Package 'BayLum'

June 14, 2017

Type Package

Title Chronological Bayesian Models Integrating Optically Stimulated Luminescence Dating

Description
Description: collection of various R functions for Bayesian analysis of Luminescence data. This includes, amongst others, data import, export, application of age models and palaeodose model.

Date 2017-05-29

Version 0.1.0

Author Claire Christophe [aut, cre], Guillaume Guerin [aut], Anne Philippe [aut]

Maintainer Claire Christophe <claire.christophe@univ-nantes.fr>

Depends R(>= 3.3.2), utils, coda, Luminescence (>= 0.7.4), ArchaeoPhases, rjags (>= 4-6)

Imports stats, graphics, grDevices, methods

LazyData true

Suggests R.rsp

VignetteBuilder R.rsp

License GPL-3

NeedsCompilation no

RoxygenNote 6.0.1

R topics documented:

BayLum-package *Chronological Bayesian Models Integrating Optically Stimulated Luminescence Dating*

Description

Chronological Bayesian Models Integrating Optically Stimulated Luminescence Dating

AgeS_Computation *Bayesian analysis for the OSL age estimation of various samples*

Description

This function computes the age of at leat two samples according to the model developed in Combes and Philippe (2017), based on an output of the Generate_DataFile function.
Samples, for which data is avalilable in several BIN files, can be analysed.

Usage

```
AgeS_Computation(DATA, Nb_sample, BinPerSample = rep(1, Nb_sample),
  SampleNames, SavePdf = FALSE, OutputFileName = c("MCMCplot", "summary",
  "2per2Laws"), OutputFilePath = c(""), SaveEstimates = FALSE,
  OutputTableName = c("DATA"), OutputTablePath = c(""), THETA = c(),
  sepTHETA = c(","), PriorAge = rep(c(0.01, 100), Nb_sample),
  StratiConstraints = c(), sepSC = c(","), LIN_fit = TRUE,
  Origin_fit = FALSE, distribution = c("cauchy"), Taille = 50000, t = 5,
  Nb_chaines = 3)
```

Arguments

DATA	list of objects: LT, sLT, ITimes, dLab, ddot_env, regDose, J, K, Nb_measurement, provided by the function Generate_DataFile. DATA contains information for more than one sample.
Nb_sample	integer **(required)**: number of samples, Nb_sampe>1.
BinPerSample	integer vector (with default): vector with the number of BIN files per sample. The length of this vector is equal to Nb_sample. BinPerSample[i] correponds to the number of BIN files for the sample whose number ID is equal to i. For more information to fill this vector, we refer to detatils in Generate_DataFile.
SampleNames	character vector: names of sample. The length of this vector is equal to Nb_sample.
SavePdf	boolean (with default): if TRUE save graph in pdf file named OutputFileName in folder OutputFilePath.
OutputFileName	character (with default): name of the pdf file that will be generated by the function if SavePdf=TRUE.
OutputFilePath	character (with default): path to the pdf file that will be generated by the function if SavePdf=TRUE.
SaveEstimates	boolean (with default): if TRUE save Bayes estimates and confidence interval at level 68% and 95%, in a csv table named OutputFileName in folder OutputFilePath.

`OutputTableName`
: character (with default): name of the table that will be generated by the function if `SaveEstimates=TRUE`.

`OutputTablePath`
: character (with default): path to the table that will be generated by the function if `SaveEstimates=TRUE`.

`THETA`
: numeric matrix or character (with default): input object for systematic and individual error. If systematic errors are considered, see the details section for instructions regarding how to correctly fill `THETA`. Otherwise, default value is suitable, and only individual error is considered.

`sepTHETA`
: character (with default): if `THETA` is character, indicate column separator in `THETA` .csv file.

`PriorAge`
: numeric vector (with default): lower and upper bounds for age parameter of each sample. `length(PriorAge)=2*Nb_sample` and `PriorAge[2i-1,2i]` correponds to the lower and upper bounds of sample whose number ID is equal to `i`.

`StratiConstraints`
: numeric matrix or character(with default): input object for the statigraphic relation between samples. If there is stratigraphic relation between samples see the details section for instructions regarding how to correctly fill `StratiConstraints`. Otherwise, default value is suitable.

`sepSC`
: character (with default): if `StratiConstraints` is character, indicate column separator in `StratiConstraints` .csv file.

`LIN_fit`
: logical (with default): if TRUE (default) allows a linear component, on top of the (default) saturating exponential curve, for the fitting of dose response curves. Please see details for more informations on the proposed dose response curves.

`Origin_fit`
: logical (with default): if TRUE, forces the dose response curves to pass through the origin. Please see details for more informations on the proposed growth curves.

`distribution`
: character (with default): type of distribution that defines how individual equivalent dose values are distributed around the palaeodose. Allowed inputs are **"cauchy"**, **"gaussian"** and **"lognormal"**.

`Taille`
: integer (with default): number of iterations for the MCMC computation (for more information see `jags.model`).

