Package 'IsoplotR'

October 20, 2017

Title Statistical Toolbox for Radiometric Geochronology

Version 1.0

Description Plots U-Pb data on Wetherill and Tera-Wasserburg concordia diagrams. Calculates concordia and discordia ages. Performs linear regression of measurements with correlated errors using 'York', 'Titterington' and 'Ludwig' approaches. Generates Kernel Density Estimates (KDEs) and Cumulative Age Distributions (CADs). Produces Multidimensional Scaling (MDS) configurations and Shepard plots of multi-sample detrital datasets using the Kolmogorov-Smirnov distance as a dissimilarity measure. Calculates 40Ar/39Ar ages, isochrons, and age spectra. Computes weighted means accounting for overdispersion. Calculates U-Th-He (single grain and central) ages, logratio plots and ternary diagrams. Processes fission track data using the external detector method and LA-ICP-MS, calculates central ages and plots fission track and other data on radial (a.k.a. 'Galbraith' plots). Constructs total Pb-U, Pb-Pb, Re-Os, Sm-Nd, Lu-Hf, Rb-Sr and 230Th-U isochrons as well as 230Th-U evolution plots.

Author Pieter Vermeesch [aut, cre]

Maintainer Pieter Vermeesch <p.vermeesch@ucl.ac.uk>

Depends R (>= 3.0.0)

Imports MASS, grDevices, graphics, stats, utils

License GPL-3

URL http://isoplotr.london-geochron.com

LazyData true

RoxygenNote 6.0.1

R topics documented:

`age` *Calculate isotopic ages*

Description

Calculates ages and propagates their analytical uncertainties.

Usage

```
age(x, ...)

## Default S3 method:
age(x, method = "U238-Pb206", exterr = TRUE, J = c(NA,
  NA), zeta = c(NA, NA), rhoD = c(NA, NA), ...)

## S3 method for class 'UPb'
age(x, type = 1, wetherill = TRUE, exterr = TRUE, i = NA,
  sigdig = NA, common.Pb = 0, ...)

## S3 method for class 'PbPb'
age(x, isochron = TRUE, i2i = TRUE, exterr = TRUE,
  i = NA, sigdig = NA, ...)

## S3 method for class 'ArAr'
age(x, isochron = FALSE, i2i = TRUE, exterr = TRUE,
  i = NA, sigdig = NA, ...)

## S3 method for class 'UThHe'
age(x, central = FALSE, i = NA, sigdig = NA, ...)
```

```
## S3 method for class 'fissiontracks'
age(x, central = FALSE, i = NA, sigdig = NA,
  exterr = TRUE, ...)

## S3 method for class 'ThU'
age(x, isochron = FALSE, i2i = TRUE, exterr = TRUE,
  i = NA, sigdig = NA, ...)

## S3 method for class 'ReOs'
age(x, isochron = TRUE, i2i = TRUE, exterr = TRUE,
  i = NA, sigdig = NA, ...)

## S3 method for class 'SmNd'
age(x, isochron = TRUE, i2i = TRUE, exterr = TRUE,
  i = NA, sigdig = NA, ...)

## S3 method for class 'RbSr'
age(x, isochron = TRUE, i2i = TRUE, exterr = TRUE,
  i = NA, sigdig = NA, ...)

## S3 method for class 'LuHf'
age(x, isochron = TRUE, i2i = TRUE, exterr = TRUE,
  i = NA, sigdig = NA, ...)
```

Arguments

`x` can be:

- a scalar containing an isotopic ratio,
- a two element vector containing an isotopic ratio and its standard error, or the spontaneous and induced track densities `Ns` and `Ni` (if `method='fissiontracks'`),
- a four element vector containing `Ar40Ar39`, `s[Ar40Ar39]`, `J`, `s[J]`,
- a six element vector containing `U`, `s[U]`, `Th`, `s[Th]`, `He` and `s[He]`,
- an eight element vector containing `U`, `s[U]`, `Th`, `s[Th]`, `He`, `s[He]`, `Sm` and `s[Sm]`
- a six element vector containing `Rb`, `s[Rb]`, `Sr`, `s[Sr]`, `Sr87Sr86`, and `s[Sr87Sr86]`
- a six element vector containing `Re`, `s[Re]`, `Os`, `s[Os]`, `Os187Os188`, and `s[Os187Os188]`
- a six element vector containing `Sm`, `s[Sm]`, `Nd`, `s[Nd]`, `Nd143Nd144`, and `s[Nd144Nd143]`
- a six element vector containing `Lu`, `s[Lu]`, `Hf`, `s[Hf]`, `Hf176Hf177`, and `s[Hf176Hf177]`
- a five element vector containing `0/8`, `s[0/8]`, `4/8`, `s[4/8]` and `cov[0/8,4/8]`

OR

- an object of class `UPb`, `PbPb`, `ArAr`, `ThU`, `RbSr`, `SmNd`, `ReOs`, `LuHf`, `UThHe` or `fissiontracks`.

`...` additional arguments

`method`	one of either `'U238-Pb206'`, `'U235-Pb207'`, `'Pb207-Pb206'`, `'Ar-Ar'`, `'Th-U'`, `'Re-Os'`, `'Sm-Nd'`, `'Rb-Sr'`, `'Lu-Hf'`, `'U-Th-He'` or `'fissiontracks'`
`exterr`	propagate the external (decay constant and calibration factor) uncertainties?
`J`	two-element vector with the J-factor and its standard error.
`zeta`	two-element vector with the zeta-factor and its standard error.
`rhoD`	two-element vector with the track density of the dosimeter glass and its standard error.
`type`	scalar flag indicating whether `1`: each U-Pb analysis should be considered separately, `2`: all the measurements should be combined to calculate a concordia age, `3`: a discordia line should be fit through all the U-Pb analyses using the maximum likelihood algorithm of Ludwig (1998), which assumes that the scatter of the data is solely due to the analytical uncertainties. `4`: a discordia line should be fitignoring the analytical uncertainties. `5`: a discordia line should be fit using a modified maximum likelihood algorithm that includes accounts for any overdispersion by adding a geological (co)variance term.
`wetherill`	logical flag to indicate whether the data should be evaluated in Wetherill (`TRUE`) or Tera-Wasserburg (`FALSE`) space. This option is only used when `type=2`
`i`	(optional) index of a particular aliquot
`sigdig`	number of significant digits for the uncertainty estimate (only used if `type=1`, `isochron=FALSE` or `central=FALSE`).
`common.Pb`	apply a common lead correction using one of three methods: `1`: use the isochron intercept as the initial Pb-composition `2`: use the Stacey-Kramer two-stage model to infer the initial Pb-composition `3`: use the Pb-composition stored in `settings('iratio','Pb206Pb204')` and `settings('iratio','Pb207Pb204')`
`isochron`	logical flag indicating whether each Ar-Ar analysis should be considered separately (`isochron=FALSE`) or an isochron age should be calculated from all Ar-Ar analyses together (`isochron=TRUE`).
`i2i`	'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') ^{40}Ar/^{36}Ar, ^{207}Pb/^{204}Pb, ^{87}Sr/^{86}Sr, ^{143}Nd/^{144}Nd, ^{187}Os/^{188}Os or ^{176}Hf/^{177}Hf ratio from an isochron fit. Setting `i2i` to `FALSE` uses the default values stored in `settings('iratio',...)` or zero (for the Pb-Pb method). When applied to data of class `ThU`, setting `i2i` to `TRUE` applies a detrital Th-correction.
`central`	logical flag indicating whether each U-Th-He analysis should be considered separately (`central=FALSE`) or a central age should be calculated from all U-Th-He analyses together (`central=TRUE`).

Value

1. if `x` is a scalar or a vector, returns the age using the geochronometer given by `method` and its standard error.

2. if x has class UPb and type=1, returns a table with the following columns: t.75, err[t.75], t.68, err[t.68], t.76, err[t.76], t.conc, err[t.conc], containing the ^{207}Pb/^{235}U-age and standard error, the ^{206}Pb/^{238}U-age and standard error, the ^{207}Pb/^{206}Pb-age and standard error, and the single grain concordia age and standard error, respectively.
3. if x has class UPb and type=1, 2, 3 or 4, returns the output of the concordia function.
4. if x has class PbPb, ArAr, RbSr, SmNd, ReOs, LuHf and isochron=FALSE, returns a table of Pb-Pb, Ar-Ar, Rb-Sr, Sm-Nd, Re-Os or Lu-Hf ages and their standard errors.
5. if x has class ThU and isochron=FALSE, returns a 5-column table with the Th-U ages, their standard errors, the initial ^{234}U/^{238}U-ratios, their standard errors, and the correlation coefficient between the ages and the initial ratios.
6. if x has class PbPb, ArAr, RbSr, SmNd, ReOs, LuHf or ThU and isochron=TRUE, returns the output of the isochron function.
7. if x has class fissiontracks and central=FALSE, returns a table of fission track ages and standard errors.
8. if x has class fissiontracks or UThHe and central=TRUE, returns the output of the central function.

See Also

concordia, isochron, central

Examples

```
data(examples)
print(age(examples$UPb))
print(age(examples$UPb,type=1))
print(age(examples$UPb,type=2))
```

---

agespectrum *Plot a (40Ar/39Ar) release spectrum*

Description

Produces a plot of boxes whose widths correspond to the cumulative amount of ^{39}Ar (or any other volume proxy), and whose heights express the analytical uncertainties. Only propagates the analytical uncertainty associated with decay constants and J-factors after computing the plateau composition. IsoplotR defines the 'plateau age' as the weighted mean age of the longest sequence (in terms of cumulative ^{39}Ar content) of consecutive heating steps that pass the modified Chauvenet criterion (see weightedmean. Note that this definition is different (and simpler) than the one used by Isoplot (Ludwig, 2003). However, it is important to mention that all definitions of an age plateau are heuristic by nature and should not be used for quantitative inference.

Usage

```
agespectrum(x, ...)

## Default S3 method:
agespectrum(x, alpha = 0.05, plateau = TRUE,
  plateau.col = rgb(0, 1, 0, 0.5), non.plateau.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, line.col = "red", lwd = 2, title = TRUE,
  xlab = "cumulative fraction", ylab = "age [Ma]", ...)

## S3 method for class 'ArAr'
agespectrum(x, alpha = 0.05, plateau = TRUE,
  plateau.col = rgb(0, 1, 0, 0.5), non.plateau.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, exterr = TRUE, line.col = "red", lwd = 2, i2i = FALSE,
  ...)
```

Arguments

`x`	a three-column matrix whose first column gives the amount of ^{39}Ar in each aliquot, and whose second and third columns give the age and its uncertainty. OR an object of class `ArAr`
`...`	optional parameters to the generic `plot` function
`alpha`	the confidence limits of the error bars/boxes and confidence intervals.
`plateau`	logical flag indicating whether a plateau age should be calculated. If `plateau=TRUE`, the function will compute the weighted mean of the largest succession of steps that yield values passing the Chi-square test for age homogeneity. If `TRUE`, returns a list with plateau parameters.
`plateau.col`	the fill colour of the rectangles used to mark the steps belonging to the age plateau.
`non.plateau.col`	if `plateau=TRUE`, the steps that do NOT belong to the plateau are given a different colour.
`sigdig`	the number of significant digits of the numerical values reported in the title of the graphical output (only used if `plateau=TRUE`).
`line.col`	colour of the isochron line
`lwd`	line width
`title`	add a title to the plot?
`xlab`	x-axis label
`ylab`	y-axis label
`exterr`	propagate the external (decay constant and calibration factor) uncertainties?
`i2i`	'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') ^{40}Ar/^{36}Ar ratio from an isochron fit. Setting `i2i` to `FALSE` uses the default values stored in `settings('iratio',...)`

Value

if `plateau=TRUE`, returns a list with the following items:

mean a 3-element vector with:

`x`: the plateau mean

`s[x]`: the estimated standard deviation of `x`

`ci[x]`: the 100(1-α)% confidence interval of `t` for the appropriate degrees of freedom

disp a 2-element vector with:

`s`: the standard deviation of the overdispersion

`ci`: the 100(1-α)% confidence interval of the overdispersion for the appropriate degrees of freedom

df the degrees of freedom for the weighted mean plateau fit

mswd the mean square of the weighted deviates of the plateau

p.value the p-value of a Chi-square test with $df = n-2$ degrees of freedom, where n is the number of steps in the plateau and 2 degrees of freedom have been removed to estimate the mean and the dispersion.

fract the fraction of ^{39}Ar contained in the plateau

tfact the t-factor for `df` degrees of freedom evaluated at $100(1-\alpha/2)\%$ confidence

plotpar plot parameters for the weighted mean (see `weightedmean`), which are not used in the age spectrum

i indices of the steps that are retained for the plateau age calculation

References

Ludwig, K. R. User's manual for Isoplot 3.00: a geochronological toolkit for Microsoft Excel. Berkeley Geochronology Center Special Pulication, 2003.

