Package 'numOSL'

July 26, 2018

Type Package

Title Numeric Routines for Optically Stimulated Luminescence Dating

Version 2.6

Date 2018-07-26

Author Jun Peng [aut, cre],
Bo Li [aut],
Jorge More [ctb],
Burton Garbow [ctb],
Kenneth Hillstrom [ctb],
John Burkardt [ctb],
Paul Gilbert [ctb],
Ravi Varadhan [ctb]

Maintainer Jun Peng <pengjun10@mails.ucas.ac.cn>

Description Package for optimizing regular numeric problems in optically stimulated luminescence dating, such as: equivalent dose calculation, dose rate determination, growth curve fitting, decay curve decomposition, statistical age model optimization, and statistical plot visualization.

License GPL-3

Depends R (>= 2.15.3)

Imports graphics, stats, utils, grDevices

URL https://CRAN.R-project.org/package=numOSL

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-07-26 05:20:03 UTC

R topics documented:

`numOSL-package` *Package for tackling basic numeric problems in optically stimulated luminescence dating*

Description

Package for routine numeric optimization and data visualization in optically stimulated luminescence dating.

Details

Package:	numOSL
Type:	Package
Version:	2.6
Date:	2018-07-26
License:	GPL-3

Author(s)

Jun Peng Hunan University of Science and Technology (HNUST), Xiangtan, China

Bo Li University of Wollongong (UOW), Wollongong, Australia

Package maintainer
Jun Peng <pengjun10@mails.ucas.ac.cn>

Related package projects
R program KMS https://github.com/pengjunUCAS/KMS
R package tgcd https://CRAN.R-project.org/package=tgcd

References

Peng J, Dong ZB, Han FQ, Long H, Liu XJ, 2013. R package numOSL: numeric routines for optically stimulated luminescence dating. Ancient TL, 31(2): 41-48.

Peng J, Li Bo, 2017. Single-aliquot Regenerative-Dose (SAR) and Standardised Growth Curve (SGC) Equivalent Dose Determination in a Batch Model Using the R Package 'numOSL'. Ancient TL, 35(2): 32-53.

`analyseBINdata` *BIN data analysis*

Description

Analysing signal data records extracted from a BIN file.

Usage

```
analyseBINdata(obj_pickBIN, nfchn, nlchn, bg = "late",
               me = 2.0, distp = "p", kph = NULL,
               kdc = NULL, dcr = NULL, FR.fchn = NULL,
               FR.mchn = NULL, FR.lchn = NULL,
               signal.type = "LxTx", outfile = NULL)
```

Arguments

`obj_pickBIN`	list(**required**): an object of S3 class `"pickBIN"` produced by function pickBINdata
`nfchn`	integer(**required**): number of the first few channels from the initial part of a decay curve. Number of counts summed over channels `(Delay+1L):(Delay+nfchn)` is calculated as the fast-component plus backgroud signal
`nlchn`	integer(**required**): number of the last few channels from the latter part of a decay curve. If `bg="late"`, number of counts averaged over channels `(Delay+On-nlchn+1L):(Delay+On)` will be calculated as the backgroud signal, if `bg="early"`, number of counts averaged over channels `(Delay+nfchn+1L):(Delay+nfchn+nlchn)` will be calculated as the backgroud signal. `Delay` and `On` are obtained internally from the BIN file.

bg	character(with default): background subtraction method, i.e., bg="early" or bg="late"
me	numeric(with default): measurement error of Lx (or Tx) in percent
distp	character(with default): distribution of photon counts, distp="p" denotes Poisson distribution, distp="op" denotes Over Poisson distribution
kph	numeric(optional): correction factor for photon counts
kdc	numeric(optional): correction factor for dark counts
dcr	numeric(optional): dark count rate
FR.fchn	vector(optional): fast-component signal channels, note that those channels are extracted internally from the "ON" channels, i.e., FR.fchn=((Delay+1):(Delay+On))[FR.fchn]. Example: FR.fchn=1:5
FR.mchn	vector(optional): medium-component signal channels, note that those channels are extracted internally from the "ON" channels, i.e., FR.mchn=((Delay+1):(Delay+On))[FR.mchn]. Example: FR.mchn=31:60
FR.lchn	vector(optional): background signal channels, note that those channels are extracted internally from the "ON" channels, i.e., FR.lchn=((Delay+1):(Delay+On))[FR.lchn]. Example: FR.lchn=201:250
signal.type	character(with default): type of signal, "LxTx", "Lx", or "Tx"
outfile	character(optional): if specified, analysis results (i.e., NO, Position, Grain, SAR.Cycle, Dose, Init, BG, Lx, seLx, TInit, TBG, Tx, seTx, LxTx, seLxTx) will be written to a CSV file named "outfile" and saved to the current work directory

Details

Function analyseBINdata is used for signal (i.e., Lx, Tx, and Lx/Tx) calculation. It provides two protocols for background subtraction (i.e., the early and late background subtraction methods).

Standard error of signals are assessed using two methods: if photon counts are assummed to follow Poisson distributions, **Eqn.(3)** of Galbraith (2002) will be applied; if photon counts are overdispersed, **Eqn.(10)** of Bluszcz et al. (2015) will be applied.

If arguments FR.fchn, FR.mchn, and FR.lchn are provided, fast ratio will be calculated according to Madsen et al. (2009).

Value

Return an invisible list of S3 class object "analyseBIN" containing the following elements:

SARdata	a data.frame containing calculated SAR data sets
criteria	values used as rejection criteria (0-1 values indicating if Tn is more than 3 sigma above BG or not, ratio of initial Tn signal to BG and associated standard error, relative standard error of Tn in percent, fast ratio of Tn and associated standard

	error), NA is produced if the value can not be calculated. Note that in this function rejection criteria are calculated but not applied
Tn	values of Tn and associated standard errors
LnTn.curve	decay curves for Ln and Tn for different aliquots (grains)
TxTn	ratios of Tx to Tn for various SAR cycles
agID	aliquot or grain ID (i.e., NO, Position, and Grain)

SARdata is a data.frame containing the following elements if signal.type="LxTx":

Element	Description
NO	aliquot (grain) number
SAR.Cycle	SAR cycle (N, R1, R2, R3, ...)
Dose	regenerative dose
LxTx	sensitivity-corrected regenerative-dose signal
seLxTx	standard error of LxTx

SARdata contains the following elements if signal.type="Lx":

Element	Description
NO	aliquot (grain) number
SAR.Cycle	SAR cycle (N, R1, R2, R3, ...)
Dose	regenerative dose
Lx	regenerative-dose signal
seLx	standard error of Lx

SARdata contains the following elements if signal.type="Tx":

Element	Description
NO	aliquot (grain) number
SAR.Cycle	SAR cycle (N, R1, R2, R3, ...)
Dose	regenerative dose
Tx	test-dose signal
seTx	standard error of Tx

Note

Though function analyseBINdata is originally designed to analyze CW-OSL data sets, IRSL data sets obtained from the SAR protocol can also be analyzed.

References

Ballarini M, Wallinga J, Wintle AG, Bos AJJ, 2007. A modified SAR protocol for optical dating of individual grains from young quartz samples. Radiation Measurements, 42(3): 360-369.

Bluszcz A, Adamiec G, Heer AJ, 2015. Estimation of equivalent dose and its uncertainty in the OSL SAR protocol when count numbers do not follow a Poisson distribution. Radiation Measurements, 81: 46-54.

Cunningham AC, Wallinga J, 2010. Selection of integration time intervals for quartz OSL decay curves. Quaternary Geochronology, 5(6): 657-666

Duller GAT, 2016. Analyst (v4.31.9), User Mannual.

Durcan JA, Duller GAT, 2011. The fast ratio: A rapid measure for testing the dominance of the fast component in the initial OSL signal from quartz. Radiation Measurements, 46(10): 1065-1072.

Galbraith R, 2002. A note on the variance of a backround-corrected OSL count. Ancient TL, 20(2): 49-51.

Madsen AT, Duller GAT, Donnelly JP, Roberts HM, Wintle AG, 2009. A chronology of hurricane landfalls at Little Sippewissett Marsh, Massachusetts, USA, using optical dating. Geomorphology, 109(1-2): 36-45.

See Also

loadBINdata; pickBINdata; pickSARdata;
calED; calSARED; calSGCED;
fitGrowth; lsNORM; BIN

Examples

```
### Example 1 (not run):
# obj_loadBIN <- loadBINdata("foo.bin", view=TRUE)
# obj_pickBIN <- pickBINdata(obj_loadBIN, Position=2, LType="OSL")
# analyseBINdata(obj_pickBIN, nfchn=3, nlchn=20)

### Example 2:
data(BIN)
obj_pickBIN <- pickBINdata(BIN, Position=c(2,4,6,8,10),
                           LType="OSL", view=FALSE)
obj_analyseBIN <- analyseBINdata(obj_pickBIN, nfchn=3, nlchn=20)
obj_analyseBIN$SARdata
```

---

as_analyseBIN *Transfom SAR data sets into S3 class object "analyseBIN"*

Description

Transfom SAR data sets into S3 class object "analyseBIN".

Usage

```
as_analyseBIN(SARdata)
```

Arguments

SARdata	matrix(**required**): SAR data set, it should contain five columns (i.e., NO, SAR.Cycle, Dose, Signal, and Signal.Err), see SARdata for details

Value

Return an invisible list of S3 class object "analyseBIN" containing the following elements:

SARdata	a data.frame containing SAR data sets
criteria	values used as rejection criteria, here it is set equal to NULL
Tn	values of Tn and associated standard errors, here it is set equal to NULL
LnTn.curve	decay curves of Ln and Tn for different aliquots (grains), here it is set equal to NULL
TxTn	ratios of Tx to Tn for various SAR cycles, here it is set equal to NULL
agID	aliquot or grain ID (i.e., NO, Position, and Grain), here both Position and Grain are set equal to 0

SARdata is a data.frame containing the following elements:

Element	Description
NO	aliquot (grain) number
SAR.Cycle	SAR cycle (N, R1, R2, R3, ...)
Dose	regenerative dose
Signal	OSL signal
Signal.Err	standard error of OSL signal

Note

Function as_analyseBIN transforms SAR data sets (see SARdata) into S3 class object "analyseBIN". The returned elements such as criteria, Tn, LnTn.curve, and TxTn are set equal to NULL.

See Also

analyseBINdata; SARdata;
calSARED; pickSARdata

Examples

```
### Example 1:
data(SARdata)
obj_analyseBIN <- as_analyseBIN(SARdata[1:8,,drop=FALSE])
res_calSARED <- calSARED(obj_analyseBIN)
res_calSARED$sarED

### Example 2 (not run):
# obj_analyseBIN <- as_analyseBIN(SARdata)
# res_calSARED <- calSARED(obj_analyseBIN, rcy1.range=c(1,1), outpdf="SARED")

### Example 3 (not run):
# obj_analyseBIN <- as_analyseBIN(SARdata)
# res_pickSARdata <- pickSARdata(obj_analyseBIN, fom.up=6, outpdf="SARdata")
# res_pickSARdata$SARdata
```

BIN *BIN data*

Description

BIN data for aeolian sample GL2-1 from the south margin of the Tengger Desert (Peng et al., 2013).

Usage

```
data(BIN)
```

Format

A S3 class object "loadBIN" produced by function loadBINdata that contains the following two elements:

records a list consists of loaded data records for each aliquot (grain)

tab a data.frame used for summarizing loaded data records

References

Peng J, Han FQ, 2013. Selections of fast-component OSL signal using sediments from the south edge of Tengger Desert. Acta Geoscientica Sinica, 34(6): 757-762.

