Chapter 2
ANALYSIS AND MODELING OF TL DATA

Abstract In this chapter we provide detailed R codes which show how researchers can analyze and model their experimental TL data. We provide R codes for the initial rise method and the method of various heating rates, which allow evaluation of both the activation energy E and the frequency factor s. We present R codes for numerically integrating the simple one trap one recombination model (OTOR), as well as for numerically integrating the equations for first, second and general order kinetics using R. We discuss the general one trap (GOT) differential equation and its analytical solution, which is based on the Lambert W function. Several examples are given for using computerized glow curve deconvolution analysis (CGCD) for single-peak and multiple-peak TL glow curves, based on the R-packages *tgcd* and the Lambert W function. Specific examples are given of using the new R package *RLumCarlo*, to simulate TL glow curves with different kinetic parameters. The chapter concludes with a list of recommended experimental protocols, which experimentalists can apply when studying TL signals.

Code 2.1: System of differential equations for OTOR

```
# Solution of the system of ODE's for the OTOR model
rm(list=ls())
library(deSolve)
TLOTOR <- function(t, x, parms) {
  with(as.list(c(parms, x)), {
    dn1 <- - n1*s*exp(-E1/(kb*(273+hr*t)))+ nc*An*(N1-n1)
```

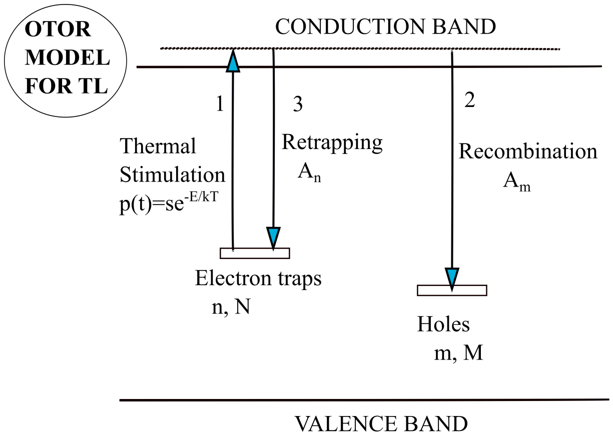

Fig. 2.1: Schematic diagram of the OTOR model for a TL process.

```
    # n1=concentration of trapped electrons
    dnc <-  n1*s*exp(-E1/(kb*(273+hr*t)))-nc*An*(N1-n1)-m*Am*nc
    # nc=concentration of conduction band electrons
    dm <- -m*Am*nc
    # m=concentration of recombination centers
    res <- c(dn1, dnc, dm)
    list(res)
  })}
## Parameters
hr<-1   # heating rate in K/s
parms  <- c(E1 =1, s=10^12, kb=8.617*10^-5, hr=hr,
            An = 10^-7,  N1 = 10^10,Am = 10^-7)
## vector of timesteps
times  <- seq(0, 250)
temps<-times*hr
## Initial conditions for the system
y <- xstart <- c(n1 = 10^9, nc = 0, m = 10^9)
## Solve system of differential equations
out <-  lsoda(xstart, times, TLOTOR, parms)
## Plotting
par(mfrow=c(1,2))
plot(temps,out[,"n1"],xlab=expression("Temperature ["^"o"*"C]"),
ylab =expression("Filled traps n(T),cm"^-3*" "),ylim=c(0,1.2e9))
legend("left",bty="n",expression("OTOR","n(T)"))
legend("topleft",bty="n", expression("(a)"))
plot(temps,out[,"m"]*parms["Am"]*out[,"nc"],pch=1,col="red",
xlab=expression("Temperature ["^"o"*"C]"),
ylim=c(0,2.8e7),ylab="TL [a.u.] and nc (t)",xlim=c(0,250))
lines(temps,10*out[,"nc"],xlab=expression("Temperature [
"^"o"*"C]"),ylab="nc(t)",typ="o",pch=2,col="blue")
legend("topright",bty="n", expression("(b)"))
legend("topleft",bty="n", pch=c(1,2),expression("TL","ncx10"),
col=c("red","blue"),lwd=1)
```

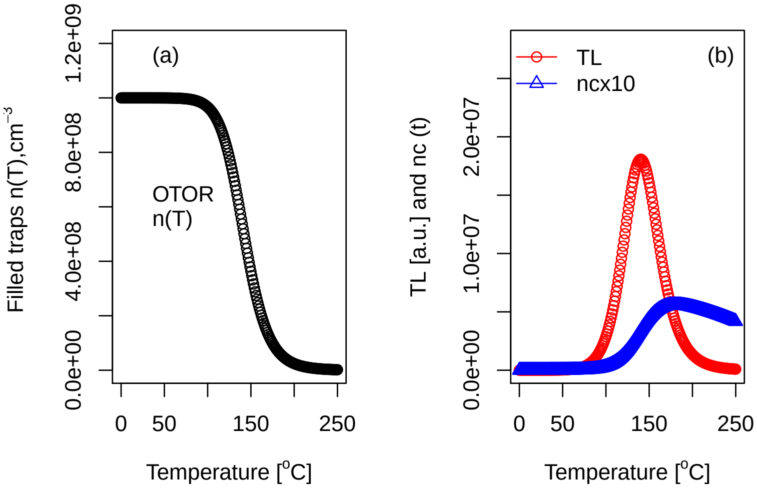

Fig. 2.2: Numerical solution of the system of differential equations for the OTOR model. (a) The concentrations of trapped electrons $n(T)$ and (b) of the conduction band electrons $n_c(T)$ (triangles), and luminescence intensity $I(T)$ (circles), during a typical TL experiment. Note that $n_c(T)$ has been multiplied by a factor of 10, for display purposes.

Code 2.2: ODE for TL: First order kinetics

```
# Numerically solve the ODE for TL: first order kinetics
rm(list=ls())
library(package ="deSolve")
```

```
# Define Parameters
k_B <- 8.617e-5  # Boltzmann constant
E <- 1 # electron trap depth [eV]
s <- 1e12 # frequency factor [1/s]
delta.t <- 1
t <- seq(0, 200, delta.t)
n.0 <- 1e10
N.traps <- 1e11
ODE <- function(t, state, parmameters){
        with(as.list(c(state, parameters)),{
        dn <-  -s*exp(-E/(k_B*(273+t))) * n
    list(c(dn)) })}
parameters <- c(N.traps = N.traps, s = s, E = E, k_B = k_B)
state <- c(n = n.0)
num_ODE <- ode(y = state, times = t, func = ODE,
parms = parameters)
# Plot remaining electrons and TL as a function of temperature
par(mfrow=c(1,2))
plot(x =num_ODE[,1],
y = num_ODE[,2], xlab=expression("Temperature ["^"o"*"C]"),
 ylab = expression("Filled traps n(T), cm"^-3*" "),
col = "red")
legend("topright",bty="n",c("(a)","n(T)"))
plot(num_ODE[-1,1], y = abs(diff(num_ODE[,2])),
xlab=expression("Temperature ["^"o"*"C]"),
ylab = "TL signal [a.u.]",xlim=c(0,170))
legend("topleft",bty="n",c("(b)","TL"))
```

