Package: NiLeDAM (via r-universe)

October 13, 2024

Title Monazite Dating for the NiLeDAM Team

Version 0.3

Date 2023-09-17

Maintainer Nathalie Vialaneix <nathalie.vialaneix@inrae.fr>

Description Th-U-Pb electron microprobe age dating of monazite, as originally described in <doi:10.1016/0009-2541(96)00024-1>.

License GPL (>= 2)

Depends R (>= 2.4.0)

Imports dplyr, ggplot2, magrittr, methods, nleqslv, rlang, scales, shiny, shinyjs, shinythemes, stats, thematic, tidyr

Suggests knitr, rmarkdown, spelling, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Encoding UTF-8

Repository CRAN

RoxygenNote 7.2.3

Collate 'init.R' 'describeNiLeDAM.R' 'describeSrilanka.R' 'ages.R' 'calculate.R' 'oneAgeTest.R' 'ageTests.R' 'tests.R' 'ages-oneAgeTest-ageTests.R' 'liveNILEDAM.R'

Language en-US

NeedsCompilation no

Author Nathalie Vialaneix [aut, trl, cre], Aurélie Mercadié [aut], Jean-Marc Montel [aut], Anne-Magali Seydoux-Guillaume [ctb]

Date/Publication 2023-09-18 06:00:02 UTC

Contents

`ages-class` *Class* `"ages"`

Description

A class for the results of the `calculateAges` function. It contains the ages calculated for each analytic point from triplets (U,Th,Pb) and corresponding errors.

Fields

`data` Object of class `"data.frame"` that contains the original triplets (U,Th,Pb) and corresponding errors.

`ages` Object of class `"vector"` that contains the ages calculated for each analytic point.

`ci` Object of class `"matrix"`, which is a two-column matrix that contains the bounds of the confidence intervals for each age of the slot `ages`. These are estimated from a MC simulation method as described in `calculateAges()`.

`sd` Object of class `"vector"` that contains the standard deviations for each age of the slot `ages`. These are estimated from a MC simulation method as described in `calculateAges()`.

`nloops` Object of class `"numeric"` that stores the number of MC simulations used to estimate the confidence intervals and the standard deviations.

`level` Object of class `"numeric"` that stores the level of the confidence intervals.

Objects from the Class

Objects can be created by calls of the function `calculateAges()`.

Methods

- print(`signature(x = "ages")`): display a summary of the results of the ages calculation.
- show(`signature(object = "ages")`): display a summary of the results of the ages calculation (same as `print.ages`).
- summary(`signature(object = "ages")`): display a summary of the results of the ages calculation (same as `print.ages`).

Author(s)

Nathalie Vialaneix, <nathalie.vialaneix@inrae.fr>

See Also

calculateAges(), tests()

Examples

```
showClass("ages")

## Example on the srilanka dataset

data(srilanka)
calculated.ages <- calculateAges(srilanka, nloops = 10)
calculated.ages
```

---

ageTests-class *Class* "ageTests"

Description

Results of χ^2-tests testing to which number of populations the ages calculated by the function calculateAges() are the most likely to come from. The test is the one described in the article Montel *et al.* (1996).

Fields

nb.pop Object of class "vector" that contains the tested numbers of populations.

best.nb Object of class "numeric" that contains the most probable number of populations, which is the smallest number for which the χ^2-test is accepted.

best.res Object of class "oneAgeTest" that contains the result of the test for best.nb populations.

Objects from the Class

Objects from this class are created by the function tests() applied to an object of class ages when the option 'nbmax' is not set to NULL but to an integer larger than 'nbmin'.

Methods

- print(signature(x = "ageTests")): displays a summary of the test's results.
- show(signature(object = "ageTests")): displays a summary of the test's results.
- summary(signature(object = "ageTests")): displays a summary of the test's results.

Author(s)

Nathalie Vialaneix, <nathalie.vialaneix@inrae.fr>

References

Montel J.M., Foret S., Veschambre M., Nicollet C., Provost A. (1996) Electron microprobe dating of monazite. *Chemical Geology*, **131**, 37–53.

See Also

ages, tests(), plot.ageTests

Examples

```
showClass("ageTests")

## Example
data(srilanka)
calculated.ages <- calculateAges(srilanka, nloops = 10)
tests(calculated.ages, nbmax = 3)
```

`calculateAges` *Calculate the ages from electron microprobe measurements.*

Description

This function calculates the ages, confidence intervals and standard deviations from triplets (U,Th,Ph), obtained by electron microprobe, given together with corresponding errors.