See Also

loadBINdata; pickBINdata; analyseBINdata

Examples

```
# Not run.
# data(BIN)
# class(BIN)
```

calDA *Annual dose rate (age) calculation*

Description

Calculating the annual dose rate and burial age.

Usage

```
calDA(dose, minGrainSize, maxGrainSize, Ucontent, Thcontent,
      Kcontent, Wct, depth, altitude, latitude, longitude,
      bulkDensity = 2.5, alphaValue = 0.03, nsim = 10000,
      rdcf = 0.05, rba = 0.05, plot = TRUE)
```

Arguments

dose	vector(**required**): equivalent dose and associated measurement error (unit, Gy)
minGrainSize	numeric(**required**): lower limit on grain size (unit, um)
maxGrainSize	numeric(**required**): upper limit on grain size (unit, um)
Ucontent	vector(**required**): uranium content and its measurement error (unit, ppm)
Thcontent	vector(**required**): thorium content and its measurement error (unit, ppm)
Kcontent	vector(**required**): potassium content and its measurement error (unit, 1 percent)
Wct	vector(**required**): water content and its measurement error (unit, 1 percent)
depth	numeric(**required**): sampling depth (unit, m)
altitude	numeric(**required**): altitude of the sampling site (unit, m above sea level)
latitude	numeric(**required**): latitude of the sampling site (unit, decimal degree)
longitude	numeric(**required**): longitude of the sampling site (unit, decimal degree)
bulkDensity	numeric(with default): average density of bulk sample (unit, g/cm^3)
alphaValue	numeric(with default): average alpha efficiency
nsim	integer(with default): number of Monte Carlo iterations
rdcf	numeric(with default): constant relative standard error for dose-rate conversion factors (unit, 1 percent)
rba	numeric(with default): constant relative standard error for beta dose absorption fraction (unit, 1 percent)
plot	logical(with default): draw a plot or not

Details

Function calDA is used for calculating the annual dose rate and burial age using concentrations of radioactive nuclides (uranium, thorium, potassium) obtained from Neutron Activation Analysis (NAA), grain size, water content, average sample density, geographical parameters (depth, altitude, latitude, longitude), and an equivalent dose value. The elemental concentrations are converted into annual dose rate according to dose-rate conversion factors (Adamiec and Aitken, 1998). Beta dose absorded fractions are calculated through linear interpolation using published data (Fain et al., 1999). The cosmic ray dose rate is estimated as a function of depth, altitude and geomagnetic latitude (Prescott and Hutton, 1994).

The standard error of an annual dose rate is estimated by a "parametric bootstrap" method. To perform the simulation, dose-rate conversion factors and beta dose absorption factor are assumed to have constant relative standard errors. Arguments such as `dose`, `Ucontent`, `Thcontent`, `Kcontent`, `wct` should be two-element vectors with the form `c(value,sd)`, as their measurement uncertainties are major error sources. Arguments such as `depth`, `altitude`, `latitude`, `longitude`, `bulkDensity`, `alphaValue` can be either a scalar of a two-element vector. This means that uncertainties from these quantities can be either ignored or taken into account during the simulation.

Value

Return a matrix that contains calculated annual dose rate and age

References

Adamiec G, Aitken M, 1998. Dose-rate conversion factors: update. Ancient TL, 16(2): 37-49.

Fain J, Soumana S, Montret M, Miallier D, Pilleyre T, Sanzelle S, 1999. Luminescence and ESR dating Beta-dose attenuation for various grain shapes calculated by a Monte-Carlo method. Quaternary Science Reviews, 18(2): 231-234.

Prescott JR, Hutton JT, 1994. Cosmic ray contributions to dose rates for Luminescence and Esr dating: large depths and long-term time variations. Radiation Measurements, 23(2-3): 497-500.

Examples

```
calDA(dose=c(25.04,0.68), minGrainSize=90,
      maxGrainSize=180, Ucontent=c(2.86,0.19),
      Thcontent=c(8.63,0.34), Kcontent=c(2.00,0.11),
      Wct=c(0.05,0.05), depth=c(1.1,0.05), altitude=c(1170,58.5),
      latitude=c(37.64,1.88), longitude=c(103.16,5.16),
      bulkDensity=c(2.5,0.1), nsim=10000, rdcf=0.05, rba=0.05)
```

---

calED *Equivalent dose calculation and error assessment*

Description

Calculating an equivalent dose and assessing its standard error.

Usage

```
calED(Curvedata, Ltx, model = "gok", origin = FALSE,
      errMethod = "sp", nsim = 500, weight = TRUE,
      trial = FALSE, plot = TRUE, nofit.rgd = NULL,
      agID = NULL, Tn = NULL, Tn3BG = NULL,
      TnBG.ratio = NULL, rseTn = NULL, FR = NULL,
      LnTn.curve = NULL, TxTn = NULL)
```

Arguments

`Curvedata`	matrix(**required**): a three-column matrix (i.e., regenerative doses, sensitivity-corrected regenerative-dose signals, and associated standard errors)
`Ltx`	vector(**required**): a two-element vector consists of sensitivity-corrected natural-dose signal and its error
`model`	character(with default): model used for growth curve fitting, see fitGrowth for available models
`origin`	logical(with default): logical value indicating if the growth curve should be forced to pass the origin
`errMethod`	character(with default): method used for equivalent dose error assessment. `"sp"` and `"mc"` denote error estimation using the Simple Transformation and Monte Carlo methods, respectively

nsim	integer(with default): desired number of randomly simulated equivalent dose obtained by Monte Carlo simulation
weight	logical(with default): logical value indicating if the growth curve should be fitted using a weighted procedure, see function fitGrowth for details
trial	logical(with default): logical value indicating if the growth curve should be fitted using other models if the given model fails, see function fitGrowth for details
plot	logical(with default): logical value indicating if the results should be plotted
nofit.rgd	integer(optional): regenerative doses that will not be used during the fitting. For example, if nofit.rgd=1 then the first regenerative dose will not be used during growth curve fitting
agID	vector(optional): a three-elemenet vector indicating aliquot (grain) ID, i.e., agID[1]=NO, agID[2]=Position, agID[3]=Grain
Tn	vector(optional): a two-element vector containing value and standard error of Tn
Tn3BG	numeric(optional): 0-1 value indicating if Tn is more than 3 sigma above BG, 1 indicates Tn>3_sigma_BG, 0 indicates Tn<=3_sigma_BG
TnBG.ratio	vector(optional): a two-element vector containing value and standard error of ratio of initial Tn signal to BG
rseTn	numeric(optional): relative standard error of Tn in percent
FR	vector(optional): a two-element vector containing value and standard error of fast ratio of Tn
LnTn.curve	list(optional): decay curve data for Ln and Tn, it should contain four elements, i.e., names(LnTn.curve)=c("Ln.x","Ln.y","Tn.x","Tn.y")
TxTn	vector(optional): ratios of Tx to Tn for various SAR cycles

Details

Function calED is used for calculating an equivalent dose and assessing its standard error. The standard errors of an equivalent dose can be assessd using the Simple Transformation or Monte Carlo method (Duller, 2007).

Interpolation is performed using a combination of golden section search and successive parabolic interpolation (***R*** function optimize in package ***stats***) (freely available Fortran 77 source code at http://www.netlib.org/fmm/fmin.f). See function fitGrowth for more details on growth curve fitting.

Value

Return an invisible list that contains the following elements:

message	return 0 if calculation succeeds, 1 if growth curve fitting fails, 2 if natural-dose signal saturates, 3 if equivalent dose calculation fails, 4 if equivalent dose error assessment fails
fitIDX	Indices of dose points used in growth curve fitting
LMpars	optimized parameters for the growth curve

value	minimized objective for the growth curve
avg.error	average fit error for the growth curve
RCS	reduced chi-square value for the growth curve
FOM	figure of merit value for the growth curve in percent
calED.method	method used for equivalent dose calculation, i.e., "Interpolation" or "Extrapolation"
mcED	randomly simulated equivalent doses
ED	calculated equivalent dose and its standard error
ConfInt	68 percent and 95 percent confidence intervals for the equivalent dose
RecyclingRatio1	the first recycling ratio and its standard error
RecyclingRatio2	the second recycling ratio and its standard error
RecyclingRatio3	the third recycling ratio and its standard error
Recuperation1	the first recuperation (i.e., ratio of the sensitivity-corrected zero-dose signal to natural-dose signal) and its standard error in percent
Recuperation2	the second recuperation (i.e., ratio of the sensitivity-corrected zero-dose signal to maximum regenerative-dose signal) and its standard error in percent

Note

Arguments agID, Tn, Tn3BG, TnBG.ratio, rseTn, FR, LnTn.curve, and TxTn have nothing to do with equivalent dose calculation. They are used only for plotting purpose.

Argument Tn3BG indicates if Tn (after background subtraction) is more than 3 sigma above BG, while argument TnBG.ratio denotes the ratio of Tn (no background subtraction) to BG.

Function calED will return message=3 (i.e.,"Failed in equivalent dose calculation") if the equivalent dose to be calculated below -50 (<Gy>/<s>).

68 percent (one sigma) and 95 percent (two sigma) confidence intervals of equivalent doses will be determined respectively using normal approximation and Monte Carlo simulation, for errMethod="sp" and errMethod="mc".

Function sgcED in previous versions was bundled to function calSGCED.

References

Duller GAT, 2007. Assessing the error on equivalent dose estimates derived from single aliquot regenerative dose measurements. Ancient TL, 25(1): 15-24.

Duller GAT, 2016. Analyst (v4.31.9), User Mannual.

Galbraith RF, Roberts RG, 2012. Statistical aspects of equivalent dose and error calculation and display in OSL dating: an overview and some recommendations. Quaternary Geochronology, 11: 1-27.

See Also

analyseBINdata; fitGrowth; calRcyRcp;
calSARED; fastED; calSGCED

Examples

```
### Example 1:
Curvedata <- cbind(c(0, 18, 36, 54, 72, 0, 18),
                   c(0.026, 1.55, 2.39, 3.46, 4.13, 0.023, 1.61),
                   c(0.005, 0.11, 0.27, 0.22, 0.20, 0.008, 0.24))
Ltx <- c(3.1,0.31)
calED(Curvedata, Ltx, model="exp", origin=FALSE)

### Example 2 (not run):
# data(BIN)
# obj_pickBIN <- pickBINdata(BIN, Position=48,
#                            LType="OSL", view=FALSE)
# obj_analyseBIN <- analyseBINdata(obj_pickBIN, nfchn=3, nlchn=20)
# Curvedata <- obj_analyseBIN$SARdata[-1,3:5]
# Ltx <- as.numeric(obj_analyseBIN$SARdata[1,4:5])
# calED(Curvedata, Ltx, model="gok", origin=FALSE)
```

`calRcyRcp` *Recycling ratio and recuperation calculation*

Description

Calculating recycling ratio, recuperation, and associated standard errors.

Usage

```
calRcyRcp(Curvedata, Ltx)
```

Arguments

`Curvedata`	matrix(**required**): a three-column matrix (i.e., regenerative doses, sensitivity-corrected regenerative-dose signals, and associated standard errors)
`Ltx`	vector(**required**): a two-element vector consists of sensitivity-corrected natural-dose signal and its error

Value

Return a list that contains the following elements:

`RecyclingRatio1`
: the first recycling ratio and its standard error

`RecyclingRatio2`
: the second recycling ratio and its standard error

`RecyclingRatio3`
: the third recycling ratio and its standard error

`Recuperation1`
: the first recuperation (i.e., ratio of the sensitivity-corrected zero-dose signal to natural-dose signal) and its standard error in percent

`Recuperation2`
: the second recuperation (i.e., ratio of the sensitivity-corrected zero-dose signal to maximum regenerative-dose signal) and its standard error in percent

Note

If the sensitivity-corrected signals for the frist, second, and third repeated regenerative doses are R1, R2, and R3, respectively, then `RecyclingRatio1=R2/R1`, `RecyclingRatio2=R3/R1`, and `RecyclingRatio3=R3/R2`.

