

Introducing an R package for luminescence dating analysis

Sebastian Kreutzer^{1,2*}, Christoph Schmidt³, Margret C. Fuchs⁴, Michael Dietze⁵, Manfred Fischer² and Markus Fuchs¹

¹Department of Geography, Justus-Liebig-University Giessen, 35390 Giessen, Germany

²Geographical Institute, Geomorphology, University of Bayreuth, 95440 Bayreuth, Germany

³Institute for Geography, University of Cologne, 50923 Cologne, Germany

⁴Department of Geology, TU Bergakademie Freiberg, 09599 Freiberg, Germany

⁵Institute of Geography, TU Dresden, 01069 Dresden, Germany

* corresponding author: sebastian.kreutzer@geogr.uni-giessen.de

(Received 16 May 2012; in final form 27 May 2012)

Abstract

For routine luminescence dating applications the commonly used Risø readers are bundled with analysis software, such as *Viewer* or *Analyst*. These software solutions are appropriate for most of the regular dating and publication jobs, and enable assessment of luminescence characteristics and provide basic statistical data treatment. However, for further statistical analysis and data treatments, this software may reach its limits. In such cases, open programming languages are a more appropriate approach. Here, we present the **R** package ‘Luminescence’ for a more flexible handling of luminescence data and related plotting purposes, using the statistical programming language **R**. The **R** language as well as the package and the source code are provided under the General Public License (GPL) conditions and are available for free. The basic functionality of the package is described along with three application examples. This package is not an alternative to the existing software (*Analyst*, *Viewer*) but may provide a collection of additional tools to analyse luminescence data and serve as a platform for further contributions.

Keywords: R, luminescence dating, LM-OSL, curve fitting, radial plot

Introduction

Since the early beginnings of luminescence dating at the end of the 1950s (Daniels et al. 1953, Houtermans and Stauffer 1957, Grögler et al. 1958), the method of luminescence dating has advanced considerably. Not least, a widespread application in geosciences has become possible with the development of commercially available, computerised and automatic measurement equipment (e.g. Risø, Daybreak, Freiberg Instruments) and reliable measurement protocols (e.g. single aliquot

regenerative (SAR) protocol by Murray and Wintle 2000). For the analysis of measured data in routine luminescence dating applications, the commonly used Risø reader is bundled with software such as the *Viewer* or *Analyst* (Duller 2007). Considering the analysis, export and graphical output functions, this software is sufficient for most of the everyday dating and publication purposes. However, experimental and exploratory luminescence studies require more computational flexibility and further statistical data treatment. In addition, more recent methods such as the linearly modulated luminescence analysis (Bulur 1996) ask for a more comprehensive analysis software. Furthermore, compiled and ready to use software solutions are black boxes to non-software engineers, even if the source code is provided. Our attempts to find an easy-to-use and scalable numerical computational environment, able to 1) import OSL measurement data, 2) process the imported data with a modular, extendable toolbox of readily available and open source functions and 3) generate high-quality numerical and graphical output was met by the statistical software **R** (R Development Core Team 2012).

The **R** programming environment was introduced in 1996 by Ihaka and Gentleman (1996) and is based on the **S** system/language developed by John Chambers (e.g. Hornik and Leisch 2002). Scripts written in **R** are in general compatible to the commercially available implementation of the **S** system called ‘S-PLUS’ (e.g. Hornik and Leisch 2002, Ligges 2008). The main advantages of using **R** are: (1) The **R** language is intuitive and easy to learn, (2) **R** is open source software and therefore free of charge, (3) a growing community offers a fast and high-quality support, (4) **R** is available for all major computer platforms and the language itself is almost independent of the operating system, (5) **R** is modular and highly extensible through user defined

Table 1: Functions in the **R** package ‘Luminescence’ (vers. 0.1.7)