See Also

`weightedmean`

Examples

```
data(examples)
agespectrum(examples$ArAr,ylim=c(0,80))
```

cad *Plot continuous data as cumulative age distributions*

Description

Empirical cumulative distribution functions or cumulative age distributions CADs (Vermeesch, 2007) are the most straightforward way to visualise the probability distribution of multiple dates. Suppose that we have a set of n dates t_i. The the CAD is a step function that sets out the rank order of the dates against their numerical value:

$CAD(t) = \sum_i 1(t < t_i)/n$

where $1(*)$ = 1 if $*$ is true and $1(*)$ = 0 if $*$ is false. CADs have two desirable properties. First, they do not require any pre-treatment or smoothing of the data. This is not the case for histograms or kernel density estimates. Second, it is easy to superimpose several CADs on the same plot. This facilitates the intercomparison of multiple samples. The interpretation of CADs is straightforward but not very intuitive. The prominence of individual age components is proportional to the steepness of the CAD. This is different from probability density estimates such as histograms, in which such components stand out as peaks. It is arguably easier to identify peaks than inflection points and this probably why CADs are not as widely used as probability density estimates. But the ease of interpretation of density estimates comes at a cost, as they require smoothing and cannot as easily be combined as CADs.

Usage

```
cad(x, ...)

## Default S3 method:
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  colmap = "heat.colors", col = "black", ...)

## S3 method for class 'detritals'
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  colmap = "heat.colors", ...)

## S3 method for class 'UPb'
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  col = "black", type = 4, cutoff.76 = 1100, cutoff.disc = c(-15, 5),
  common.Pb = 0, ...)

## S3 method for class 'PbPb'
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  col = "black", i2i = FALSE, ...)

## S3 method for class 'ArAr'
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  col = "black", i2i = FALSE, ...)
```

```
## S3 method for class 'ThU'
cad(x, pch = NA, verticals = TRUE, xlab = "age [ka]",
  col = "black", i2i = FALSE, ...)

## S3 method for class 'ReOs'
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  col = "black", i2i = TRUE, ...)

## S3 method for class 'SmNd'
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  col = "black", i2i = TRUE, ...)

## S3 method for class 'RbSr'
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  col = "black", i2i = TRUE, ...)

## S3 method for class 'LuHf'
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  col = "black", i2i = TRUE, ...)

## S3 method for class 'UThHe'
cad(x, pch = NA, verticals = TRUE, xlab = "age [Ma]",
  col = "black", ...)

## S3 method for class 'fissiontracks'
cad(x, pch = NA, verticals = TRUE,
  xlab = "age [Ma]", col = "black", ...)
```

Arguments

`x`	a numerical vector OR an object of class `UPb`, `PbPb`, `ArAr`, `UThHe`, `fissiontracks`, `ReOs`, `RbSr`, `SmNd`, `LuHf`, `ThU` or `detritals`
`...`	optional arguments to the generic `plot` function
`pch`	plot character to mark the beginning of each CAD step
`verticals`	logical flag indicating if the horizontal lines of the CAD should be connected by vertical lines
`xlab`	x-axis label
`colmap`	an optional string with the name of one of `R`'s built-in colour palettes (e.g., `heat.colors`, `terrain.colors`, `topo.colors`, `cm.colors`), which are to be used for plotting data of class `detritals`.
`col`	colour to give to single sample datasets (not applicable if `x` has class `detritals`)
`type`	scalar indicating whether to plot the ^{207}Pb/^{235}U age (`type`=1), the ^{206}Pb/^{238}U age (`type`=2), the ^{207}Pb/^{206}Pb age (`type`=3), the ^{207}Pb/^{206}Pb-^{206}Pb/^{238}U age (`type`=4), or the (Wetherill) concordia age (`type`=5)
`cutoff.76`	the age (in Ma) below which the ^{206}Pb/^{238}U-age and above which the ^{207}Pb/^{206}Pb-age is used. This parameter is only used if `type=4`.

cutoff.disc
: two element vector with the maximum and minimum percentage discordance allowed between the $^{207}Pb/^{235}U$ and $^{206}Pb/^{238}U$ age (if $^{206}Pb/^{238}U$ < cutoff.76) or between the $^{206}Pb/^{238}U$ and $^{207}Pb/^{206}Pb$ age (if $^{206}Pb/^{238}U$ > cutoff.76). Set `cutoff.disc=NA` if you do not want to use this filter.

common.Pb
: apply a common lead correction using one of three methods:

  `1`: use the isochron intercept as the initial Pb-composition

  `2`: use the Stacey-Kramer two-stage model to infer the initial Pb-composition

  `3`: use the Pb-composition stored in `settings('iratio','Pb206Pb204')` and `settings('iratio','Pb207Pb204')`

i2i
: 'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') $^{40}Ar/^{36}Ar$, $^{207}Pb/^{204}Pb$, $^{87}Sr/^{86}Sr$, $^{143}Nd/^{144}Nd$, $^{187}Os/^{188}Os$ or $^{176}Hf/^{177}Hf$ ratio from an isochron fit. Setting `i2i` to `FALSE` uses the default values stored in `settings('iratio',...)` or zero (for the Pb-Pb method). When applied to data of class `ThU`, setting `i2i` to `TRUE` applies a detrital Th-correction.

References

Vermeesch, P., 2007. Quantitative geomorphology of the White Mountains (California) using detrital apatite fission track thermochronology. Journal of Geophysical Research: Earth Surface, 112(F3).

See Also

kde, radialplot

Examples

```
data(examples)
cad(examples$DZ,verticals=FALSE,pch=20)
```

---

central — *Calculate U-Th-He and fission track central ages and compositions*

Description

Computes the geometric mean composition of a set of fission track or U-Th-He data or any other kind of heteroscedastic data, and returns the corresponding age and fitting parameters. The central age assumes that the observed age distribution is the combination of two sources of scatter: analytical uncertainty and true geological dispersion.

1. For fission track data, the analytical uncertainty is assumed to obey Poisson counting statistics and the geological dispersion is assumed to follow a lognormal distribution.
2. For U-Th-He data, the U-Th-(Sm)-He compositions are assumed to follow a logistic normal normal distribution with lognormal measurement uncertainties.
3. For all other data types, both the analytical uncertainties and the true ages are assumed to follow lognormal distributions.

The difference between the central age and the weighted mean age is usually small unless the data are imprecise and/or strongly overdispersed.

Usage

```
central(x, ...)

## Default S3 method:
central(x, alpha = 0.05, ...)

## S3 method for class 'UThHe'
central(x, alpha = 0.05, model = 1, ...)

## S3 method for class 'fissiontracks'
central(x, mineral = NA, alpha = 0.05, ...)
```

Arguments

`x`	an object of class `UThHe` or `fissiontracks`, OR a 2-column matrix with (strictly positive) values and uncertainties
`...`	optional arguments
`alpha`	cutoff value for confidence intervals
`model`	choose one of the following statistical models: `1`: weighted mean. This model assumes that the scatter between the data points is solely caused by the analytical uncertainty. If the assumption is correct, then the MSWD value should be approximately equal to one. There are three strategies to deal with the case where MSWD>1. The first of these is to assume that the analytical uncertainties have been underestimated by a factor $\sqrt{MSWD}$. Alternative approaches are described below. `2`: unweighted mean. A second way to deal with over- or underdispersed datasets is to simply ignore the analytical uncertainties. `3`: weighted mean with overdispersion: instead of attributing any overdispersion (MSWD > 1) to underestimated analytical uncertainties (model 1), one could also attribute it to the presence of geological uncertainty, which manifests itself as an added (co)variance term.
`mineral`	setting this parameter to either `apatite` or `zircon` changes the default efficiency factor, initial fission track length and density to preset values (only affects results if `x$format=2`.)

Value

if `x` has class `UThHe`, a list containing the following items:

uvw (if the input data table contains Sm) or **uv** (if it doesn't): the geometric mean log[U/He], log[Th/He] (, and log[Sm/He]) composition.

covmat the covariance matrix of `uvw` or `uv`.

mswd the reduced Chi-square statistic of data concordance, i.e. $mswd = SS/df$, where SS is the sum of squares of the log[U/He]-log[Th/He] compositions.

model the fitting model.

df the degrees of freedom $(2n - 2)$ of the fit (only reported if `model=1`).

p.value the p-value of a Chi-square test with `df` degrees of freedom (only reported if `model=1`.

tfact the $100(1-\alpha/2)\%$ percentile of the t- distribution for `df` degrees of freedom (not reported if `model=2`.

age a three- or four-element vector with:

`t`: the central age.

`s[t]`: the standard error of `s[t]`.

`ci[t]`: the $100(1-\alpha/2)\%$ confidence interval for `t` for the appropriate number of degrees of freedom.

`disp[t]`: the $100(1-\alpha/2)\%$ confidence interval enhanced by a factor of $\sqrt{mswd}$ (only reported if `model=1`).

`w`: the geological overdispersion term. If `model=3`, this is a two-element vector with the standard deviation of the (assumedly) Normal dispersion and the corresponding $100(1-\alpha/2)\%$ confidence interval. If codemodel<3 `w=0`.

OR, otherwise:

age a three-element vector with:

`t`: the central age

`s[t]`: the standard error of `s[t]`

`ci[t]`: the $100(1-\alpha/2)\%$ confidence interval for `t` for the appropriate number of degrees of freedom

disp a two-element vector with the overdispersion (standard deviation) of the excess scatter, and the corresponding $100(1-\alpha/2)\%$ confidence interval for the appropriate degrees of freedom.

mswd the reduced Chi-square statistic of data concordance, i.e. $mswd = X^2/df$, where X^2 is a Chi-square statistic of the EDM data or ages

df the degrees of freedom $(n-2)$

p.value the p-value of a Chi-square test with `df` degrees of freedom

References

Galbraith, R.F. and Laslett, G.M., 1993. Statistical models for mixed fission track ages. Nuclear tracks and radiation measurements, 21(4), pp.459-470.

Vermeesch, P., 2008. Three new ways to calculate average (U-Th)/He ages. Chemical Geology, 249(3), pp.339-347.

See Also

`weightedmean`, `radialplot`, `helioplot`

Examples

```
data(examples)
print(central(examples$UThHe)$age)
```

concordia *Concordia diagram*

Description

The concordia diagram is a graphical means of assessing the internal consistency of U-Pb data. It sets out the measured ^{206}Pb/^{238}U- and ^{207}Pb/^{235}U-ratios against each other ('Wetherill' diagram) or, equivalently, the ^{207}Pb/^{206}Pb- and ^{206}Pb/^{238}U-ratios ('Tera-Wasserburg' diagram). The space of concordant isotopic compositions is marked by a curve, the 'concordia line'. Isotopic ratio measurements are shown as 100(1-alpha)% confidence ellipses. Concordant samples plot near to, or overlap with, the concordia line. They represent the pinnacle of geochronological robustness. Samples that plot away from the concordia line but are aligned along a linear trend form an isochron (or 'discordia' line) that can be used to infer the composition of the non-radiogenic ('common') lead or to constrain the timing of prior lead loss.

The `concordia` function plots U-Pb data on Wetherill and Tera-Wasserburg concordia diagrams, calculates concordia ages and compositions, evaluates the equivalence of multiple U-Pb compositions, computes the weighted mean isotopic composition and the corresponding concordia age using the method of maximum likelihood, computes the MSWD of equivalence and concordance and their respective Chi-squared p-values. Performs linear regression and computes the upper and lower intercept ages (for Wetherill) or the lower intercept age and the ^{207}Pb/^{206}Pb intercept (for Tera-Wasserburg), taking into account error correlations and decay constant uncertainties.