References

Wintle AG, Murray AS, 2006. A review of quartz optically stimulated luminescence characteristics and their relevance in single-aliquot regeneration dating protocols. Radiation Measurements, 41(4): 369-391.

See Also

calED; fastED; calSARED; pickSARdata

`calSARED` *SAR equivalent doses calculation and selection*

Description

Calculating and selecting a series of equivalent doses in a batch mode according to the single aliquot regenerative-dose (SAR) method (Murray and Wintle, 2000).

Usage

```
calSARED(obj_analyseBIN, model = "gok", origin = FALSE,
         errMethod = "sp", nsim = 500, weight = TRUE,
         trial = TRUE, nofit.rgd = NULL, Tn.above.3BG = TRUE,
         TnBG.ratio.low = NULL, rseTn.up = NULL, FR.low = NULL,
         rcy1.range = NULL, rcy2.range = NULL, rcy3.range = NULL,
         rcp1.up = NULL, rcp2.up = NULL, fom.up = NULL,
         rcs.up = NULL, calED.method = NULL, rseED.up = NULL,
         use.se = TRUE, outpdf = NULL, outfile = NULL)
```

Arguments

obj_analyseBIN	list(**required**): an object of S3 class "analyseBIN" produced by function analyseBINdata or as_analyseBIN
model	character(with default): model used for growth curve fitting, see fitGrowth for available models
origin	logical(with default): logical value indicating if the growth curve should be forced to pass the origin
errMethod	character(with default): method used for equivalent dose error assessment. See function calED for details
nsim	integer(with default): desired number of randomly simulated equivalent dose obtained by Monte Carlo simulation
weight	logical(with default): logical value indicating if the growth curve should be fitted using a weighted procedure, see function fitGrowth for details
trial	logical(with default): logical value indicating if the growth curve should be fitted using other models if the given model fails, see function fitGrowth for details
nofit.rgd	integer(optional): regenerative doses that will not be used during the fitting. For example, if `nofit.rgd=6` then the sixth regenerative dose will not be used during growth curve fitting
Tn.above.3BG	logical(with default): logical value indicating if aliquot (grain) with Tn below 3 sigma BG should be rejected
TnBG.ratio.low	numeric(optional): lower limit on ratio of initial Tn signal to BG
rseTn.up	numeric(optional): upper limit on relative standard error of Tn in percent
FR.low	numeric(optional): lower limit on fast ratio of Tn
rcy1.range	vector(optional): a two-element vector indicating the lower and upper limits on recycling ratio 1, Example: `rcy1.range=c(0.9,1.1)`
rcy2.range	vector(optional): a two-element vector indicating the lower and upper limits on recycling ratio 2
rcy3.range	vector(optional): a two-element vector indicating the lower and upper limits on recycling ratio 3
rcp1.up	numeric(optional): upper limit on recuperation 1 (i.e., ratio of the sensitivity-corrected zero-dose signal to natural-dose signal) in percent
rcp2.up	numeric(optional): upper limit on recuperation 2 (i.e., ratio of the sensitivity-corrected zero-dose signal to maximum regenerative-dose signal) in percent
fom.up	numeric(optional): upper limit on figure of merit (FOM) values of growth curves in percent
rcs.up	numeric(optional): upper limit on reduced chi-square (RCS) values of growth curves
calED.method	character(optional): method used for equivalent dose calculation, i.e., `"Interpolation"` or `"Extrapolation"`
rseED.up	numeric(optional): upper limit on the relative standard error of equivalent dose in percent

use.se	logical(with default): logical value indicating if standard errors of values should be used during application of rejection criteria
outpdf	character(optional): if specified, results of SAR equivalent dose calculation will be written to a PDF file named "outpdf" and saved to the current work directory
outfile	character(optional): if specified, SAR equivalent doses related quantities will be written to a CSV file named "outfile" and saved to the current work directory

Value

Return an invisible list that contains the following elements:

LMpars	a list containing optimized parameters of growth curves of calculated (selected) SAR equivalent doses
Tn	values and standard errors of Tn of calculated (selected) SAR equivalent doses
Ltx	sensitivity-corrected natural-dose signals and associated standard errors used for SAR equivalent dose calculation
sarED	calculated (selected) SAR equivalent doses and associated standard errors
ConfInt	68 percent (one sigma) and 95 percent (two sigma) confidence intervals of SAR equivalent doses
agID	aliquot (grain) ID of calculated (selected) SAR equivalent doses
summary.info	a summary of the SAR equivalent dose calculation

Note

Rejection criteria used to select reliable SAR equivalent dose estimates can be catergorized into three groups:
(1) signal-related criteria, such as Tn.above.3BG, TnBG.ratio.low, rseTn.up, and FR.low;
(2) growth-curve-related criteria, such as rcy1.range, rcy2.range, rcy3.range, rcp1.up, rcp2.up, fom.up, and rcs.up;
(3) equivalent-dose-related criteria, such as calED.method and rseED.up.

References

Duller GAT, 2016. Analyst (v4.31.9), User Mannual.

Murray AS, Wintle AG, 2000. Luminescence dating of quartz using improved single-aliquot regenerative-dose protocol. Radiation Measurements, 32(1): 57-73.

Wintle AG, Murray AS, 2006. A review of quartz optically stimulated luminescence characteristics and their relevance in single-aliquot regeneration dating protocols. Radiation Measurements, 41(4): 369-391.

See Also

analyseBINdata; fitGrowth;
calED; calSGCED; pickSARdata

Examples

```
data(BIN)
obj_pickBIN <- pickBINdata(BIN, Position=c(2,4,6,8,10), Grain=0,
                           LType="OSL", view=FALSE)
obj_analyseBIN <- analyseBINdata(obj_pickBIN, nfchn=3, nlchn=20)
res_SARED <- calSARED(obj_analyseBIN, model="exp", origin=FALSE)
# plot(res_SARED$Tn[,1], res_SARED$sarED[,1], xlab="Tn", ylab="ED (<Gy>|<s>)")
```

---

calSGCED *SGC Equivalent dose calculation and selection*

Description

Calculating and selecting equivalent doses in a batch model according to the "standardised growth curve" (SGC) method suggested by Roberts and Duller (2004) or the "global standardised growth curve" (gSGC) method suggested by Li et al. (2015, 2016).

Usage

```
calSGCED(obj_analyseBIN, SGCpars, model, origin, avgDev,
         method = "SGC", SAR.Cycle = "N", errMethod = "sp",
         Tn.above.3BG = TRUE, TnBG.ratio.low = NULL,
         rseTn.up = NULL, FR.low = NULL, rseED.up = NULL,
         use.se = TRUE, outpdf = NULL, outfile = NULL)
```

Arguments

`obj_analyseBIN`	list(**required**): an object of S3 class "analyseBIN" produced by function analyseBINdata or as_analyseBIN
`SGCpars`	vector(**required**): optimized parameters of the SGC obtained using function lsNORM (or fitGrowth)
`model`	character(**required**): fitting model used for obtaining `SGCpars`
`origin`	logical(**required**): logical value indicating if established SGC passes the origin
`avgDev`	numeric(**required**): average deviation (i.e., average fit error) of the SGC obtained using function fitGrowth or lsNORM. This quantity stands for the uncertainty of established SGC when assessing the equivalent dose error using the simple transformaion method
`method`	character(with default): method used for equivalent dose calculation, i.e., `method="SGC"` (for the original SGC method) or `method="gSGC"` (for the improved SGC method)
`SAR.Cycle`	character(with default): SAR cycles used for SGC equivalent dose calculation. Example: `SAR.Cycle=c("N","R3")`
`errMethod`	character(with default): method used for equivalent dose error assessment
`Tn.above.3BG`	logical(with default): logical value indicating if aliquot (grain) with Tn below 3 sigma BG should be rejected

TnBG.ratio.low	numeric(optional): lower limit on ratio of initial Tn signal to BG
rseTn.up	numeric(optional): upper limit on relative standard error of Tn in percent
FR.low	numeric(optional): lower limit on fast ratio of Tn
rseED.up	numeric(optional): upper limit on the relative standard error of equivalent dose in percent
use.se	logical(with default): logical value indicating if standard errors of values should be used during application of rejection criteria
outpdf	character(optional): if specified, results of SGC equivalent dose calculation will be written to a PDF file named "outpdf" and saved to the current work directory
outfile	character(optional): if specified, SGC equivalent doses related quantities will be written to a CSV file named "outfile" and saved to the current work directory

Value

Return an invisible list that contains the following elements:

scale.Ltx	scaled standardised natural-dose signals and associated standard errors used for SGC equivalent dose calculation. Note that standardised natural-dose signals will remain un-scaled if method="SGC"
sgcED	calculated SGC equivalent doses
ConfInt	68 percent (one sigma) and 95 percent (two sigma) confidence intervals of SGC equivalent doses
agID	aliquot (grain) ID of calculated (selected) SGC equivalent doses
summary.info	a summary of the SGC equivalent dose calculation

References

Li B, Roberts RG, Jacobs Z, Li SH, 2015. Potential of establishing a "global standardised growth curve" (gSGC) for optical dating of quartz from sediments. Quaternary Geochronology, 27: 94-104.

Li B, Jacobs Z, Roberts RG, 2016. Investigation of the applicability of standardised growth curves for OSL dating of quartz from Haua Fteah cave, Libya. Quaternary Geochronology, 35: 1-15.

Roberts HM, Duller GAT, 2004. Standardised growth curves for optical dating of sediment using multiple-grain aliquots. Radiation Measurements, 38(2): 241-252.

See Also

fitGrowth; lsNORM; SARdata;
scaleSGCN; calED; calSARED

Examples

```
data(SARdata)
### (1) gSGC ED calculation:
### gSGCpars were obtained using function "lsNORM".
gSGCpars <- c(137.440874251, 0.007997863, 2.462035263, -0.321536177)
avg.error2 <- 0.1111623
```

```
res <- calSGCED(as_analyseBIN(SARdata), gSGCpars, method="gSGC",
                model="gok", origin=FALSE, avgDev=avg.error2,
                SAR.Cycle=c("N","R3"))
print(res$sgcED)

### (2) SGC ED calculation (not run):
### SGCpars were obtained using function "fitGrowth".
# SGCpars <- c(183.474322547,  0.007038048,  4.254287622, -0.337734151)
# avg.error <- 0.3156259
# calSGCED(as_analyseBIN(SARdata), SGCpars, method="SGC", model="gok",
#          origin=FALSE, avgDev=avg.error, SAR.Cycle="N", outpdf="SGCED")

### (3) gSGC ED calculation and signal-related
###     rejection criteria application (not run):
# data(BIN)
# res_pickBIN <-pickBINdata(BIN, LType="OSL")
# res_analyseBIN <- analyseBINdata(res_pickBIN, nfchn=4, nlchn=30)
# res_lsNORM <- lsNORM(res_analyseBIN$SARdata, model="gok", origin=FALSE)

# calSGCED(res_analyseBIN, SGCpars=res_lsNORM$LMpars2[,1],
#         model="gok", origin=FALSE, avgDev=res_lsNORM$avg.error2,
#         method="gSGC", SAR.Cycle=c("N","R3"), Tn.above.3BG=TRUE,
#         TnBG.ratio.low=4, rseTn.up=10, outpdf="foo", outfile="foo")
```

---

dbED *De distribution summarization*

Description

Calculating statistical parameters (skewness, kurtosis, quantiles) for a number of equivalent dose values.