Name	Description
Analyse_SAR.OSLdata()	Analyse SAR CW-OSL measurements
Calc_FadingCorr()	Applying a fading correction according to Huntley and Lamothe (2001) for a given age and a given g-value
Calc_FuchsLang2001()	Calculate D_e applying the method of Fuchs and Lang (2001)
Calc_OSLLxTxRatio()	Calculate L_x/T_x ratio for a given set of OSL curves
CW2pLM()	Transform a CW-OSL curve into a pseudo-LM (pLM) curve (e.g. Bulur, 2000)
fit_LMCurve()	Non-linear Least Squares (NLS) fit for LM-OSL curves
plot_BINfileData()	Plot single luminescence curves from a BIN-file object (readBIN2R())
plot_DeDistribution()	Plot D_e distribution with a kernel density estimate (KDE)
plot_GrowthCurve()	Fit and plot a growth curve for luminescence data
plot_Histogram()	Plot a histogram with a separate error plot
plot_RadialPlot()	Plot a Galbraith's radial plot
readBIN2R()	Import Risø BIN-file into R
Second2Gray()	Converting values from seconds (s) to Gray (Gy)

packages. Especially the latter advantage points to the great potential in terms of experimental and increasingly wide application of luminescence investigations and is therefore the motivation for our contribution. Additionally, (6) already existing statistical tools of data analysis (e.g. the minimum age model of Galbraith et al. 1999) can directly be applied or implemented using the same software: **R**.

In general, **R** can be described as software for statistical computing by default without a graphical user interface but with a command line or terminal. It provides a powerful tool, once the **R** language has been assimilated. Our first **R** scripts were written to produce plots from *.csv (comma separated values) files, e.g. probability density plots, or to calculate statistical parameters from extensive data sets. However, when sharing the scripts with a growing community, the consequence is to encapsulate the **R** code into functions and bundle them within a separate **R** package. A package can be installed and loaded into the **R** environment. The provided functions work as small applications within the **R** environment while the source code of the scripts is always available and the calculations are therefore transparent for the user. Along with the **R** code itself, a documentation file for every function is available as well as additional example data, which have been thoroughly tested in advance.

Here we present the **R** package ‘Luminescence’ in its latest version (0.1.7) including a set of functions for data import, analysis of luminescence properties, statistical data processing and result plotting, e.g. D_e distributions, growth curves or calculate L_x/T_x values for a given set of curves (Table 1). The basic features are presented using three examples of typical

applications: (1) Importing BIN-files, (2) plotting D_e distributions, (3) fitting LM-OSL curves. Along with these functions, example data sets are delivered. Nevertheless, this paper is not an introduction to the **R** language nor is it possible to present all functions bundled within the package in detail. We tried to keep all examples as simple as possible. The given **R** code is intended to: (1) provide a short insight into how **R** and the package work (for readers who are not familiar with the **R** language) and (2) allow **R** users to roughly assess the basic functionality of the package.

Examples

To provide an overview of the basic functionality we focus on three application examples. For further functions the reader is referred to the manual of the package. The chosen examples are given as **R** code with the corresponding (reduced) terminal output. Code, terminal output and **R**-relating functions are indicated by monospaced letters. All examples were produced using the freely available integrated development environment (IDE) *RStudio™ Desktop* (<http://rstudio.org>) but the reader may also use other **R** IDEs or the **R** terminal.

Example 1: Importing *.bin files

A crucial question for data analysis usually is how to import the measured data to the preferred software. Substantial time is often spent before starting the data analysis undertaking data transformation and format conversion. The measurement results of the Risø readers are stored in a binary file format (*.bin, BIN-file) containing various data of the measurements. There are several ways to export

