

# R package *numOSL*: numeric routines for optically stimulated luminescence dating

Jun Peng<sup>1,2\*</sup>, ZhiBao Dong<sup>1</sup>, FengQing Han<sup>3</sup>, Hao Long<sup>4</sup>, XiangJun Liu<sup>3</sup>

<sup>1</sup>Cold and Arid Regions Environmental and Engineering Research Institute, Chinese Academy of Sciences, 730000 Lanzhou, China

<sup>2</sup>University of Chinese Academy of Sciences, 100049 Beijing, China

<sup>3</sup>Qinghai Institute of Salt Lakes, Chinese Academy of Sciences, 810008 Xining, China

<sup>4</sup>Nanjing Institute of Geography and Limnology, Chinese Academy of Sciences, 210008 Nanjing, China

\*Corresponding author: pengjun10@mails.ucas.ac.cn

(Received 27 August 2013; in final form 23 December 2013)

## Abstract

In luminescence dating, there are various software packages that can be used to do data analysis, such as *Analyst*, *Java RadialPlotter*, *SigmaPlot*<sup>TM</sup> and so on. There is also an open source **R** package *Luminescence*, which has the capacity to tackle not only lots of basic statistical analysis but also provide more advanced data treatment. In this article, we present another **R** package, *numOSL*, the aim of which is neither to present a comprehensive numeric tool for luminescence dating just as the **R** package *Luminescence* did, nor to cover all numeric topics regarding luminescence dating. We focus only on the most frequently encountered numeric problems concerning luminescence dating, including equivalent dose calculation and error estimation, decay signal decomposition, fast-component equivalent dose calculation, and statistical age model optimization. Almost all our code is written in Fortran and is linked to **R** using an interface in order to improve algorithms, generality and flexibility. This makes it faster and perhaps more powerful when comparing to other numeric software.

## Introduction

Basic numeric techniques are an important aspect of luminescence dating, which are frequently encountered in daily handling of luminescence data. These numeric techniques may include: 1) interpolation; 2) non-linear parameter optimization; 3) Monte-Carlo simulation; and 4) maximum likelihood estimation. Many software packages can be used to perform a special kind of numeric analysis for luminescence dating. For example, *Analyst* (Duller, 2007a) is mainly used for basic data handling such as data import/export, equivalent dose calculation, and plotting of graphs. *Java RadialPlotter* (Vermeesch, 2009) focuses on optimizing parameters in Galbraith's statistical age

models and drawing radial plots (Galbraith, 1988). *SigmaPlot*<sup>TM</sup> has been frequently employed to carry out decay curve fitting (Choi et al, 2006). However, if one wants to perform more flexible and more comprehensive analysis on a series of different kinds of data, such as curve fitting with decay signal data, or statistical age model analysis with equivalent dose (ED) values, no single software mentioned above can satisfy all the requests. There also is an open source **R** package *Luminescence* (Kreutzer et al, 2012, Dietze et al, 2013) written in pure **R** language, which contains more comprehensive numeric routines to analyze various kinds of luminescence data. Though flexible, sometimes pure **R** function run very slowly, which may impede its implementation in problems that need a great number of iterations or Monte-Carlo simulations.

**R** serves as excellent statistical software (Ihaka and Gentleman, 1996; R Core Team, 2013), and is free of charge. Most importantly, code written in **R** is available to the user and might be modified and redistributed. Another significant characteristic of **R** is that it can easily communicate with other programming languages such as *Fortran* and *C++*. Hence one can write the time-consuming part of a program in *Fortran* or *C++* and then link it to **R** using an interface to achieve acceptable running speed. This remains **R**'s flexibility and also makes it really powerful. In this paper, we introduce the **R** package *numOSL* as a toolbox for numeric optimizations that are frequently encountered in luminescence dating. To make it flexible and powerful, almost all our functions are written in *Fortran* and are linked to **R** using interface. The aims of this paper are: 1) unifying regular numeric routines in luminescence dating into handy functions and make them available to all users; 2) introducing the usage of these functions with examples.