Usage

```
dbED(EDdata, plot = TRUE, typ = "pdf", from = NULL,
     to = NULL, step = NULL, nbin = 15, pcolor = "grey",
     psize = 1.5, outfile = NULL)
```

Arguments

EDdata	matrix(**required**): a two-column matrix (i.e., equivalent dose values and associated standard errors)
plot	logical(with default): draw a plot or not
typ	character(with default): type of plot, typ="pdf" means a probability density plot and typ="hist" means a histogram plot
from	numeric(optional): desired lower limit on x-axis

to	numeric(optional): desired upper limit on x-axis
step	numeric(optional): a step-size used for constructing the probability density plot (if typ="pdf"). Smaller step value gives smoother density curve
nbin	integer(with default): desired number of intervals for the histogram (if typ="hist")
pcolor	character(with default): color of data points, input colors() to see available colors
psize	numeric(with default): size of data points
outfile	character(optional): if specified, calculated probability densities (if typ="pdf") will be written to a CSV file named "outfile" and saved to the current work directory

Value

Return a list that contains the following elements:

weight.ED	weigthed mean of equivalent dose values and associated standard error
skewness	weighted skewness of equivalent dose values and associated standard error
kurtosis	kurtosis of equivalent dose values and associated standard error
quantile.ED	quantiles of equivalent dose values

References

Galbraith RF, 2010. On plotting OSL equivalent doses. Ancient TL, 28(1): 1-10.

Galbraith RF, Roberts RG, 2012. Statistical aspects of equivalent dose and error calculation and display in OSL dating: an overview and some recommendations. Quaternary Geochronology, 11: 1-27.

See Also

psRadialPlot; RadialPlotter; EDdata

Examples

```
data(EDdata)
dbED(EDdata$gl11,typ="pdf")
```

---

decomp *OSL decay curve decomposition*

Description

Decomposing a CW-OSL or LM-OSL decay curve to a given number of first-order exponential components using a combination of differential evolution and Levenberg-Marquardt algorithm suggested by Bluszcz and Adamiec (2006).

Usage

```
decomp(Sigdata, delay.off = c(0,0), ncomp = 2,
       constant = TRUE, typ = "cw", control.args = list(),
       weight = FALSE, plot = TRUE, log = "x", lwd = 2,
       curve.no = NULL, SAR.Cycle = NULL, irr.dose = NULL,
       outfile = NULL, transf = TRUE)
```

Arguments

`Sigdata`	matrix(**required**): a two-column matrix (i.e., stimulation time and photon count values)
`delay.off`	vector(with default): a two-elment vector indicating the "Delay" and "Off" values of the decay curves, i.e., `delay.off[1]=Delay`,`delay.off[2]=Off`
`ncomp`	integer(with default): number of decomposed components
`constant`	logical(with default): logical value indicating if a constant component should be subtracted from the decay curve
`typ`	character(with default): type of a decay curve (i.e., `typ="cw"` or `typ="lm"`)
`control.args`	list(with default): arguments used in the differential evolution algorithm, see details
`weight`	logical(with default): logical value indicating if the fit should be performed using a weighted procedure
`plot`	logical(with default): logical value indicating if the results should be plotted
`log`	character(with default): a character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic
`lwd`	numeric(with default): width of curves (lines)
`curve.no`	numeric(optional): decay curve number
`SAR.Cycle`	numeric(optional): SAR cycle of the decay curve, Example: `SAR.Cycle="R1"`
`irr.dose`	numeric(optional): irradiation dose of the decay curve
`outfile`	character(optional): if specified, decomposed signal values will be written to a CSV file named `"outfile"` and saved to the current work directory
`transf`	logical(with default): do not use (for backward compatibility purpose)

Details

Function decomp decomposes an OSL decay curve to a specified number of components using a combination of differential evolution and Levenberg-Marquardt algorithm. Both CW-OSL and LM-OSL decay curves can be decomposed.

For a CW-OSL decay curve, the fitting model (Bluszcz and Adamiec, 2006) is:
`I(t)=a1*b1*exp(-b1*t)+...+ak*bk*exp(-bk*t)`,
where `k=1, 2, ..., 7`, `I(t)` is the luminescence intensity as a function of time, `a` is the number of trapped electrons, and `b` is the detrapping rate. The constant component is `c` if `constant=TRUE`.

For a LM-OSL decay curve, the fitting model (Bulur, 2000) is:
`I(t)=a1*b1*(t/P)*exp[-b1*t^2/(2*P)]+...+ak*bk*(t/P)*exp[-bk*t^2/(2*P)]`,
where `k=1, 2, ..., 7`, and `I(t)` is the luminescence intensity as a function of time, `P` is the total stimulation time, `a` is the number of trapped electrons, and `b` is the detrapping rate. The constant component is `c*(t/P)` if `constant=TRUE`.

Parameters are initialized using a differential evolution method suggested by Bluszcz and Adamiec (2006), then the Levenberg-Marquardt algorithm (minpack: Fortran 90 version by John Burkardt, freely available at http://people.sc.fsu.edu/~jburkardt/f_src/minpack/minpack.html) will be performed to optimize the parameters. If `weight=TRUE`, the photon counts will be assumed to follow a Possion distribution with a standard error equal to the square root of the number of photon counts, and the decay curve will be fitted using a weighted procedure. Setting `weight=TRUE` gives more weight to photon counts from slower decaying components.

Arguments in `control.args` that control the differential evolution algorithm include:
(1) *factor*: the number of population members, `np=factor*ncomp`, default `factor=20`;
(2) f: a weighting factor that lies between `0` and `1.2`, default `f=0.5`;
(3) *cr*: a crossover constant that lies between `0` and `1`, default `cr=0.99`;
(4) *maxiter*: maximum number of iterations, default `maxiter=500`;
(5) *tol*: tolerance for stopping the iteration, the procedure will be terminated if all relative standard deviations of parameters are smaller than `tol`, defalut `tol=0.1`.

Value

Return an invisible list containing the following elements:

`message`	return 0 if fit succeeds, else 1
`comp.sig`	a matrix containing time, signal, and fitted signal values for each component
`LMpars`	optimized parameters for the decay curve
`constant`	estimated constant component, it returns `0` if `constant=FALSE`
`value`	minimized objective for the decay curve
`FOM`	figure of merit value for the decay curve in percent

Note

Arguments `curve.no`, `SAR.Cycle`, and `irr.dose` have nothing to do with decay curve fitting. They are used only for plotting purpose.

The model to be optimized should not be underdetermined. This means that the number of data points should exceed (or at least be equal to) the number of parameters. For a given model, this routine will return an error if any standard errors of parameters cannot be estimated by numerical difference-approximation. Function `decompc` in previous versions was bundled to function decomp.

We would like to thank Professor Andrzej Bluszcz who helps us a lot during the programming of this function. Dr Jeong-Heon Choi is thanked for providing published data sets to test this routine.

References

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(7-8): 886-891.

Bulur E, 2000. A simple transformation for converting CW-OSL curves to LM-OSL curves. Radiation Measurements, 32(2): 141-145.

Differential evolution algorithm, http://en.wikipedia.org/wiki/Differential_evolution

Jain M, Murray AS, Boetter-Jensen L, 2003. Characterisation of blue-light stimulated luminescence components in different quartz samples: implications for dose measurement. Radiation Measurements, 37(4-5): 441-449.

More JJ, 1978. "The Levenberg-Marquardt algorithm: implementation and theory," in Lecture Notes in Mathematics: Numerical Analysis, Springer-Verlag: Berlin. 105-116.

Further reading

Adamiec G, 2005. OSL decay curves-relationship between single- and multiple-grain aliquots. Radiation Measurements, 39(1): 63-75.

Balian HG, Eddy NW, 1977. Figure-of-merit (FOM), an improved criterion over the normalized chi-squared test for assessing goodness-of-fit of gamma-ray spectral peaks. Nuclear Instruments and Methods, 145(2): 389-95.

Choi JH, Duller GAT, Wintle AG, 2006. Analysis of quartz LM-OSL curves. Ancient TL, 24(1): 9-20.

Li SH, Li B, 2006. Dose measurement using the fast component of LM-OSL signals from quartz. Radiation Measurements, 41(5): 534-541.

Peng J, Dong ZB, Han FQ, Han YH, Dai XL, 2014. Estimating the number of components in an OSL decay curve using the Bayesian Information Criterion. Geochronometria, 41(4): 334-341.

See Also

Signaldata; pickBINdata; fastED

Examples

```
### Example 1:
data(Signaldata)
decomp(Signaldata$lm,ncomp=3,typ="lm",
       control.args=list(maxiter=10))

### Example 2 (not run):
# data(BIN)
# obj_pickBIN <- pickBINdata(BIN, Position=2, Run=2, view=TRUE,
#                            LType="OSL", force.matrix=TRUE)
# decomp(obj_pickBIN$BINdata[[1]], ncomp=2, log="xy")
```

EDdata *Equivalent dose values*

Description

Two sets of equivalent dose values.

Usage

```
data(EDdata)
```

Format

A list that contains two sets of equivalent dose values:

gl11 35 equivalent dose values of a sand sample from the Tengger Desert (Peng and Han, 2013)

al3 84 equivalent dose values of an alluvial deposit from the andean precordillera (Schmidt et al., 2012)

References

Peng J, Han FQ, 2013. Selections of fast-component OSL signal using sediments from the south edge of Tengger Desert. Acta Geoscientica Sinica, 34(6): 757-762.

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(1): 62-75.

See Also

dbED; psRadialPlot; RadialPlotter;
mcFMM; mcMAM

Examples

```
# Not run.
# data(EDdata)
# names(EDdata)
```

fastED *Fast-component equivalent dose calculation*

Description

Estimating a fast-component equivalent dose using decay curves obtained from the single aliquot regenerative-dose (SAR) method.

Usage

```
fastED(Sigdata, Redose, delay.off = c(0,0), ncomp = 2,
       constant = TRUE, control.args = list(), typ = "cw",
       model = "gok", origin = FALSE, errMethod = "sp",
       nsim = 500, weight.decomp = FALSE,
       weight.fitGrowth = TRUE, trial = TRUE,
       nofit.rgd = NULL, outpdf = NULL, log = "x",
       lwd = 2, test.dose = NULL, agID = NULL)
```

Arguments

Sigdata	matrix(**required**): a series of decay curves stored in a matrix column by column, the first column denotes stimulation time values, see details. Data structure of this kind can be obtained using function pickBINdata by setting argument force.matrix=TRUE, see examples
Redose	vector(**required**): regenerative dose values. Example: Redose=c(1,2,3,4,0,1)
delay.off	vector(with default): a two-elment vector indicating the "Delay" and "Off" values of the decay curves, i.e., delay.off[1]=Delay,delay.off[2]=Off
ncomp	integer(with default): number of decomposed components
constant	logical(with default): logical value indicating if a constant background should be subtracted from the decay curve, see function decomp for details
control.args	list(with default): arguments used in the differential evolution algorithm, see function decomp for details
typ	character(with default): type of an OSL decay curve, only CW-OSL decay curve can be analyzed currently
model	character(with default): model used for growth curve fitting, see function fitGrowth for available models
origin	logical(with default): logical value indicating if the growth curve should be forced to pass the origin
errMethod	character(with default): method used for equivalent dose error assessment. See function calED for details
nsim	integer(with default): desired number of randomly simulated equivalent dose obtained by Monte Carlo simulation
weight.decomp	character(with default): logical value indicating if the decay curve should be fitted using a weighted procedure, see function decomp for details

`weight.fitGrowth`
: character(with default): logical value indicating if the growth curve should be fitted using a weighted procedure, see function fitGrowth for details

`trial`
: logical(with default): logical value indicating if the growth curve should be fitted using other models if the given model fails, see function fitGrowth for details

`nofit.rgd`
: integer(optional): regenerative doses that will not be used during the fitting. For example, if `nofit.rgd=1` then the first regenerative dose will not be used during fast-component growth curve fitting

`outpdf`
: character(optional): if specified, results of fast-component equivalent dose calculation will be written to a PDF file named `"outpdf"` and saved to the current work directory

`log`
: character(with default): a character string which contains "x" if the x axis is to be logarithmic, "y" if the y axis is to be logarithmic and "xy" or "yx" if both axes are to be logarithmic

`lwd`
: numeric(with default): width of curves (lines)

`test.dose`
: numeric(optional): test dose of decay curves

`agID`
: vector(optional): a three-elemenet vector indicating aliquot (grain) ID, i.e., `agID[1]=NO`, `agID[2]=Position`, `agID[3]=Grain`

Details

Function fastED is used to estimate a fast-component equivalent dose using data sets obtained from the SAR protocol (Murray and Wintle, 2000). The routine trys to decompose a series of decay curves to a specified number of components, then the numbers of trapped electrons from the fast-component will be used to construct the growth curve to estimate a fast-component equivalent dose. See function decomp, fitGrowth, and calED for more details concerning decay curve decomposition, growth curve fitting, and equivalent dose calculation, respectively.