binary measurement data to more general exchange formats such as the ASCII format. The *Analyst* software as well as the *Risø Viewer* offers export functionalities of the raw data into an ASCII file. An alternative is the *MS WindowsTM* terminal program '*Bin2Ascii*' written in C (Bailey 2008). It produces separate ASCII files for every measured curve in one run. Furthermore, there may be numerous individual software solutions, written and used by various researchers throughout the luminescence community. In general, at least one additional step is needed to import the measured data into **R** but the metadata stored in the BIN-file (e.g. read temperature, type of stimulation) may get lost for further usage. However, the format of the BIN-file is described in detail in the *Analyst* manual (Duller 2007) which allowed us to develop an **R** function that directly imports the entire BIN-file into the **R** environment to maintain all provided data. The corresponding function is called `readBIN2R()` and has been successfully tested for BIN-files produced by the *Sequence Editor* of version 3.x and 4.x. To import a BIN-file into the **R** environment the function requires one argument containing the path and the name of the BIN-file:

```
> temp<-readBIN2R("~/Desktop/test.BIN")
> temp
Risoe.BINfileData Object
  Version:      03
  Object Date:  270520
  User:         x
  System ID:    0
  Overall Records: 1104
  Records Type: OSL=720;TL=336;IRSL=48;
  Position Range: 1 : 48
  Run Range:    1 : 45
  Set Range:    1 : 16
```

The `readBIN2R()` function returns an **R** object with two slots: `@METADATA` and `@DATA`. The first one is a `data.frame` object containing the metadata of the measurement as shown in the *Analyst*, the second slot contains a `list` object containing vectors with the measured count data.

```
> temp@METADATA[1:2,1:6]
  ID SEL VERSION LENGTH PREVIOUS NPOINTS
1  TRUE  03   672      0      100
2  TRUE  03   672     672      100

> unlist(temp@DATA[1])[1:10]
[1] 140 138 133 143 129 123 134 134 129 143
```

The provided metadata equals what is shown when using the 'Display Information' options in the *Analyst*. The metadata can be used to select certain curve data for further analysis. For example, to select the record IDs (which are identical to the IDs in the

slot `@DATA`) of all measured OSL data from the measurement, in **R** the following code is required:

```
> temp@DATA[temp@METADATA[temp@METADATA[,
+"LTYPE"]=="OSL", "ID"]]
```

Note: Working with the package does not mean that the input data has to be a BIN-file. For other formats, e.g. *.csv or *.txt, **R** provides generic functions that can be used instead (e.g. `read.csv()` or `read.table()`).

Example 2: Plotting D_e distributions

The next example was chosen to focus on what is of most general relevance and an essential advantage of **R**, the production of high-quality plots in different file formats. Creating figures for publication or presentation might be a time consuming and sometimes frustrating process. Once the plotting functions for a desired figure design have been created in **R**, the code can be reused almost universally with only minor modifications (e.g. modifying the input arguments). In the package we bundled a few plot functions, which produce plots that are commonly used to present luminescence dating results. Here we focus on two frequently used plot alternatives: (1) empirical cumulative distribution along with kernel density estimates (KDE) and (2) the radial plot. Within the package the function to produce an empirical distribution along with the KDE is called `plot_DeDistribution()` and the radial plot function `plot_RadialPlot()`. Both require as input a two column `data.frame` containing x-y-values (e.g. D_e and D_e error). Here a fine grain quartz loess sample from Saxony/Germany (BT998, unpublished data) is used as an example data set:

```
> ExampleData.DeValues[1:3,]
      x      y
1 207.5040 7.3062
2 208.3326 7.0470
3 227.1216 9.3216
```

The plot output (Figure 1) gives the D_e values with their corresponding D_e error in ascending order and additionally shows the basic D_e distribution parameters (number of D_e , mean, standard deviation). The KDE is shown as a blue line. It is produced with:

```
> plot_DeDistribution(ExampleData.DeValues)
```

For plotting a radial plot (e.g. Galbraith 1988) on a log z-scale the **R** code is quite similar:

```
> plot_RadialPlot(ExampleData.DeValues)
```