Usage

```
concordia(x, tlim = NULL, alpha = 0.05, wetherill = TRUE,
  show.numbers = FALSE, levels = NA, ellipse.col = c("#00FF0080",
  "#FF000080"), concordia.col = "darksalmon", exterr = FALSE,
  show.age = 0, sigdig = 2, common.Pb = 0, ticks = NULL, ...)
```

Arguments

`x`	an object of class `UPb`
`tlim`	age limits of the concordia line
`alpha`	probability cutoff for the error ellipses and confidence intervals
`wetherill`	logical flag (`FALSE` for Tera-Wasserburg)
`show.numbers`	logical flag (`TRUE` to show grain numbers)
`levels`	a vector with additional values to be displayed as different background colours within the error ellipses.
`ellipse.col`	a vector of two background colours for the error ellipses. If `levels=NA`, then only the first colour will be used. If `levels` is a vector of numbers, then `ellipse.col` is used to construct a colour ramp.
`concordia.col`	colour of the concordia line
`exterr`	show decay constant uncertainty?

show.age one of either:

0: plot the data without calculating an age

1: fit a concordia composition and age

2: fit a discordia line through the data using the maximum likelihood algorithm of Ludwig (1998), which assumes that the scatter of the data is solely due to the analytical uncertainties. In this case, IsoplotR will either calculate an upper and lower intercept age (for Wetherill concordia), or a lower intercept age and common (^{207}Pb/^{206}Pb)-ratio intercept (for Tera-Wasserburg). If mswd>0, then the analytical uncertainties are augmented by a factor $\sqrt{mswd}$.

3: fit a discordia line ignoring the analytical uncertainties

4: fit a discordia line using a modified maximum likelihood algorithm that includes accounts for any overdispersion by adding a geological (co)variance term.

sigdig number of significant digits for the concordia/discordia age

common.Pb apply a common lead correction using one of three methods:

1: use the Stacey-Kramer two-stage model to infer the initial Pb-composition

2: use the isochron intercept as the initial Pb-composition

3: use the Pb-composition stored in settings('iratio','Pb206Pb204') and settings('iratio','Pb207Pb204')

ticks an optional vector of age ticks to be added to the concordia line.

... optional arguments to the generic plot function

Value

if show.age=1, returns a list with the following items:

x a named vector with the (weighted mean) U-Pb composition

cov the covariance matrix of the (mean) U-Pb composition

mswd a vector with three items (equivalence, concordance and combined) containing the MSWD (Mean of the Squared Weighted Deviates, a.k.a the reduced Chi-squared statistic outside of geochronology) of isotopic equivalence, age concordance and combined goodness of fit, respectively.

p.value a vector with three items (equivalence, concordance and combined) containing the p-value of the Chi-square test for isotopic equivalence, age concordance and combined goodness of fit, respectively.

df the number of degrees of freedom used for the mswd calculation. These values are useful when expanding the analytical uncertainties when mswd>1.

age a 4-element vector with:

t: the concordia age (in Ma)

s[t]: the estimated uncertainty of t

ci[t]: the 95% confidence interval of t for the appropriate degrees of freedom

disp[t]: the 95% confidence interval for t augmented by $\sqrt{mswd}$ to account for overdispersed datasets.

if show.age=2, 3 or 4, returns a list with the following items:

model the fitting model (`model=type-2`).

x a two element vector with the upper and lower intercept ages (if `wetherill=TRUE`) or the lower intercept age and ^{207}Pb/^{206}Pb intercept (for Tera-Wasserburg).

cov the covariance matrix of the elements in `x`.

err a `2 x 2` or `3 x 2` matrix with the following rows:

`s`: the estimated standard deviation for `x`

`ci`: the 95% confidence interval of `x` for the appropriate degrees of freedom

`disp[t]`: the 95% confidence interval for `x` augmented by $\sqrt{mswd}$ to account for overdispersed datasets (only reported if `type=3`).

df the degrees of freedom of the concordia fit (concordance + equivalence)

p.value p-value of a Chi-square test for age homogeneity (only reported if `type=3`).

mswd mean square of the weighted deviates – a goodness-of-fit measure. `mswd > 1` indicates overdispersion w.r.t the analytical uncertainties (not reported if `type=4`).

w two-element vector with the standard deviation of the (assumedly) Normal overdispersion term and the corresponding $100(1-\alpha)\%$ confidence interval (only important if `type=5`).

References

Ludwig, K.R., 1998. On the treatment of concordant uranium-lead ages. Geochimica et Cosmochimica Acta, 62(4), pp.665-676.

Examples

```
data(examples)
concordia(examples$UPb)
```

`ellipse` *Get coordinates of error ellipse for plotting*

Description

Construct an error ellipse age a given confidence level from its centre and covariance matrix

Usage

```
ellipse(x, y, covmat, alpha = 0.05, n = 50)
```

Arguments

`x`	x-coordinate (scalar) for the centre of the ellipse
`y`	y-coordinate (scalar) for the centre of the ellipse
`covmat`	covariance matrix of the x-y coordinates
`alpha`	the probability cutoff for the error ellipses
`n`	the resolution of the error ellipses

Value

an [n x 2] matrix of plot coordinates

Examples

```
x = 99; y = 101;
covmat <- matrix(c(1,0.9,0.9,1),nrow=2)
ell <- ellipse(x,y,covmat)
plot(c(90,110),c(90,110),type='l')
polygon(ell,col=rgb(0,1,0,0.5))
points(x,y,pch=21,bg='black')
```

---

evolution *Th-U evolution diagram*

Description

Plots Th-U data on a ^{234}U/^{238}U-^{230}Th/^{238}U evolution diagram, a ^{234}U/^{238}U-age diagram, or (if ^{234}U/^{238}U is assumed to be in secular equilibrium), a ^{230}Th/^{232}Th-^{238}U/^{232}Th diagram, calculates isochron ages.

Usage

```
evolution(x, xlim = NA, ylim = NA, alpha = 0.05, transform = FALSE,
  detrital = FALSE, show.numbers = FALSE, levels = NA,
  ellipse.col = c("#00FF0080", "#FF000080"), line.col = "darksalmon",
  isochron = FALSE, model = 1, exterr = TRUE, sigdig = 2, ...)
```

Arguments

`x`	an object of class `ThU`
`xlim`	x-axis limits
`ylim`	y-axis limits
`alpha`	probability cutoff for the error ellipses and confidence intervals
`transform`	if `TRUE`, plots ^{234}U/^{238}U vs. Th-U age.
`detrital`	apply a detrital Th correction by projecting the compositions along an isochron?
`show.numbers`	label the error ellipses with the grain numbers?
`levels`	a vector with additional values to be displayed as different background colours within the error ellipses.
`ellipse.col`	a vector of two background colours for the error ellipses. If `levels=NA`, then only the first colour will be used. If `levels` is a vector of numbers, then `ellipse.col` is used to construct a colour ramp.
`line.col`	colour of the age grid
`isochron`	fit a 3D isochron to the data?

`model` if `isochron=TRUE`, choose one of three regression models:

`1`: maximum likelihood regression, using either the modified error weighted least squares algorithm of York et al. (2004) for 2-dimensional data, or the Maximum Likelihood formulation of Ludwig and Titterington (1994) for 3-dimensional data. These algorithms take into account the analytical uncertainties and error correlations, under the assumption that the scatter between the data points is solely caused by the analytical uncertainty. If the assumption is correct, then the MSWD value should be approximately equal to one. There are three strategies to deal with the case where MSWD>1. The first of these is to assume that the analytical uncertainties have been underestimated by a factor $\sqrt{MSWD}$. Alternative approaches are described below.

`2`: ordinary least squares regression: a second way to deal with over- or underdispersed datasets is to simply ignore the analytical uncertainties.

`3`: maximum likelihood regression with overdispersion: instead of attributing any overdispersion (MSWD > 1) to underestimated analytical uncertainties (model 1), one can also attribute it to the presence of geological uncertainty, which manifests itself as an added (co)variance term.

`exterr` propagate the decay constant uncertainty in the isochron age?

`sigdig` number of significant digits for the isochron age

`...` optional arguments to the generic `plot` function

References

Ludwig, K.R. and Titterington, D.M., 1994. Calculation of ^{230}Th/U isochrons, ages, and errors. Geochimica et Cosmochimica Acta, 58(22), pp.5031-5042.

Ludwig, K.R., 2003. Mathematical-statistical treatment of data and errors for ^{230}Th/U geochronology. Reviews in Mineralogy and Geochemistry, 52(1), pp.631-656.

Examples

```
data(examples)
evolution(examples$ThU)
```

`examples` *Example datasets for testing* `IsoplotR`

Description

U-Pb, Pb-Pb, Ar-Ar, Re-Os, Sm-Nd, Rb-Sr, Lu-Hf, U-Th-He, Th-U, fission track and detrital datasets

Details

`examples` an 18-item list containing:

`UPb`: an object of class `UPb` containing a high precision U-Pb dataset of Kamo et al. (1996) packaged with Ken Ludwig's `Isoplot` program.

`PbPb`: an object of class `PbPb` containing a Pb-Pb dataset from Connelley et al. (2017).

`DZ`: an object of class `detrital` containing a detrital zircon U-Pb dataset from Namibia (Vermeesch et al., 2015).

`ArAr`: an object of class `ArAr` containing a ^{40}Ar/^{39}Ar spectrum of Skye basalt produced by Sarah Sherlock (Open University).

`UThHe`: an object of class `UThHe` containing a U-Th-Sm-He dataset of Fish Lake apatite produced by Daniel Stockli (UT Austin).

`FT1`: an object of class `fissiontracks` containing a synthetic external detector dataset.

`FT2`: an object of class `fissiontracks` containing a synthetic LA-ICP-MS-based fission track dataset using the zeta calibration method.

`FT3`: an object of class `fissiontracks` containing a synthetic LA-ICP-MS-based fission track dataset using the absolute dating approach.

`ReOs`: an object of class `ReOs` containing a ^{187}Os/^{187}Re-dataset from Selby (2007).

`SmNd`: an object of class `SmNd` containing a ^{143}Nd/^{147}Sm-dataset from Lugmair et al. (1975).

`RbSr`: an object of class `RbSr` containing an ^{87}Rb/^{86}Sr-dataset from Compston et al. (1971).

`LuHf`: an object of class `LuHf` containing an ^{176}Lu/^{177}Hf-dataset from Barfod et al. (2002).

`ThU`: an object of class `ThU` containing a synthetic 'Osmond-type' dataset from Titterington and Ludwig (1994).

`LudwigMean`: an object of class `other` containing a collection of ^{206}Pb/^{238}U-ages and errors of the example dataset by Ludwig (2003).

`LudwigKDE`: an object of class `'other'` containing the ^{206}Pb/^{238}U-ages (but not the errors) of the example dataset by Ludwig (2003).

`LudwigSpectrum`: an object of class `'other'` containing the ^{39}Ar abundances, ^{40}Ar/^{39}Ar-ages and errors of the example dataset by Ludwig (2003).

`LudwigMixture`: an object of class `'other'` containing a dataset of dispersed zircon fission track ages of the example dataset by Ludwig (2003).

References

Barfod, G.H., Albarede, F., Knoll, A.H., Xiao, S., Telouk, P., Frei, R. and Baker, J., 2002. New Lu-Hf and Pb-Pb age constraints on the earliest animal fossils. Earth and Planetary Science Letters, 201(1), pp.203-212.

Compston, W., Berry, H., Vernon, M.J., Chappell, B.W. and Kaye, M.J., 1971. Rubidium-strontium chronology and chemistry of lunar material from the Ocean of Storms. In Lunar and Planetary Science Conference Proceedings (Vol. 2, p. 1471).

Connelly, J.N., Bollard, J. and Bizzarro, M., 2017. Pb-Pb chronometry and the early Solar System. Geochimica et Cosmochimica Acta, 201, pp.345-363.

Galbraith, R. F. and Green, P. F., 1990: Estimating the component ages in a finite mixture, Nuclear Tracks and Radiation Measurements, 17, 197-206.

Kamo, S.L., Czamanske, G.K. and Krogh, T.E., 1996. A minimum U-Pb age for Siberian flood-basalt volcanism. Geochimica et Cosmochimica Acta, 60(18), 3505-3511.