Argument `Sigdata` is a column-matrix made up with stimulation time values and a number of decay curves:

Column.no	Description
`I`	Stimulation time values
`II`	Natural-dose signal values
`III`	Test-dose signal values for the natural-dose
`IV`	The 1th Regenerative-dose signal values
`V`	Test-dose signal values for the 1th regenerative-dose
`VI`	The 2th regenerative-dose signal values
`VII`	Test-dose signal values for the 2th regenerative-dose
`...`	...

Value

Return an invisible list containing the following elements:

`decomp.pars`
: a list containing optimized parameters of successfully fitted decay curves

`Curvedata`
: data sets used for building the fast-component growth curve

`Ltx`	sensitivity-corrected natural-dose fast-component signal and its standard error
`LMpars`	optimizaed parameters for the fast-component growth curve
`value`	minimized objective for the fast-component growth curve
`avg.error`	average fit error for the fast-component growth curve
`RCS`	reduced chi-square value for the fast-component growth curve
`FOM`	figure of merit value for the fast-component growth curve in percent
`calED.method`	method used for fast-component equivalent dose calculation, i.e., `"Interpolation"` or `"Extrapolation"`
`mcED`	randomly simulated fast-component equivalent doses
`ED`	fast-component equivalent dose and its standard error
`ConfInt`	68 percent and 95 percent confidence interval of fast-component equivalent dose
`RecyclingRatio1`	the first fast-component recycling ratio and its standard error
`RecyclingRatio2`	the second fast-component recycling ratio and its standard error
`RecyclingRatio3`	the third fast-component recycling ratio and its standard error
`Recuperation1`	the first fast-component recuperation (i.e., ratio of the sensitivity-corrected zero-dose signal to natural-dose signal) and its standard error in percent
`Recuperation2`	the second fast-component recuperation (i.e., ratio of the sensitivity-corrected zero-dose signal to maximum regenerative-dose signal) and its standard error in percent

Note

Argument `test.dose` and `agID` have nothing to do with fast-component equivalent dose calculation. They are used only for plotting purpose.

The number of trapped electrons that corresponds to the largest decay rate will be regarded as the fast-component signal, which cannot always ensure that a pure fast-component signal be extracted if an ultra-fast decaying component appears.

The authors thank Professor Sheng-Hua Li and Professor Geoff Duller for their helpful discussions concerning fast-component equivalent dose calculation.

References

Li SH, Li B, 2006. Dose measurement using the fast component of LM-OSL signals from quartz. Radiation Measurements, 41(5): 534-541.

Murray AS, Wintle AG, 2000. Luminescence dating of quartz using improved single-aliquot regenerative-dose protocol. Radiation Measurements, 32(1): 57-73.

See Also

pickBINdata; Signaldata;
fitGrowth; decomp; calED

Examples

```
### Example 1 (not run):
# data(Signaldata)
# fastED(Signaldata$cw,Redose=c(80,160,240,320,0, 80)*0.13,
#        ncomp=3, constant=FALSE, outpdf="fastED1")

### Example 2 (not run):
# data(BIN)
# obj_pickBIN <- pickBINdata(BIN, Position=6, Grain=0,
#                            LType="OSL", force.matrix=TRUE)
# fastED(obj_pickBIN$BINdata[[1]], ncomp=2, constant=TRUE,
#        Redose=c(100,200,300,400,0,100)*0.13, outpdf="fastED2")
```

`fitGrowth` *Growth curve fitting*

Description

Fitting growth curves using the Levenberg-Marquardt algorithm.

Usage

```
fitGrowth(Curvedata, model = "gok", origin = FALSE,
          weight = TRUE, trial = FALSE, plot = TRUE,
          nofit.rgd = NULL, agID = NULL, Tn = NULL,
          Tn3BG = NULL, TnBG.ratio = NULL, rseTn = NULL,
          FR = NULL, RecyclingRatio1 = NULL,
          RecyclingRatio2 = NULL, RecyclingRatio3 = NULL,
          Recuperation1 = NULL, Recuperation2 = NULL,
          LnTn.curve = NULL, TxTn = NULL)
```

Arguments

`Curvedata`	matrix(**required**): a three-column matrix (i.e., regenerative doses, sensitivity-corrected regenerative-dose signals, and associated standard errors)
`model`	character(with default): model used for growth curve fitting, see details
`origin`	logical(optional): logical value indicating if the growth curve should be forced to pass the origin
`weight`	logical(with default): logical value indicating if the growth curve should be fitted using a weighted procedure, see details
`trial`	logical(with default): logical value indicating if the growth curve should be fitted using other models if the given model fails, see details
`plot`	logical(with default): logical value indicating if the results should be plotted

nofit.rgd	integer(optional): regenerative doses that will not be used during the fitting. For example, if nofit.rgd=c(5,6) then both the fifth and sixth regenerative doses will not be used during growth curve fitting
agID	vector(optional): a three-elemenet vector indicating aliquot (grain) ID, i.e., agID[1]=NO, agID[2]=Position, agID[3]=Grain
Tn	vector(optional): a two-element vector containing value and standard error of Tn
Tn3BG	numeric(optional): 0-1 value indicating if Tn is more than 3 sigma above BG, 1 indicates Tn>3_sigma_BG, 0 indicates Tn<=3_sigma_BG
TnBG.ratio	vector(optional): a two-element vector containing value and standard error of ratio of initial Tn signal to BG
rseTn	numeric(optional): relative standard error of Tn in percent
FR	vector(optional): a two-element vector containing value and standard error of fast ratio of Tn
RecyclingRatio1	vector(optional): a two-element vector containing value and standard error of the first recycling ratio
RecyclingRatio2	vector(optional): a two-element vector containing value and standard error of the second recycling ratio
RecyclingRatio3	vector(optional): a two-element vector containing value and standard error of the third recycling ratio
Recuperation1	vector(optional): a two-element vector containing value and standard error of the first recuperation
Recuperation2	vector(optional): a two-element vector containing value and standard error of the second recuperation
LnTn.curve	list(optional): decay curve data for Ln and Tn, it should contain four elements, i.e., names(LnTn.curve)=c("Ln.x","Ln.y","Tn.x","Tn.y")
TxTn	vector(optional): ratios of Tx to Tn for various SAR cycles

Details

In growth curve fitting using function fitGrowth, five models are available:
(1) *"line"*: a linear model, y=a*x+b;
(2) *"exp"*: a single saturation exponential model, y=a*[1-exp(-b*x)]+c;
(3) *"lexp"*: a single saturation exponential plus linear model, y=a*[1-exp(-b*x)]+c*x+d;
(4) *"dexp"*: a double saturation exponential model, y=a*[1-exp(-b*x)]+c*[1-exp(-d*x)]+e;
(5) *"gok"*: a general order kinetic model (Guralnik et al., 2015), y=a*[1-(1+b*c*x)^(-1/c)]+d.

The fitting process is performed using the Levenberg-Marquardt algorithm (minpack: Fortran 90 source code by John Burkardt, freely available at http://people.sc.fsu.edu/~jburkardt/f_src/minpack/minpack.html). If weight=TRUE, a weighted procedure will be performed through weighting each data point by its inverse variance. User is advised to set argument plot=TRUE if possible to visualize the quality of fit.

Argument `trial=TRUE` means that if the growth curve can not be fitted successfully using the user-supplied model, then the procedure will try to fit other models instead:

Model	Tried model
"dexp"	c("dexp", "gok", "exp", "line")
"lexp"	c("lexp", "gok", "exp", "line")
"gok"	c("gok", "exp", "line")
"exp"	c("exp", "line")
"line"	c("line")

For example, if `model="dexp"` and `trial=TRUE`, then a number of models from sequence `c("dexp", "gok", "exp", "line")` will be applied one after another until the fit succeeds.

The required number of data points for each model is (value inside the parentheses denotes the required number of observations if the model passes the origin):

Model	Required NPoints
"dexp"	>=5 (>=4)
"lexp"	>=4 (>=3)
"gok"	>=4 (>=3)
"exp"	>=3 (>=2)
"line"	>=2 (>=1)

If user-provided data is not enough for model fitting (i.e., the number of data points is less than the number of parameters of a given model), then a model with less parameters will be fitted. For example, if 4 data points are fitted using a "dexp" (`origin=FALSE`) model that actually needs at least 5 data points, then a 4-parameter "gok" model will be fitted instead.

Value

Return a list that contains the following elements:

`message`	return 0 if the fit succeeds, else 1
`fitIDX`	Indices of dose points used in growth curve fitting
`LMpars`	optimized parameters for the growth curve
`value`	minimized objective for the growth curve
`avg.error`	average fit error for the growth curve
`RCS`	reduced chi-square value for the growth curve
`FOM`	figure of merit value for the growth curve in percent

Note

Arguments `agID`, `Tn`, `Tn3BG`, `TnBG.ratio`, `rseTn`, `FR`, `RecyclingRatio1`, `RecyclingRatio2`, `RecyclingRatio3`, `Recuperation1`, `Recuperation2`, `LnTn.curve`, `TxTn` have nothing to do with growth curve fitting. They are used only for plotting purpose.

The model to be optimized should not be underdetermined. This means that the number of data points should exceed (or at least be equal to) the number of parameters. For a given model, the procedure will return an error if any standard errors of parameters cannot be estimated by numerical difference-approximation.

References

Balian HG, Eddy NW, 1977. Figure-of-merit (FOM), an improved criterion over the normalized chi-squared test for assessing goodness-of-fit of gamma-ray spectral peaks. Nuclear Instruments and Methods, 145(2): 389-95.

Guralnik B, Li B, Jain M, Chen R, Paris RB, Murray AS, Li SH, Pagonis V, Valla PG, Herman F, 2015. Radiation-induced growth and isothermal decay of infrared-stimulated luminescence from feldspar. Radiation Measurements, 81: 224-231.

More JJ, 1978. "The Levenberg-Marquardt algorithm: implementation and theory" in Lecture Notes in Mathematics: Numerical Analysis, Springer-Verlag: Berlin. 105-116.