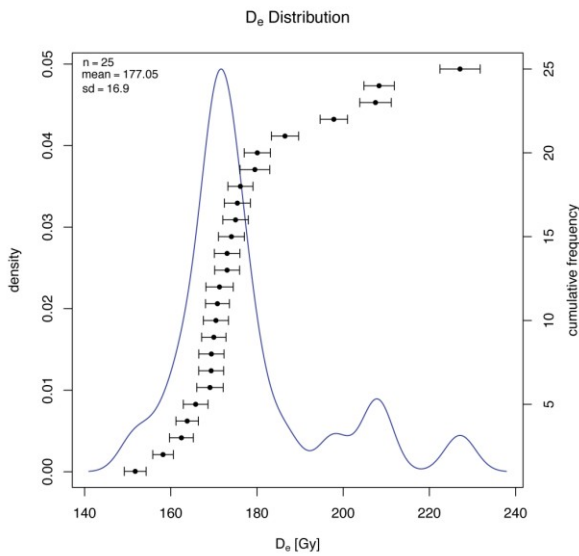


Figure 1: D_e distribution with kernel density plot for a fine grain quartz sample (BT998). Plot output for the function `plot_DeDistribution()`, the aliquots are shown in ascending D_e order. Note: the function does not check if the input values are given in e.g. Gray or seconds, or if the given values are D_e values or not.

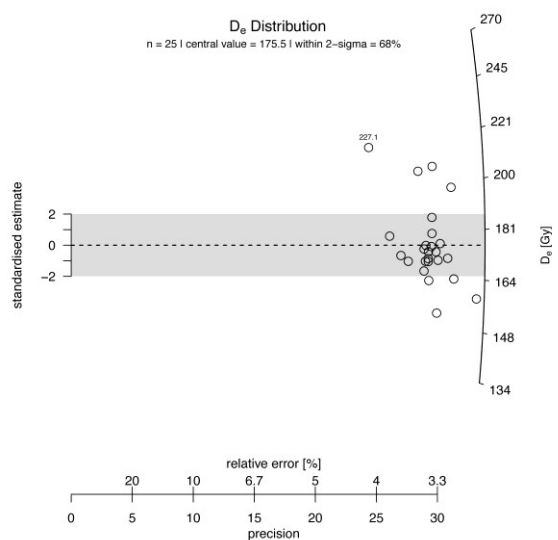


Figure 2: D_e distribution shown as a radial plot for a fine grain quartz sample (BT998) as output of the function `plot_RadialPlot()`. The numeric value within the plot indicates the highest D_e value.

The resulting radial plot is shown in Figure 2. The function is based on an **S** script of Rex Galbraith, but has been rewritten to reduce the needed manual adjustments.

Using generic functions of **R**, the plot can be further exported into several formats (e.g. JPEG, SVG, PDF). For the given function an export as PDF requires just a minor extension, e.g. by stating the desired file format with the arguments of output path and plot size:

```
> pdf("~/Desktop/Figure1.pdf", paper=
+ "special")
+ plot_DeDistribution(ExampleData.DeValues,
+ xlab="s")
+ dev.off()
```

For a more general discussion of the advantages and disadvantages of the different plot types the reader is referred to Galbraith and Roberts (in press).

Example 3: LM-OSL curve fitting

Since the introduction of optically stimulated luminescence (OSL) dating by Huntley et al. (1985), it has been reported that the continuous wave (CW) OSL signals from quartz decay non-exponentially. Bailey et al. (1997) postulated that the quartz signal consists of, at least, three distinct components (termed as fast, medium and slow) with different bleaching and dose-response characteristics. In 1996, Bulur (1996) presented an alternative read-out method by ramping the stimulation intensity over time, termed the linear modulation technique (LMT or LM-OSL). The obtained peak-shaped curve is supposed to be associated with the successive release of electrons from traps/components with increasing optical stability (i.e. decreasing detrapping probability) during measurement (e.g. Bulur 2000).

As a result of the different and sometimes disadvantageous component characteristics (e.g. bleachability, thermal stability), for routine quartz sediment dating the underlying assumption is that the chosen integral of the bulk luminescence signal is dominated by an easy-to-bleach signal component and the contribution of other components can be minimised, e.g. by early background subtraction (Ballarini et al. 2007). Another approach is the isolation of the fast-component by direct measurement (Bailey 2010) or decomposition of the signal by mathematical curve fitting (CW- or LM-OSL curve fitting).