## Introduction to functions in R package *numOSL*

### Function: *calED*

Currently, there are eight functions in package *numOSL* (Table 1). Function “*calED*” is used to fit a dose-response curve using the Levenberg-Marquardt algorithm and calculate an ED value by interpolation. Three models are provided (linear, exponential, linear plus exponential model) to do the fitting. For the linear model the fitting will not be a problem, but when it comes to an exponential or an exponential plus linear model, care needs to be taken regarding the choice of initial parameters. There are two options in function “*calED*” that can be used to initialize non-linear parameters: “*nstart*” and “*upb*”. “*upb*” is used to control the upper boundary of the *b* value in the non-linear model, which will be generated uniformly in the space (0, *upb*); then other initial values can be obtained with linear algebra method. For example, in an exponential model of the formula  $y = a(1 - e^{-bx}) + c$ , combining paired observations ( $x_i, y_i, i=1, \dots, n$ ) with a uniformly distributed random *b* value, *a* and *c* can be calculated using a linear algebraic method. Then the Levenberg-Marquardt algorithm will be called to optimize those parameters. The process will be carried out repeatedly until stopping conditions are satisfied. The maximum allowed number of attempts is set to be “*nstart*”. The standard error of the ED value can be assessed using two methods (simple transform and Monte-Carlo simulation) outlined by Duller (2007b). Figure 1 is the outputted plot if we input the following commands to the **R** console:

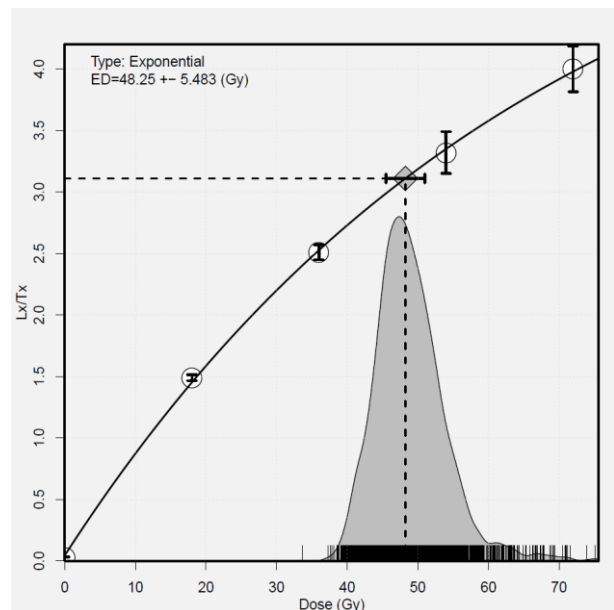
```
library(numOSL)
Curvedata<-data.frame(cbind( c(0, 18, 36, 54, 72),
  c(0.03,1.49,2.51,3.32,4.0),
  c(0.002,0.05,0.12,0.34,0.37) ) )
Ltx<-c(3.11, 0.131)
res<-calED(Curvedata, Ltx, model = "exp", nsim = 3000)
res
```

### Function: *decomp*

Function “*decomp*” is a unified function for decay curve decomposition and can be applied either to CW-OSL or LM-OSL decay curves. Fitting decay curves is an ill-conditioned problem, and it is very sensitive to the choice of initial parameters. To make the procedure flexible and practicable, CW-OSL decay curves are fitted using a combination of a differential evolution method and the Levenberg-Marquardt algorithm as suggested by Bluszcz and Adamiec (2006). The general procedure for CW-OSL

**Table 1:** Available functions in version 1.0 of package *numOSL*

Function	Description
<i>calED</i>	Calculate an ED value and assess its standard error
<i>dbED</i>	Summarize the statistical characteristics of the distribution of ED values
<i>decomp</i>	Decompose CW-OSL or LM-OSL decay curves
<i>decompc</i>	Decompose CW-OSL or LM-OSL decay curves (plus a constant component)
<i>fastED</i>	Calculate a CW-OSL ED value using the fastest component
<i>print.RadialPlotter</i>	Print an object of class “ <i>RadialPlotter</i> ”
<i>RadialPlotter</i>	Optimize parameters of Galbraith’s statistical age models
<i>sgcED</i>	Calculate ED values using the standardized grow curve method



**Figure 1:** Plot output for function “*calED*”. The shaded area shows a gaussian kernel density plot for 3,000 ED values simulated with a Monte-Carlo technique.