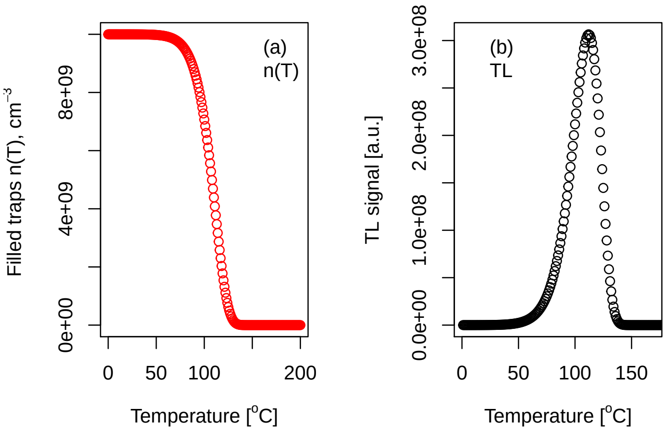

Fig. 2.3: Numerical solution of the Randall and Wilkins Eq.(**??**), for first order kinetics TL. (a) Remaining trapped electrons $n(T)$, and (b) The corresponding TL signal.

Code 2.3: First-order TL by varying the initial trap concentrations (tgcd)

```
# Simulate first-order glow peaks with various
# initial electron trap concentrations (n0).
rm(list=ls())
# library(tgcd)
library(package ="tgcd")
temps <- seq(300, 440, by=1)
peak1 <- simPeak(temps,n0=0.2e10,Nn=1e10,ff=1e12, ae=1.0, hr=1,
                 typ="f",plot=FALSE)
peak2 <- simPeak(temps,n0=0.4e10,Nn=1e10,ff=1e12, ae=1.0, hr=1,
                 typ="f",plot=FALSE)
peak3 <- simPeak(temps,n0=0.6e10,Nn=1e10,ff=1e12, ae=1.0, hr=1,
                 typ="f",plot=FALSE)
peaks<-cbind(peak1$tl, peak2$tl, peak3$tl)
matplot(temps, peaks, type="o",  pch=c(0,1,2),
        col=c("black","red","blue"),lwd=2,
        xlab="Temperature (K)", ylab="TL intensity [a.u.]")
legend("topleft",bty="n", pch=c(NA,NA,NA,0,1,2),
       c(expression('First order peaks',' ','n'[o]*'/N'),
"0.2","0.4","0.6"),col=c(NA,NA,NA,"black","red","blue"))
```

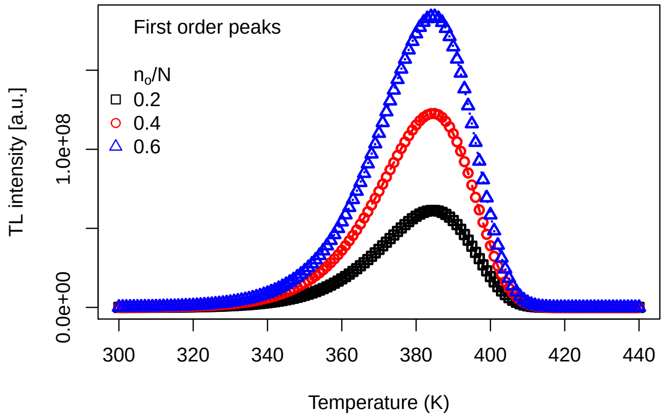

Fig. 2.4: Simulation of first-order TL glow peaks with three different initial electron trap concentrations $n_0 = 0.2\,, 0.4,\ 0.6 \times 10^{10}\ cm^{-3}$. The location of the maximum TL intensity does not shift significantly as n_0 changes, and the peak height is proportional to n_0.

Code 2.4: Second-order TL by varying the initial trap concentrations (tgcd)

```
# Simulate second-order glow peaks with various
# initial electron trap concentrations (n0).
rm(list=ls())
library(tgcd)
temps <- seq(300, 500, by=1)
peak1 <- simPeak(temps,n0=0.2e10,Nn=1e10,ff=1e12, ae=1.0, hr=1,
typ="s",plot=FALSE)
peak2 <- simPeak(temps,n0=0.4e10,Nn=1e10,ff=1e12, ae=1.0, hr=1,
typ="s",plot=FALSE)
peak3 <- simPeak(temps,n0=0.6e10,Nn=1e10,ff=1e12, ae=1.0, hr=1,
typ="s",plot=FALSE)
peaks<-cbind(peak1$tl, peak2$tl, peak3$tl)
matplot(temps, peaks, type="o",  pch=c(0,1,2),
        col=c("black","red","blue"),lwd=2,
        xlab="Temperature (K)", ylab="TL intensity [a.u.]")
```

```
legend("topright",bty="n", pch=c(NA,NA,NA,0,1,2),
       c(expression('Second order peaks',' ','n'[0]*'/N'),
"0.2","0.4","0.6"),col=c(NA,NA,NA,"black","red","blue"))
```

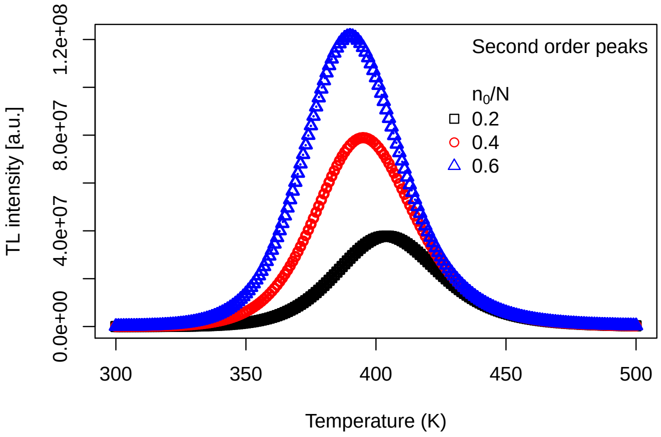

Fig. 2.5: Simulation of second order TL glow peaks with three different initial electron trap concentrations $n_0 = 0.2\,, 0.4,\ 0.6 \times 10^{10}\ cm^{-3}$. The location of the maximum TL intensity shifts with n_0, and the *area* under the peak is proportional to n_0.

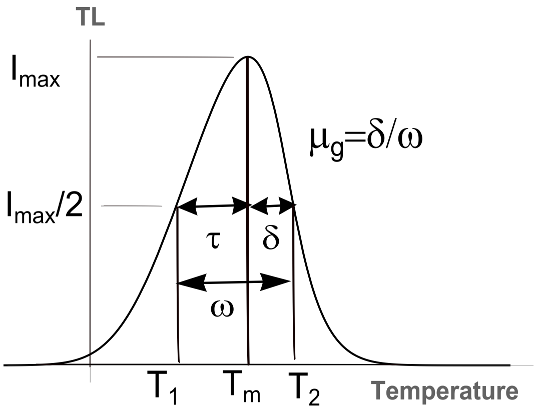

Fig. 2.6: Schematic diagram of TL peak. The half-widths τ, δ, ω define the geometrical shape factor $\mu_g = \delta/\omega$. First-order peaks have an asymmetric form, while second-order peaks are very nearly symmetric.