Besides the discussion of the difficulties resulting from the mathematical fitting itself (e.g. Istratov and Vyvenko 1999, Bailey 2010, Bailey et al. 2011), the limitation of this method for routine applications seems to be more of a practical nature. The decomposition of a single LM-OSL curve usually involves a multistep process using different

programs. Therefore, attempting to fit multiple curves is a time consuming process. During the last years a few attempts were presented based on proprietary or self-written software (e.g. Singarayer 2002, Choi et al. 2006, Bailey 2008) to fit LM-OSL curves or CW-OSL curves (e.g. Rowan et al. 2012). To offer an alternative, and to fit a batch of curves automatically, we decided to implement a fitting routine, making the fitting process much more flexible in the way the background is subtracted or the plot output is produced. This approach is transparent and applicable for extensive data sets and further analysis. To avoid any confusion: We did not develop a new mathematical fitting algorithm, but rather used the implemented functions of **R** to write a new function (`fit_LMData()`) that covers the needs in the context of OSL dating of quartz with references to the cited literature. Therefore, we employed the internal non-linear least-square fitting function `nls()` with the `port` algorithm (<http://www.netlib.org/port/>) and the 1st order kinetic function given by Kitis and Pagonis (2008). More information is provided on the help pages of the package. As input file, at least one `data.frame` containing LM-curve data (measured data) with two columns (time, counts) is required:

```
> values.curve
      time counts
1      1     48
2      2     19
3      3     23
.      .      .
.      .      .
4000 4000    530
```

The values from an LM-OSL measurement of a coarse grain (90 – 250 μm) quartz sample from Norway (BT900, Fuchs et al. in press) are shown. To run a fit to the `data.frame` (`values.curve`) trying a 3-component function on a log-time scale (x-axis) the code in **R** is:

```
> fit_LMCurve(values=values.curve,n.
+ components=3,log_scale="x")
```

The resulting plot for sample BT900 is shown in Figure 3. If an additional curve for the measured background (2nd LM-curve) is provided as a two column `data.frame`, here termed as `values.curveBG`, the background is fitted using a polynomial function and subtracted automatically from the first curve. The result is shown in Figure 4, and the corresponding call in **R** is:

```
>fit_LMCurve(values=values.curve, values.bg=
+ values.curveBG,+n.components=3,
+ log_scale="x", output.plotBG=TRUE)
```

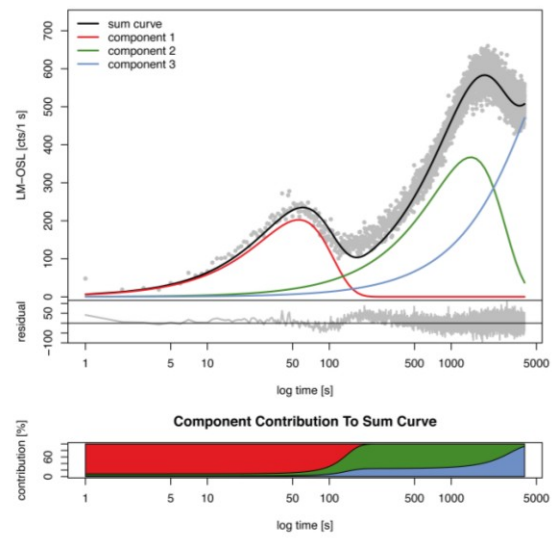


Figure 3: LM-OSL curve from a coarse grain quartz sample (BT900) fitted with a 3-component function plotted using the function `plot_LMCurve()` without background subtraction. The lower plot shows the contribution of each component to the light sum.

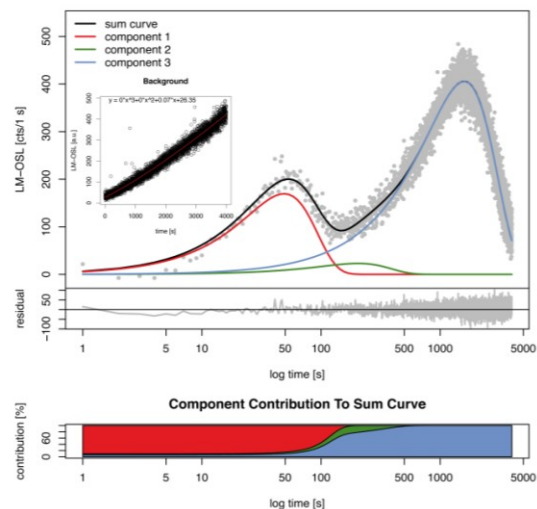


Figure 4: LM-OSL curve from a coarse grain quartz sample (BT900) fitted with a 3-component function using the function `plot_LMCurve()` including background subtraction. The lower plot shows the contribution of each component to the light sum. The fitted background signal (separate curve) is shown in the inset.

