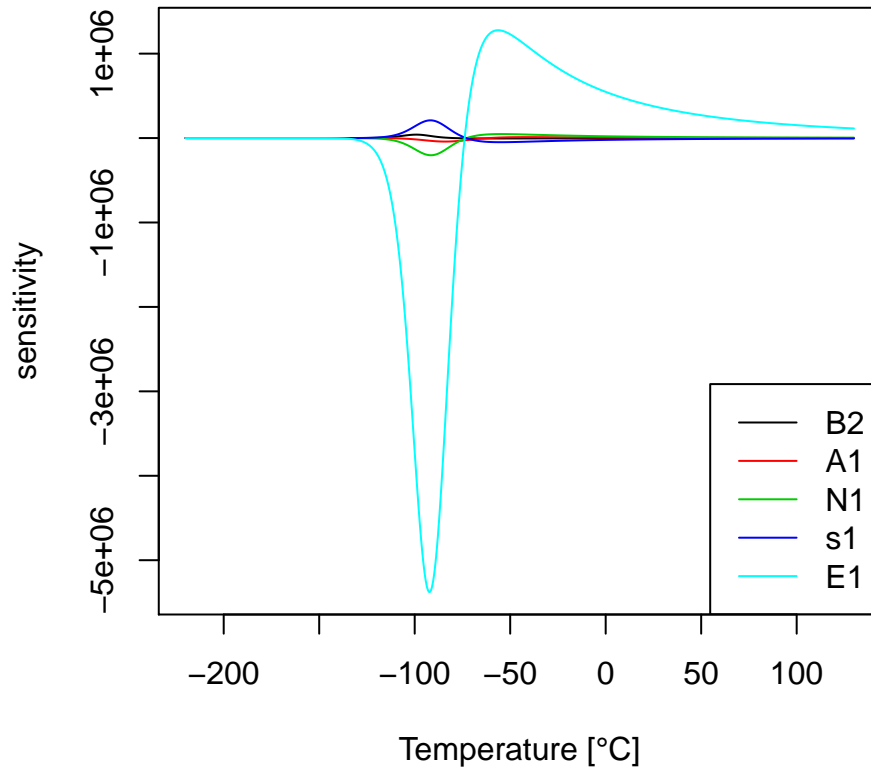


Figure 8: Local sensitivity analysis with own parameter sets

```
SensR <- FME::sensFun(func = func_FME,
  parms = parms,
  senspar = c("B2", "A1", "N1", "s1", "E1"),
  varscale = 1
)

plot(SensR,
  legpos = "bottomright",
  xlab = "Temperature [\u00B0C]",
  main = "")
```

### 3 Summary

This vignette showed the potential of the R package ‘RLumModel’ to use own parameter sets simulating quartz luminescence behaviour. Quartz as well as  $Al_2O_3$  luminescence phenomena can be numerically described and graphically plotted. Another feature is the possibility to perform sensitivity analysis and inverse modelling with own data sets.

## References

- Bailey, R., 2001. Towards a general kinetic model for optically and thermally stimulated luminescence of quartz. *Radiation Measurements* 33, 17–45.
- Chen, R., Pagonis, V., 2013. On the quasi-equilibrium assumptions in the theory of thermoluminescence (TL). *Journal of Luminescence* 143, 734–740.
- Eddelbuettel, D., 2013. *Seamless R and C++ integration with Rcpp*. Springer.
- Friedrich, J., Kreutzer, S., Schmidt, C., 2016. Solving ordinary differential equations to understand luminescence: 'RLumModel', an advanced research tool for simulating luminescence in quartz using R. *Quaternary Geochronology*.
- Lawless, J., Chen, R., Pagonis, V., 2009. Sublinear dose dependence of thermoluminescence and optically stimulated luminescence prior to the approach to saturation level. *Radiation Measurements* 44, 606–610.
- Pagonis, V., Lawless, J., Chen, R., Andersen, C., 2009. Radioluminescence in Al<sub>2</sub>O<sub>3</sub>:C – analytical and numerical simulation results. *Journal of Physics D: Applied Physics* 42, 175107.
- Wintle, A., 1975. Thermal Quenching of Thermoluminescence in Quartz. *Geophysical Journal International* 41, 107–113.