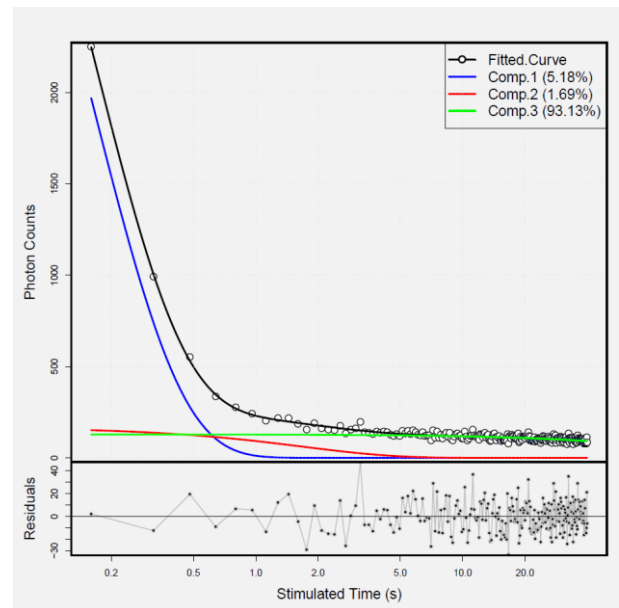
fitting is: parameters are initialized using a differential evolution algorithm, and then using these initial parameters the Levenberg-Marquardt algorithm is employed for optimization. As pointed out by Bluszcz and Adamiec (2006), in some radical cases, the differential evolution algorithm (Storn and

Price, 1997) may fail, and we also note that sometimes parameters can be initialized with the differential evolution algorithm successfully but the Levenberg-Marquardt algorithm fails. If any of the two situations mentioned above occurs, a series of simple trials will be carried out, that is: detrapping rates are initialized to values outlined by Jain et al (2003) using permutation and combinations. For example, if we wish to decompose a given CW-OSL decay curve to 3 components, there will be  $C(3,7)=35$  possible combinations of detrapping rates. Then for each possible combination of detrapping rates the number of trapped electrons that correspond to those detrapping rates can be obtained using a linear algebra method (Bluszcz, 1996) and these values will be used as initial parameters for the Levenberg-Marquardt algorithm. The above process is performed repeatedly until an acceptable result is found. Since the program is written in *Fortran* and wrapped with *R*, the whole procedure can be executed very quickly. The simple trial method described above is exactly the tactic we adopt in LM-OSL decay curves decomposing. Because we find it difficult to use the same method as used for fitting CW-OSL decay curves. The longer stimulation time makes it so time-consuming to conduct the differential evolution algorithm, even if all parts of the program are written in pure *Fortran*, that it becomes impracticable. Using the following commands in the *R* console we can obtain the plot shown in Figure 2 and Figure 3, respectively. The optional parameter “outfile” can be used to output decomposed signal values to a file and save it into the current work directory in “.csv” format (here the file will be “lmsig.csv”).

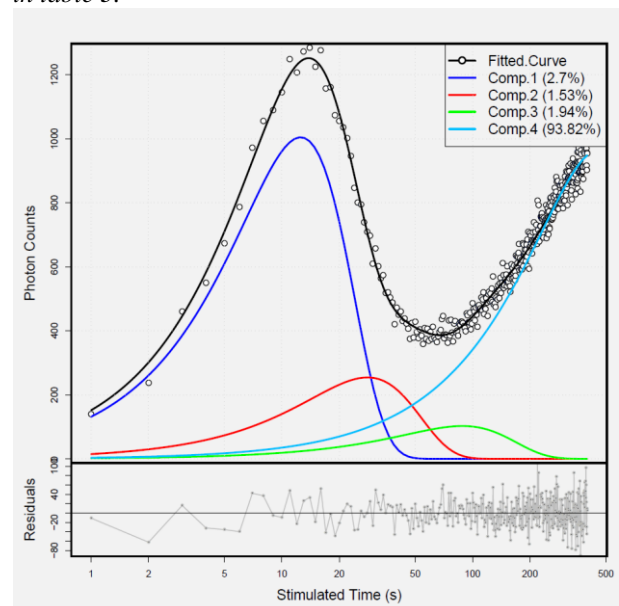
```
data(Signaldata)
print(decomp(Signaldata$cw[1:2], ncomp=3, lwd=2))
res<-decomp(Signaldata$lms, ncomp=4, typ="lm",
            transf=TRUE, lwd=2, outfile="lmsig")
res
```

#### Function: fastED

A basic requirement for the application of the single-aliquot regenerative dose protocol (Murray and Wintle, 2000) is that the initial part of the OSL signal is dominated by the fast component (Li and Li, 2006). Generally, a fast-component ED value can be obtained through direct measurement (Bailey, 2010), component isolation with curve fitting (Cunningham and Wallinga, 2009) or integral channels selection (Cunningham and Wallinga, 2010). The function “fastED” attempts to build a fast-component growth curve to estimate a fast-component ED value using data obtained by the single-aliquot regenerative dose protocol (a series of decay curves). The number of