Ludwig, K. R., and D. M. Titterington., 1994. "Calculation of ^{230}Th/U isochrons, ages, and errors." Geochimica et Cosmochimica Acta 58.22, 5031-5042.

Ludwig, K. R., 2003. User's manual for Isoplot 3.00: a geochronological toolkit for Microsoft Excel. No. 4.

Lugmair, G.W., Scheinin, N.B. and Marti, K., 1975. Sm-Nd age and history of Apollo 17 basalt 75075-Evidence for early differentiation of the lunar exterior. In Lunar and Planetary Science Conference Proceedings (Vol. 6, pp. 1419-1429).

Selby, D., 2007. Direct Rhenium-Osmium age of the Oxfordian-Kimmeridgian boundary, Staffin bay, Isle of Skye, UK, and the Late Jurassic time scale. Norsk Geologisk Tidsskrift, 87(3), p.291.

Vermeesch, P. and Garzanti, E., 2015. Making geological sense of 'Big Data' in sedimentary provenance analysis. Chemical Geology, 409, pp.20-27.

Vermeesch, P., 2008. Three new ways to calculate average (U-Th)/He ages. Chemical Geology, 249(3),pp.339-347.

Examples

```
data(examples)

concordia(examples$UPb)

agespectrum(examples$ArAr)

isochron(examples$ReOs)

radialplot(examples$FT1)

helioplot(examples$UThHe)

evolution(examples$ThU)

kde(examples$DZ)

radialplot(examples$LudwigMixture)

agespectrum(examples$LudwigSpectrum)

weightedmean(examples$LudwigMean)
```

helioplot *Visualise U-Th-He data on a logratio plot or ternary diagram*

Description

Plot U-Th(-Sm)-He data on a (log[He/Th] vs. log[U/He]) logratio plot or U-Th-He ternary diagram

Usage

```
helioplot(x, logratio = TRUE, model = 1, show.central.comp = TRUE,
  show.numbers = FALSE, alpha = 0.05, contour.col = c("white", "red"),
  levels = NA, ellipse.col = c("#00FF0080", "#0000FF80"), sigdig = 2,
  xlim = NA, ylim = NA, fact = NA, ...)
```

Arguments

`x` an object of class `UThHe`

`logratio` Boolean flag indicating whether the data should be shown on bivariate log[He/Th] vs. log[U/He] diagramme, or a U-Th-He ternary diagramme.

`model` choose one of the following statistical models:

`1`: weighted mean. This model assumes that the scatter between the data points is solely caused by the analytical uncertainty. If the assumption is correct, then the MSWD value should be approximately equal to one. There are three strategies to deal with the case where MSWD>1. The first of these is to assume that the analytical uncertainties have been underestimated by a factor $\sqrt{MSWD}$. Alternative approaches are described below.

`2`: unweighted mean. A second way to deal with over- or underdispersed datasets is to simply ignore the analytical uncertainties.

`3`: weighted mean with overdispersion: instead of attributing any overdispersion (MSWD > 1) to underestimated analytical uncertainties (model 1), one could also attribute it to the presence of geological uncertainty, which manifests itself as an added (co)variance term.

`show.central.comp` show the geometric mean composition as a white ellipse?

`show.numbers` show the grain numbers inside the error ellipses?

`alpha` probability cutoff for the error ellipses and confidence intervals

`contour.col` two-element vector with the fill colours to be assigned to the minimum and maximum age contour

`levels` a vector with additional values to be displayed as different background colours within the error ellipses.

`ellipse.col` a vector of two background colours for the error ellipses. If `levels=NA`, then only the first colour will be used. If `levels` is a vector of numbers, then `ellipse.col` is used to construct a colour ramp.

`sigdig` number of significant digits for the central age

xlim	optional limits of the x-axis (log[U/He]) of the logratio plot. If xlim=NA, the axis limits are determined automatically.
ylim	optional limits of the y-axis (log[Th/He]) of the logratio plot. If ylim=NA, the axis limits are determined automatically.
fact	three-element vector with the scaling factors of the ternary diagram if fact=NA, these will be determined automatically
...	optional arguments to the generic plot function

References

Vermeesch, P., 2010. HelioPlot, and the treatment of overdispersed (U-Th-Sm)/He data. Chemical Geology, 271(3), pp.108-111.

Examples

```
data(examples)
helioplot(examples$UThHe)
dev.new()
helioplot(examples$UThHe,logratio=FALSE)
```

---

isochron *Calculate and plot isochrons*

Description

Plots cogenetic Ar-Ar, Pb-Pb, Rb-Sr, Sm-Nd, Re-Os, Lu-Hf, U-Th-He or Th-U data as X-Y scatterplots, fits an isochron curve through them using the york function, and computes the corresponding isochron age, including decay constant uncertainties.

Usage

```
isochron(x, ...)

## Default S3 method:
isochron(x, xlim = NA, ylim = NA, alpha = 0.05,
  sigdig = 2, show.numbers = FALSE, levels = NA,
  ellipse.col = c("#00FF0080", "#FF000080"), line.col = "red", lwd = 2,
  title = TRUE, model = 1, ...)

## S3 method for class 'ArAr'
isochron(x, xlim = NA, ylim = NA, alpha = 0.05,
  sigdig = 2, show.numbers = FALSE, levels = NA,
  ellipse.col = c("#00FF0080", "#FF000080"), inverse = TRUE,
  line.col = "red", lwd = 2, plot = TRUE, exterr = TRUE, model = 1,
  ...)

## S3 method for class 'PbPb'
```

```
isochron(x, xlim = NA, ylim = NA, alpha = 0.05,
  sigdig = 2, show.numbers = FALSE, levels = NA,
  ellipse.col = c("#00FF0080", "#FF000080"), inverse = TRUE,
  line.col = "red", lwd = 2, plot = TRUE, exterr = TRUE, model = 1,
  ...)

## S3 method for class 'RbSr'
isochron(x, xlim = NA, ylim = NA, alpha = 0.05,
  sigdig = 2, show.numbers = FALSE, levels = NA,
  ellipse.col = c("#00FF0080", "#FF000080"), line.col = "red", lwd = 2,
  plot = TRUE, exterr = TRUE, model = 1, ...)

## S3 method for class 'ReOs'
isochron(x, xlim = NA, ylim = NA, alpha = 0.05,
  sigdig = 2, show.numbers = FALSE, levels = NA,
  ellipse.col = c("#00FF0080", "#FF000080"), line.col = "red", lwd = 2,
  plot = TRUE, exterr = TRUE, model = 1, ...)

## S3 method for class 'SmNd'
isochron(x, xlim = NA, ylim = NA, alpha = 0.05,
  sigdig = 2, show.numbers = FALSE, levels = NA,
  ellipse.col = c("#00FF0080", "#FF000080"), line.col = "red", lwd = 2,
  plot = TRUE, exterr = TRUE, model = 1, ...)

## S3 method for class 'LuHf'
isochron(x, xlim = NA, ylim = NA, alpha = 0.05,
  sigdig = 2, show.numbers = FALSE, levels = NA,
  ellipse.col = c("#00FF0080", "#FF000080"), line.col = "red", lwd = 2,
  plot = TRUE, exterr = TRUE, model = 1, ...)

## S3 method for class 'ThU'
isochron(x, type = 2, xlim = NA, ylim = NA, alpha = 0.05,
  sigdig = 2, show.numbers = FALSE, levels = NA,
  ellipse.col = c("#00FF0080", "#FF000080"), line.col = "red", lwd = 2,
  plot = TRUE, exterr = TRUE, model = 1, ...)

## S3 method for class 'UThHe'
isochron(x, xlim = NA, ylim = NA, alpha = 0.05,
  sigdig = 2, show.numbers = FALSE, line.col = "red", lwd = 2,
  plot = TRUE, model = 1, ...)
```

Arguments

`x` EITHER a matrix with the following five columns:

- **X** the x-variable
- **sX** the standard error of `X`
- **Y** the y-variable
- **sY** the standard error of `Y`

- **rXY** the correlation coefficient of `X` and `Y`

  OR

  an object of class `ArAr`, `PbPb`, `ReOs`, `RbSr`, `SmNd`, `LuHf`, `UThHe` or `ThU`.

`...`	optional arguments to be passed on to the generic plot function if `model=2`
`xlim`	2-element vector with the plot limits of the x-axis
`ylim`	2-element vector with the plot limits of the y-axis
`alpha`	confidence cutoff for the error ellipses and confidence intervals
`sigdig`	the number of significant digits of the numerical values reported in the title of the graphical output
`show.numbers`	logical flag (`TRUE` to show grain numbers)
`levels`	a vector with additional values to be displayed as different background colours within the error ellipses.
`ellipse.col`	a vector of two background colours for the error ellipses. If `levels=NA`, then only the first colour will be used. If `levels` is a vector of numbers, then `ellipse.col` is used to construct a colour ramp.
`line.col`	colour of the isochron line
`lwd`	line width
`title`	add a title to the plot?
`model`	construct the isochron using either: 1. Error-weighted least squares regression 2. Ordinary least squares regression 3. Error-weighted least squares with overdispersion term
`inverse`	if `TRUE` and `x` has class `ArAr`, plots ^{36}Ar/^{40}Ar vs. ^{39}Ar/^{40}Ar. if `TRUE` and `x` has class `PbPb`, plots ^{207}Pb/^{206}Pb vs. ^{204}Pb/^{206}Pb.
`plot`	if `FALSE`, suppresses the graphical output
`exterr`	propagate external sources of uncertainty (J, decay constant)?
`type`	following the classification of Ludwig and Titterington (1994), one of either: 1. 'Rosholt type-II' isochron, setting out ^{230}Th/^{232}Th vs. ^{238}U/^{232}Th 2. 'Osmond type-II' isochron, setting out ^{230}Th/^{238}U vs. ^{232}Th/^{238}U 3. 'Rosholt type-II' isochron, setting out ^{234}U/^{232}Th vs. ^{238}U/^{232}Th 4. 'Osmond type-II' isochron, setting out ^{234}U/^{238}U vs. ^{232}Th/^{238}U

Value

if `x` has class `PbPb`, `ArAr`, `RbSr`, `SmNd`, `ReOs` or `LuHf`, or `UThHe`, returns a list with the following items:

a the intercept of the straight line fit and its standard error.

b the slope of the fit and its standard error.

cov.ab the covariance of the slope and intercept

df the degrees of freedom of the linear fit ($df = n - 2$)

y0 a four-element list containing:

y: the atmospheric ^{40}Ar/^{36}Ar or initial ^{207}Pb/^{204}Pb, ^{187}Os/^{188}Os, ^{87}Sr/^{86}Sr, ^{143}Nd/^{144}Nd or ^{176}Hf/^{177}Hf ratio.

s[y]: the propagated uncertainty of y

ci[y]: the $100(1-\alpha/2)\%$ confidence interval for y given the appropriate degrees of freedom.

disp[y]: the $100(1-\alpha/2)\%$ confidence interval for y enhanced by $\sqrt{mswd}$ (only applicable if model=1).

age a four-element list containing:

t: the ^{207}Pb/^{206}Pb, ^{40}Ar/^{39}Ar, ^{187}Os/^{187}Re, ^{87}Sr/^{86}Sr, ^{143}Nd/^{144}Nd or ^{176}Hf/^{177}Hf age.

s[t]: the propagated uncertainty of t

ci[t]: the $100(1-\alpha/2)\%$ confidence interval for t given the appropriate degrees of freedom.

disp[t]: the $100(1-\alpha/2)\%$ confidence interval for t enhanced by $\sqrt{mswd}$ (only applicable if model=1).

tfact the $100(1-\alpha/2)\%$ percentile of a t-distribution with df degrees of freedom.

additionally, if model=1:

mswd the mean square of the residuals (a.k.a 'reduced Chi-square') statistic (omitted if model=2).

p.value the p-value of a Chi-square test for linearity

OR, if x has class ThU:

par if type=1 or type=3: the best fitting ^{230}Th/^{232}Th intercept, ^{230}Th/^{238}U slope, ^{234}U/^{232}Th intercept and ^{234}U/^{238}U slope, OR, if type=2 or type=4: the best fitting ^{234}U/^{238}U intercept, ^{230}Th/^{232}Th slope, ^{234}U/^{238}U intercept and ^{234}U/^{232}Th slope.