Code 2.5: 1st and 2nd order TL with the same parameters

```
# Simulate a first and a second  order TL glow peak
# with the same kinetic parameters
rm(list=ls())
library(package ="tgcd")
temps <- seq(300, 500, by=.2)
peak1 <- simPeak(temps, n0=1e10, Nn=1e10, ff=1e12, ae=1, hr=1,
                 typ="f",plot=FALSE)
peak2 <- simPeak(temps, n0=1e10, Nn=1e10,  ff=1e12, ae=1, hr=1,
                 typ="s",plot=FALSE)
n<-cbind(peak1$n, peak2$n)
par(mfrow=c(1,2))
matplot(temps, n, type="l", lwd=3,lty=c(1,2),
xlab="Temperature (K)",
ylab=expression("Filled traps (cm"^"-3"*")"))
legend("topright",bty="n", expression("(a)"))
legend("right",bty="n", lty=c(1,2),expression("b=1","b=2"),
       col=c("black","red"),lwd=2)
peaks<-cbind(peak1$tl, peak2$tl)
matplot(temps, peaks, type="l", lwd=3, lty=c(1,2),
xlab="Temperature (K)",ylab="TL intensity [a.u.])")
legend("right",bty="n", lty=c(1,2),expression("b=1","b=2"),
       col=c("black","red"),lwd=2)
legend("topright",bty="n", expression("(b)"))

print.noquote("Parameters for first order TL peak")
print(c(peak1$sp[1],peak1$sp[2],peak1$sp[3]))
print(c(peak1$sp[4],peak1$sp[5],peak1$sp[6]))
cat("\nShape factor=",round(peak1$sp[7],3))

print.noquote("Parameters for second order TL peak")
print(c(peak2$sp[1],peak2$sp[2],peak2$sp[3]))
print(c(peak2$sp[4],peak2$sp[5],peak2$sp[6]))
cat("\nShape factor=",round(peak2$sp[7],3))
```

```
## [1] Parameters for first order TL peak
##       T1       T2       Tm
## 367.2004 397.0405 384.6000
##       d1       d2      thw
## 17.39956 12.44053 29.84009
##
## Shape factor= 0.417[1] Parameters for second order TL peak
##        T1        T2        Tm
## 363.5701 405.6734 383.8000
##        d1        d2       thw
## 20.22992 21.87338 42.10330
##
## Shape factor= 0.52
```

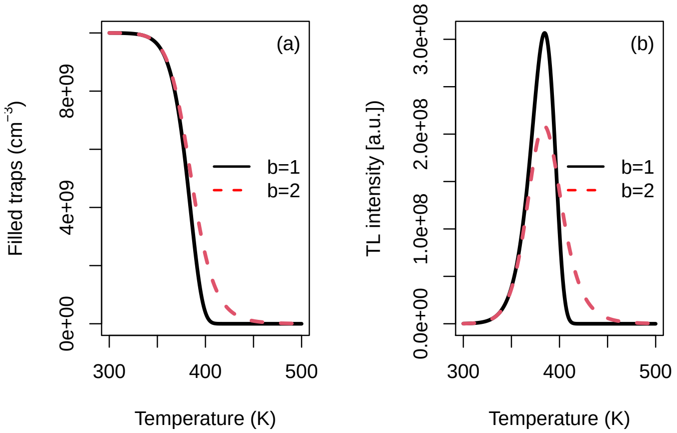

Fig. 2.7: Comparison of simulated first and second order TL peaks (solid and dashed lines respectively), evaluated with the same kinetic parameters. The second order process is slower, and the peak shape for second order is almost symmetric.(a) Filled traps $n(T)$ (b) The corresponding TL signal.

Code 2.6: The initial rise method: find energy E from TL data

```
# Apply the initial rise method to find the activation energy E
# Load the data from txt file, which  contains pairs of
# data  in the form: (Temperature_in_C,TL_Intensity (any units)
rm(list=ls())
library(tgcd)
data("Kitis")
```

```
x<-Kitis$x001[,1]
y<-Kitis$x001[,2]
mydata<-data.frame(x,y)
kB<-8.617*1e-5 # Boltzmann constant in eV/K
initialPos<-270  #analyze data points from #270 to #320
finalPos<-320
x<- mydata[,1][initialPos:finalPos]
y<- mydata[,2][initialPos:finalPos]
rangeData<-cbind(x,y)
y<-log(y)
x<-1/(kB*x)
bestfit<-lm(y~x)
summary(bestfit)
coefficients(bestfit)
par(mfrow=c(1,2))
plot(mydata,col="blue",
xlab=expression("Temperature [K]"),ylab = "TL (a.u.)")
lines(rangeData,col="red",lwd = 3)
legend("topright",bty="n","(a)")
plot(x, y, xlab = "1/(kT)  [1/eV]",ylab = "ln(TL)")
abline(lm(y~x))
legend("topright",bty="n",c("(b)"," ","Initial","Rise"))
```

```
  ##
  ## Call:
  ## lm(formula = y ~ x)
  ##
  ## Residuals:
  ##       Min        1Q    Median        3Q       Max
  ## -0.158165 -0.030093  0.001636  0.027663  0.172387
  ##
  ## Coefficients:
  ##             Estimate Std. Error t value Pr(>|t|)
  ## (Intercept) 39.97106    0.44678   89.46   <2e-16 ***
  ## x           -1.23111    0.01616  -76.17   <2e-16 ***
  ## ---
  ## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
  ##
  ## Residual standard error: 0.05592 on 49 degrees of freedom
  ## Multiple R-squared:  0.9916,Adjusted R-squared:  0.9915
  ## F-statistic:  5802 on 1 and 49 DF,  p-value: < 2.2e-16
  ##
  ## (Intercept)           x
  ##   39.971063   -1.231108
```

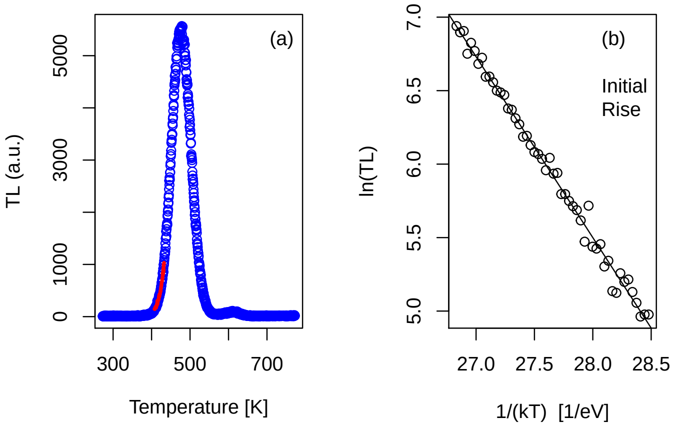

Fig. 2.8: Example of applying the initial rise method to find the activation energy E. (a) The red line indicates the initial rise area being analyzed; (b) The slope of the best line line gives the activation energy ($-E$).