Usage

```
calculateAges(measures, nloops = 1000, level = 0.05, verbose = TRUE, seed = NULL)
```

Arguments

`measures`	a data.frame object with one electron microprobe measurement by row and with columns U, corresponding error for U, Th, corresponding error for Th, Pb, corresponding error for Pb, all expressed in ppm. See an example with `data(srilanka)`.
`nloops`	the number of Monte Carlo (MC) simulations used to estimate the confidence intervals for the ages. Default is `1000`.
`level`	the level of significance of the confidence intervals for the ages. Default is `0.05`.
`verbose`	logical; activates the verbose mode. Default is `TRUE`.
`seed`	if supplied, initialize the random seed. Default is `NULL` (the random seed is not initialized).

Details

The ages are calculated by solving the Equation (1) of Montel *et al.* (1996). The equation is solved by the Broyden method implemented in the `nleqslv()` function.

The standard deviations and the confidence intervals are calculated using a MC approach: randomized observations of the triplets are generated from normal distributions with standard deviations equal to $\frac{\text{error}}{2}$ where 'error' denotes the error in the measurement of Th, U or Pb, passed in '`measures`'. Standard deviations are estimated by the empirical standard deviations and confidence intervals by quantiles for probabilities $\frac{\texttt{level}}{2}$ and $1 - \frac{\texttt{level}}{2}$, respectively.

Value

An object of class `ages`.

Note

You should use at least 1000 MC simulations otherwise the estimated confidence intervals and standard deviations will not be reliable. Such simulations can take a few seconds/minutes for fifty or so triplets and corresponding errors.

Author(s)

Jean-Marc Montel, <jean-marc.montel@ensg.inpl-nancy.fr>

Nathalie Vialaneix, <nathalie.vialaneix@inrae.fr>

References

Montel J.M., Foret S., Veschambre M., Nicollet C., Provost A. (1996) Electron microprobe dating of monazite. *Chemical Geology*, **131**, 37–53.

See Also

`ages` `tests`

Examples

```
## Load the data
data(srilanka)

## Calculate the ages
calculateAges(srilanka, nloops=10)
```

liveNiLeDAM *Graphical Web User Interface for 'NiLeDAM'*

Description

This function starts the graphical user interface with the default system browser. This interface is more likely to work properly with Firefox https://www.mozilla.org/fr/firefox/new/. In case Firefox is not your default browser, copy/paste http://localhost:8100 into the address bar.

Usage

```
liveNiLeDAM()
```

Value

Starts the 'NiLeDAM' GUI.

Author(s)

Aurélie Mercadié, <aurelie.mercadie@inrae.fr>

Nathalie Vialaneix, <nathalie.vialaneix@inrae.fr>

NiLeDAM *Monazite Dating for the 'NiLeDAM' Team*

Description

The aim here is to provide facilities to date monazite using electron microprobe analysis (EPMA). More precisely, the 'NiLeDAM' package has three main features:

- **ages calculation** from a set of analyses of the triplets (U,Th,Pb) contents and corresponding errors, it calculates the age of each analytic point by solving Equation (1) of Montel *et al.* (1996)
- **common age(s) estimation** from these results, it estimates the common age(s) under the assumption that the analyses come from a given number of age populations and test for this assumption (χ^2-test)
- **number of populations estimation** from the calculated ages, it estimates the most probable number of age populations from which the analyses come.

The package contains two main functions: `calculateAges()`, that calculate the ages and `tests()`, that test for the number of age populations.

An example is provided with the data `srilanka`, which can be run using `demo(srilanka)`.

Author(s)

Aurélie Mercadié, <aurelie.mercadie@inrae.fr>

Jean-Marc Montel, <jean-marc.montel@ensg.inpl-nancy.fr>

Nathalie Vialaneix, <nathalie.vialaneix@inrae.fr>

Maintainer: Nathalie Vialaneix

References

Montel J.M., Foret S., Veschambre M., Nicollet C., Provost A. (1996) Electron microprobe dating of monazite. *Chemical Geology*, **131**, 37–53.

Examples

```
## Not run: demo(srilanka)
```

---

oneAgeTest-class *Class* "oneAgeTest"

Description

Results of a χ^2-test testing if the ages calculated by the function `calculateAges()` come from a given number of populations. The test is the one described in the article Montel *et al.* (1996).

Fields

`data` Object of class `"ages"` as created by the function `calculateAges()`. This is the object passed as an argument to the function `tests()`.

`ages` Object of class `"vector"` that contains the estimated common age(s) of the population(s). Its size is equal to the argument 'nbmin' passed to the function `tests()`.