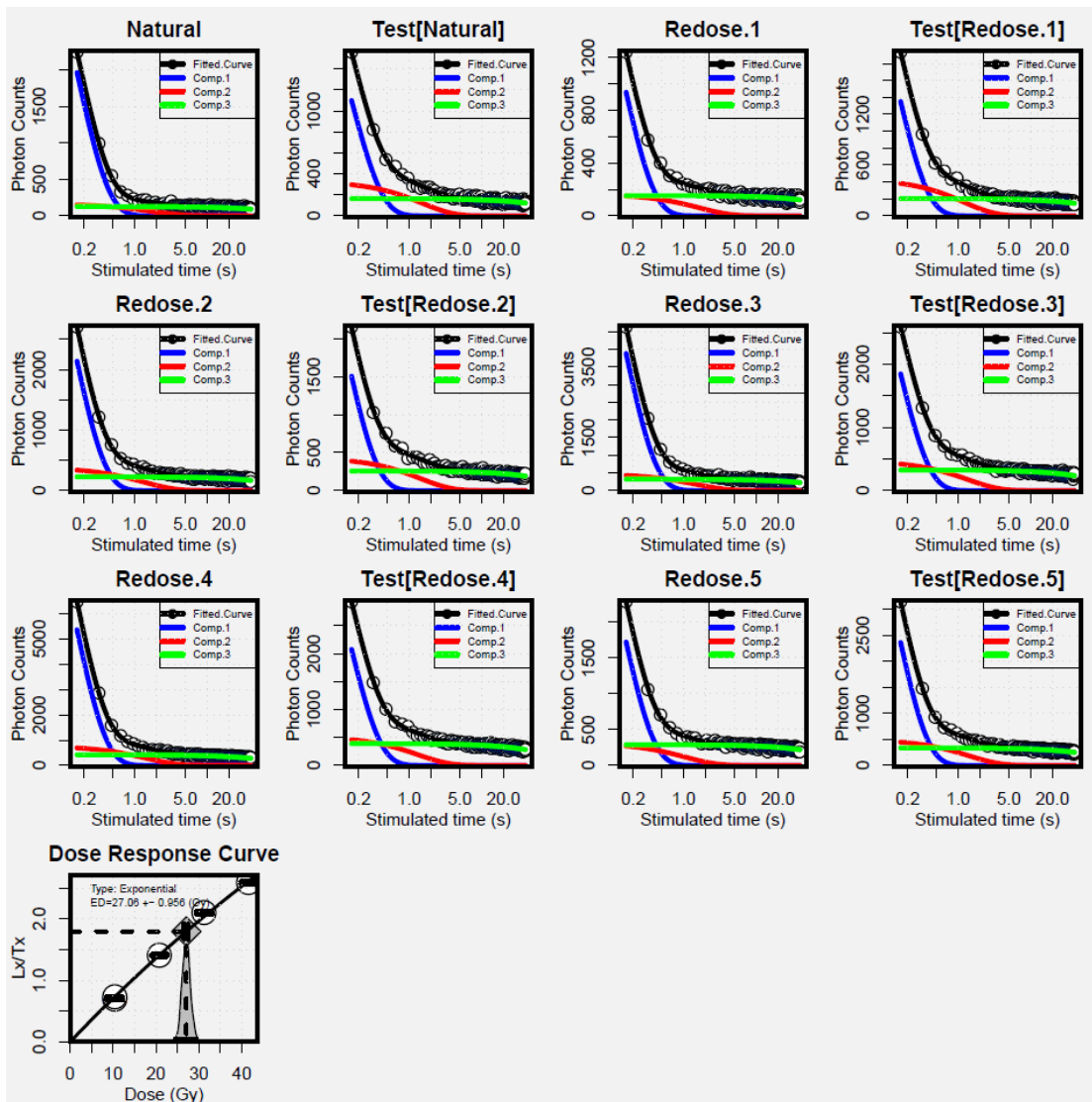


**Figure 2:** Decomposed 3 components for natural CW-OSL decay curve of sample GL1-1 (Peng and Han, 2013) using function “decomp”. The estimated parameters (magnitudes and decay rates) are given in table 3.



**Figure 3:** Decomposed 4 components for a LM-OSL decay curve (Li and Li, 2006) using function “decomp”. The estimated parameters are given in table 3.

trapped electrons that corresponds to the largest decay rate will be regarded as the fast-component signal, which cannot always ensure a pure fast-component signal to be extracted if ultra-fast component appears. The number of components to be decomposed is specified by the argument “ncomp”.



**Figure 4:** Result of fast-component ED estimation for sample GL1-1 with 3 components using function “fastED”. A total of 12 decay curves are decomposed and the growth curve is constructed with the extracted standardized fast-component signals.