`t`
: integer (with default): 1 every `t` iterations of the MCMC is considered for sampling the posterior distribution (for more information see `jags.model`).

`Nb_chaines`
: integer (with default): number of independent chains for the model (for more information see `jags.model`).

Details

**** How to fill `StratiConstraints` ? ****

If there is stratigraphic relations between samples, information in DATA must be ordered by order of increasing ages. The user can fill the `StratiConstraints` matrix as follow.
First, concerning the **size of the matrix**, row number of `StratiConstraints` matrix is equal to `Nb_sample+1`, and column number is equal to `Nb_sample`.
Secondly, conserning the **first line of the matrix**, for all `i in {1,...,Nb_Sample}`, `StratiConstraints[1,i]=1` that means the lower bound of the sample age given in `PriorAge[2i-1]` for the sample whose number ID is equal to `i`, is taken into account.
Thirdly, concerning the **sample relations**, for all `j in {2,...,Nb_Sample+1}` and all `i in {j,...,Nb_Sample}`,

StratiConstraints[j,i]=1 if sample age whose number ID is equal to j-1 is lower than sample age whose number ID is equal to i. Otherwise, StratiConstraints[j,i]=0.
Note that StratiConstraints_{2:Nb_sample+A,1:Nb_sample} is a upper triangular matrix.

The user can also use SCMatrix function to construc the StratiConstraints matrix.

The user can also refer to a .csv file that containts the relation between samples as defined above.

**** How to fill THETA concerning common and individual error? ****

If systematic errors are considered, the user can fill the THETA matrix as follow.
Note that row number of THETA is equal the column number. For all i in {1,...,Nb_sample}, THETA[i,i] containts individual error plus systematic error of the sample whose number ID is equal to i. For all i,j in {1,...,Nb_sample} and i different from j , THETA[i,j] containts common error between samples whose number ID are equal to i and j.
Note that THETA[i,j] is a symetric matrix.

The user can also refer to a .csv file that containts the errors as defined above.

**** Different growth curves ****

As for Age_Computation, the user can choose from 4 dose response curves:

- **Saturating exponential plus linear growth** (AgeMultiBF_EXPLIN):
  for all x in IR+, f(x)=a(1-exp(-x/b))+cx+d; select
  - LIN_fit=TRUE
  - Origin_fit=FALSE
- **Saturating exponential growth** (AgeMultiBF_EXP):
  for all x in IR+, f(x)=a(1-exp(-x/b))+d; select
  - LIN_fit=FALSE
  - Origin_fit=FALSE
- **Saturating exponential plus linear growth and fitting through the origin** (AgeMultiBF_EXPLINZO):
  for all x in IR+, f(x)=a(1-exp(-x/b))+cx; select
  - LIN_fit=TRUE
  - Origin_fit=TRUE
- **Saturating exponential growth and fitting through the origin** (AgeMultiBF_EXPZO):
  for all x in IR+, f(x)=a(1-exp(-x/b)); select
  - LIN_fit=FALSE
  - Origin_fit=TRUE

Value

1- NUMERICAL OUTPUT

A list containing the following objects:

- **Sampling** that corresponds to a sample of the posterior distributions of the Age, palaeodose and equivalent dose dispersion parameters.
- **Model_GrowthCurve**, stating which dose response fitting option was chosen;
- **Distribution**, stating which distribution was chosen to model the dispersion of individual equivalent dose values around the palaeodose of the sample;
- **PriorAge**, stating the priors used for the age parameter.

The Gelman and Rudin test of convergency

Prints the result of the Gelman and Rudin test of convergency for the age, palaeodose and equivalent dose dispersion parameters for each sample. A result close to one is expected.
In addition, the user must visually assess the convergency of the trajectories by looking at the pdf file generated by the function (see 2- for more informations).
If both convergencies (Gelman and Rudin test and plot checking) are satisfactory, the user can consider the printed estimates as valid. Otherwise, the user may try increasing the number of MCMC interations (`Taille`) to reach convergency.

Credible intervals and Bayes estimates

Prints the Bayes esitmates, the credible intervals at level 95% and 68% for the age, palaeodose and equivalent dose dispersion parameters for each sample.

2- PLOT OUTPUT

MCMC trajectories

A pdf file with the MCMC trajectories and posterior distributions of the age, palaeodose and equivalent dose dispersion parameters. There is one page per sample.
The first line of the figure correponds to the age parameter, the second to the palaeodose parameter and the third to the equivalent dose dispersion parameter. On each line, the plot on the left represents the MCMC trajectories, and the one on the right the posterior distribution of the parameter.

Summary of sample ages

Plot credible intervals and bayes estimate of each sample age on a same graph.

Stratigraphic constraints

If there are stratigraphic constraints, plot sampling of each sample two by two...