Code 2.7: TL glow curve for four different heating rates

```
# Plot the same TL glow curve for four different heating rates
rm(list=ls())
library(tgcd)
temps <- seq(300, 440, by=1)  # temperature in K
hRates<-c(1,2,3,4)
## function to calculate TL for different hehating rates####
findTL<-function(hRate)
{peak<-simPeak(temps,n0=0.2e10,Nn=1e10,ff=1e12,ae=1.0, hr=hRate,
typ="f",plot=FALSE)
peak$tl}
### Calculate TL with different heating rates
TLs<-sapply(hRates,findTL)
matplot(temps,TLs,type="o", lty=c(1,1,1,1),lwd=1,  pch=1:4,
col=1:4,xlab="Temperature (K)", ylim=c(0,8e7),
ylab=expression(paste("TL/ ",beta," (counts/K)")))
legend("topright",bty="n","Heating rate method")
legend("topleft",bty="n", pch=c(1:4,NA),expression(
  "1 K/s","2","3","4"," "),col=c(1:4,NA),lwd=1)
```

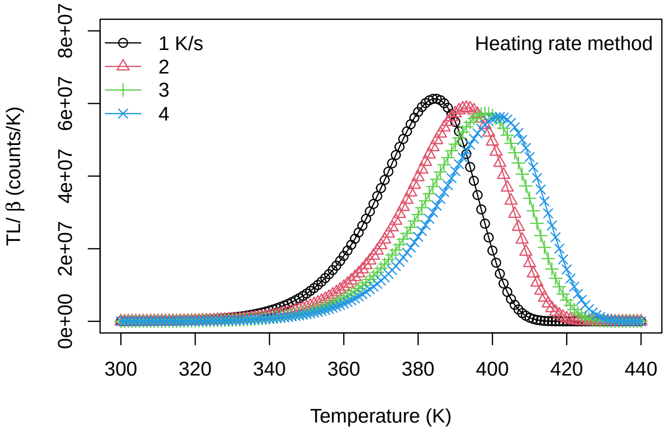

Fig. 2.9: Simulation of the same TL glow curve measured with four different heating rates $\beta = 1, 2, 3, 4$ K/s. As the heating rate increases, the TL peak shifts to the right towards higher temperatures, and the intensity decreases. The area under the curves stays the same. Notice that the y-scale is TL/β, i.e. in counts/K, and not in counts/s.

Code 2.8: Apply heating rate method to TL data, to find E,s

```
# Apply the heating rate method to find E,s
rm(list=ls())
library(tgcd)
library(scales)
kB<-8.617*1e-5 # Boltzmann constant in eV/K
temps <- seq(340, 420, by=2)
#### function to calculate TL for different heating rates####
findTL<-function(hRate)
{peak<-simPeak(temps,n0=0.2e10,Nn=1e10,ff=1e12,ae=1.0, hr=hRate,
typ="f",plot=FALSE)
peak$tl}
######## function to find Tmax ######
```

```
findTmax<-function(hRate)
{peak<-simPeak(temps,n0=0.2e10,Nn=1e10,ff=1e12,ae=1.0, hr=hRate,
typ="f",plot=FALSE)
temps[[match(max(peak$tl),peak$tl)]]}
########## calculate 1/kTmax and log(Tmax^2/beta)
hRates<-c(1,2,3,4)
maxTL<-sapply(hRates,findTmax)
TLs<-sapply(hRates,findTL)
####### plots
par(mfrow=c(1,2))
matplot(temps,TLs,type="o",lty="solid",lwd=1,pch=1:4,
 col=c(1:4),xlab="Temperature (K)",  ylim=c(0,8e7),
ylab=expression(paste("TL/ ",beta," (counts/K)")))
y<-log(maxTL^2/hRates)
x<-1/(kB*maxTL)
coefficients(lm(y~x))
energy<-coefficients(lm(y~x))[[2]]
intercept<-coefficients(lm(y~x))[[1]]
cat('\nFrequency factor s=',
    scientific(exp(-intercept)*energy/kB),' s^-1')
legend("topright",bty="n","(a)")
legend("topleft",bty="n", pch=c(1:4,NA),expression(
  "1 K/s","2","3","4"," "),col=c(1:4,NA),lwd=1)
plot(x,y,xlab="1/(kT)   (1/eV)",
ylab=expression(paste('ln(Tmax'^'2'*')/',beta)),ylim=c(10.4,12))
abline(lm(y ~ x))
legend("topleft",bty="n",legend=c("(b)","Heating rate",
"method"))
```

```
  ## (Intercept)           x
  ## -17.1459974   0.9619712
  ##
  ## Frequency factor s= 3.12e+11  s^-1
```

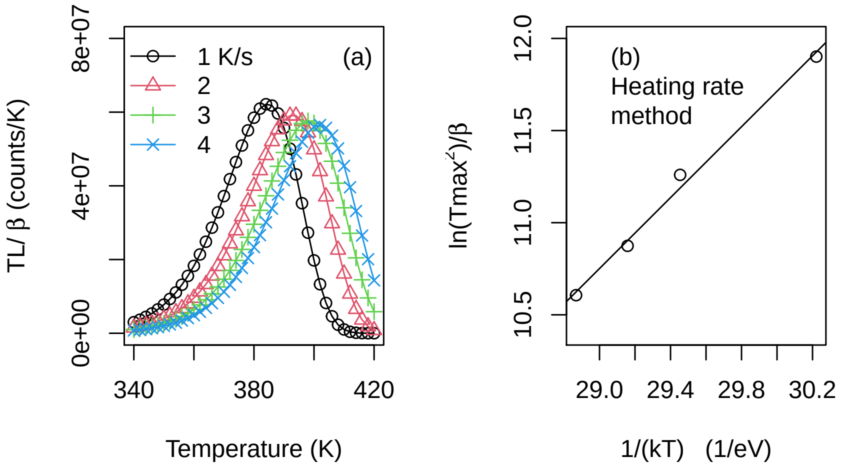

Fig. 2.10: Applying the heating rate method to obtain both the activation energy E and the frequency factor s. (a) The simulated TL glow curves (b) The slope and intercept of the best fit line yield both parameters (E, s).