During the calculation, the model used for growth curve fitting is chosen automatically with the principle of the minimum sum of square of residuals. Only CW-OSL data can be analyzed currently. We use sample GL1-1 (Peng and Han, 2013) as an example to show the usage of this function, the two decay curves that correspond to 0 regenerative dose are precluded before the analysis and each decay curve is decomposed to 3 components. The plot is shown in Figure 4:

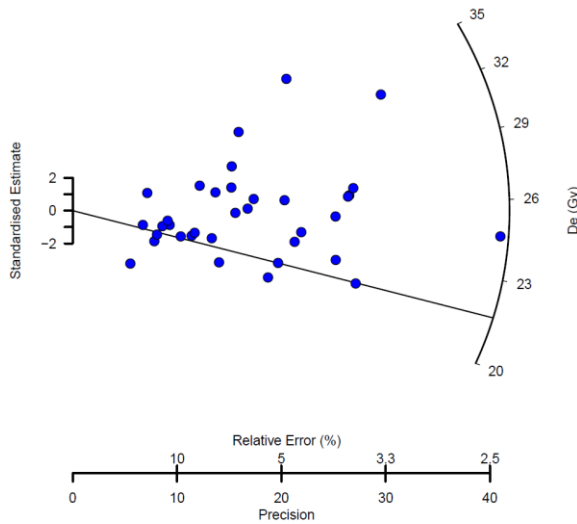
```
fastED(Signaldata$sw[,c(-12,-13)],
ncomp=3, constant=FALSE,
Redose=c(80,160,240,320,80)*0.13)
```

The running time of the above process can be checked by **R** inner function “system.time” as:

```
system.time(fastED(Signaldata$sw[,c(-12,-13)],
ncomp=3, constant=FALSE,
Redose=c(80,160,240,320,80)*0.13))
```

*Function: RadialPlotter*

Function “RadialPlotter” is a unified function for optimizing a number of statistical age models as reviewed in Galbraith and Roberts (2012). It provides routines for the central age model (Galbraith et al, 1999), the finite mixture age model (Galbraith, 1988, Galbraith and Laslett, 1993) and the minimum age model (Galbraith et al, 1999). Depending on the specified model, it draws a radial plot automatically,

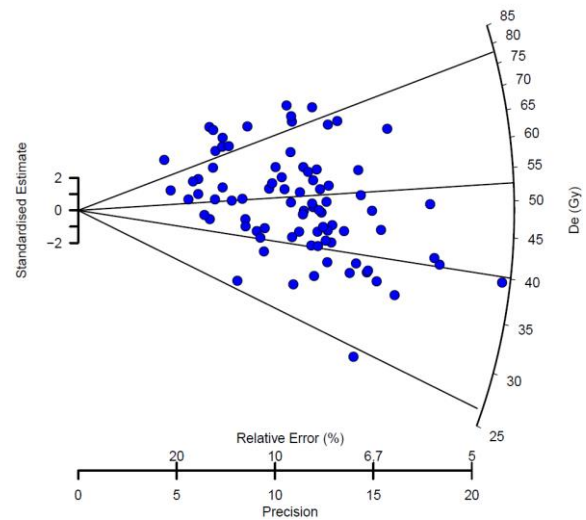


**Figure 5:** Radial plot output for 3-parameter minimum age model of sample GL1-1 with function “RadialPlotter”. The black line indicates estimated minimum ED value.

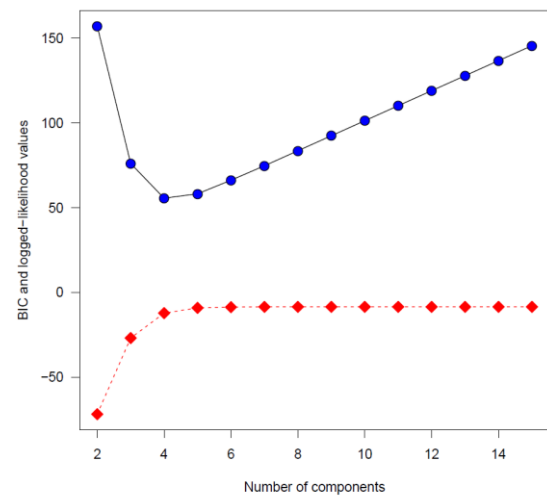
which works just like the *Java RadialPlotter* written by Vermeesch (2009). Both the central age model and the finite mixture age model are analyzed using the maximum likelihood estimation method outlined by Galbraith (1988). For minimum age models, it provides two numeric procedures to do the task: the “L-BFGS-B” algorithm (Zhu et al, 1994) and the “port” routine (the “port” algorithm is conducted using *R* inner function “nlminb” in package *stats*). The result returned by the function “RadialPlotter” is an object of class “RadialPlotter”. An example of using this function to fit a three-parameter minimum age model with sample GL1-1 is shown below, and the output is Figure 5:

```
data(EDdata)
obj<-RadialPlotter(EDdata$gl11,ncomp=-1,
  zscale=seq(20,37,3),kratio=0.6)
unclass(obj)
```