cov the covariance matrix of par.

df the degrees of freedom for the linear fit, i.e. $(3n-3)$ if x$format=1 or x$format=2, and $(2n-2)$ if x$format=3 or x$format=4

a if type=1: the ^{230}Th/^{232}Th intercept; if type=2: the ^{230}Th/^{238}U intercept; if type=3: the ^{234}Th/^{232}Th intercept; if type=4: the ^{234}Th/^{238}U intercept and its propagated uncertainty.

b if type=1: the ^{230}Th/^{238}U slope; if type=2: the ^{230}Th/^{232}Th slope; if type=3: the ^{234}U/^{238}U slope; if type=4: the ^{234}U/^{232}Th slope and its propagated uncertainty.

cov.ab the covariance between a and b.

mswd the mean square of the residuals (a.k.a 'reduced Chi-square') statistic.

p.value the p-value of a Chi-square test for linearity.

y0 a four-element vector containing:

y: the initial ^{234}U/^{238}U-ratio

s[y]: the propagated uncertainty of y

ci[y]: the $100(1-\alpha/2)\%$ confidence interval for y

disp[y]: the $100(1-\alpha/2)\%$ confidence interval for y enhanced by $\sqrt{mswd}$.

age a four-element vector containing:

t: the initial ^{234}U/^{238}U-ratio

s[t]: the propagated uncertainty of t

ci[t]: the $100(1-\alpha/2)\%$ confidence interval for t

disp[t]: the $100(1-\alpha/2)\%$ confidence interval for t enhanced by $\sqrt{mswd}$.

d a matrix with the following columns: the X-variable for the isochron plot, the analytical uncertainty of X, the Y-variable for the isochron plot, the analytical uncertainty of Y, and the correlation coefficient between X and Y.

xlab the x-label of the isochron plot

ylab the y-label of the isochron plot

References

Nicolaysen, L.O., 1961. Graphic interpretation of discordant age measurements on metamorphic rocks. Annals of the New York Academy of Sciences, 91(1), pp.198-206.

Ludwig, K.R. and Titterington, D.M., 1994. Calculation of ^{230}Th/U isochrons, ages, and errors. Geochimica et Cosmochimica Acta, 58(22), pp.5031-5042.

Examples

```
data(examples)
isochron(examples$ArAr)
```

`IsoplotR` *library(IsoplotR)*

Description

`IsoplotR` is a toolkit for isotope geochronology. Plots U-Pb data on Wetherill and Tera-Wasserburg concordia diagrams. Calculates concordia and discordia ages. Performs linear regression of measurements with correlated errors using 'York', 'Titterington' and 'Ludwig' approaches. Generates Kernel Density Estimates (KDEs) and Cumulative Age Distributions (CADs). Produces Multidimensional Scaling (MDS) configurations and Shepard plots of multi-sample detrital datasets using the Kolmogorov-Smirnov distance as a dissimilarity measure. Calculates ^{40}Ar/^{39}Ar ages, isochrons, and age spectra. Computes weighted means accounting for overdispersion. Calculates U-Th-He (single grain and central) ages, logratio plots and ternary diagrams. Processes fission track data using the external detector method and LA-ICP-MS, calculates central ages and plots fission track and other data on radial (a.k.a. 'Galbraith' plots). Constructs total Pb-U, Pb-Pb, Re-Os, Sm-Nd, Lu-Hf, Rb-Sr and ^{230}Th-U isochrons as well as ^{230}Th-U evolution plots.

Details

A list of documented functions may be viewed by typing `help(package='IsoplotR')`. Detailed instructions are provided at `http://isoplotr.london-geochron.com`. A manuscript with the theoretical background is in preparation.

Author(s)

Maintainer: Pieter Vermeesch `<p.vermeesch@ucl.ac.uk>`

See Also

Useful links:

- http://isoplotr.london-geochron.com

---

kde *Create (a) kernel density estimate(s)*

Description

Creates one or more kernel density estimates using a combination of the Botev (2010) bandwidth selector and the Abramson (1982) adaptive kernel bandwidth modifier.

Usage

```
kde(x, ...)

## Default S3 method:
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, ...)

## S3 method for class 'UPb'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, type = 4,
  cutoff.76 = 1100, cutoff.disc = c(-15, 5), common.Pb = 0, ...)

## S3 method for class 'detritals'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA,
  samebandwidth = TRUE, normalise = TRUE, ...)

## S3 method for class 'PbPb'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, i2i = FALSE,
  ...)

## S3 method for class 'ArAr'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
```

```
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, i2i = FALSE,
  ...)

## S3 method for class 'ThU'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [ka]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, i2i = FALSE,
  ...)

## S3 method for class 'ReOs'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, i2i = TRUE,
  ...)

## S3 method for class 'SmNd'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, i2i = TRUE,
  ...)

## S3 method for class 'RbSr'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, i2i = TRUE,
  ...)

## S3 method for class 'LuHf'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, i2i = TRUE,
  ...)

## S3 method for class 'UThHe'
kde(x, from = NA, to = NA, bw = NA, adaptive = TRUE,
  log = FALSE, n = 512, plot = TRUE, pch = NA, xlab = "age [Ma]",
  ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2),
  show.hist = TRUE, bty = "n", binwidth = NA, ncol = NA, ...)

## S3 method for class 'fissiontracks'
kde(x, from = NA, to = NA, bw = NA,
  adaptive = TRUE, log = FALSE, n = 512, plot = TRUE, pch = NA,
```

```
xlab = "age [Ma]", ylab = "", kde.col = rgb(1, 0, 1, 0.6),
hist.col = rgb(0, 1, 0, 0.2), show.hist = TRUE, bty = "n",
binwidth = NA, ncol = NA, ...)
```

Arguments

`x`	a vector of numbers OR an object of class `UPb`, `PbPb`, `ArAr`, `ReOs`, `SmNd`, `RbSr`, `UThHe`, `fissiontracks`, `ThU` or `detrital`
`...`	optional arguments to be passed on to `density`
`from`	minimum age of the time axis. If `NULL`, this is set automatically
`to`	maximum age of the time axis. If `NULL`, this is set automatically
`bw`	the bandwidth of the KDE. If `NULL`, `bw` will be calculated automatically using `botev()`
`adaptive`	logical flag controlling if the adaptive KDE modifier of Abramson (1982) is used
`log`	transform the ages to a log scale if `TRUE`
`n`	horizontal resolution of the density estimate
`plot`	show the KDE as a plot
`pch`	the symbol used to show the samples. May be a vector. Set `pch=NA` to turn them off.
`xlab`	the x-axis label
`ylab`	the y-axis label
`kde.col`	the fill colour of the KDE specified as a four element vector of `r, g, b, alpha` values
`hist.col`	the fill colour of the histogram specified as a four element vector of r, g, b, alpha values
`show.hist`	logical flag indicating whether a histogram should be added to the KDE
`bty`	change to `"o"`, `"l"`, `"7"`, `"c"`, `"u"`, or `"]"` if you want to draw a box around the plot
`binwidth`	scalar width of the histogram bins, in Myr if `x$log = FALSE`, or as a fractional value if `x$log = TRUE`. Sturges' Rule is used if `binwidth = NA`
`ncol`	scalar value indicating the number of columns over which the KDEs should be divided. This option is only used if `x` has class `detritals`.
`type`	scalar indicating whether to plot the ^{207}Pb/^{235}U age (`type`=1), the ^{206}Pb/^{238}U age (`type`=2), the ^{207}Pb/^{206}Pb age (`type`=3), the ^{207}Pb/^{206}Pb-^{206}Pb/^{238}U age (`type`=4), or the (Wetherill) concordia age (`type`=5)
`cutoff.76`	the age (in Ma) below which the ^{206}Pb/^{238}U and above which the ^{207}Pb/^{206}Pb age is used. This parameter is only used if `type=4`.
`cutoff.disc`	two element vector with the maximum and minimum percentage discordance allowed between the ^{207}Pb/^{235}U and ^{206}Pb/^{238}U age (if ^{206}Pb/^{238}U < `cutoff.76`) or between the ^{206}Pb/^{238}U and ^{207}Pb/^{206}Pb age (if ^{206}Pb/^{238}U > `cutoff.76`). Set `cutoff.disc=NA` if you do not want to use this filter.

`common.Pb` apply a common lead correction using one of three methods:

1: use the isochron intercept as the initial Pb-composition

2: use the Stacey-Kramer two-stage model to infer the initial Pb-composition

3: use the Pb-composition stored in `settings('iratio','Pb206Pb204')` and `settings('iratio','Pb207Pb204')`

`samebandwidth` logical flag indicating whether the same bandwidth should be used for all samples. If `samebandwidth = TRUE` and `bw = NULL`, then the function will use the median bandwidth of all the samples.

`normalise` logical flag indicating whether or not the KDEs should all integrate to the same value.

`i2i` 'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') ^{40}Ar/^{36}Ar, ^{207}Pb/^{204}Pb, ^{87}Sr/^{86}Sr, ^{143}Nd/^{144}Nd, ^{187}Os/^{188}Os or ^{176}Hf/^{177}Hf ratio from an isochron fit. Setting `i2i` to `FALSE` uses the default values stored in `settings('iratio',...)` or zero (for the Pb-Pb method). When applied to data of class `ThU`, setting `i2i` to `TRUE` applies a detrital Th-correction.

Value

if `plot = TRUE`, returns an object of class `KDE`, i.e. a list containing the following items:

x horizontal plot coordinates

y vertical plot coordinates

bw the base bandwidth of the density estimate

ages the data values from the input to the `kde` function

or, if `x` has class `=detritals`, an object of class `KDEs`, i.e. a list containing the following items:

kdes a named list with objects of class `KDE`

from the beginning of the common time scale

to the end of the common time scale

themax the maximum probability density of all the KDEs

xlabel the x-axis label to be used by `plot.KDEs`

References

Abramson, I.S., 1982. On bandwidth variation in kernel estimates-a square root law. The annals of Statistics, pp.1217-1223.

Botev, Z. I., J. F. Grotowski, and D. P. Kroese. "Kernel density estimation via diffusion." The Annals of Statistics 38.5 (2010): 2916-2957.

Vermeesch, P., 2012. On the visualisation of detrital age distributions. Chemical Geology, 312, pp.190-194.

Examples

```
data(examples)
kde(examples$DZ[['N1']],kernel="epanechnikov")
kde(examples$DZ,from=0,to=3000)
```

`ludwig` *Linear regression of X,Y,Z-variables with correlated errors, taking into account decay constant uncertainties.*

Description

Implements the maximum likelihood algorithm of Ludwig (1998)

Usage

```
ludwig(x, ...)

## Default S3 method:
ludwig(x, ...)

## S3 method for class 'UPb'
ludwig(x, exterr = FALSE, alpha = 0.05, model = 1, ...)
```

Arguments

`x`	an object of class `UPb`
`...`	optional arguments
`exterr`	propagate external sources of uncertainty (e.g., decay constant)?
`alpha`	cutoff value for confidence intervals
`model`	one of three regression models: `1`: fit a discordia line through the data using the maximum likelihood algorithm of Ludwig (1998), which assumes that the scatter of the data is solely due to the analytical uncertainties. In this case, IsoplotR will either calculate an upper and lower intercept age (for Wetherill concordia), or a lower intercept age and common (207Pb/206Pb)o-ratio intercept (for Tera-Wasserburg). If MSWD>0, then the analytical uncertainties are augmented by a factor $\sqrt{MSWD}$. `2`: fit a discordia line ignoring the analytical uncertainties `3`: fit a discordia line using a modified maximum likelihood algorithm that includes accounts for any overdispersion by adding a geological (co)variance term.

Value

par a two-element vector with the lower concordia intercept and initial ^{207}Pb/^{206}Pb-ratio.

cov the covariance matrix of `par`

df the degrees of freedom of the model fit ($3n - 3$, where n is the number of aliquots).

mswd the mean square of weighted deviates (a.k.a. reduced Chi-square statistic) for the fit.

p.value p-value of a Chi-square test for the linear fit

References

Ludwig, K.R., 1998. On the treatment of concordant uranium-lead ages. Geochimica et Cosmochimica Acta, 62(4), pp.665-676.