Code 2.9: The GOT equation for TL in OTOR (deSolve)

```
# Numerical solution of the GOT equation for TL
rm(list=ls())
library("deSolve")
# Define Parameters
A_n <- 1e-10 # coefficient of retrapping
A_m <- 1e-8 # coefficient of recombination
k_B <- 8.617e-5  # Boltzmann constant
E <- 1 # electron trap depth [eV]
s <- 1e12 # frequenc factor [1/s]
delta.t <- 1
t <- seq(0, 300, delta.t)
# time = temperature, i.e. heating rate= 1 K/s
n.0 <- 1e10  # initial concentration of filled traps
N.traps <- 1e11   # total concentrations of available traps
# Calculate numerical ODE solution
```

```
ODE <- function(t, state, parmameters){
with(as.list(c(state, parameters)),{
dn <- -s*exp(-E/(k_B*(273+t))) * n^2 * A_m / ((N.traps - n) *
A_n + n * A_m)
list(c(dn))  })}
parameters <- c(N.traps = N.traps, s = s, E = E, k_B = k_B,
A_m = A_m, A_n = A_n)
state <- c(n = n.0)
num_ODE <- ode(y = state, times = t, func = ODE,
parms = parameters)
# Plot filled traps n(T) and TL as a function of temperature
par(mfrow=c(1,2))
plot(x = num_ODE[,1],
    y = num_ODE[,2],xlab=expression("Temperature ["^"o"*"C]"),
    ylab =expression("Filled traps (cm"^"-3"*")"),col = "red")
legend("topright",bty="n",expression('(a)',' ','GOT','n(T)'))
plot(x = num_ODE[-1,1],  y = abs(diff(num_ODE[,2])),
xlab=expression("Temperature ["^"o"*"C]"),
     ylab = "TL signal [a.u.]")
legend("topright",bty="n",c("(b)"," ","TL"))
```

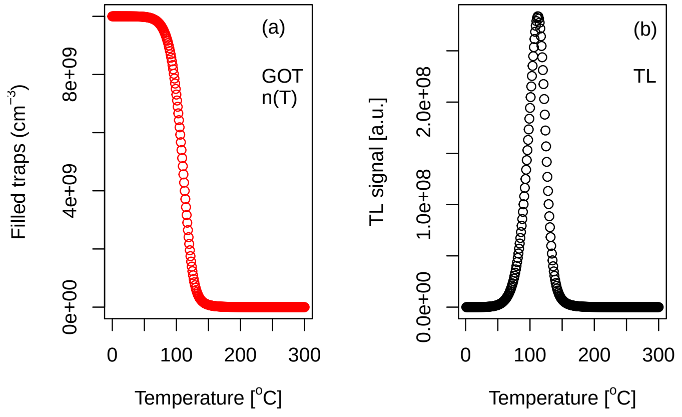

Fig. 2.11: Numerical solution of the GOT equation for TL. (a) Filled traps $n(T)$ (b) The TL glow curve.

Code 2.10: Plot the W0-Lambert solution of GOT equation

```r
rm(list=ls())
library(lamW)
##  Example of plot for Lambert W-function from x=0 to x=100
xs <- seq(0, 100, by=1)
ys <- lambertW0(xs)
## Plot the analytical solution of GOT, using Lambert W-function
x1<-300:450   # temperatures in K
kB<-8.617E-5
no<-1E8
N<-1E10
R<-.01
c<-(no/N)*(1-R)/R
En<-1
s<-1E12
beta<-1
k<-function(u) {integrate(function(p){exp(-En/(kB*p))},
                          300,u)[[1]]}
y1<-lapply(x1,k)
x<-unlist(x1)
y<-unlist(y1)
zTL<-(1/c)-log(c)+(s*no/(c*N*R))*y
# plots
par(mfrow=c(1,2))
plot(xs, ys, type="l", col="red", lwd=3, ylim=c(0,5),
     xlab="x", ylab="Lambert W0(x)")
legend("topleft",bty="n",c("(a)"," ","Lambert",
"function W0(x)"))
plot(x,(N*R/((1-R)^2))*s*exp(-En/(kB*x))/(lambertW0(exp(zTL))
+lambertW0(exp(zTL))^2),xlab="Temperature, K",col="blue",
     ylab="TL with W0(T)")
legend("topleft",bty="n",c("(b)"," ","TL", "using", "W0(x)"))
```

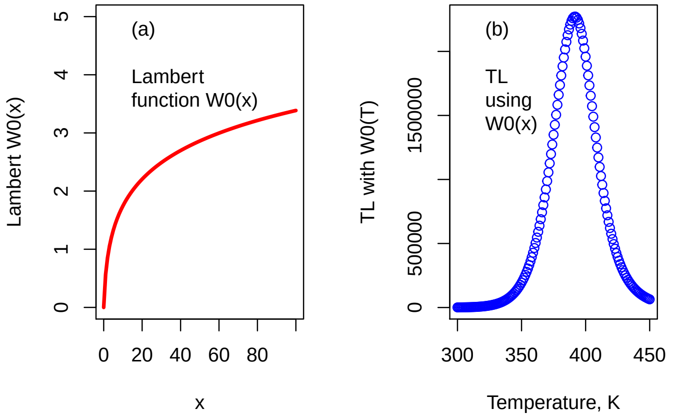

Fig. 2.12: (a) Plot of the Lambert function $W(x)$ between $x = 0$ and $x = 100$. (b) Plot of the analytical solution of the GOT Eq.(**??**), using the Lambert W function.

Code 2.11: Deconvolution of Glocanin glow curve (tgcd)

```
# Deconvolution of Reference glow curve #1 (project GLOCANIN)
rm(list=ls())
library("tgcd")
data(Refglow)   # Load the data
# Deconvolve data with 1 peak using the LAMBERT W function
startingPars <-
  cbind(c(15.0),c(1.0),c(520), c(0.1))   # Im, E, Tm, R
invisible(capture.output(TL1 <- tgcd(Refglow$x001, npeak=1,
model="lw",inisPAR=startingPars,nstart=10,edit.inis=FALSE)))
print.noquote("Best fit parameters")
TL1$pars
cat("\nGeometrical shape factor=",
round(TL1$sp[,7],2))
cat("\nFigure Of Merit FOM=",round(TL1$FOM,4))

  ## [1] Best fit parameters
  ##          INTENS(Im) ENERGY(E) TEMPER(Tm) rValue(r)
## 1th-Peak   10968.39   1.182267   490.4689      1e-16
##
## Geometrical shape factor= 0.43
## Figure Of Merit FOM= 0.0097
```

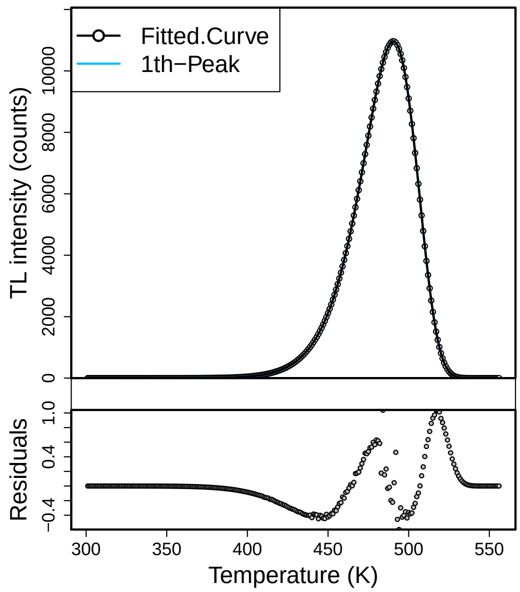

Fig. 2.13: Deconvolution of TL data containing a single peak, using the Lambert *W* function. The data is from Reference glow curve #1 in the intercomparison project GLOCANIN (Bos et al. [3]).