`sd` Object of class `"vector"` that contains the estimated standard deviations of the common age(s) of the population(s). Its size is equal to the argument 'nbmin' passed to the function `tests()`.

`ic` Object of class `"matrix"` having 'nbmin' rows and two columns corresponding to the confidence intervals at level 'level' as passed as an argument to the function `tests()`.

`S` Object of class `"numeric"` χ^2 test statistic of the test.

`thres` Object of class `"numeric"` probability threshold of the test.

`df` Object of class `"numeric"` degree of freedom of the test.

`level` Object of class `"numeric"` confidence level of the test. Passed as an argument to the function `tests()`.

`h0` Object of class `"logical"` result of the test: if `h0` is `TRUE`, the test is accepted at level 'level' and if `h0` is `FALSE`, the test is rejected at level 'level'.

`which.pop` Object of class `"vector"` that contains the population number for each analytic point. `1` corresponds to the population with the first age in `ages`, `2` to the population with the second age in `ages`...

Objects from the Class

Objects from this class are created by the function `tests()` applied to an object of class `ages` when the option 'nbmax' is set to NULL.

Methods

- print(`signature(x = "oneAgeTest")`): displays a summary of the test's result.
- show(`signature(object = "oneAgeTest")`): displays a summary of the test's result.
- summary(`signature(object = "oneAgeTest")`): displays a summary of the test' results.

Author(s)

Nathalie Vialaneix, <nathalie.vialaneix@inrae.fr>

References

Montel J.M., Foret S., Veschambre M., Nicollet C., Provost A. (1996) Electron microprobe dating of monazite. *Chemical Geology*, **131**, 37–53.

See Also

`ages`, `tests()`, `plot.oneAgeTest`

Examples

```
showClass("oneAgeTest")

## Example

data(srilanka)
calculated.ages <- calculateAges(srilanka, nloops=10)
tests(calculated.ages)
```

---

plot-methods *Method* "plot" *for Classes* "oneAgeTest" *and* "ageTests"

Description

Display densities of the ages calculated by the function `calculateAges()`, as well as the common age(s) density(ies) as returned by the function `tests()`.

Usage

```
## S4 method for signature 'oneAgeTest'
plot(x, y = NULL, main = "", col = "red")
```

Arguments

x	an object of class "oneAgeTest" or "ageTests" (see section Methods)
y	NUll
main	a character string, title of the plot
col	a character string, color of the common age(s) density(ies)

Details

The estimated densities are Gaussian, as supposed by the model described in Montel *et al.* (1996). Each density is referenced by its number to help identify outliers or invalid measurements.

Also note that if an object of class `oneAgeTest` is passed to the function, the common age(s) densities is displayed even if the test of the function `tests()` is rejected.

If argument 'col' is supplied, it is used to display the density(ies) of the common age(s).

Usage

plot(x, ...)

Methods

- `signature(object = "oneAgeTest")`: an object of type `oneAgeTest` as produced by the function `tests()`.
- `...`: further arguments passed to the function `plot()`.
- `signature(object = "ageTests")`: an object of type `ageTests` as produced by the function `tests()`.
- `...`: further arguments passed to the function `plot()`.

Author(s)

Nathalie Vialaneix, <nathalie.vialaneix@inrae.fr>

References

Montel J.M., Foret S., Veschambre M., Nicollet C., Provost A. (1996) Electron microprobe dating of monazite. *Chemical Geology*, **131**, 37–53.

See Also

`tests()`, `oneAgeTest`, `ageTests`

Examples

```
data(srilanka)
calculated.ages <- calculateAges(srilanka, nloops = 10)
res.tests <- tests(calculated.ages, 1, 3)
plot(res.tests, main = "Densities", col = "lightseagreen")
```

popline-methods *Method* "popline" *for Classes* "oneAgeTest" *and* "ageTests"

Description

Display observations as well as their reference's population as calculated by the function tests() given their value of Pb (ppm) and Th*(ppm), where Th* is a function of Th and U variables.

Usage

```
## S4 method for signature 'oneAgeTest'
popline(x, main2 = "")
```

Arguments

x	an object of class "oneAgeTest" or "ageTests" (see section Methods)
main2	a character string, title of the plot

Details

Th* is computed thanks to the following formula: $Th + 232 * U * [238(exp(4.9475 * 10^{-5} * Age) - 1)]^{-1} * [[exp(9.8485 * 10^{-4} * Age + 138 * exp(1.55125 * 10^{-4} * Age))][139] - 1]$

Slopes, depending directly of the ages computed with the function tests(), are obtained through the following equation: $[exp(4.9475 * 10^{-5} * Age) - 1] * 207.3/232$.