Author(s)

Claire Christophe, Guillaume Guerin

References

Combes, Benoit and Philippe, Anne, 2017. Bayesian analysis of multiplicative Gaussian error for multiple ages estimation in optically stimulated luminescence dating. Quaternary Geochronology (in press)

Combes, B., Philippe, A., Lanos, P., Mercier, N., Tribolo, C., Guerin, G., Guibert, P., Lahaye, C., 2015. A Bayesian central equivalent dose model for optically stimulated luminescence dating. Quaternary Geochronology 28, 62-70. doi:10.1016/j.quageo.2015.04.001

See Also

`Generate_DataFile`, `rjags` package, `MCMC_plot`, `Age_Computation`, `SCMatrix`

Examples

```
## Load data
# data(DATA1,envir = environment())
# data(DATA2,envir = environment())
# DATA=Concat_DataFile(DATA2,DATA1)

## Age computation of samples GDB5 and GDB3,
## without common error and without stratigraphic constraints
# Age=AgeS_Computation(DATA,Nb_sample=2,SampleNames=c("GDB5","GDB3"),SavePdf=FALSE,Taille=1000)
```

```
## Age computation of samples GDB5 and GDB3,
## without common error, assuming GDB5 age younder than GDB3 age
# SC=matrix(data=c(1,1,0,1,0,0),ncol=Nb_sample,nrow = (Nb_sample+1),byrow = T)
# Age=AgeS_Computation(DATA,Nb_sample=2,SampleNames=c("GDB5","GDB3"),
#    StratiConstraints=SC,Taille=10000)
```

`Age_Computation` *Bayesian analysis for the OSL age estimation of one sample*

Description

This function computes the age of a sample according to the model developed in Combes and Philippe (2017), based on an output of the `Generate_DataFile` function.
A sample, for which data is avalilable in several BIN files, can be analysed.

Usage

```
Age_Computation(DATA, samplename, SavePdf = FALSE,
  OutputFileName = c("MCMCplot"), OutputFilePath = c(""),
  SaveEstimates = FALSE, OutputTableName = c("DATA"),
  OutputTablePath = c(""), BinPerSample = c(1), PriorAge = c(0.01, 100),
  LIN_fit = TRUE, Origin_fit = FALSE, distribution = c("cauchy"), I = 1,
  Taille = 50000, t = 5, Nb_chaines = 3)
```

Arguments

`DATA`	list of objects: LT, sLT, ITimes, dLab, ddot_env, regDose, J, K, Nb_measurement, provided by the function `Generate_DataFile`. `DATA` can contain information for more than one sample.
`samplename`	character: name of the sample.
`SavePdf`	boolean (with default): if TRUE save graph in pdf file named `OutputFileName` in folder `OutputFilePath`.
`OutputFileName`	character (with default): name of the pdf file that will be generated by the function if `SavePdf`=TRUE.
`OutputFilePath`	character (with default): path to the pdf file that will be generated by the function if `SavePdf`=TRUE.
`SaveEstimates`	boolean (with default): if TRUE save Bayes estimates and confidence interval at level 68% and 95%, in a csv table named `OutputFileName` in folder `OutputFilePath`.
`OutputTableName`	character (with default): name of the table that will be generated by the function if `SaveEstimates`=TRUE.
`OutputTablePath`	character (with default): path to the table that will be generated by the function if `SaveEstimates`=TRUE.
`BinPerSample`	integer vector (with default): vector with the number of BIN files per sample. If in `DATA` there is more than one sample, the `BinPerSample` vector must be the same as that used to run the function `Generate_DataFile` for generating the `DATA` object.

PriorAge	numeric vector (with default): lower and upper bounds for the sample age parameter. length(PriorAge)=2.
LIN_fit	logical (with default): if TRUE (default) allows a linear component, on top of the (default) saturating exponential curve, for the fitting of dose response curves. Please see details for more informations on the proposed dose response curves.
Origin_fit	logical (with default): if TRUE, forces the dose response curves to pass through the origin. Please see details for more informations on the proposed growth curves.
distribution	character (with default): type of distribution that defines how individual equivalent dose values are distributed around the palaeodose. Allowed inputs are **"cauchy"**, **"gaussian"** and **"lognormal"**.
I	integer (with default): if DATA contains data from more than one sample, I indicates the ID number of the sample to be analysed.
Taille	integer (with default): number of iterations for the MCMC computation (for more information see jags.model).
t	integer (with default): 1 every t iterations of the MCMC is considered for sampling the posterior distribution (for more information see jags.model).
Nb_chaines	integer (with default): number of independent chains for the model (for more information see jags.model).