Examples

```
f <- system.file("UPb4.csv",package="IsoplotR")
d <- read.data(f,method="U-Pb",format=4)
fit <- ludwig(d)
```

`mds` *Multidimensional Scaling*

Description

Performs classical or nonmetric Multidimensional Scaling analysis

Usage

```
mds(x, ...)

## Default S3 method:
mds(x, classical = FALSE, plot = TRUE, shepard = FALSE,
  nnlines = FALSE, pos = NULL, col = "black", bg = "white", xlab = "",
  ylab = "", ...)

## S3 method for class 'detritals'
mds(x, classical = FALSE, plot = TRUE,
  shepard = FALSE, nnlines = FALSE, pos = NULL, col = "black",
  bg = "white", xlab = "", ylab = "", ...)
```

Arguments

`x`	a dissimilarity matrix OR an object of class `detrital`
`...`	optional arguments to the generic `plot` function
`classical`	logical flag indicating whether classical (`TRUE`) or nonmetric (`FALSE`) MDS should be used
`plot`	show the MDS configuration (if `shepard=FALSE`) or Shepard plot (if `shepard=TRUE`) on a graphical device
`shepard`	logical flag indicating whether the graphical output should show the MDS configuration (`shepard=FALSE`) or a Shepard plot with the 'stress' value. This argument is only used if `plot=TRUE`.
`nnlines`	if `TRUE`, draws nearest neighbour lines
`pos`	a position specifier for the labels (if `pch!=NA`). Values of 1, 2, 3 and 4 indicate positions below, to the left of, above and to the right of the MDS coordinates, respectively.

col	plot colour (may be a vector)
bg	background colour (may be a vector)
xlab	a string with the label of the x axis
ylab	a string with the label of the y axis

Value

returns an object of class MDS, i.e. a list containing the following items:

points a two column vector of the fitted configuration

classical a logical flag indicating whether the MDS configuration was obtained by classical (TRUE) or nonmetric (FALSE) MDS

diss the dissimilarity matrix used for the MDS analysis

stress (only if classical=TRUE) the final stress achieved (in percent)

References

Vermeesch, P., 2013. Multi-sample comparison of detrital age distributions. Chemical Geology, 341, pp.140-146.

Examples

```
data(examples)
mds(examples$DZ,nnlines=TRUE,pch=21,cex=5)
dev.new()
mds(examples$DZ,shepard=TRUE)
```

---

peakfit *Finite mixture modelling of geochronological datasets*

Description

Implements the discrete mixture modelling algorithms of Galbraith and Green (1993) and applies them to fission track and other geochronological datasets.

Usage

```
peakfit(x, ...)

## Default S3 method:
peakfit(x, k = "auto", sigdig = 2, log = TRUE,
  alpha = 0.05, ...)

## S3 method for class 'fissiontracks'
peakfit(x, k = 1, exterr = TRUE, sigdig = 2,
  log = TRUE, alpha = 0.05, ...)
```

```
## S3 method for class 'UPb'
peakfit(x, k = 1, type = 4, cutoff.76 = 1100,
  cutoff.disc = c(-15, 5), exterr = TRUE, sigdig = 2, log = TRUE,
  alpha = 0.05, ...)

## S3 method for class 'PbPb'
peakfit(x, k = 1, exterr = TRUE, sigdig = 2, log = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'ArAr'
peakfit(x, k = 1, exterr = TRUE, sigdig = 2, log = TRUE,
  i2i = FALSE, alpha = 0.05, ...)

## S3 method for class 'ReOs'
peakfit(x, k = 1, exterr = TRUE, sigdig = 2, log = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'SmNd'
peakfit(x, k = 1, exterr = TRUE, sigdig = 2, log = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'RbSr'
peakfit(x, k = 1, exterr = TRUE, sigdig = 2, log = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'LuHf'
peakfit(x, k = 1, exterr = TRUE, sigdig = 2, log = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'ThU'
peakfit(x, k = 1, exterr = FALSE, sigdig = 2, log = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'UThHe'
peakfit(x, k = 1, sigdig = 2, log = TRUE, alpha = 0.05,
  ...)
```

Arguments

x — either a `[2 x n]` matrix with measurements and their standard errors, or an object of class `fissiontracks`, `UPb`, `PbPb`, `ArAr`, `ReOs`, `SmNd`, `RbSr`, `LuHf`, `ThU` or `UThHe`

... — optional arguments (not used)

k — the number of discrete age components to be sought. Setting this parameter to `'auto'` automatically selects the optimal number of components (up to a maximum of 5) using the Bayes Information Criterion (BIC).

sigdig	number of significant digits to be used for any legend in which the peak fitting results are to be displayed.
log	take the logs of the data before applying the mixture model?
alpha	cutoff value for confidence intervals
exterr	propagate the external sources of uncertainty into the component age errors?
type	scalar indicating whether to plot the ^{207}Pb/^{235}U age (type=1), the ^{206}Pb/^{238}U age (type=2), the ^{207}Pb/^{206}Pb age (type=3), the ^{207}Pb/^{206}Pb-^{206}Pb/^{238}U age (type=4), or the (Wetherill) concordia age (type=5)
cutoff.76	the age (in Ma) below which the ^{206}Pb/^{238}U and above which the ^{207}Pb/^{206}Pb age is used. This parameter is only used if type=4.
cutoff.disc	two element vector with the maximum and minimum percentage discordance allowed between the ^{207}Pb/^{235}U and ^{206}Pb/^{238}U age (if ^{206}Pb/^{238}U < cutoff.76) or between the ^{206}Pb/^{238}U and ^{207}Pb/^{206}Pb age (if ^{206}Pb/^{238}U > cutoff.76). Set cutoff.disc=NA if you do not want to use this filter.
i2i	'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') ^{40}Ar/^{36}Ar, ^{207}Pb/^{204}Pb, ^{87}Sr/^{86}Sr, ^{143}Nd/^{144}Nd, ^{187}Os/^{188}Os or ^{176}Hf/^{177}Hf ratio from an isochron fit. Setting i2i to FALSE uses the default values stored in settings('iratio',...) or zero (for the Pb-Pb method). When applied to data of class ThU, setting i2i to TRUE applies a detrital Th-correction.

Value

returns a list with the following items:

peaks a 3 x k matrix with the following rows:

- t: the ages of the k peaks
- s[t]: the estimated uncertainties of t
- ci[t]: the $100(1-\alpha/2)\%$ confidence interval for t

props a 2 x k matrix with the following rows:

- p: the proportions of the k peaks
- s[p]: the estimated uncertainties of p

L the log-likelihood of the fit

tfact the $100(1-\alpha/2)\%$ percentile of the t-distribution with $(n-2k+1)$ degrees of freedom

legend a vector of text expressions to be used in a figure legend

References

Galbraith, R.F. and Laslett, G.M., 1993. Statistical models for mixed fission track ages. Nuclear tracks and radiation measurements, 21(4), pp.459-470.

Examples

```
data(examples)
peakfit(examples$FT1,k=2)
```

radialplot *Visualise heteroscedastic data on a radial plot*

Description

Implementation of a graphical device developed by Rex Galbraith to display several estimates of the same quantity that have different standard errors.

Usage

```
radialplot(x, ...)

## Default S3 method:
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", sigdig = 2, show.numbers = FALSE, pch = 21,
  levels = NA, bg = c("white", "red"), title = TRUE, k = 0,
  markers = NULL, alpha = 0.05, ...)

## S3 method for class 'fissiontracks'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "arcsin", sigdig = 2, show.numbers = FALSE, pch = 21,
  levels = NA, bg = c("white", "red"), title = TRUE, markers = NULL,
  k = 0, exterr = TRUE, alpha = 0.05, ...)

## S3 method for class 'UPb'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", type = 4, cutoff.76 = 1100,
  cutoff.disc = c(-15, 5), show.numbers = FALSE, pch = 21, levels = NA,
  bg = c("white", "red"), markers = NULL, k = 0, exterr = TRUE,
  common.Pb = 0, alpha = 0.05, ...)

## S3 method for class 'PbPb'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", show.numbers = FALSE, pch = 21, levels = NA,
  bg = c("white", "red"), markers = NULL, k = 0, exterr = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'ArAr'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", show.numbers = FALSE, pch = 21, levels = NA,
  bg = c("white", "red"), markers = NULL, k = 0, exterr = TRUE,
  i2i = FALSE, alpha = 0.05, ...)

## S3 method for class 'UThHe'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", show.numbers = FALSE, pch = 21, levels = NA,
  bg = c("white", "red"), markers = NULL, k = 0, alpha = 0.05, ...)
```

```
## S3 method for class 'ReOs'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", show.numbers = FALSE, pch = 21, levels = NA,
  bg = c("white", "red"), markers = NULL, k = 0, exterr = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'SmNd'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", show.numbers = FALSE, pch = 21, levels = NA,
  bg = c("white", "red"), markers = NULL, k = 0, exterr = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'RbSr'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", show.numbers = FALSE, pch = 21, levels = NA,
  bg = c("white", "red"), markers = NULL, k = 0, exterr = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'LuHf'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", show.numbers = FALSE, pch = 21, levels = NA,
  bg = c("white", "red"), markers = NULL, k = 0, exterr = TRUE,
  i2i = TRUE, alpha = 0.05, ...)

## S3 method for class 'ThU'
radialplot(x, from = NA, to = NA, t0 = NA,
  transformation = "log", show.numbers = FALSE, pch = 21, levels = NA,
  bg = c("white", "red"), markers = NULL, k = 0, i2i = TRUE,
  alpha = 0.05, ...)
```

Arguments

`x`	Either an nx2 matix of (transformed) values z and their standard errors s OR and object of class `fissiontracks`, `UThHe`, `ArAr`, `ReOs`, `SmNd`, `RbSr`, `LuHf`, `ThU`, `PbPb` or `UPb`
`...`	additional arguments to the generic `points` function
`from`	minimum age limit of the radial scale
`to`	maximum age limit of the radial scale
`t0`	central value
`transformation`	one of either `log`, `linear` or (if `x` has class `fissiontracks`)
`sigdig`	the number of significant digits of the numerical values reported in the title of the graphical output.
`show.numbers`	boolean flag (`TRUE` to show grain numbers)
`pch`	plot character (default is a filled circle)

`levels` a vector with additional values to be displayed as different background colours of the plot symbols.

`bg` a vector of two background colours for the plot symbols. If `levels=NA`, then only the first colour will be used. If `levels` is a vector of numbers, then `bg` is used to construct a colour ramp.

`title` add a title to the plot?

`k` number of peaks to fit using the finite mixture models of Galbraith and Green (1993). Setting `k='auto'` automatically selects an optimal number of components based on the Bayes Information Criterion (BIC). Setting `k='min'` estimates the minimum value using a three parameter model consisting of a Normal distribution truncated by a discrete component.

`markers` vector of ages of radial marker lines to add to the plot.

`alpha` cutoff value for confidence intervals

`exterr` propagate the external sources of uncertainty into the mixture model errors?

`type` scalar indicating whether to plot the ^{207}Pb/^{235}U age (`type`=1), the ^{206}Pb/^{238}U age (`type`=2), the ^{207}Pb/^{206}Pb age (`type`=3), the ^{207}Pb/^{206}Pb-^{206}Pb/^{238}U age (`type`=4), or the (Wetherill) concordia age (`type`=5)

`cutoff.76` the age (in Ma) below which the ^{206}Pb/^{238}U and above which the ^{207}Pb/^{206}Pb age is used. This parameter is only used if `type=4`.

`cutoff.disc` two element vector with the maximum and minimum percentage discordance allowed between the ^{207}Pb/^{235}U and ^{206}Pb/^{238}U age (if ^{206}Pb/^{238}U < `cutoff.76`) or between the ^{206}Pb/^{238}U and ^{207}Pb/^{206}Pb age (if ^{206}Pb/^{238}U > `cutoff.76`). Set `cutoff.disc=NA` if you do not want to use this filter.