See Also

analyseBINdata; SARdata;
calED; calSARED; fastED;
pickSARdata; lsNORM; calSGCED

Examples

```
### Example 1:
Curvedata <- cbind(c(0, 18, 36, 54, 72, 0, 18),
                   c(0.026, 1.55, 2.39, 3.46, 4.13, 0.023, 1.61),
                   c(0.005, 0.11, 0.27, 0.22, 0.20, 0.008, 0.24))
fitGrowth(Curvedata, model="gok", origin=FALSE, plot=TRUE)

### Example 2 (not run):
# data(SARdata)
# Curvedata <- SARdata[!is.element(SARdata[,2], "N"),3:5]
# fitGrowth(Curvedata, model="gok", origin=FALSE)
```

---

loadBINdata *BIN file loading (importing)*

Description

Loading (importing) a BIN file into the R platform.

Usage

```
loadBINdata(filename, view = TRUE)
```

Arguments

`filename`	character(**required**): name(s) of file(s) (with file extension ".BIN", ".bin", "BINX", or "binx"), the file(s) must be available from the current working directory. Example: `filename=c("foo1.bin","foo2.binx")`
`view`	logical(optional): logical value indicating if the loaded data should be visualized in a Summary Table

Details

Function loadBINdata is used for loading BIN (BINX) files into the R platform. Five versions of binary files (V3, V4, V6, V7, and V8) are loadable. It can load a single BIN (BINX) file or a number of files into R simultaneously.
Items reserved during the loading process include:
(1) *Position*: Carousel position;

(2) *Grain*: Grain number;

(3) *Run*: Run number;

(4) *Set*: Set number;

(5) *DType*: Data type, includes: `Natural`, `N+dose`, `bleach`, `Bleach+dose`, `Natural(Bleach)`, `N+dose(Bleach)`, `Dose`, `Background`;

(6) *IRRTime*: Irradiation time;

(7) *NPoints*: number of data points;

(8) *LType*: Luminescence type, includes: `TL`, `OSL`, `IRSL`, `M-IR`, `M-VIS`, `TOL`, `TRPOSL`, `RIR`, `RBR`, `USER`, `POSL`, `SGOSL`, `RL`, `XRF`;

(9) *Low*: Low (temperature, time, wavelength);

(10) *High*: High (temperature, time, wavelength);

(11) *Rate*: Rate (temperature, time, wavelength);

(12) *Temperature*: Sample temperature;

(13) *Delay*: TOL "delay" channels;

(14) *On*: TOL "on" channels;

(15) *Off*: TOL "off" channels;

(16) *LightSource*: Light source, includes: `None`, `Lamp`, `IRDiodes`, `CalibraitionLED`, `BlueDiodes`, `WhiteLight`, `GreenLaser`, `IRLaser`;

(17) *AnTemp*: Annealing temperature;

(18) *TimeSinceIrr*: Time since irradiation;

(19) *Time*: Data collection time;

(20) *Date*: Data collection date.

Value

Return an invisible list of S3 class object "loadBIN" containing the following elements:

records	a list containing loaded data records
tab	a table (data.frame) summarizing items of loaded data records

Note

We would like to appreciate Dr Lei Gao who prompts me to write this function and provides measured data sets to test this procedure.

References

Duller GAT, 2016. Analyst (v4.31.9), User Mannual.

See Also

pickBINdata; analyseBINdata; BIN

Examples

```
### Not run.
### Ensure that file "foo.bin" is available
### from the current working directory.
# obj_loadBIN <- loadBINdata("foo.bin", view=TRUE)
# class(obj_loadBIN)
# obj_loadBIN$records
```

---

lsNORM *Regenerative-dose signal optimization using the LS-normalisation procedure*

Description

Optimizing standardised regenerative-dose signals according to the least-squares normalisation (LS-normalisation) procedure using an iterative scaling and fitting procedure proposed by Li et al. (2016).

Usage

```
lsNORM(SARdata, model = "gok", origin = FALSE,
       weight = FALSE, natural.rm = TRUE,
       norm.dose = NULL, maxiter = 10,
       plot = TRUE)
```

Arguments

SARdata	matrix(**required**): SAR data used for performing the LS-normalisation procedure, it should contain five columns (i.e., NO, SAR.Cycle, Dose, Signal, and Signal.Err), see SARdata for details
model	character(with default): model used for growth curve fitting, see fitGrowth for available models
origin	logical(with default): logical value indicating if the growth curve should be forced to pass the origin
weight	logical(with default): logical value indicating if the growth curve should be fitted using a weighted procedure, see function fitGrowth for details
natural.rm	logical(with default): logical value indicating if the standardised natural-dose signal should be removed from SARdata
norm.dose	numeric(optional): regenerative-dose used for re-scaling established gSGC. For example, if norm.dose=100, then the signal value for a dose value of 100 (Gyls) will be re-scaled to unity
maxiter	integer(with default): allowed maximum number of iterations during the LS-normalisation optimization process
plot	logical(with default): logical value indicating if the results should be plotted

Details

Function lsNORM is used for optimizing regenerative-dose signal data from a number of grains (aliquots) according to the least-squares normalisation (LS-normalisation) procedure outlined by Li et al. (2016) using an iterative scaling and fitting procedure.

The LS-normalisation procedure for growth curve optimization involves the following steps:
(1) Fit standardised regenerative-dose signals from all of the aliquots;
(2) Re-scale the individual growth curve from each aliquot using a scaling factor. The scaling factor for each aliquot is determined in a way such that the sum of squared residuals is minimized. Each aliquots is treated individually, and different scaling factors are calculated for different aliquots.
(3) Iterate the fitting (step 1) and re-scaling (step 2). The iterative procedure is performed repeatedly until there is negligible change in the relative standard deviation of the normalised growth curve data.

Value

Return an invisible list that contains the following elements:

norm.SARdata	SAR data sets optimized using the LS-normalisation procedure
sf	scaling factor of standardised regenerative-dose signals
iter	number of iterations required
LMpars1	optimized parameters for the growth curve before LS-normalisation
value1	minimized objective for the growth curve before LS-normalisation
avg.error1	average fit error for the growth curve before LS-normalisation
RCS1	reduced chi-square value for the growth curve before LS-normalisation

FOM1	figure of merit value for the growth curve before LS-normalisation in percent
LMpars2	optimized parameters for the growth curve after LS-normalisation
value2	minimized objective for the growth curve after LS-normalisation
avg.error2	average fit error for the growth curve after LS-normalisation
RCS2	reduced chi-square value for the growth curve after LS-normalisation
FOM2	figure of merit value for the growth curve after LS-normalisation in percent

References

Li B, Jacobs Z, Roberts RG, 2016. Investigation of the applicability of standardised growth curves for OSL dating of quartz from Haua Fteah cave, Libya. Quaternary Geochronology, 35: 1-15.

See Also

analyseBINdata; fitGrowth; SARdata
scaleSGCN; calSGCED

Examples

```
### Example 1:
data(SARdata)
# Use only the first five aliquots of SARdata.
Data <- SARdata[1:40,]
res_lsNORM <- lsNORM(Data, model="gok")
res_lsNORM$norm.SARdata

### Example 2 (not run):
# data(BIN)
# obj_pickBIN <- pickBINdata(BIN, Position=1:48, Grain=0,
#                            LType="OSL", view=FALSE)
# obj_analyseBIN <- analyseBINdata(obj_pickBIN, nfchn=3, nlchn=20)
# lsNORM(obj_analyseBIN$SARdata, norm.dose=300)
```

---

mcFMM — *Finite mixture age model optimization (using a Markov chain Monte Carlo method)*

Description

Sampling from the joint-likelihood functions of finite mixture age models (include the central age model) using a Markov chain Monte Carlo (MCMC) method.

Usage

```
mcFMM(EDdata, ncomp = 1, addsigma = 0, iflog = TRUE,
      nsim = 50000, inis = list(), control.args = list())
```

Arguments

EDdata	matrix(**required**): a two-column matrix (i.e., equivalent dose values and associated standard errors)
ncomp	integer(with default): number of components (1 denotes the central age model)
addsigma	numeric(with default): additional uncertainty
iflog	logical(with default): transform equivalent dose values to log-scale or not
nsim	integer(with default): deseried number of iterations
inis	list(with default): initial state of parameters. Example: `inis=list(p1=1,p2=1,mu1=5,mu2=10)` in FMM2 (the sum of `p1` and `p2` will be normalized to 1 during the simulation)
control.args	list(with default): arguments used in the Slice Sampling algorithm, see details

Details

Function mcFMM is used for sampling from the joint-likelihood functions of finite mixture age models (include the central age model) using a Markov chain Monte Carlo sampling algorithm called Slice Sampling (Neal, 2003). Three arguments (`control.args`) are used for controling the sampling process:
(1) *w*: size of the steps for creating an interval from which to sample, default `w=1`;
(2) *m*: limit on steps for expanding an interval, `m<=1` means no limit on the expandation, `m>1` means the interval is expanded with a finite number of iterations, default `m=-100`;
(3) *nstart*: maximum number of trials for updating a variable in an iteration. It can be used for monitoring the stability of the simulation. For example, a MAM4 is likely to crash down for data sets with small numbers of data points or less dispersed distributions (see section 8.3 of Galbraith and Roberts, 2012 for a discussion), and sometimes more than one trial (i.e., using `nstart>1`) is required to complete the sampling process, default `nstart=1`.

Value

Return an invisible list of S3 class object `"mcAgeModels"` including the following elements:

EDdata	equivalent dose values
addsigma	additional uncertainty
model	fitting model
iflog	transform equivalent dose values to log-scale or not
nsim	number of iterations
chains	simulated samples

References

Galbraith RF, Green P, 1990. Estimating the component ages in a finite mixture. International Journal of Radiation Applications and Instrumentation. Part D. Nuclear Tracks and Radiation Measurements, 17: 197-206.

Neal RM, 2003. "Slice sampling" (with discussion). Annals of Statistics, 31(3): 705-767. Software is freely available at http://www.cs.utoronto.ca/~radford/slice.software.html.

See Also

mcMAM; reportSAM;
RadialPlotter; EDdata

Examples

```
# Not run.
# data(EDdata)
# Construct a MCMC chain for FMM3.
# obj<-mcFMM(EDdata$gl11,ncomp=3,nsim=5000)
# reportSAM(obj,thin=2,burn=1e3)
```

`mcMAM` *Minimum age model optimization (using a Markov chain Monte Carlo method)*

Description

Sampling from the joint-likelihood function of the minimum age model using a Markov chain Monte Carlo (MCMC) method .

Usage

```
mcMAM(EDdata, ncomp = -1, addsigma = 0, iflog = TRUE,
      nsim = 50000, inis = list(), control.args = list())
```

Arguments

`EDdata`	matrix(**required**): a two-column matrix (i.e., equivalent dose values and associated standard errors)
`ncomp`	integer(with default): number of components, `-1=MAM3`, `-2=MAM4`
`addsigma`	numeric(with default): additional uncertainty
`iflog`	logical(with default): transform equivalent dose values to log-scale or not
`nsim`	integer(with default): deseried number of iterations
`inis`	list(with default): initial state of parameters. Example: `inis=list(p=0.1,gamma=20,sigma=0.5)` in MAM3
`control.args`	list(with default): arguments used by the Slice Sampling algorithm, see function mcFMM for details

Value

Return an invisible list of S3 class object `"mcAgeModels"`. See mcFMM for details.

References

Galbraith RF, Roberts RG, Laslett GM, Yoshida H, Olley JM, 1999. Optical dating of single grains of quartz from Jinmium rock shelter, northern Australia. Part I: experimental design and statistical models. Archaeometry, 41(2): 339-364.

Neal RM, 2003. "Slice sampling" (with discussion). Annals of Statistics, 31(3): 705-767. Software is freely available at http://www.cs.utoronto.ca/~radford/slice.software.html.