Along with the optional plot and terminal output (not shown) a `nls()` object is returned. This **R** object contains all data from the fit (e.g. parameters,

residuals) and can be used for further analysis using generic functions of **R**, e.g. calculating confidence intervals for the fitted parameters (`confint()`). The start parameters for the fitting are estimated automatically, but also manual start parameters will be accepted. With this, for instance, the function can be used in a loop to test different sets of start parameters and compare them with the obtained quality of the fit (i.e. pseudo- R^2) or to fit many curves in one run. An example for a fitting loop over all LM-curves of a measurement is given in the supplementary data.

To compare the results with other software solutions (obtained b_i and n_i values for a given number of components), we conducted a survey comparing the results from this **R** package with results returned by the software from Diana Bailey (FitBin9, Bailey 2008) and Grzegorz Adamiec (Hybfit, principle described in Bluszcz and Adamiec 2006). Three different samples (Supplementary Table 1) were used for this study: (a) a sample from a beach deposit in Norway (BT900, Fuchs et al. in press), (b) a Mol-sand sample from Belgium (MOL1, Gullentops and Vandenberghe 1995) and (c) an archaeological artefact (chalcedony) from Romania (Rom16, Schmidt et al., in prep.). The LM-OSL curves of the samples were fitted with and without background subtraction using the automatic start parameter recognition option of the software for a given number of components. For the investigated LM-OSL curves without background subtraction we observed that the shapes of the curves were indistinguishable. The use of the software's background subtraction option results in differences in the fitting parameters, probably due to the way of background subtraction (Supplementary Table 2). However, if no background subtraction is applied, the output of the software visually appears to be very similar. A typical LM fit for sample BT900 is shown in Figure 5. Detailed results of the comparison are given in the supplementary.

Discussion

This paper briefly describes an **R** package for luminescence dating and reports the advantages compared to existing software. The **R** programming language, although intuitive to users having basic knowledge in numerical programming languages (e.g. *MATLAB*TM or *Mathematica*TM), is not as user friendly as the *Analyst*, the *Risø Viewer*, *MS Excel*TM or *SigmaPlot*TM. However such a comparison would be misleading. **R** allows the creation of complex and transparent data analysis routines for experimental protocols that are not available in existing software. The presented package is not standalone software and is not intended to be such. The presented package provides a collection of functions, serving as a

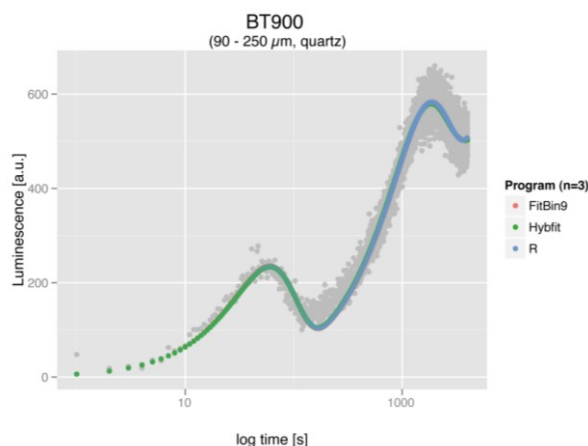


Figure 5: LM-OSL fitting using a 3-component function for the quartz sample BT900 on a log scale. The sum curves show that all programs produce similar fits and therefore the three curves are indistinguishable.

flexible, modular and easy to extend toolbox for the purpose of luminescence dating with special focus on the analysis of non-standardised measurements. The comparison of the fitting programs (**R**, FitBin9 and Hybfit) showed that by default **R** is bundled with powerful libraries and functions like the used `nls()` function producing reliable fitting results. Nevertheless, this is no out-of-the box tool and has to be adjusted for the purpose of luminescence dating. In contrast, using the *Analyst* software is a straightforward process for routine luminescence data analysis and therefore every day in use in our laboratories.