Usage

popline(x, main2)

Methods

- signature(object = "oneAgeTest"): an object of type oneAgeTest as produced by the function tests().
- main2: a character string to entitle the plot
- signature(object = "ageTests"): an object of type ageTests as produced by the function tests().
- main2: a character string to entitle the plot

Author(s)

Aurélie Mercadié, <aurelie.mercadie@inrae.fr>

Jean-Marc Montel, <jean-marc.montel@ensg.inpl-nancy.fr>

Nathalie Vialaneix, <nathalie.vialaneix@inrae.fr>

References

Montel J.M., Foret S., Veschambre M., Nicollet C., Provost A. (1996) Electron microprobe dating of monazite. *Chemical Geology*, **131**, 37–53.

See Also

`tests()`, `oneAgeTest`, `ageTests`

Examples

```
data(srilanka)
calculated.ages <- calculateAges(srilanka, nloops = 10)
res.tests <- tests(calculated.ages, 1, 3)
popline(res.tests, main2 = "Populations")
```

`srilanka` *An example data set: electron microprobe data.*

Description

This dataset is kindly provided by Anne-Magali Seydoux-Guillaume <anne.magali.seydoux@univ-st-etienne.fr> and has been published in Seydoux-Guillaume *et al.* (2012).

Usage

```
data(srilanka)
```

Format

A data frame with 32 observations on the following 6 variables:

- **U**: U concentration (ppm)
- **errU**: error on the measurement of U concentration
- **Th**: Th concentration (ppm)
- **ErrTh**: error on the measurement of Th concentration
- **Pb**: Pb concentration (ppm)
- **ErrPb**: error on the measurement of Pb concentration

Details

The first 8 observations are group control data (more precisely, they correspond to standard reference analyses). When testing if all the observations are issued from a single population, the assumption is thus rejected. Removing the first 8 observations leads to obtain a positive answer when testing if the observations come from the same population. See `demo(srilanka)`.

References

Seydoux-Guillaume A.M., Montel J.M., Bingen B., Bosse V., de Parseval P., Paquette J.L., Janots E., Wirth R. (2012) Low-temperature alteration of monazite: fluid mediated coupled dissolution-precipitation, irradiation damage and disturbance of the U-Pb and Th-Pb chronometers. *Chemical Geology*, **330–331**, 140–158.

Examples

```
data(srilanka)

# With control group data
summary(srilanka)

# Without control group data
summary(srilanka[9:32,])
```

`tests-methods` *Method* `"tests"` *for Class* `"ages"`

Description

Test if a set of ages belongs to a given number of populations or alternatively find out the most probable number of populations within a range.

Usage

```
## S4 method for signature 'ages'
tests(object, nbmin = 1, nbmax = NULL, level = object@level, verbose = TRUE)
```

Arguments

`object`	an object of type `ages` as produced by the function `calculateAges()`
`nbmin`	a numeric. Minimum tested number of populations.
`nbmax`	a numeric. Maximum tested number of populations.
`level`	a numeric. Confidence interval level. Default is equal to `object@level`, i.e., to the level of confidence used to calculate the ages.
`verbose`	logical. Activates the verbose mode.

Details

The function successively tests all values from `nbmin` to `nbmax` and keeps the smallest accepted one (returns an error if all supplied values are rejected). In this case, the output value is an object of class `ageTests`.

If `nbmax` is `NULL`, the results are kept even if the test is rejected. In this case, the output value is an object of class `oneAgeTest`.

The performed tests are those described in Montel *et al.* (1996).

Value

An object of class ageTests or oneAgeTest depending on the value of nbmax (see section "Details").

Usage

```
tests(object, nbmin=1, nbmax=NULL, level=object@level, verbose=TRUE)
```

Methods

- signature(object = "ages"): an object of type ages as produced by the function calculateAges().
- ...: other arguments passed to the method's function (see section Usage)

Author(s)

Jean-Marc Montel, <jean-marc.montel@ensg.inpl-nancy.fr>

Nathalie Vialaneix, <nathalie.vialaneix@inrae.fr>

References

Montel J.M., Foret S., Veschambre M., Nicollet C., Provost A. (1996) Electron microprobe dating of monazite. *Chemical Geology*, **131**, 37–53.

See Also

calculateAges(), oneAgeTest, ageTests

Examples

```
data(srilanka)
calculated.ages <- calculateAges(srilanka, nloops = 10)
tests(calculated.ages)
tests(calculated.ages, 1, 3)
```

Index