Code 2.12: Deconvolution of TL user data (.txt file, tgcd)

```
# Deconvolve data with 2 peaks using the LAMBERT W function
rm(list=ls())
library("tgcd")
# Load the data
mydata = read.table("lbodata.txt")
startingPars <-
```

```
cbind(c(105.0,5.0),c(1.1,1.4),c(460,550),c(0.01,.01)) #Im,E,Tm,R
invisible(capture.output(TL1 <- tgcd(mydata, npeak=2,
model="lw",inisPAR=startingPars, nstart=10, edit.inis=FALSE)))
print.noquote("Best fit parameters")
TL1$pars
cat("\nGeometrical shape factors"," ")
round(TL1$sp[,7],2)
```

```
## [1] Best fit parameters
##          INTENS(Im) ENERGY(E) TEMPER(Tm)    rValue(r)
## 1th-Peak  115.97882  1.209594   464.2374 5.133850e-11
## 2th-Peak   22.96337  1.158213   516.7809 2.233305e-01
##
## Geometrical shape factors  1th-Peak 2th-Peak
##     0.45     0.43
```

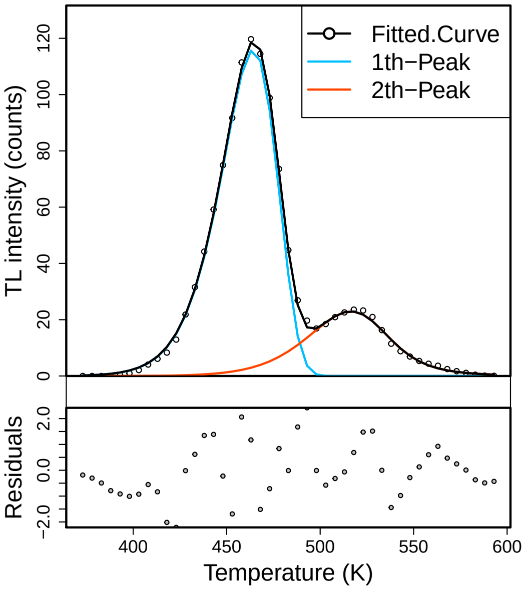

Fig. 2.14: Deconvolution of experimental data for dosimetric material LBO with 2 peaks (Kitis et al. [18]), by using the Lambert W function.

Code 2.13: Deconvolution of 9-peak Glocanin TL data (tgcd)

```r
# Deconvolve TL signal using 9 peaks (no background subtraction)
# a GOK model using user-supplied initial kinetic parameters.
rm(list=ls())
library("tgcd")
data(Refglow)
knPars <-
cbind(c(9824,21009,27792,50520,7153, 5496,6080,1641,2316), # Im
c(1.24, 1.36, 2.10, 2.65, 1.43, 1.16, 2.48, 2.98, 2.25), # E
c(387, 428, 462, 488, 493, 528, 559, 585, 602), # Tm
c(1.02, 1.15, 1.99, 1.20, 1.28, 1.19, 1.40, 1.01, 1.18)) # b
invisible(capture.output(TL1 <- tgcd(Refglow$x009, npeak=9,
model="g1",inisPAR=knPars, nstart=10, edit.inis=FALSE)))
print.noquote("Best fit parameters")
TL1$pars
```

```
## [1] Best fit parameters
##          INTENS(Im) ENERGY(E) TEMPER(Tm) bValue(b)
## 1th-Peak   9820.004  1.237294   387.3132  1.025077
## 2th-Peak  20959.877  1.361432   428.1756  1.149763
## 3th-Peak  27324.883  2.058686   462.1433  1.825707
## 4th-Peak  52194.441  2.498238   488.1280  1.119436
## 5th-Peak   7519.826  1.411680   495.8060  1.045013
## 6th-Peak   5351.507  1.474229   524.2461  1.388807
## 7th-Peak   7077.083  2.177145   557.5652  1.334783
## 8th-Peak   1547.388  3.343393   586.0483  1.057578
## 9th-Peak   2311.923  2.249152   602.6770  2.000000
```

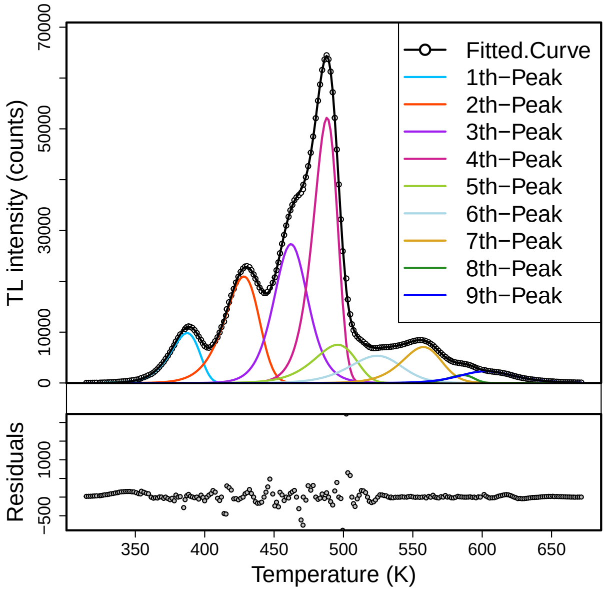

Fig. 2.15: Deconvolution of a glow curve from the GLOCANIN project using 9 peaks with a GOK model, and with user-supplied initial kinetic parameters (Bos et al. [3]).

```
# This is a continuation of R code above)
hist(TL1$residuals,lwd=3,xlab="Residuals",
ylab="Frequency",main=NULL)
legend("topright",bty="n",c("Fitting","Residuals"))
```

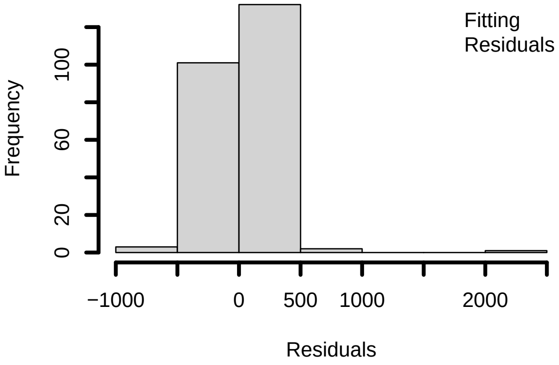

Fig. 2.16: Histogram of residuals $y_i^{expt} - y_i^{fit}$ from the best fit shown in Fig.2.15, for the nine-peak TL glow curve of the GLOCANIN project, with a GOK model.