When attempting to fit the finite mixture model there is a parameter “maxcomp” that can be used to control the maximum allowed number of components. The function will search for the number of components that gives a minimum Bayesian Information Criterion (BIC) value (Schwarz, 1978) in the finite mixture model. Here we use sample AL3 (Schmidt et al, 2012) as an example to find out the most appropriate number of components. In the following codes, the maximum number of components is set to be 15. The radial plot created by this function is shown in Figure 6. A plot for BIC and maximized logged likelihood values against the number of components is shown in Figure 7, which



**Figure 6:** Radial plot output for finite mixture age model of sample AL3 with function “RadialPlotter”. The appropriate number of components that gives a minimum BIC value is found automatically at 4. The black lines are characteristic ED values for the estimated 4 components, respectively. The estimated parameters (proportion and characteristic ED value for each component) are summarized in table 4.



**Figure 7:** Variations of BIC and maximized logged-likelihood values with the number of components for sample AL3. The blue points are the BIC values while the red points are logged-likelihood values.

shows that 4 components given the minimum BIC value and should be regarded as the most appropriate number of components.

```
obj<-RadialPlotter(EDdata$al3, ncomp=0, maxcomp=15,
  zscale=seq(25,85,5), kratio=0.6)
print(obj)
```

**Table 2:** Comparing Duller's (2007b) results with that obtained using function "calED" in R package numOSL (version 1.0).

	R package numOSL			Duller (2007b)		
	ED	Curve fitting	Monte Carlo	ED	Curve fitting	Monte Carlo
Data 1	0.7	0.7±0.06	0.7±0.07	0.7	0.7±0.06	0.7±0.06
Data 2	28.5	28.5±0.67	28.5±0.76	28.5	28.5±0.67	28.5±0.75
Data 3	0.043	0.043±0.0056	0.043±0.002	0.046	0.046±0.004	0.046±0.001
Data 4	0.71	0.71±0.11	0.71±0.14	0.71	0.71±0.11	0.71±0.13

**Table 3:** Comparing results of decay curve decompositions obtained using function "decomp" in R package numOSL (version 1.0) with that obtained using SigmaPlot™ (version 10.0) and R package Luminescence (version 0.2.4) (Kreutzer et al, 2012) for natural CW-OSL decay curve of sample GL1-1 and LM-OSL decay curve of a quartz sample (Li and Li, 2006). a and b are estimated magnitude and decay rate for each signal component, respectively.

Component		R package numOSL		SigmaPlot™		R package Luminescence	
		a	b	a	b	a	b
CW-	#1	5240 ± 115	6.11 ± 0.16	5240 ± 104	6.11 ± 0.14	5240 ± 213	6.11 ± 0.30
OSL	#2	166.6 ± 25.8	0.59 ± 0.11	166.6 ± 21.0	0.59 ± 0.08	166.6 ± 6.28	0.59 ± 0.002
	#3	128.3 ± 3.40	0.008 ± 0.001	128.3 ± 2.94	0.008 ± 0.001	128.3 ± 48.0	0.008 ± 0.20
	residual	=43454		residual =43453.65		residual=43454	
LM-	#1	20769 ± 668	2.54 ± 0.065	20800 ± 663	2.54 ± 0.068	20771 ± 1759	2.54 ± 0.16
OSL	#2	11787 ± 520	0.51 ± 0.053	11798 ± 619	0.5 ± 0.048	11787 ± 3685	0.51 ± 0.13
	#3	14941 ± 1038	0.05 ± 0.007	14995 ± 1158	0.05 ± 0.006	14943 ± 4633	0.05 ± 0.018
	#4	721057 ± 25497	0.002 ± 0.0001	722659 ± 26461	0.002 ± 0.0001	721118 ± 65749	0.002 ± 0.0002
	residual	=373062.8		residual =373066.02		residual=373063	

**Table 4:** Comparing results of finite mixture age model obtained using Java RadialPlotter (version 4.4) (Vermeesch, 2009) and that obtained using function "RadialPlotter" in R package numOSL (version 1.0) for sample AL3 and sample GL1-1. The numbers of components (k) that give minimum BIC values are estimated automatically in the finite mixture age model. The spreads (sigma) in ED values are set to be 0.0.