`common.Pb` apply a common lead correction using one of three methods:

`1`: use the isochron intercept as the initial Pb-composition

`2`: use the Stacey-Kramer two-stage model to infer the initial Pb-composition

`3`: use the Pb-composition stored in `settings('iratio','Pb206Pb204')` and `settings('iratio','Pb207Pb204')`

`i2i` 'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') ^{40}Ar/^{36}Ar, ^{207}Pb/^{204}Pb, ^{87}Sr/^{86}Sr, ^{143}Nd/^{144}Nd, ^{187}Os/^{188}Os or ^{176}Hf/^{177}Hf ratio from an isochron fit. Setting `i2i` to `FALSE` uses the default values stored in `settings('iratio',...)` or zero (for the Pb-Pb method). When applied to data of class `ThU`, setting `i2i` to `TRUE` applies a detrital Th-correction.

References

Galbraith, R.F., 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), pp.207-214.

Examples

```
data(examples)
radialplot(examples$FT1)
```

`read.data` *Read geochronology data*

Description

Cast a `.csv` file or a matrix into one of `IsoplotR`'s data classes

Usage

```
read.data(x, ...)

## Default S3 method:
read.data(x, method = "U-Pb", format = 1, ...)

## S3 method for class 'matrix'
read.data(x, method = "U-Pb", format = 1, ...)
```

Arguments

`x` either a file name (`.csv` format) OR a matrix

`...` optional arguments to the `read.csv` function

`method` one of `'U-Pb'`, `'Pb-Pb'`, `'Ar-Ar'`, `'detritals'`, `Rb-Sr`, `Sm-Nd`, `Re-Os`, `Th-U`, `'U-Th-He'`, `'fissiontracks'` or `'other'`

`format` formatting option, depends on the value of `method`.

if `method='U-Pb'`, then `format` is one of either:

1. `7/5, s[7/5], 6/8, s[6/8], rho`
2. `8/6, s[8/6], 7/6, s[7/6] (, rho)`
3. `X=7/6, s[X], Y=7/5, s[Y], Z=6/8, s[Z] (, rho[X,Y]) (, rho[Y,Z])`
4. `X=7/5, s[X], Y=6/8, s[Y], Z=4/8, rho[X,Y], rho[X,Z], rho[Y,Z]`
5. `X=8/6, s[X], Y=7/6, s[Y], Z=4/6, rho[X,Y], rho[X,Z], rho[Y,Z]`
6. `7/5, s[7/5], 6/8, s[6/8], 4/8, s[4/8], 7/6, s[7/6], 4/7, s[4/7], 4/6, s[4/6]`

where optional columns are marked in round brackets

if `method='Pb-Pb'`, then `format` is one of either:

1. `6/4, s[6/4], 7/4, s[7/4], rho`
2. `4/6, s[4/6], 7/6, s[7/6], rho`
3. `6/4, s[6/4], 7/4, s[7/4], 7/6, s[7/6]`

if `method='Ar-Ar'`, then `format` is one of either:

1. `9/6, s[9/6], 0/6, s[0/6], rho (, 39)`
2. `6/0, s[6/0], 9/0, s[9/0] (, rho) (, 39)`
3. `9/0, s[9/0], 6/0, s[6/0], 9/6, s[9/6] (, 39)`

if `method='Rb-Sr'`, then `format` is one of either:

1. `Rb87/Sr86, s[Rb87/Sr86], Sr87/Sr86, s[Sr87/Sr86] (, rho)`

2. `Rb, s[Rb], Sr, s[Sr], Sr87/Sr86, s[Sr87/Sr86]`

where `Rb` and `Sr` are in ppm

if `method='Sm-Nd'`, then `format` is one of either:

1. `Sm147/Nd144, s[Sm147/Nd144], Nd143/Nd144, s[Nd143/Nd144] (, rho)`
2. `Sm, s[Sm], Nd, s[Nd], Nd143/Nd144, s[Nd143/Nd144]`

where `Sm` and `Nd` are in ppm

if `method='Re-Os'`, then `format` is one of either:

1. `Re187/Os188, s[Re187/Os188], Os187/Os188, s[Os187/Os188] (, rho)`
2. `Re, s[Re], Os, s[Os], Os187/Os188, s[Os187/Os188]`

where `Re` and `Os` are in ppm

if `method='Lu-Hf'`, then `format` is one of either:

1. `Lu176/Hf177, s[Lu176/Hf177], Hf176/Hf177, s[Hf176/Hf177] (, rho)`
2. `Lu, s[Lu], Hf, s[Hf], Hf176/Hf177, s[Hf176/Hf177]`

where `Lu` and `Hf` are in ppm

if `method='Th-U'`, then `format` is one of either:

1. `X=8/2, s[X], Y=4/2, s[Y], Z=0/2, s[Z], rho[X,Y], rho[X,Z], rho[Y,Z]`
2. `X=2/8, s[X], Y=4/8, s[Y], Z=0/8, s[Z], rho[X,Y], rho[X,Z], rho[Y,Z]`

where all values are activity ratios

if `method='fissiontracks'`, then `format` is one of either:

1. the External Detector Method (EDM), which requires a ζ-calibration constant and its uncertainty, the induced track density in a dosimeter glass, and a table with the spontaneous and induced track densities.
2. LA-ICP-MS-based fission track data using the ζ-calibration method, which requires a 'session ζ' and its uncertainty and a table with the number of spontaneous tracks, the area over which these were counted and one or more U/Ca- or U-concentration measurements and their analytical uncertainties.
3. LA-ICP-MS-based fission track data using the 'absolute dating' method, which only requires a table with the the number of spontaneous tracks, the area over which these were counted and one or more U/Ca- or U-concentration measurements and their analytical uncertainties.

Details

IsoplotR provides the following example input files:

- U-Pb: `UPb1.csv`, `UPb2.csv`, `UPb3.csv`, `UPb4.csv`, `UPb5.csv`, `UPb6.csv`
- Pb-Pb: `PbPb1.csv`, `PbPb2.csv`, `PbPb3.csv`
- Ar-Ar: `ArAr1.csv`, `ArAr2.csv`, `ArAr3.csv`
- Re-Os: `ReOs1.csv`, `ReOs2.csv`
- Sm-Nd: `SmNd1.csv`, `SmNd2.csv`
- Rb-Sr: `RbSr1.csv`, `RbSr2.csv`
- Lu-Hf: `LuHf1.csv`, `LuHf2.csv`

- Th-U: `ThU1.csv`, `ThU2.csv`, `ThU3.csv`, `ThU4.csv`
- fissiontracks: `FT1.csv`, `FT2.csv`, `FT3.csv`
- U-Th-He: `UThHe.csv`, `UThSmHe.csv`
- detritals: `DZ.csv`
- other: `LudwigMixture.csv`, `LudwigMean.csv`, `LudwigKDE.csv` `LudwigSpectrum.csv`

The contents of these files can be viewed using the `system.file(...)` function.

Value

an object of class `UPb`, `PbPb`, `ArAr`, `UThHe`, `ReOs`, `SmNd`, `RbSr`, `LuHf`, `detritals`, `fissiontracks`, `ThU` or `other`

Examples

```
file.show(system.file("spectrum.csv",package="IsoplotR"))

f1 <- system.file("UPb1.csv",package="IsoplotR")
d1 <- read.data(f1,method="U-Pb",format=1)
concordia(d1)

f2 <- system.file("ArAr1.csv",package="IsoplotR")
d2 <- read.data(f2,method="Ar-Ar",format=1)
agespectrum(d2)

f3 <- system.file("ReOs1.csv",package="IsoplotR")
d3 <- read.data(f3,method="Re-Os",format=1)
isochron(d2)

f4 <- system.file("FT1.csv",package="IsoplotR")
d4 <- read.data(f4,method="fissiontracks",format=1)
radialplot(d4)

f5 <- system.file("UThSmHe.csv",package="IsoplotR")
d5 <- read.data(f5,method="U-Th-He")
helioplot(d5)

f6 <- system.file("ThU2.csv",package="IsoplotR")
d6 <- read.data(f6,method="Th-U",format=2)
evolution(d6)

#  one detrital zircon U-Pb file (detritals.csv)
f7 <- system.file("DZ.csv",package="IsoplotR")
d7 <- read.data(f7,method="detritals")
kde(d7)

#  four 'other' files (LudwigMixture.csv, LudwigSpectrum.csv,
#  LudwigMean.csv, LudwigKDE.csv)
f8 <- system.file("LudwigMixture.csv",package="IsoplotR")
d8 <- read.data(f8,method="other")
radialplot(d8)
```

set.zeta *Calculate the zeta calibration coefficient for fission track dating*

Description

Determines the zeta calibration constant of a fission track dataset (EDM or LA-ICP-MS) given its true age and analytical uncertainty.

Usage

```
set.zeta(x, tst = c(0, 0), exterr = TRUE, update = TRUE, sigdig = 2)
```

Arguments

x	an object of class `fissiontracks`
tst	a two-element vector with the true age and its standard error
exterr	logical flag indicating whether the external uncertainties associated with the age standard or the dosimeter glass (for the EDM) should be accounted for when propagating the uncertainty of the zeta calibration constant.
update	logical flag indicating whether the function should return an updated version of the input data, or simply return a two-element vector with the calibration constant and its standard error.
sigdig	number of significant digits

Value

an object of class `fissiontracks` with an updated `x$zeta` value

References

Vermeesch, P., 2017. Statistics for LA-ICP-MS based fission track dating. Chemical Geology, 456, pp.19-27.

Examples

```
data(examples)
print(examples$FT1$zeta)
FT <- set.zeta(examples$FT1,tst=c(250,5))
print(FT$zeta)
```

`settings` *Load settings to and from json*

Description

Get and set preferred values for decay constants, isotopic abundances, molar masses, fission track etch efficiences, and etchable lengths, and mineral densities, either individually or via a `.json` file format.

Usage

```
settings(setting = NA, ..., fname = NA)
```

Arguments

`setting` unless `fname` is provided, this should be one of either:

`'lambda'`: to get and set decay constants

`'iratio'`: isotopic ratios

`'imass'`: isotopic molar masses

`'mindens'`: mineral densities

`'etchfact'`: fission track etch efficiency factors

`'tracklength'`: equivalent isotropic fission track length

`...` depends on the value for `setting`:

- for `'lambda'`: the isotope of interest (one of either `"fission"`, `"U238"`, `"U235"`, `"U234"`, `"Th232"`, `"Th230"`, `"Re187"`, `"Sm147"`, `"Rb87"`, `"Lu176"`, or `"K40"`) PLUS (optionally) the decay constant value and its analytical error. Omitting these two numbers simply returns the existing values.
- for `'iratio'`: the isotopic ratio of interest (one of either `"Ar40Ar36"`, `"Ar38Ar36"`, `"Rb85Rb87"`, `"Sr88Sr86"`, `"Sr87Sr86"`, `"Sr84Sr86"`, `"Re185Re187"`, `"Os184Os192"` `"Os186Os192"`, `"Os187Os192"`, `"Os188Os192"`, `"Os189Os192"`, `"Os190Os192"`, `"U238U235"`, `"Sm144Sm152"`, `"Sm147Sm152"`, `"Sm148Sm152"`, `"Sm149Sm152"`, `"Sm150Sm152"`, `"Sm154Sm152"`, `"Nd142Nd144"`, `"Nd143Nd144"`, `"Nd145Nd144"`, `"Nd146Nd144"`, `"Nd148Nd144"`, `"Nd150Nd144"`, `"Lu176Lu175"`, `"Hf174Hf177"`, `"Hf176Hf177"`, `"Hf178Hf177"`, `"Hf179Hf177"`, `"Hf180Hf177"`) PLUS (optionally) the isotopic ratio and its analytical error. Omitting these two numbers simply returns the existing values.
- for `'imass'`: the (isotopic) molar mass of interest (one of either `"U"`, `"Rb"`, `"Rb85"`, `"Rb87"`, `"Sr84"`, `"Sr86"`, `"Sr87"`, `"Sr88"`, `"Re"`, `"Re185"`, `"Re187"`, `"Os"`, `"Os184"`, `"Os186"`, `"Os187"`, `"Os188"`, `"Os189"`, `"Os190"`, `"Os192"`, `"Sm"`, `"Nd"`, `"Lu"`, `"Hf"`) PLUS (optionally) the molar mass and its analytical error. Omitting these two numbers simply returns the existing values.
- for `'mindens'`: the mineral of interest (one of either `"apatite"` or `"zircon"`) PLUS the mineral density. Omitting this number simply returns the existing value.