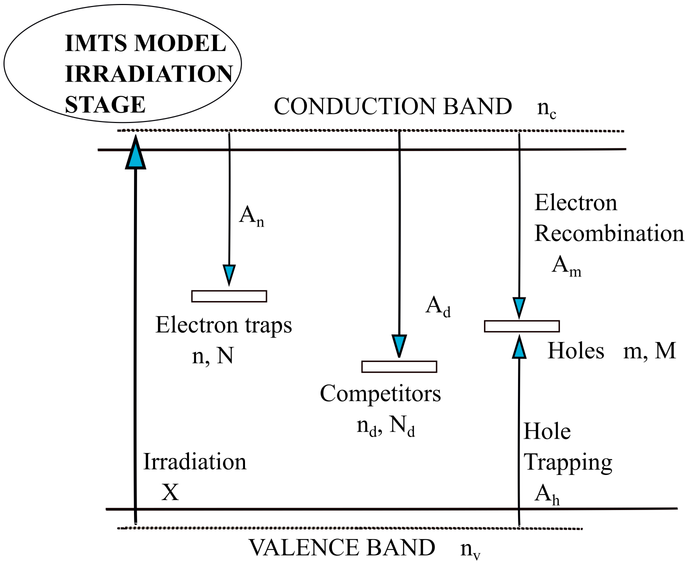

Fig. 2.17: The general phenomenological interactive multitrap system (IMTS) model, describing stimulated luminescence effects, during the irradiation stage.

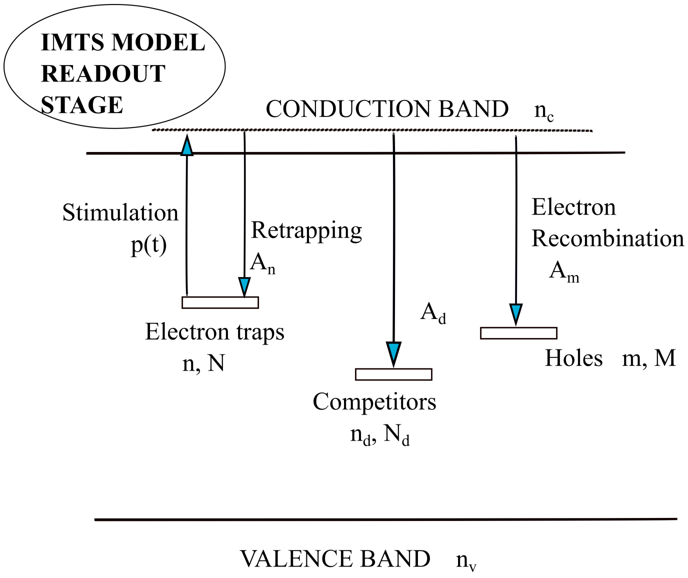

Fig. 2.18: Schematic diagram of the (IMTS) model, describing stimulated luminescence effects during the thermal or optical stimulation stage.

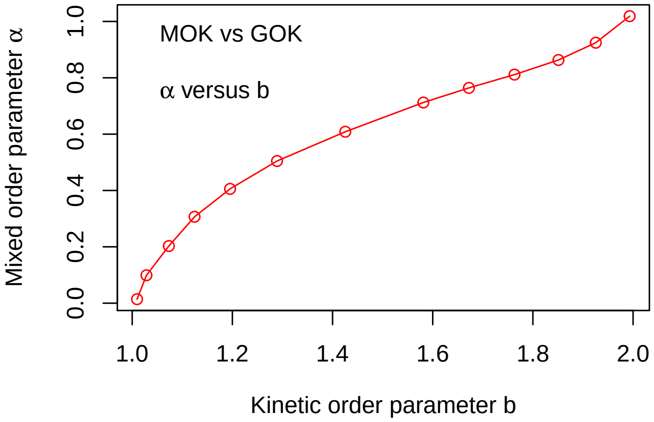

Fig. 2.19: The mixed order kinetics parameter α as a function of the general order kinetics parameter b.

Code 2.14: MOK deconvolution of Glocanin TL (tgcd)

```
# Deconvolution of Reference GLOCANIN glow curve #1 with MOK
rm(list=ls())
library("tgcd")
data(Refglow)
# Load the data.
# Deconvolve data with 1 peak using the MOK expression
startingPars <-
  cbind(c(15.0),  c(1.0), c(520), c(0.1)) # Im, E, Tm, R
invisible(capture.output(TL1 <- tgcd(Refglow$x001, npeak=1,
model="m1",inisPAR=startingPars, nstart=10, edit.inis=FALSE)))
print.noquote("Best fit parameters")
TL1$pars
```

```
## [1] Best fit parameters
##           INTENS(Im) ENERGY(E) TEMPER(Tm)   aValue(a)
## 1th-Peak   10967.93  1.182694    490.341 0.000499707
```

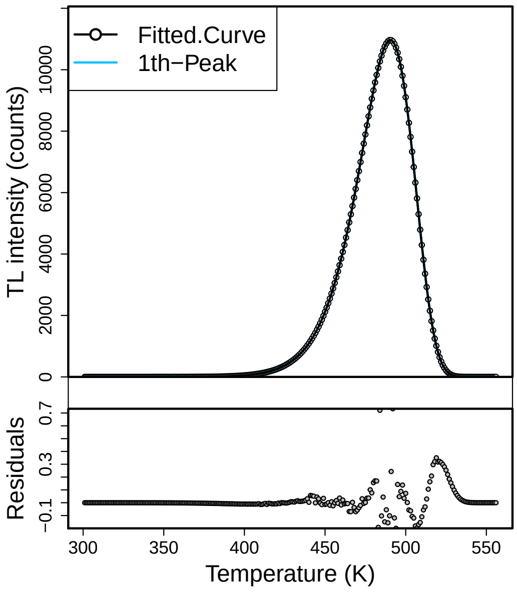

Fig. 2.20: Deconvolution of Reference glow curve #1 in the GLOCANIN project, using Mixed Order Kinetics (MOK).

Function Name	**Description**
plot_RLumCarlo	Plots 'RLumCarlo' modeling results (the averaged signal or the number of remaining electrons), with modeling uncertainties.
run_MC_ISO_DELOC	Simulation of ITL signals using the one trap one recombination center (OTOR) model.
run_MC_LM_OSL_DELOC	Simulation of LM-OSL signals using the delocalized OTOR model.
run_MC_CW_OSL_DELOC	Simulation of CW-OSL signals using the OTOR model.
run_MC_TL_DELOC	Simulation of TL signals using the OTOR model.