In summary, our contribution is intended to point to the possibilities provided by **R**, the existence of the 'Luminescence' package and to invite other **R** users to contribute and further improve the package. Not least our work may encourage others to start working with **R**.

Conclusions

An **R** package ('Luminescence') for luminescence dating data analysis has been presented and the basic features were shown by three examples. The **R** package is provided under the General Public Licence (GPL)¹ conditions and is available for free. The package in its latest version including the manual is distributed over the Comprehensive **R** Archive Network (CRAN)².

The reader may also have noticed that in the luminescence dating community several **R/S** scripts

¹ <http://www.gnu.org/licenses/gpl-3.0.html>

² <http://cran.r-project.org/web/packages/Luminescence/index.html>

(e.g. from Rex Galbraith or Lee Arnold) are circulating. These scripts are not (yet) part of the presented **R** package. Furthermore, herewith we invite everyone to contribute to the package.

Acknowledgments

We are thankful to Rex Galbraith for his permission to publish the radial plot function under GPL and his useful comments on the rewritten plot function. Grzegorz Adamiec is thanked for helpful discussions on the fitting procedure using the program Hybfit. We are furthermore thankful to Diana Bailey for providing the fitting program FitBin9 for the comparison of the programs. The work of the corresponding author was gratefully funded by the DFG (FU 417/7-2).

Supplementary Information

Supplementary Information for this article can be found at www.aber.ac.uk/ancient-tl

References

- Bailey, D.J.E. (2008). Development of LM OSL analysis techniques for applications to optical dating. Unpublished DPhil thesis, University of Oxford.
- Bailey, R.M. (2010). Direct measurement of the fast component of quartz optically stimulated luminescence and implications for the accuracy of optical dating. *Quaternary Geochronology*, **5**, 559–568.
- Bailey, R.M., Smith, B.W., Rhodes, E.J. (1997). Partial bleaching and the decay form characteristics of quartz OSL. *Radiation Measurements*, **27**, 123–136.
- Bailey, R.M., Yukihara, E.G., McKeever, S.W.S. (2011). Separation of quartz optically stimulated luminescence components using green (525 nm) stimulation. *Radiation Measurements*, **46**, 643–648.
- Ballarini, M., Wallinga, J., Wintle, A.G., Bos, A.J.J. (2007). A modified SAR protocol for optical dating of individual grains from young quartz samples. *Radiation Measurements*, **42**, 360–369.
- Bluszcz, A., Adamiec, G. (2006). Application of differential evolution to fitting OSL decay curves. *Radiation Measurements*, **41**, 886–891.
- Bulur, E. (1996). An alternative technique for optically stimulated luminescence (OSL) experiment. *Radiation Measurements*, **26** (5), 701–709.
- Bulur, E. (2000). A simple transformation for converting CW-OSL curves to LM-OSL curves. *Radiation Measurements*, **32**, 141–145.
- Choi, J.H., Duller, G.A.T., Wintle, A.G. (2006). Analysis of quartz LM-OSL curves. *Ancient TL*, **24**, 9–20.
- Daniels, F., Boyd, C.A., Saunders, D.F. (1953). Thermoluminescence as a Research Tool. *Science*, **117** (3040), 343–349.
- Duller, G.A.T. (2007). *Analyst. Manual*, 1–45.
- Fuchs, M., Kreutzer, S., Fischer, M., Sauer, D., Sørensen, R. (in press). OSL and IRSL dating of raised beach sand deposits along the southeastern coast of Norway. *Quaternary Geochronology*, doi:10.1016/j.quageo.2011.11.009.
- Fuchs, M., Lang, A. (2001). OSL dating of coarse-grain fluvial quartz using single-aliquot protocols on sediments from NE Peloponnese, Greece. *Quaternary Science Reviews*, **20**, 783–787.
- Galbraith, R.F. (1988). Graphical display of estimates having differing standard errors. *Technometrics*, **30**, 271–281.
- Galbraith, R. F., Roberts, R. G., Laslett, G. M., Yoshida, H., Olley, J. M. (1999). Optical dating of single and multiple grains of Quartz from Jinnium Rock Shelter, Northern Australia: Part I, Experimental design and statistical models, *Archaeometry*, **41**, 339–364.
- Galbraith, R. F., Roberts, R. G. (in press). Statistical aspects of equivalent dose and error calculation and display in OSL dating: An overview and some recommendations, *Quaternary Geochronology* doi:10.1016/j.quageo.2012.04.020.
- Grögler, N., Houtermans, F.G., Stauffer, H. (1958). Radiation damage as a research tool for geology and prehistory. 5^o Rassegna Internazionale Elettronica E Nucleare, Supplemento Agli Atti Del Congresso Scientifico, **1**, 5–15.
- Gullentops, F., Vandenberghe, N. (1995). Toelichtingen Bij De Geologische Kaart Van Belgie Vlaams Gewest: Kaartblad 17 Mol. 1–66.
- Hornik, K., Leisch, F. (2002). Vienna and R: Love, marriage and the future. *Festschrift - 50 Jahre Österreichische Statistische Gesellschaft*, 61–20.
- Houtermans, F.G., Stauffer, H. (1957). Thermolumineszenz als mittel zur Untersuchung der Temperatur - und Strahlungsgeschichte von Mineralien und Gesteinen. *Helvetica Physica Acta*, **30**, 274–277.
- Huntley, D.J., Godfrey-Smith, D.I., Thewalt, M.L.W. (1985). Optical dating of sediments. *Nature*, **313**, 105–107.
- Huntley, D.J., Lamothe, M. (2001). Ubiquity of anomalous fading in K-feldspars and the measurement and correction for it in optical dating. *Canadian Journal of Earth Sciences*, **38**, 1093–1106.
- Ihaka, R., Gentleman, R. (1996). R: A Language for Data Analysis and Graphics. *J. Computational and Graphical Statistics*, **5**, 299–314.