	sigma=0.0	R package numOSL		Java RadialPlotter	
		k		k	
AL3		k=4		k=4	
		p <sub>1</sub> =2.1%±1.8%	u <sub>1</sub> =26.4±1.9	p <sub>1</sub> =2.1%±1.8%	u <sub>1</sub> =26.4±2.0
		p <sub>2</sub> =35.1%±6.5%	u <sub>2</sub> =40.1±0.7	p <sub>2</sub> =35.2%±6.4%	u <sub>2</sub> =40.1±0.7
		p <sub>3</sub> =40.8%±6.7%	u <sub>3</sub> =52.6±1.2	p <sub>3</sub> =40.7%±6.7%	u <sub>3</sub> =52.7±1.2
		p <sub>4</sub> =22.0%±4.9%	u <sub>4</sub> =77.9±2.2	p <sub>4</sub> =22.0%±9.4%	u <sub>4</sub> =77.9±2.2
GL1-1		k=3		k=3	
		p <sub>1</sub> =37.6%±10.9%	u <sub>1</sub> =22.0±0.4	p <sub>1</sub> =38.0%±11.0%	u <sub>1</sub> =22.0±0.5
		p <sub>2</sub> =50.9%±11.3%	u <sub>2</sub> =25.7±0.4	p <sub>2</sub> =51.0%±11.0%	u <sub>2</sub> =25.7±0.4
		p <sub>3</sub> =11.5%±5.9%	u <sub>3</sub> =33.7±0.9	p <sub>3</sub> =11.0%±16.0%	u <sub>3</sub> =33.8±0.9

#### Comparison with other software

The reliability of these routines was checked by comparing the results with those displayed in published articles or obtained using other software. The function "calED" was tested using the same data and results that were published in Duller (2007b) (Table 2); results obtained by function "decomp" are

compared to that obtained using SigmaPlot™ and R package Luminescence (Kreutzer et al, 2012) (Table 3); function "RadialPlotter" is checked by comparing the results with that estimated using Java RadialPlotter (Vermeesch, 2009) (Table 4). The results of all the tests are comparable between the different software packages.



## Discussion and conclusion

Numeric techniques such as Monte-Carlo error simulation, and differential evolution are time-consuming tasks. Decomposing decay curves to various components and maximum likelihood estimations of Galbraith's statistical age models (especially the minimum age model) are ill-conditioned mathematical problems; whether it is possible to obtain a reasonable result depends heavily on the choice of initial parameters. Hence to make the program have the capacity to be applicable to a wide range of data sets, trying various initial parameters is indispensable, which, on the other hand, slows down the running speed. This problem will be especially significant if one wants to use the results to do some further analysis. For example, one may need to calculate ED values and assess their standard errors repeatedly when using the SGC method (Roberts and Duller, 2004). Alternatively if one hopes to decompose a series of decay curves to build a dose-response curve in order to calculate a fast-component ED value for data this requires general and fast routines.

In this paper, we present an **R** package (*numOSL*) for tackling regular numeric problems that are frequently encountered in analyzing luminescence data. This is the second **R** package concerning luminescence dating that is available on the Comprehensive **R** Archive Network (**CRAN**), but the work presented here is not a duplicated version for **R** package *Luminescence*. We focus only on regular numeric routines in luminescence dating and aim to make those routines more general and robust through a mixture programming language of **R** and *Fortran*. The tests show that results generated by package *numOSL* are comparable to other numeric packages. Flexible as it is, however, we must admit that users may need some basic knowledge about the **R** programming language when attempting to use an **R** package. So if one only needs to do some basic handling or plotting of luminescence data, the best choice would be *Analyst*, which is sufficient for rudimentary data processing and is user-friendly. The **R** package *numOSL* is provided under the General Public Licence (GPL3) conditions, which is free software and can be downloaded freely from <http://CRAN.R-project.org/package=numOSL>.

Anyone who finds a bug during the use of this package is encouraged to contact the corresponding author.

## Acknowledgments

We thank Rex Galbraith and Sebastian Kreutzer for their permission to bundle their **R** code for radial plot drawing to **R** package *numOSL*. We also thank Andrzej Bluszcz, who offers lots of help during the writing of the differential evolution algorithm.

Sheng-Hua Li and Geoff Duller are thanked for helpful discussions concerning fast-component equivalent dose calculation. Silke Schmidt, Jeong-Heon Choi and Bo Li are thanked for providing published data to test our programs. This work was funded by the Ministry of Science and Technology of the People's Republic of China (2013CB956000 and 2012CB426501).