See Also

mcFMM; reportSAM; RadialPlotter; EDdata

Examples

```
# Not run.
# data(EDdata)
# Construct a MCMC chain for MAM3.
# obj<-mcMAM(EDdata$al3,ncomp=-1,addsigma=0.1,nsim=5000)
# reportSAM(obj,burn=1e3,thin=2)
#
# The convergence of the simulations may be diagnosed with
# the Gelman and Rubin's convergence diagnostic.
# library(coda)   # Only if package "coda" has been installed.
# args<-list(nstart=50)
# inis1<-list(p=0.01,gamma=26,mu=104,sigma=0.01)
# inis2<-list(p=0.99,gamma=100,mu=104,sigma=4.99)
# obj1<-mcMAM(EDdata$al3,ncomp=-2,nsim=3000,inis=inis1,control.args=args)
# obj2<-mcMAM(EDdata$al3,ncomp=-2,nsim=3000,inis=inis2,control.args=args)
# chain1<-mcmc(obj1$chains)
# chain2<-mcmc(obj2$chains)
# chains<-mcmc.list(chain1,chain2)
# gelman.plot(chains)
```

---

pickBINdata *BIN data set selection*

Description

Extracting data sets from a loaded BIN (BINX) file.

Usage

```
pickBINdata(obj_loadBIN, Position = NULL, Grain = NULL,
            Run = NULL, Set = NULL, DType = NULL,
            IRRTime = NULL, NPoints = NULL, LType = NULL,
            Low = NULL, High = NULL, Rate = NULL,
            Temperature = NULL, Delay = NULL, On = NULL,
            Off = NULL, LightSource = NULL, AnTemp = NULL,
            TimeSinceIrr = NULL, view = TRUE,
            manual.select = FALSE, force.matrix = FALSE)
```

Arguments

obj_loadBIN	list(**required**): an object of S3 class "loadBIN" produced by function loadBINdata
Position	vector(optional): carousel position, Example: Position=1:48
Grain	vector(optional): grain number
Run	vector(optional): run number
Set	vector(optional): set number
DType	character(optional): a character vector indicating data type, Example: DType=c("Natural", "N+dose")
IRRTime	vector(optional): irradiation time
NPoints	vector(optional): number of data points
LType	character(optional): a character vector indicating luminescence types, Example: LType="OSL"
Low	vector(optional): lower limit on temperature, time, or wavelength
High	vector(optional): upper limit on temperature, time, or wavelength
Rate	vector(optional): increasing rate of temperature, time, or wavelength
Temperature	vector(optional): a vector indicating the sample temperatures
Delay	vector(optional): TOL "delay" channels
On	vector(optional): TOL "on" channels
Off	vector(optional): TOL "off" channels
LightSource	character(optional): a character vector indicating light source, Example: LightSource="BlueDiodes"
AnTemp	vector(optional): annealing temperature
TimeSinceIrr	vector(optional): time since irradiation
view	logical(with default): logical value indicating if the loaded data should be visualized in a Summary Table
manual.select	logical(with default): logical value indicating if the loaded data should be selected manually using a Summary Table
force.matrix	logical(with default): logical value indicating if the picked data sets of an aliquot (grain) should be transformed into a matrix

Details

Function pickBINdata is used for pick up data sets from an object of S3 class "loadBIN" generated using function loadBINdata. Set force.matrix=TRUE will transform data sets of an aliquot (grain) into a matrix, the transformation fails if data sets have different x (temperature, time, or wavelength) coordinates. Transformed data sets stored in a matrix can be visualize via matplot (see examples).

Value

Return an invisible list of S3 class object "pickBIN" containing two elements:

BINdata	selected BIN data
agID	Aliquot or grain ID (i.e., c("NO","Position","Grain")) of selected data sets, it returns NULL if force.matrix=TRUE

References

Duller GAT, 2016. Analyst (v4.31.9), User Mannual.

See Also

loadBINdata; analyseBINdata; BIN;
decomp; fastED

Examples

```
 ### Example 1 (visualize decay curves for Position=2):
 data(BIN)
 obj_pickBIN <- pickBINdata(BIN, Position=2, view=FALSE,
                            LType="OSL", force.matrix=TRUE)
 matplot(obj_pickBIN$BINdata[[1]][,1],
         obj_pickBIN$BINdata[[1]][,-1],
         main="Decay curves",
         xlab="Stimulation time (s)",
         ylab="Photon counts",
         type="l", log="xy")

### Example 2 (visualize test-dose decay curves for Position=2):
obj_pickBIN <- pickBINdata(BIN, Position=2, Run=c(5,11,17,23,29,34,40),
                           view=FALSE, LType="OSL", force.matrix=TRUE)
matplot(obj_pickBIN$BINdata[[1]][,1],
        obj_pickBIN$BINdata[[1]][,-1],
        main="Test-dose decay curves",
        xlab="Stimulation time (s)",
        ylab="Photon counts",
        type="l", log="xy")

### Example 3 (visualize regenerative-dose decay curves for Position=2):
obj_pickBIN <- pickBINdata(BIN, Position=2, Run=c(8,14,20,26,31,37),
                           view=FALSE, LType="OSL", force.matrix=TRUE)
matplot(obj_pickBIN$BINdata[[1]][,1],
        obj_pickBIN$BINdata[[1]][,-1],
        main="Regenerative-dose decay curves",
        xlab="Stimulation time (s)",
        ylab="Photon counts",
        type="l", log="xy")
```

---

pickSARdata *SAR data set selection*

Description

Selecting SAR data sets (growth curves) in a batch model according to specified rejection criteria.

Usage

```
pickSARdata(obj_analyseBIN, model = "gok", origin = FALSE,
            weight = TRUE, trial = TRUE, nofit.rgd = NULL,
            Tn.above.3BG = TRUE, TnBG.ratio.low = NULL,
            rseTn.up = NULL, FR.low = NULL, rcy1.range = NULL,
            rcy2.range = NULL, rcy3.range = NULL,
            rcp1.up = NULL, rcp2.up = NULL, fom.up = NULL,
            rcs.up = NULL, use.se = TRUE, norm.dose = NULL,
            outpdf = NULL, outfile = NULL)
```

Arguments

`obj_analyseBIN`	list(**required**): an object of S3 class "analyseBIN" produced by function analyseBINdata or as_analyseBIN
`model`	character(with default): model used for growth curve fitting, see fitGrowth for available models
`origin`	logical(with default): logical value indicating if the growth curve should be forced to pass the origin
`weight`	logical(with default): logical value indicating if the growth curve should be fitted using a weighted procedure, see function fitGrowth for details
`trial`	logical(with default): logical value indicating if the growth curve should be fitted using other models if the given model fails, see function fitGrowth for details
`nofit.rgd`	integer(optional): regenerative doses that will not be used during the fitting. For example, if `nofit.rgd=2` then the second regenerative dose will not be used during growth curve fitting
`Tn.above.3BG`	logical(with default): logical value indicating if aliquot (grain) with Tn below 3 sigma BG should be rejected
`TnBG.ratio.low`	numeric(optional): lower limit on ratio of initial Tn signal to BG
`rseTn.up`	numeric(optional): upper limit on relative standard error of Tn in percent
`FR.low`	numeric(optional): lower limit on fast ratio of Tn
`rcy1.range`	vector(optional): a two-element vector indicating the lower and upper limits on recycling ratio 1, Example: `rcy1.range=c(0.9,1.1)`
`rcy2.range`	vector(optional): a two-element vector indicating the lower and upper limits on recycling ratio 2
`rcy3.range`	vector(optional): a two-element vector indicating the lower and upper limits on recycling ratio 3
`rcp1.up`	numeric(optional): upper limit on recuperation 1 (i.e., ratio of the sensitivity-corrected zero-dose signal to natural-dose signal) in percent
`rcp2.up`	numeric(optional): upper limit on recuperation 2 (i.e., ratio of the sensitivity-corrected zero-dose signal to maximum regenerative-dose signal) in percent
`fom.up`	numeric(optional): upper limit on figure of merit (FOM) values of growth curves in percent

rcs.up	numeric(optional): upper limit on reduced chi-square (RCS) values of growth curves
use.se	logical(with default): logical value indicating if standard errors of values should be used during application of rejection criteria
norm.dose	numeric(optional): dose value used for SAR data set re-normalisation, for example, if norm.dose=100, then sensitivity-corrected signal for Redose=100 obtained through growth curve fitting will be used to re-normalise a SAR data set
outpdf	character(optional): if specified, results of growth curve fitting will be written to a PDF file named "outpdf" and saved to the current work directory
outfile	character(optional): if specified, SAR data related quantities will be written to a CSV file named "outfile" and saved to the current work directory

Value

Return an invisible list that contains the following elements:

LMpars	a list containing optimized parameters of growth curves of selected SAR data sets
SARdata	a data.frame containing selected SAR data sets
norm.SARdata	a data.frame containing re-normalised SAR data sets, it returns NULL if norm.dose=NULL
agID	aliquot or grain ID (i.e., c("NO","Position","Grain")) of selected SAR data
summary.info	a summary of the SAR data selection

See Also

analyseBINdata; fitGrowth;
lsNORM; calSARED

Examples

```
# Not run.
# data(BIN)
# obj_pickBIN <- pickBINdata(BIN, Position=1:48, Grain=0,
#                            LType="OSL", view=FALSE)
# obj_analyseBIN <- analyseBINdata(obj_pickBIN, nfchn=3, nlchn=20)
# pickSARdata(obj_analyseBIN, model="gok", fom.up=3, outpdf="SARdata")
```

---

psRadialPlot *Pseudo radial plot drawing*

Description

Drawing a pseudo (simplified) radial plot.

Usage

```
psRadialPlot(EDdata, addsigma = 0, dose = NULL,
             zmin = NULL, zmax = NULL, ntick = 6,
             digits = 2, pcolor = "blue", psize = 1,
             rg = 2, zlabel = "De (Gy)")
```

Arguments

EDdata	matrix(**required**): a two-column matrix (i.e., equivalent dose values and associated standard errors)
addsigma	numeric(with default): additional uncertainty
dose	vector(optional): dose population(s) to be drawn
zmin	numeric(with default): lower limit on z-axis
zmax	numeric(with default): upper limit on z-axis
ntick	integer(with default): desired number of ticks in z-axis
digits	integer(with default): number of decimal places or significant digits for values shown in z-axis
pcolor	character(with default): color of a data point, input colors() to see more available colors
psize	numeric(with default): size of a data point
rg	integer(with default): range of a dose population, 0=dose, 1=dose+/-sigma, 2=dose+/-2sigma
zlabel	character(with default): title for the z-axis

Details

Function psRadialPlot is used for drawing a simplified radial plot in which the z-axis is a straight line. The pseudo radial plot is easier to construct compared to the regular radial plot. This function can be adopted to display estimates that have different error estimates in any field of the analytical sciences. Note that the function handles datasets in log-scale, so any minus observation is not allowed.

Value

Return a pseudo radial plot

References

Galbraith RF, 1988. Graphical display of estimates having differing standard errors. Technometrics, 30(3): 271-281.

Galbraith RF, 1994. Some applications of radial plots. Journal of the American Statistical Association, 89(428): 1232-1242.

Galbraith RF, 2010. On plotting OSL equivalent doses. Ancient TL, 28(1): 1-10.

Galbraith RF, Roberts RG, Laslett GM, Yoshida H, Olley JM, 1999. Optical dating of single grains of quartz from Jinmium rock shelter, northern Australia. Part I: experimental design and statistical models. Archaeometry, 41(2): 339-364.