Details

For more flexibility, the user can choose from 4 dose response curves:

- **Saturating exponential plus linear growth** (AgeMultiBF_EXPLIN):
  for all x in IR+, f(x)=a(1-exp(-x/b))+cx+d; select
  - LIN_fit=TRUE
  - Origin_fit=FALSE
- **Saturating exponential growth** (AgeMultiBF_EXP):
  for all x in IR+, f(x)=a(1-exp(-x/b))+d; select
  - LIN_fit=FALSE
  - Origin_fit=FALSE
- **Saturating exponential plus linear growth and fitting through the origin** (AgeMultiBF_EXPLINZO):
  for all x in IR+, f(x)=a(1-exp(-x/b))+cx; select
  - LIN_fit=TRUE
  - Origin_fit=TRUE
- **Saturating exponential growth and fitting through the origin** (AgeMultiBF_EXPZO):
  for all x in IR+, f(x)=a(1-exp(-x/b)); select
  - LIN_fit=FALSE
  - Origin_fit=TRUE

Value

1- NUMERICAL OUTPUT

A list containing the following objects:

- **Sampling** that corresponds to a sample of the posterior distributions of the Age, palaeodose and equivalent dose dispersion parameters.
- **Model_GrowthCurve**, stating which dose response fitting option was chosen;
- **Distribution**, stating which distribution was chosen to model the dispersion of individual equivalent dose values around the palaeodose of the sample;
- **PriorAge**, stating the priors used for the age parameter.

The Gelman and Rudin test of convergency

Prints the result of the Gelman and Rudin test of convergency for the age, palaeodose and equivalent dose dispersion parameters. A result close to one is expected.
In addition, the user must visually assess the convergency of the trajectories by looking at the pdf file generated by the function (see 2- for more informations).
If both convergencies (Gelman and Rudin test and plot checking) are satisfactory, the user can consider the printed estimates as valid. Otherwise, the user may try increasing the number of MCMC interations (`Taille`) to reach convergency.

Credible intervals and Bayes estimates

Prints the Bayes esitmates, the credible intervals at level 95% and 68% for the age, palaeodose and equivalent dose dispersion parameters of the sample.

2- PLOT OUTPUT

A pdf file with the MCMC trajectories and posterior distributions of the age, palaeodose and equivalent dose dispersion parameters.
The first line of the figure correponds to the age parameter, the second to the palaeodose parameter and the third to the equivalent dose dispersion parameter. On each line, the plot on the left represents the MCMC trajectories, and the one on the right the posterior distribution of the parameter.

Author(s)

Claire Christophe, Guillaume Guerin

References

Combes, Benoit and Philippe, Anne, 2017. Bayesian analysis of multiplicative Gaussian error for multiple ages estimation in optically stimulated luminescence dating. Quaternary Geochronology (in press)

Combes, B., Philippe, A., Lanos, P., Mercier, N., Tribolo, C., Guerin, G., Guibert, P., Lahaye, C., 2015. A Bayesian central equivalent dose model for optically stimulated luminescence dating. Quaternary Geochronology 28, 62-70. doi:10.1016/j.quageo.2015.04.001

See Also

`Generate_DataFile`, `rjags` package, `MCMC_plot`

Examples

```
## load data file generated by the function Generate_DataFile
# data(DATA1,envir = environment())
# Age=Age_Computation(DATA1,samplename="GDB3",SavePdf=FALSE,Taille=1000)
```

Concat_DataFile *Concatenates two outputs of the function* Generate_DataFile

Description

This function allows concatenating two lists provided as output of the Generate_DataFile function.
Only concatenation of 2 files is possible.

Usage

```
Concat_DataFile(u1, u2)
```

Arguments

u1	list of objects: LT, sLT, ITimes, dLab, ddot_env, regDose, J, K, Nb_measurement.
u2	list of objects: LT, sLT, ITimes, dLab, ddot_env, regDose, J, K, Nb_measurement.

Value

A List of objects: LT, sLT, ITimes, dLab, ddot_env, regDose, J, K, Nb_measurement.

Author(s)

Claire Christophe, Guillaume Guerin

See Also

Generate_DataFile

Examples

```
# load data files
data(DATA1,envir = environment())
data(DATA2,envir = environment())
# concatenate two data files
DATA3=Concat_DataFile(DATA1,DATA2)
str(DATA3)
```

DATA1 *DATA on sample named GDB3*

Description

list of objects: LT, sLT, ITimes, dLab, ddot_env, regDose, J,K,Nb_measurement obtained using Generate_DataFile function from luminescence data.

Usage

```
data("DATA1")
```

Format

A data frame with 0 observations on the following 2 variables.

`LT` (one list per sample); each list contains all L/T values for the corresponding sample;

`sLT` (one list per sample); each list contains all uncertainties on L/T values for the corresponding sample;

`ITimes` (one list per sample); each list contains irradiation time values for the corresponding sample;

`dLab` , a matrix containing in line `i`, the laboratory dose rate and its variance for sample `i`;

`ddot_env` , a matrix containing in line `i`, the environmental dose rate and its variance (excluding the common error terms) for sample `i`;

`regDose` (one list per sample); each list contains all regenerated doses;

`J` , a vector giving, for each BIN file, the number of aliquots selected for the analysis;

`K` , a vector giving, for each BIN file, the number of regenerative doses in the SAR protocol;

`Nb_measurement` , a vector giving, for each BIN file, the number of measurements;

Details

A FAIRE

Source

A FAIRE

References

A FAIRE

Examples

```
data(DATA1)
str(DATA1)
```

DATA2 *DATA on sample named GDB5*

Description

list of objects: LT, sLT, ITimes, dLab, ddot_env, regDose, J,K,Nb_measurement obtained using Generate_DataFile function from luminescence data.