- Istratov, A.A., Vyvenko, O.F. (1999). Exponential analysis in physical phenomena. *Review of Scientific Instruments*, **70**, 1233–1257.
- Kitis, G., Pagonis, V. (2008). Computerized curve deconvolution analysis for LM-OSL. *Radiation Measurements*, **43**, 737–741.
- Ligges, U. (2008). Programmieren mit R (Statistik und ihre Anwendungen). Springer.
- Murray, A.S., Wintle, A.G. (2000). Luminescence dating of quartz using an improved single-aliquot regenerative-dose protocol. *Radiation Measurements*, **32**, 57–73.
- R Development Core Team (2012). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, <http://www.R-project.org/>.
- Rowan, A.V., Roberts, H.M., Jones, M. A., Duller, G.A.T., Covey-Crump, S.J., Brocklehurst, S.H. (2012). Optically stimulated luminescence dating of glaciofluvial sediments on the Canterbury Plains, South Island, New Zealand, *Quaternary Geochronology*, **8**, 10-22.
- Schmidt, C., Sitlivy, V., Anghelinu, M., Chabai, V., Kels, H., Uthmeier, T., Hauck, T., Bălcean, I., Hilgers, A., Richter, J., Radtke, U. (in prep). Unexpectedly old: thermo-luminescence dating of heated artefacts from the Aurignacian open-air site of Românești-Dumbrăvița I, Romania.
- Singarayer, J.S. (2002). Linearly modulated optically stimulated luminescence of sedimentary quartz: physical mechanisms and implications for dating. Unpublished DPhil thesis, University of Oxford.

Reviewer

G.A.T. Duller

Reviewers Comments

R is increasingly being used for a wide range of scientific applications, both in Earth and Physical Sciences. This excellent package is a very welcome contribution and I hope that it will encourage many luminescence colleagues to explore the use of this powerful software.

The authors have also provided a very valuable help file as part of the Luminescence Package, and this can be accessed from within **R** by typing:

```
help(Luminescence)
```