See Also

dbED; RadialPlotter; EDdata

Examples

```
data(EDdata)
psRadialPlot(EDdata$al3, addsigma=0.10,
             dose=c(39.14, 51.27, 79.14), digits=1,
             zmin=30, zmax=100, ntick=10, rg=1)
```

`RadialPlotter` *Statistical age model optimization (with a Maximum Likelihood Estimation method)*

Description

Depending on the specified number of components, this function performs statistical age models analysis reviewed in Galbraith and Roberts (2012) dynamically using a Maximum Likelihood Estimation method. Age models that can be applied include: central age model (CAM), minimum age model (MAM), and finite mixture age model (FMM).

Usage

```
RadialPlotter(EDdata, ncomp = 0, addsigma = 0,
              maxcomp = 6, algorithm = c("port","lbfgsb"),
              plot = TRUE, pcolor = "blue", psize = 1.5,
              kratio = 0.3, zscale = NULL)
```

Arguments

`EDdata`	matrix(**required**): a two-column matrix (i.e., equivalent dose values and associated standard errors)
`ncomp`	integer(with default): number of components, `-1=MAM3`, `-2=MAM4`, `1=CAM`, `0` means fitting FMM automatically, and `>=1` means fitting FMM with a given number of components
`addsigma`	numeric(with default): additional uncertainty
`maxcomp`	integer(with default): maximum number of components in FMM
`algorithm`	character(with default): algorithm used for optimizing MAM, default `algorithm="port"`
`plot`	logical(with default): draw a radial plot or not
`pcolor`	character(with default): color of a data point, input colors() to see more available colors
`psize`	numeric(with default): size of a data point
`kratio`	numeric(with default): argument controlling the shape of zscale
`zscale`	vector(optional): argument controlling the scale of z-axis. Example: `zscale=seq(min(EDdata),max(EDdata),by=3L)`

Details

Both CAM and FMM are fitted using a iterative Maximum Likelihood Estimation procedure outlined by Galbraith (1988), while MAM can be estimated using either the "L-BFGS-B" algorithm (***R*** function optim in package ***stats***) or the "port" algorithm (***R*** function nlminb in package ***stats***).

Value

Return an object of S3 class `"RadialPlotter"` that contains the following elements:

`pars`	optimizaed parameters
`bic`	calculated Bayesian Information Criterion (BIC) value
`maxlik`	optimized maximum logged likelihood value

Note

Function RadialPlotter was given the same name as the ***Java*** package ***RadialPlotter*** written by Pieter Vermeesch (Vermeesch, 2009). Note that this function fits a model in log-scale, hence any minus equivalent dose value is not allowed, and that the procedure will return an error if any standard error of a parameter cannot be estimated by numerical difference-approximation.

The original ***S*** code for drawing a radial plot was written by Rex Galbraith and was transformed to ***R*** by Sebastian Kreutzer. The code for drawing radial plot in this function was modified from package ***Luminescence*** written by Kreutzer et al. (2012). We thank Dr Rex Galbraith for his permission to modify and bundle the code to this function. We also thank Dr Silke Schmidt, Dr Helena Rodnight, Dr Xian-Jiao Ou, and Dr Amanda Keen-Zebert for providing published OSL data sets to test this routine.

References

Galbraith RF, 1988. Graphical display of estimates having differing standard errors. Technometrics, 30(3): 271-281.

Galbraith RF, 1990. The radial plot: Graphical assessment of spread in ages. International Journal of Radiation Applications and Instrumentation. Part D. Nuclear Tracks and Radiation Measurements, 17(3): 207-214.

Galbraith RF, Green P, 1990. Estimating the component ages in a finite mixture. International Journal of Radiation Applications and Instrumentation. Part D. Nuclear Tracks and Radiation Measurements, 17: 197-206.

Galbraith RF, Laslett GM, 1993. Statistical models for mixed fission track ages. Nuclear Tracks And Radiation Measurements, 21(4): 459-470.

Galbraith RF, 1994. Some applications of radial plots. Journal of the American Statistical Association, 89(428): 1232-1242.

Galbraith RF, Roberts RG, Laslett GM, Yoshida H, Olley JM, 1999. Optical dating of single grains of quartz from Jinmium rock shelter, northern Australia. Part I: experimental design and statistical models. Archaeometry, 41(2): 339-364.

Galbraith RF, 2005. Statistics for fission track analysis. Chapman & Hall/CRC Press.

Galbraith RF, 2010. On plotting OSL equivalent doses. Ancient TL, 28(1): 1-10.

Galbraith RF, Roberts RG, 2012. Statistical aspects of equivalent dose and error calculation and display in OSL dating: an overview and some recommendations. Quaternary Geochronology, 11: 1-27.

Further reading

Duller GAT, 2008. Single-grain optical dating of Quaternary sediments: why aliquot size matters in luminescence dating. Boreas, 37(4): 589-612.

Kreutzer S, Schmidt C, Fuchs MC, Dietze M, Fischer M, Fuchs M, 2012. Introducing an R package for luminescence dating analysis. Ancient TL, 30(1): 1-8. Software is freely available at https://CRAN.R-project.org/package=Luminescence.

Rodnight H, 2008. How many equivalent dose values are needed to obtain a reproducible distribution? Ancient TL, 26(1): 3-10.

Rodnight H, Duller GAT, Wintle AG, Tooth S, 2006. Assessing the reproducibility and accuracy of optical dating of fluvial deposits. Quaternary Geochronology, 1(2): 109-120.

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(1): 62-75.

Vermeesch P, 2009. RadialPlotter: a Java application for fission track, luminescence and other radial plots. Radiation Measurements, 44: 409-410. Software is freely available at http://www.ucl.ac.uk/~ucfbpve/radialplotter/.

See Also

mcMAM; mcFMM;
dbED; psRadialPlot; EDdata

Examples

```
data(EDdata)
# Find out the appropriate number of components
# in FMM and fit automatically.
RadialPlotter(EDdata$al3,zscale=seq(24,93,7))

# Fit MAM3 (not run).
# RadialPlotter(EDdata$gl11,ncomp=-1,zscale=seq(20,37,3))
```

---

reportSAM *Reporting MCMC outputs for statistical age models*

Description

Summarizing distributions of parameters simulated from statistical age models.

Usage

```
reportSAM(obj, burn = 10000, thin = 5,
          plot = TRUE, outfile = NULL, ...)
```

Arguments

`obj`	list(**required**): an object of S3 class `"mcAgeModels"`, which is produced by function mcFMM or mcMAM
`burn`	integer(with default): number of iterations (i.e., the initial, non-stationary portion of the chain) to be discarded
`thin`	integer(with default): take every `thin-th` iteration
`plot`	logical(with default): plot the MCMC output or not
`outfile`	character(optional): if specified, simulated parameters will be written to a CSV file named `"outfile"` and saved to the current work directory
`...`	do not use

Details

Function reportSAM summarizes the output of a Markov chain (the mean values, the standard deviations, and the 95 percent confidence intervals). The initial *i* (`burn=i`) samples may have been affected by the inital state and has to be discarded ("burn-in"). Autocorrelation of simulated samples can be reduced by taking every *j-th* (`thin=j`) iteration ("thining").

Value

Return a list that contains the following elements:

`pars`	parameters and relevant statistics
`quantile`	quantiles of simulated parameters
`maxlik`	calculated maximum logged likelihood value

References

Lunn D, Jackson C, Best N, Thomas A, Spiegelhalter D, 2013. The BUGS book: a practical introduction to bayesian analysis. Chapman & Hall/CRC Press.

Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB, 2013. Bayesian data analysis. Chapman & Hall/CRC Press.

See Also

mcFMM; mcMAM

SARdata *Data sets used for SAR equivalent dose calculation*

Description

SAR data sets for individual aliquots from the "later" group of sample HF11 from Haua Fteah cave, Libya (Li et al., 2016).

Usage

```
data(SARdata)
```

Format

A data.frame with 840 observations containing the following 5 variables:

NO aliquot (grain) number

SAR.Cycle SAR cycles

Dose regenerative doses

Signal OSL signals

Signal.Err standard error of OSL signals

References

Li B, Jacobs Z, Roberts RG, 2016. Investigation of the applicability of standardised growth curves for OSL dating of quartz from Haua Fteah cave, Libya. Quaternary Geochronology, 35: 1-15.

See Also

fitGrowth; lsNORM; calSGCED;
as_analyseBIN

Examples

```
# Not run.
# data(SARdata)
# head(SARdata)
```

scaleSGCN *Natural-dose signal re-scaling*

Description

Re-scaling sensitivity-corrected natural-dose signals according to the "global standardised growth curve" (gSGC) method suggested by Li et al. (2015, 2016).

Usage

```
scaleSGCN(obj_analyseBIN, SGCpars, model, origin,
          SAR.Cycle, Tn.above.3BG = TRUE,
          TnBG.ratio.low = NULL, rseTn.up = NULL,
          FR.low = NULL, use.se = TRUE, outfile = NULL)
```

Arguments

obj_analyseBIN	list(**required**): an object of S3 class "analyseBIN" produced by function analyseBINdata or as_analyseBIN
SGCpars	vector(**required**): optimized parameters of the SGC obtained using function fitGrowth or lsNORM
model	character(**required**): fitting model used for obtaining SGCpars
origin	logical(**required**): logical value indicating if established SGC passes the origin
SAR.Cycle	character(**required**): a two-element character vector containing SAR cycles used for natural-dose signal re-scaling. Example: SAR.Cycle=c("N","R3")
Tn.above.3BG	logical(with default): logical value indicating if aliquot (grain) with Tn below 3 sigma BG should be rejected
TnBG.ratio.low	numeric(optional): lower limit on ratio of initial Tn signal to BG
rseTn.up	numeric(optional): upper limit on relative standard error of Tn in percent
FR.low	numeric(optional): lower limit on fast ratio of Tn
use.se	logical(with default): logical value indicating if standard errors of values should be used during application of rejection criteria
outfile	character(optional): if specified, scaled SGC data related quantities will be written to a CSV file named "outfile" and saved to the current work directory

Details

Sensitivity-corrected natural-dose signals are re-scaled according to **Eqn.(10)** of Li et al. (2015).

Value

Return an invisible list that contains the following elements:

scale.Ltx	scaled natural-dose signals and associated standard errors
agID	aliquot (grain) ID of scaled natural-dose signals

References

Li B, Roberts RG, Jacobs Z, Li SH, 2015. Potential of establishing a "global standardised growth curve" (gSGC) for optical dating of quartz from sediments. Quaternary Geochronology, 27: 94-104.

Li B, Jacobs Z, Roberts RG, 2016. Investigation of the applicability of standardised growth curves for OSL dating of quartz from Haua Fteah cave, Libya. Quaternary Geochronology, 35: 1-15.

See Also

lsNORM; calSGCED

Examples

```
# Not run.
data(SARdata)
gSGCpars <- c(137.440874251, 0.007997863, 2.462035263, -0.321536177)
scaleSGCN(as_analyseBIN(SARdata), SGCpars=gSGCpars, model="gok",
          origin=FALSE, SAR.Cycle=c("N","R3"))
```

Signaldata *Decay curves datasets*

Description

CW-OSL and LM-OSL decay curves.

Usage

```
data(Signaldata)
```

Format

A list that contains CW-OSL and LM-OSL decay curves:

cw a number of CW-OSL decay curves of a sand sample from the Tengger Desert in northern china (Peng and Han, 2013)

lm a LM-OSL decay curve from Li and Li (2006)

References

Li SH, Li B, 2006. Dose measurement using the fast component of LM-OSL signals from quartz. Radiation Measurements, 41(5): 534-541.

Peng J, Han FQ, 2013. Selections of fast-component OSL signal using sediments from the south edge of Tengger Desert. Acta Geoscientica Sinica, 34(6): 757-762.

See Also

decomp; fastED

Examples

```
# Not run.
# data(Signaldata)
# names(Signaldata)
```

Index