Usage

```
data("DATA2")
```

Format

A data frame with 0 observations on the following 2 variables.

`LT` (one list per sample); each list contains all L/T values for the corresponding sample;

`sLT` (one list per sample); each list contains all uncertainties on L/T values for the corresponding sample;

`ITimes` (one list per sample); each list contains irradiation time values for the corresponding sample;

`dLab` , a matrix containing in line `i`, the laboratory dose rate and its variance for sample `i`;

`ddot_env` , a matrix containing in line `i`, the environmental dose rate and its variance (excluding the common error terms) for sample `i`;

`regDose` (one list per sample); each list contains all regenerated doses;

`J` , a vector giving, for each BIN file, the number of aliquots selected for the analysis;

`K` , a vector giving, for each BIN file, the number of regenerative doses in the SAR protocol;

`Nb_measurement` , a vector giving, for each BIN file, the number of measurements;

Details

A FAIRE

Source

A FAIRE

References

A FAIRE

Examples

```
data(DATA2)
str(DATA2)
```

Generate_DataFile *Generates, from one (or several) BIN file(s), a list of luminescence data and information before statistical analysis*

Description

This function is used to generate, from the BIN file(s), a list of values of: OSL intensities and associated uncertainties, regenerative doses, etc., which will be the input of the Bayesian models. To be easy-to-use, this function requires a rigorous organisation - all needed files should be arranged in one folder - of informations concerning each BIN file. It is possible to process data for various samples simultaneously and to consider more than one BIN file per sample.

Usage

```
Generate_DataFile(Path, Names, Nb_sample, Nb_binfile = length(Names),
  BinPerSample = rep(1, Nb_sample), sepDP = c(","), sepDE = c(","),
  sepDS = c(","), sepR = c("="))
```

Arguments

Path	character **(required)**: the path to the project folder, containing one or more subfolders in which the BIN files are located.
Names	character vector **(required)**: list of names of the sub-folders containing the BIN files - each subfolder must contain a BIN file and associated .csv files. See details for more informations on associated .csv files required in the subfolders. If there is more than one BIN file per sample, see the details section for instructions regarding how to correctly fill the Names vector.
Nb_sample	integer **(required)**: number of samples.
Nb_binfile	integer (with default): number of BIN files. It must be equal to, or greater than Nb_sample.
BinPerSample	integer vector (with default): vector with the number of BIN files per sample. The length of this vector must be equal to Nb_sample and the sum of entries of this vector must be equal to Nb_binfile. If there is more than one BIN file per sample, see the details section for instructions regarding how to correctly fill BinPerSample vector. Otherwise, this vector must contain a list of 1 values.
sepDP	character (with default): column separator in the DiscPose.csv files.
sepDE	character (with default): column separator in the DoseEnv.csv files.
sepDS	character (with default): column separator in the DoseLab.csv files.
sepR	character (with default): column separator in the Rule.csv files.

Details

With Path and Names, this function goes to the subfolders containing the BIN files and associated information to compute the luminescence data.

**** What are the required files in each subfolder? ****

Each subfolder can be named, for example, as the sample name followed by a number; it must contain:

- **bin.BIN**, the bin file renamed as bin.BIN (note: the name of all files matters);
- **DiscPos.csv**, a two columns .csv file containing the list of disc and grain position number of the previously selected grains (typically this list will include the position of grains based on their sensitivity, recycling or other properties);
- **DoseEnv.csv**, a two columns file containing the observation of the natural (or environmental), dose rate, and its non-shared variance (i.e. after removing all shared errors). Note: the user shall provide the squared value of the error associated with the dose rate experienced by the sample grains in nature;
- **DoseSourve.csv**, a two columns file containing the observation of the laboratory dose rate, and its variance (squared error);
- **rule.csv**, a .csv file containing information on
  - beginSignal= the first channel for summing the natural or regenerative OSL signal (typically 1 or 6);
  - endSignal= the last channel for summing the natural or regenerative OSL signal (typically 5 or 10);
  - beginBackground= the first channel for background estimation of the natural or regenerative OSL signal (typically 76 or 81);
  - endBackground= the last channel for background estimation of the natural or regenerative OSL signal (typically 95 or 100);

- beginTest,
- endTest,
- beginTestBackground,
- endTestBackground= same values as above, for the test dose response (typically the same values should be used);
- inflatePercent= uncertainty arising from the instrument reproducibility (typically 0.02, i.e. 2%);
- nbOfLastCycleToRemove= number of cycles at the end of the SAR protocol which should not be included in the dose response curve fitting (typically 1 if only a recycling test is performed, or 2 if both recycling and IR depletion are tested).