## References

- Bailey, R.M. (2010). Direct measurement of the fast component of quartz optically stimulated luminescence and implications for the accuracy of optical dating. *Radiation Measurements* 5: 559-568.
- Bluszcz, A. (1996). Exponential function fitting to TL growth data and similar applications. *Geochronometria* 13: 135-141.
- Bluszcz, A., Adamiec, G. (2006). Application of differential evolution to fitting OSL decay curves. *Radiation Measurements* 41: 886-891.
- Choi, J.H., Duller G.A.T., Wintle, A.G. (2006). Analysis of quartz LM-OSL curves. *Ancient TL* 24: 9-20.
- Cunningham, A.C., Wallinga, J. (2009). Optically stimulated luminescence dating of young quartz using the fast component. *Radiation Measurements* 44: 423-428.
- Cunningham, A.C., Wallinga, J. (2010). Selection of integration time intervals for quartz OSL decay curves. *Quaternary Geochronology* 5: 657-666.
- Dietze M., Kreutzer, S., Fuchs, M.C., Burow, C., Fischer M., Schmidt, C. (2013). A practical guide to the R package Luminescence. *Ancient TL* 31: 11-18.
- Duller, G.A.T. (2007a). *Analyst*. Manual, p.1-45.
- Duller, G.A.T. (2007b). Assessing the error on equivalent dose estimates derived from single aliquot regenerative dose measurements. *Ancient TL* 25: 15-24.
- Galbraith, R.F. (1988). Graphical display of estimates having differing standard errors. *Technometrics* 30: 271-281.
- Galbraith, R.F., Laslett, G.M. (1993). Statistical models for mixed fission track ages. *Nuclear Tracks and Radiation Measurements* 21: 459-470.
- Galbraith, R.F., Roberts, R.G. (2012). Statistical aspects of equivalent dose and error calculation and display in OSL dating: An overview and some recommendations. *Quaternary Geochronology* 11: 1-27.
- Galbraith, R.F., Roberts, R.G., Laslett, G.M., Yoshida, H., Olley, J.M. (1999). Optical dating of single grains of quartz from Jinmium rock shelter, northern Australia. Part I: experimental design and statistical models. *Archaeometry* 41: 339-364.

- Ihaka, R., Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *J. Computational and Graphical Statistics* 5: 299-314.
- Jain, M., Murray, A.S., Boetter-Jensen, L. (2003). Characterisation of blue-light stimulated luminescence components in different quartz samples: implications for dose measurement. *Radiation Measurements* 37: 441-449.
- Kreutzer, S., Schmidt, C., Fuchs, M.C., Dietze, M., Fischer, M., Fuchs, M. (2012). Introducing an R package for luminescence dating analysis. *Ancient TL* 30: 1-8.
- Li, S.H., Li, B. (2006). Dose measurement using the fast component of LM-OSL signals from quartz. *Radiation Measurements* 41: 534-541.
- Murray, A.S., Wintle, A.G. (2000). Luminescence dating of quartz using improved single-aliquot regenerative-dose protocol. *Radiation Measurements* 32: 57-73.
- Peng, J., Han, F.Q. (2013). Selections of fast-component OSL signal using sediments from the south edge of Tengger Desert. *Acta Geoscientica Sinica* 34: 757-762 (In Chinese with English abstract).
- R Core Team. (2013). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL <http://www.R-project.org/>.
- Roberts, H.M., Duller, G.A.T. (2004). Standardised growth curves for optical dating of sediment using multiple-grain aliquots. *Radiation Measurements* 38: 241-252.
- Schmidt, S., Tsukamoto, S., Salomon, E., Frechen, M., Hetzel, R. (2012). Optical dating of alluvial deposits at the orogenic front of the Andean precordillera (Mendoza, Argentina). *Geochronometria* 39: 62-75.
- Schwarz, G. (1978). Estimating the dimension of a model. *Annals of statistics* 6: 461-464.
- Storn, R., Price, K. (1997). Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces. *J. Global Optim.* 11: 341-359.
- Vermeesch, P. (2009). RadialPlotter: a Java application for fission track, luminescence and other radial plots. *Radiation Measurements* 44: 409-410.
- Zhu, C., Byrd, R.H., Lu, P., Nocedal, J. (1994). "L-BFGS-B: Fortran subroutines for large scale bound constrained optimization" Tech. Report, NAM-11, EECS Department, Northwestern University.

**Reviewer**

G.A.T. Duller