Table 2.1: Table of DELOCalized functions available in the package *RLumCarlo.*

Code 2.15: Combine 3 plots for isothermal experiment

```
##=========================================================##
## COMBINE 3 PLOTS  FOR DELOCALIZED ITL
##=========================================================##
rm(list = ls(all=T))
suppressMessages(library(RLumCarlo))
## set time vector
times <- seq(0, 400)
run_MC_ISO_DELOC(T=220, E=1.4, s=1e12,R=1e-3, times = times) %>%
plot_RLumCarlo(norm = TRUE, col="red",legend = F)
run_MC_ISO_DELOC(T=230, E=1.4, s=1e12,R=1e-3, times = times) %>%
plot_RLumCarlo(norm = TRUE, col="green",add = TRUE)
run_MC_ISO_DELOC(T=240, E=1.4, s=1e12,R=1e-3, times = times) %>%
plot_RLumCarlo(norm = TRUE, col="blue",add = TRUE, times= times)
legend("top",bty="n",legend=c("Isothermal TL signal",
```

Process	**Symbol**	**Parameter in RLumCarlo function**	**Units**	**Typical values**
Delocalized TL	E	Thermal activation energy of the trap	eV	0.5-3
	s	Frequency factor of the trap	1/s	1E8-1E16
	times	Sequence of time steps for simulation (heating rate is 1 K/s)	s	0-700
	clusters	Number of MC runs	1	1E1-1E4
	N_e	Total number of electron traps available	1	2-1E5
	n_filled	Number of filled electron traps at the beginning of the simulation	1	1-1E5
	R	Delocalized retrapping ratio	1	0-1
Delocalized CW-IRSL	A	Optical excitation rate from trap to conduction band	1/s	1E-3-1
	times	Sequence of time steps for simulation	s	0-500
	clusters	Number of MC runs	1	1E1-1E4
	N_e	Total number of electron traps available	1	2-1E5
	n_filled	Number of filled electron traps at the beginning of the simulation	1	1-1E5
	R	Delocalized retrapping ratio	1	0-1
Delocalized ISO	E	Thermal activation energy of the trap	eV	0.5-3
	s	Frequency factor of the trap	1/s	1E8-1E16
	T	Temperature of the isothermal process	°C	20-300
	times	Sequence of time steps for simulation	s	0-1000
	clusters	Number of MC runs	1	1E1-1E4
	N_e	Number of electrons	1	2-1E5
	n_filled	Number of filled electron traps at the beginning of the simulation	1	1-1E5
	R	Delocalized retrapping ratio	1	0-1
Delocalized LM-OSL	A	Optical excitation rate from trap to conduction band	1/s	1E-3-1
	times	Sequence of time steps for simulation	s	0-3000
	clusters	Number of MC runs	1	1E1-1E4
	N_e	Total number of electron traps available	1	2-1E5
	n_filled	Number of filled electron traps at the beginning of the simulation	1	1-1E5
	R	Delocalized retrapping ratio	1	0-1

Table 2.2: Table of input parameters for DELOC functions in *RLumCarlo*

```
expression("at 220, 230, 240"^"o"*"C")))
```

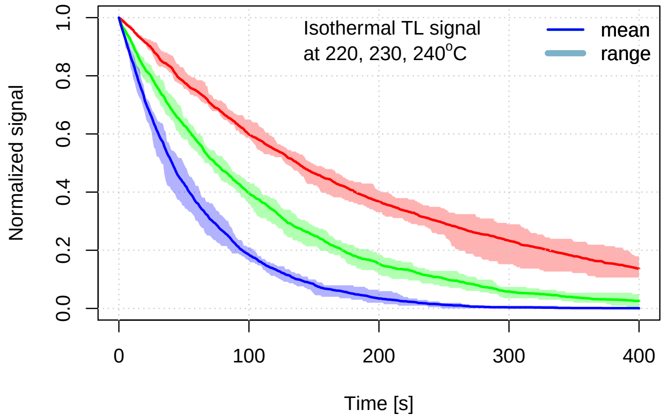

Fig. 2.21: Example of combining 3 plots for ITL signals from DELOCalized transitions, within the OTOR model.

Code 2.16: Single MC plot for delocalized TL

```
##==================================================##
## Example 1: Single MC Plot  for delocalized TL
##==================================================##
rm(list = ls(all=T))
library(RLumCarlo)
run_MC_TL_DELOC(
s = 3.5e12,
E = 1.45,
R = 0.1,
times = 100:350
) %>%
#Plot results of the MC simulation
plot_RLumCarlo
legend("topleft",bty="n",c("TL signal", "using
DELOC","function"))
```

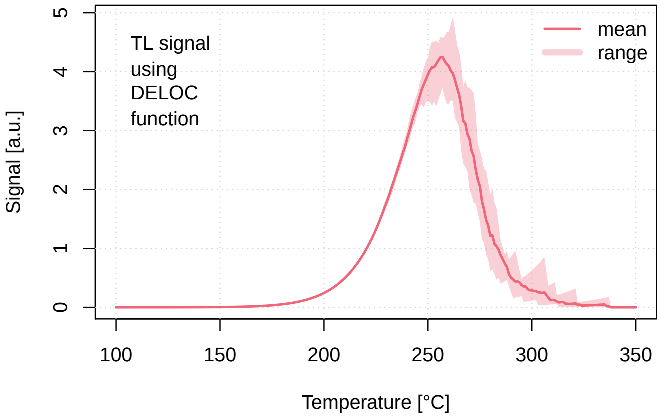

Fig. 2.22: MC simulation of TL glow curve from the OTOR delocalized transition model.

Code 2.17: MC for delocalized TL: multiple parameters

```
##===============================================##
##===============================================##
## Plot multiple TL  curves with varying params
##===============================================##
# define your parameters
rm(list = ls(all=T))
library(RLumCarlo)
times=seq(150,350,1)
s=rep(3.5e12,4)
E=rep(1.45,4)
R<-c(0.7e-6,1e-6,0.01,0.1)
clusters=100
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 colors for the individual curves
plot_uncertainty <- c(TRUE,TRUE,TRUE,TRUE) # plot  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], ylim=c(0,20))
}
legend("left",bty="n",c("TL","DELOC","many variables"))
legend("topright",bty="n",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)
```

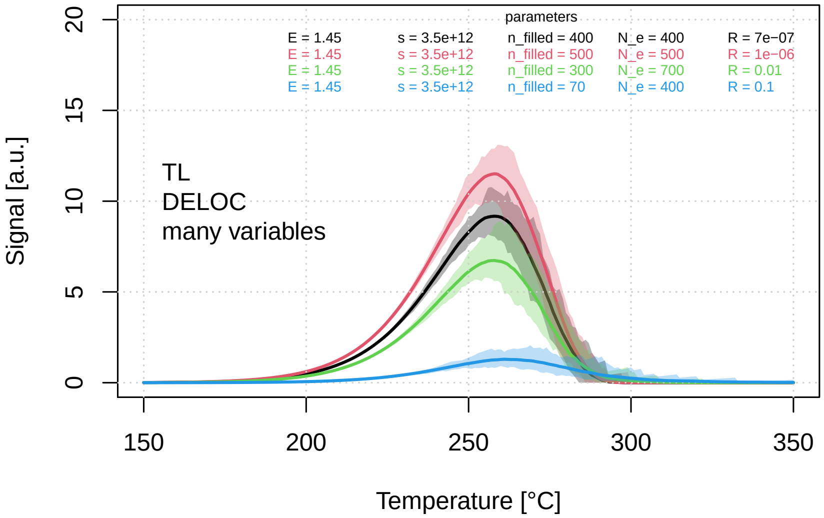

Fig. 2.23: Simulations of DELOCalized TL transitions, by varying many parameters in the model.