** **How to fill the** `Names` **vector?** **

`Names` is a vector of length `Nb_binfile`. `Names`[i] is the name (e.g., Sample1-File1, or successive names separated by "/" signs, if BIN files are in subfolders, e.g. Sample1/File1) of the subfolder containing all informations on the BIN file of ID number `i`. The names in `Names` are ordered following two rules:

- The names in the `Names` vector must be ordered following the sample order (the names of subfolders containing BIN files for the same sample should follow each other in the `Names` vector, e.g. Sample1, Sample2-File1, Sample2-File2, etc.).

- If stratigraphic constraints apply to samples, and so a **Bayesian model with stratigraphic constraints** is implemented, then the names in the `Names` vector must be ordered by order of increasing ages.
  For example, `Names=c(noun1,noun2)`, in which case `noun1` (respectively, `noun2`) corresponds to the subfolder name containing the BIN file of sample 1 (respectively of sample 2). In addition, if we know that sample 1 is younger than sample 2, then `Names` vector is correctly filled. If conversely, `Names=c(noun2,noun1)`, the analysis performed by `AgeS_Computation` would not be consistent.

** **How to fill** the `BinPerSample` **vector?** **

`BinPerSample`[i] correponds to the number of BIN files for the sample whose number ID is equal to `i`.
For example, let us consider a case with two samples (Sample1 and Sample2), with 2 BIN files for Sample1 and 1 for Sample2. In this case, `Nb_binfile=3` and `Nb_sample=2`. The user may then set `Names=c("Sample1-File1", "Sample1-File2", "Sample2-File1")`, in which case `"Sample1-1"` is the name of the subfolder containing the first BIN file for Sample1, `"Sample1-File2"` the name of the subfolder for the second BIN file of Sample1; eventually, `"Sample2-1"` is the name of the subfolder containing the BIN file for the second sample. In this case, `BinPerSample=c(2,1)`.

For the general BIN-file structure, the reader is referred to the following website: `http://www.nutech.dtu.dk/`

The function `read_BIN2R` developped in `Luminescence` package is used to read the BIN files.

Value

A list containing the following objects:

- **LT** (one list per sample); each list contains all L/T values for the corresponding sample;
- **sLT** (one list per sample); each list contains all uncertainties on L/T values for the corresponding sample;
- **ITimes** (one list per sample); each list contains irradiation time values for the corresponding sample;

- **dLab**, a matrix containing in line `i`, the laboratory dose rate and its variance for sample `i`;
- **ddot_env**, a matrix containing in line `i`, the environmental dose rate and its variance (excluding the common error terms) for sample `i`;
- **regDose** (one list per sample); each list contains all regenerated doses;
- **J**, a vector giving, for each BIN file, the number of aliquots selected for the analysis;
- **K**, a vector giving, for each BIN file, the number of regenerative doses in the SAR protocol;
- **Nb_measurement**, a vector giving, for each BIN file, the number of measurements;

Author(s)

Claire Christophe, Guillaume Guerin

See Also

`read_BIN2R`, `Concat_DataFile`, `LT_RegenDose`

Examples

```
## 1) Example for one sample with one Bin File
## give the name of the path to data folder
# Path=c("inst/extdata/")
## give the name of the folder datat
# Names=c("GDB3")
## give the number of sample
# Nb_sample=1
# DATA=Generate_DataFile(Path,Names,Nb_sample)
# str(DATA)

## 2) Example for 2 samples and one Bin file each
## give the name of the path to data folder
# Path=c("inst/extdata/")
## give the name of the folder datat
# Names=c("GDB5","GDB3")
## give the number of sample
# Nb_sample=2
# DATA=Generate_DataFile(Path,Names,Nb_sample)
# str(DATA)
```

LT_RegenDose *Plots L/T as a function of regenerative dose*

Description

This function plots L/T values as a function of Regenerative Dose, for every selected aliquot.

Usage

```
LT_RegenDose(DATA, Path, Names, BinPerSample = rep(1), SampleNumber = 1,
  sepDP = c(","))
```

Arguments

DATA	list of objects: LT, sLT, ITimes, dLab, ddot_env, regDose, J, K, Nb_measurement, provided by the function Generate_DataFile. DATA can contain information from more than one sample.
Path	character: path to the project folder (the same as the one used in Generate_DataFile to provide DATA)
Names	character vector: list of names of the sub-folders containing the BIN files, which were used by Generate_DataFile to generate the DATA object.
BinPerSample	integer vector (with default): vector with the number of BIN files per sample, which was used in Generate_DataFile to generate the DATA object.
SampleNumber	integer(with default): ID number (in [1,Nb_sample]) of the sample selected for plotting L/T as a function of regenerative doses. Required if the DATA object contains information for more than one sample.
sepDP	character(with default): column separator in the DiscPose.csv file.

Details

To fill Names and BinPerSample, we refer to the detail section from the Generate_DataFile function.

Value

L/T plots; there are as many plots as selected aliquots in the DiscPos.csv file. There are 9 plots per page. There is not interpolation.

Author(s)

Claire Christophe, Guillaume Guerin

See Also

Generate_DataFile

Examples

```
## load data file generated by the function Generate_DataFile
# data(DATA1,envir = environment())
# LT_RegenDose(DATA1,Path=c('inst/extdata/'),Names=c("GDB3"))
```

MCMCsample *MCMC sample from the posterior distribution of the dataset GDB5*

Description

MCMC samples from the posterior distribution of "A" for age, "D" for palaeodose and "sD" for dispersion of equivalent doses around "D", of the data set GDB5.

Usage

```
data("MCMCsample")
```

Format

It is a matric with 6000 row and tree column.

`A` The first column of the matrice are sampled from the posterior distribution of the paramete `A`

`D` The first column of the matrice are sampled from the posterior distribution of the paramete `D`

`sD` The first column of the matrice are sampled from the posterior distribution of the paramete `sD`

Details

A FAIRE

Source

A FAIRE

References

A FAIRE

Examples

```
data(MCMCsample)
## maybe str(MCMCsample) ; plot(MCMCsample[,1],type="l") ...
```

MCMC_plot *MCMC trajectories and posterior distributions plot*

Description

This function uses the ouput of `coda.samples` to plot the trajectories of MCMC and densities of the posterior distributions of the age - if it is calculated, palaeodose and equivalent dose dispersion parameters of the sample. This function is used in the function `Age_Computation`.

Usage

```
MCMC_plot(sample, size, SampleNames, Nb_sample = 1, Nb_chaines = 3,
  value = c(0, Nb_sample, 2 * Nb_sample), nom = c("A", "D", "sD"))
```

Arguments

`sample`	MCMC.list: this is generated by `jags.model` in `Age_Computation`.
`size`	integer: length of each chain.
`SampleNames`	character vector: names of the samples, used in the figure titles.
`Nb_sample`	integer (with default): number of analysed samples.
`Nb_chaines`	integer (with default): number of independent chains for the model (for more informations, see `jags.model`).
`value`	integer vector (with default): position idex used to select age (if available), palaeodose and equivalent dose dispersion parameters for the first sample.
`nom`	character vector (with default): names of the selected parameters with value `A` for age (if available), `D` for palaeodose and `sD` for equivalent dose dispersion.

Value

A pdf file with the MCMC trajectories and posterior distributions for each parameter defined in nom. There is one page per sample, which is divided by length(nom) vertically and by 2 horizontally. The first line of the figure correponds to the first parameter defined in nom, and so on. On each line, the plot on the left represents the MCMC trajectories, and the one on the right the posterior distributions of the parameter.

Author(s)

Claire Christophe, Guillaume Guerin

See Also

Age_Computation, coda.samples and rjags packages.

Examples

```
data(MCMCsample,envir = environment())
MCMC_plot(MCMCsample,2000,SampleNames="GDB3",Nb_sample=1,Nb_chaines=3)
```

PalaeodoseComputation *Bayesian analysis for the palaeodose estimation of various samples*

Description

This function computes the palaeodose of one or various samples according to the model developed in Combes et al (2015), based on an output of the Generate_DataFile function.
Samples, for which data is avalilable in several BIN files, can be analysed.

Usage

```
PalaeodoseComputation(DATA, Nb_sample, BinPerSample = rep(1, Nb_sample),
  SampleNames, SavePdf = FALSE, OutputFileName = c("MCMCplot", "summary"),
  OutputFilePath = c(""), SaveEstimates = FALSE,
  OutputTableName = c("DATA"), OutputTablePath = c(""), LIN_fit = TRUE,
  Origin_fit = FALSE, distribution = c("cauchy"), Taille = 50000, t = 5,
  Nb_chaines = 3)
```

Arguments

DATA	list of objects: LT, sLT, ITimes, dLab, ddot_env, regDose, J, K, Nb_measurement, provided by the function Generate_DataFile. DATA contains information for more than one sample.
Nb_sample	integer **(required)**: number of samples, Nb_sampe>1.
BinPerSample	integer vector (with default): vector with the number of BIN files per sample. The length of this vector is equal to Nb_sample. BinPerSample[i] correponds to the number of BIN files for the sample whose number ID is equal to i. For more information to fill this vector, we refer to detatils in Generate_DataFile.
SampleNames	character vector: names of sample. The length of this vector is equal to Nb_sample.

SavePdf	boolean (with default): if TRUE save graph in pdf file named OutputFileName in folder OutputFilePath.
OutputFileName	character (with default): name of the pdf files that will be generated by the function.
OutputFilePath	character (with default): path to the pdf files that will be generated by the function.
SaveEstimates	boolean (with default): if TRUE save Bayes estimates and confidence interval at level 68 in a csv table named OutputFileName in folder OutputFilePath.
OutputTableName	character (with default): name of the table that will be generated by the function if SaveEstimates=TRUE.
OutputTablePath	character (with default): path to the table that will be generated by the function if SaveEstimates=TRUE.
LIN_fit	logical (with default): if TRUE (default) allows a linear component, on top of the (default) saturating exponential curve, for the fitting of dose response curves. Please see details for more informations on the proposed dose response curves.
Origin_fit	logical (with default): if TRUE, forces the dose response curves to pass through the origin. Please see details for more informations on the proposed growth curves.
distribution	character (with default): type of distribution that defines how individual equivalent dose values are distributed around the palaeodose. Allowed inputs are **"cauchy"**, **"gaussian"** and **"lognormal"**.
Taille	integer (with default): number of iterations for the MCMC computation (for more information see jags.model).
t	integer (with default): 1 every t iterations of the MCMC is considered for sampling the posterior distribution (for more information see jags.model).
Nb_chaines	integer (with default): number of independent chains for the model (for more information see jags.model).

Details

As for Age_Computation and AgeS_Computation, the user can choose from 4 dose response curves:

- **Saturating exponential plus linear growth** (AgeMultiBF_EXPLIN):
  for all x in IR+, f(x)=a(1-exp(-x/b))+cx+d; select
  - LIN_fit=TRUE
  - Origin_fit=FALSE
- **Saturating exponential growth** (AgeMultiBF_EXP):
  for all x in IR+, f(x)=a(1-exp(-x/b))+d; select
  - LIN_fit=FALSE
  - Origin_fit=FALSE
- **Saturating exponential plus linear growth and fitting through the origin** (AgeMultiBF_EXPLINZO):
  for all x in IR+, f(x)=a(1-exp(-x/b))+cx; select
  - LIN_fit=TRUE
  - Origin_fit=TRUE

- **Saturating exponential growth and fitting through the origin** (`AgeMultiBF_EXPZO`): for all x in IR+, `f(x)=a(1-exp(-x/b))`; select
  - `LIN_fit=FALSE`
  - `Origin_fit=TRUE`

Value

1- NUMERICAL OUTPUT

A list containing the following objects:

- **Sampling** that corresponds to a sample of the posterior distributions of palaeodose and equivalent dose dispersion parameters.
- **Model_GrowthCurve**, stating which dose response fitting option was chosen;
- **Distribution**, stating which distribution was chosen to model the dispersion of individual equivalent dose values around the palaeodose of the sample;
- **PriorAge**, stating the priors used for the age parameter.

The Gelman and Rudin test of convergency

Prints the result of the Gelman and Rudin test of convergency for palaeodose and equivalent dose dispersion parameters for each sample. A result close to one is expected.
In addition, the user must visually assess the convergency of the trajectories by looking at the pdf file generated by the function (see 2- for more informations).
If both convergencies (Gelman and Rudin test and plot checking) are satisfactory, the user can consider the printed estimates as valid. Otherwise, the user may try increasing the number of MCMC interations (`Taille`) to reach convergency.

Credible intervals and Bayes estimates

Prints the Bayes esitmates, the credible intervals at level 95% and 68% for the palaeodose and equivalent dose dispersion parameters for each sample.

2- PLOT OUTPUT

MCMC trajectories

A pdf file with the MCMC trajectories and posterior distributions of the palaeodose and equivalent dose dispersion parameters. There is one page per sample.
The first line of the figure correponds to the palaeodose parameter and the second to the equivalent dose dispersion parameter. On each line, the plot on the left represents the MCMC trajectories, and the one on the right the posterior distribution of the parameter.

Author(s)

Claire Christophe, Guillaume Guerin

References

Combes, B., Philippe, A., Lanos, P., Mercier, N., Tribolo, C., Guerin, G., Guibert, P., Lahaye, C., 2015. A Bayesian central equivalent dose model for optically stimulated luminescence dating. Quaternary Geochronology 28, 62-70. doi:10.1016/j.quageo.2015.04.001

See Also

`Generate_DataFile`, `rjags` package, `MCMC_plot`, `Age_Computation`, `SCMatrix`

Examples

```
## Load data
# data(DATA1,envir = environment())
## Palaeodose computation of samples GDB3
# P=PalaeodoseComputation(DATA1,Nb_sample=1,SampleNames=c("GDB5"),Taille=1000)
```

SCMatrix *Definition of the stratigraphic constraint matrix*

Description

This function helps to define the stratigraphic relation between samples, with questions. The output of this function can be used in function AgeS_Computation.

Usage

```
SCMatrix(Nb_sample, Names)
```

Arguments

Nb_sample	interger: the sample number.
Names	charcater vector: sample names.

Value

A Matrix that summarise the ordered relation between samples. This matrix can be intergrate in AgeS_Computation function. We refer to detail on AgeS_Computation for more information concerning this matrix.

Author(s)

Claire Christophe, Guillaume Guerin

See Also

AgeS_Computation

Examples

```
## Assume that "sample1" is younger than "sample2"
# SCMatrix(2,c("sample1","sample2"))
# 1
```

Index