- `'etchfact'`: the mineral of interest (one of either `"apatite"` or `"zircon"`) PLUS the etch efficiency factor. Omitting this number simply returns the existing value.
- `'tracklength'`: the mineral of interest (one of either `"apatite"` or `"zircon"`) PLUS the equivalent isotropic fission track length. Omitting this number simply returns the existing value.

`fname` the path of a `.json` file

Value

if `setting=NA` and `fname=NA`, returns a `.json` string

if `...` contains only the name of an isotope, isotopic ratio, element, or mineral and no new value, `settings` returns either a scalar with the existing value, or a two-element vector with the value and its uncertainty.

References

1. Decay constants:
   - ^{238}U, ^{235}U: Jaffey, A. H., et al. "Precision measurement of half-lives and specific activities of U^{235} and U^{238}." Physical Review C 4.5 (1971): 1889.
   - ^{232}Th: Le Roux, L. J., and L. E. Glendenin. "Half-life of ^{232}Th. "Proceedings of the National Meeting on Nuclear Energy, Pretoria, South Africa. 1963.
   - ^{234}U, ^{230}Th: Cheng, H., Edwards, R.L., Shen, C.C., Polyak, V.J., Asmerom, Y., Woodhead, J., Hellstrom, J., Wang, Y., Kong, X., Spotl, C. and Wang, X., 2013. Improvements in ^{230}Th dating, ^{230}Th and ^{234}U half-life values, and U-Th isotopic measurements by multi-collector inductively coupled plasma mass spectrometry. Earth and Planetary Science Letters, 371, pp.82-91.
   - Sm: Lugmair, G. W., and K. Marti. "Lunar initial ^{143}Nd/^{144}Nd: differential evolution of the lunar crust and mantle." Earth and Planetary Science Letters 39.3 (1978): 349-357.
   - Nd: Zhao, Motian, et al. "Absolute measurements of neodymium isotopic abundances and atomic weight by MC-ICPMS." International Journal of Mass Spectrometry 245.1 (2005): 36-40.
   - Re: Selby, D., Creaser, R.A., Stein, H.J., Markey, R.J. and Hannah, J.L., 2007. Assessment of the 187Re decay constant by cross calibration of Re-Os molybdenite and U-Pb zircon chronometers in magmatic ore systems. Geochimica et Cosmochimica Acta, 71(8), pp.1999-2013.
   - Ar: Renne, Paul R., et al. "Response to the comment by WH Schwarz et al. on "Joint determination of ^{40}K decay constants and ^{40}Ar*/^{40}K for the Fish Canyon sanidine standard, and improved accuracy for ^{40}Ar/^{39}Ar geochronology" by PR Renne et al.(2010)." Geochimica et Cosmochimica Acta 75.17 (2011): 5097-5100.
   - Rb: Villa, I.M., De Bievre, P., Holden, N.E. and Renne, P.R., 2015. "IUPAC-IUGS recommendation on the half life of ^{87}Rb". Geochimica et Cosmochimica Acta, 164, pp.382-385.
   - Lu: Soederlund, Ulf, et al. "The ^{176}Lu decay constant determined by Lu-Hf and U-Pb isotope systematics of Precambrian mafic intrusions." Earth and Planetary Science Letters 219.3 (2004): 311-324.

2. Isotopic ratios:

   - Ar: Lee, Jee-Yon, et al. "A redetermination of the isotopic abundances of atmospheric Ar." Geochimica et Cosmochimica Acta 70.17 (2006): 4507-4512.
   - Rb: Catanzaro, E. J., et al. "Absolute isotopic abundance ratio and atomic weight of terrestrial rubidium." J. Res. Natl. Bur. Stand. A 73 (1969): 511-516.
   - Sr: Moore, L. J., et al. "Absolute isotopic abundance ratios and atomic weight of a reference sample of strontium." J. Res. Natl.Bur. Stand. 87.1 (1982): 1-8.
   - Sm: Chang, Tsing-Lien, et al. "Absolute isotopic composition and atomic weight of samarium." International Journal of Mass Spectrometry 218.2 (2002): 167-172.
   - Re: Gramlich, John W., et al. "Absolute isotopic abundance ratio and atomic weight of a reference sample of rhenium." J. Res. Natl. Bur. Stand. A 77 (1973): 691-698.
   - Os: Voelkening, Joachim, Thomas Walczyk, and Klaus G. Heumann. "Osmium isotope ratio determinations by negative thermal ionization mass spectrometry." Int. J. Mass Spect. Ion Proc. 105.2 (1991): 147-159.
   - Lu: De Laeter, J. R., and N. Bukilic. "Solar abundance of ^{176}Lu and s-process nucleosynthesis." Physical Review C 73.4 (2006): 045806.
   - Hf: Patchett, P. Jonathan. "Importance of the Lu-Hf isotopic system in studies of planetary chronology and chemical evolution." Geochimica et Cosmochimica Acta 47.1 (1983): 81-91.
   - U: Hiess, Joe, et al. "^{238}U/^{235}U systematics in terrestrial uranium-bearing minerals." Science 335.6076 (2012): 1610-1614.

Examples

```
# load and show the default constants that come with IsoplotR
json <- system.file("constants.json",package="IsoplotR")
settings(fname=json)
print(settings())

# use the decay constant of Kovarik and Adams (1932)
settings('lambda','U238',0.0001537,0.0000068)
print(settings('lambda','U238'))

# returns the 238U/235U ratio of Hiess et al. (2012):
print(settings('iratio','U238U235'))
# use the 238U/235U ratio of Steiger and Jaeger (1977):
settings('iratio','U238U235',138.88,0)
print(settings('iratio','U238U235'))
```

---

titterington *Linear regression of X,Y,Z-variables with correlated errors*

Description

Implements the maximum likelihood algorithm of Ludwig and Titterington (1994)

Usage

```
titterington(x, alpha = 0.05)
```

Arguments

x	a [9 x n] matrix with the following columns: X, sX, Y, sY, Z, sZ, rhoXY, rhoXZ, rhoYZ.
alpha	cutoff value for confidence intervals

Value

a four-element list of vectors containing:

par 4-element vector c(a,b,A,B) where a is the intercept of the X-Y regression, b is the slope of the X-Y regression, A is the intercept of the X-Z regression, and B is the slope of the X-Z regression.

cov [4 x 4]-element covariance matrix of par

mswd the mean square of the residuals (a.k.a 'reduced Chi-square') statistic

p.value p-value of a Chi-square test for linearity

df the number of degrees of freedom for the Chi-square test ($3n$-3)

References

Ludwig, K.R. and Titterington, D.M., 1994. Calculation of ^{230}Th/U isochrons, ages, and errors. Geochimica et Cosmochimica Acta, 58(22), pp.5031-5042.

Examples

```
d <- matrix(c(0.1677,0.0047,1.105,0.014,0.782,0.015,0.24,0.51,0.33,
              0.2820,0.0064,1.081,0.013,0.798,0.015,0.26,0.63,0.32,
              0.3699,0.0076,1.038,0.011,0.819,0.015,0.27,0.69,0.30,
              0.4473,0.0087,1.051,0.011,0.812,0.015,0.27,0.73,0.30,
              0.5065,0.0095,1.049,0.010,0.842,0.015,0.27,0.76,0.29,
              0.5520,0.0100,1.039,0.010,0.862,0.015,0.27,0.78,0.28),
            nrow=6,ncol=9)
colnames(d) <- c('X','sX','Y','sY','Z','sZ','rXY','rXZ','rYZ')
titterington(d)
```

---

weightedmean *Calculate the weighted mean age*

Description

Models the data as a Normal distribution with two sources of variance. Estimates the mean and 'overdispersion' using the method of Maximum Likelihood. Computes the MSWD of a Normal fit without overdispersion. Implements Chauvenet's Criterion to detect and reject outliers. Only propagates the analytical uncertainty associated with decay constants and J-factors after computing the weighted mean isotopic composition.

Usage

```
weightedmean(x, ...)

## Default S3 method:
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, ...)

## S3 method for class 'UPb'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, type = 4, cutoff.76 = 1100, cutoff.disc = c(-15, 5),
  alpha = 0.05, exterr = TRUE, common.Pb = 0, ...)

## S3 method for class 'PbPb'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, exterr = TRUE, i2i = FALSE, ...)

## S3 method for class 'ThU'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, i2i = TRUE, ...)

## S3 method for class 'ArAr'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, exterr = TRUE, i2i = FALSE, ...)

## S3 method for class 'ReOs'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, exterr = TRUE, i2i = TRUE, ...)

## S3 method for class 'SmNd'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, exterr = TRUE, i2i = TRUE, ...)

## S3 method for class 'RbSr'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, exterr = TRUE, i2i = TRUE, ...)

## S3 method for class 'LuHf'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, exterr = TRUE, i2i = TRUE, ...)
```

```
## S3 method for class 'UThHe'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, ...)

## S3 method for class 'fissiontracks'
weightedmean(x, detect.outliers = TRUE, plot = TRUE,
  rect.col = rgb(0, 1, 0, 0.5), outlier.col = rgb(0, 1, 1, 0.5),
  sigdig = 2, alpha = 0.05, exterr = TRUE, ...)
```

Arguments

- `x` — a two column matrix of values (first column) and their standard errors (second column) OR an object of class `UPb`, `PbPb`, `ArAr`, `ReOs`, `SmNd`, `RbSr`, `LuHf`, `ThU`, `fissiontracks` or `UThHe`
- `...` — optional arguments
- `detect.outliers` — logical flag indicating whether outliers should be detected and rejected using Chauvenet's Criterion.
- `plot` — logical flag indicating whether the function should produce graphical output or return numerical values to the user.
- `rect.col` — the fill colour of the rectangles used to show the measurements or age estimates.
- `outlier.col` — if `detect.outliers=TRUE`, the outliers are given a different colour.
- `sigdig` — the number of significant digits of the numerical values reported in the title of the graphical output.
- `alpha` — the confidence limits of the error bars/rectangles.
- `type` — scalar indicating whether to plot the ^{207}Pb/^{235}U age (`type`=1), the ^{206}Pb/^{238}U age (`type`=2), the ^{207}Pb/^{206}Pb age (`type`=3), the ^{207}Pb/^{206}Pb-^{206}Pb/^{238}U age (`type`=4), or the (Wetherill) concordia age (`type`=5)
- `cutoff.76` — the age (in Ma) below which the ^{206}Pb/^{238}U age and above which the ^{207}Pb/^{206}Pb age is used. This parameter is only used if `type=4`.
- `cutoff.disc` — two element vector with the maximum and minimum percentage discordance allowed between the ^{207}Pb/^{235}U and ^{206}Pb/^{238}U age (if ^{206}Pb/^{238}U < `cutoff.76`) or between the ^{206}Pb/^{238}U and ^{207}Pb/^{206}Pb age (if ^{206}Pb/^{238}U > `cutoff.76`). Set `cutoff.disc=NA` if you do not want to use this filter.
- `exterr` — propagate decay constant uncertainty?
- `common.Pb` — apply a common lead correction using one of three methods:

  `1`: use the isochron intercept as the initial Pb-composition

  `2`: use the Stacey-Kramer two-stage model to infer the initial Pb-composition

  `3`: use the Pb-composition stored in `settings('iratio','Pb206Pb204')` and `settings('iratio','Pb207Pb204')`
- `i2i` — 'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') ^{40}Ar/^{36}Ar, ^{207}Pb/^{204}Pb, ^{87}Sr/^{86}Sr, ^{143}Nd/^{144}Nd, ^{187}Os/^{188}Os or ^{176}Hf/^{177}Hf

ratio from an isochron fit. Setting `i2i` to `FALSE` uses the default values stored in `settings('iratio',...)` or zero (for the Pb-Pb method). When applied to data of class `ThU`, setting `i2i` to `TRUE` applies a detrital Th-correction.

Value

returns a list with the following items:

mean a three element vector with:

`x`: the weighted mean

`s[x]`: the estimated analytical uncertainty of `x`

`ci[x]`: the $100(1-\alpha/2)\%$ confidence interval for `x` given the appropriate degrees of freedom

disp a two element vector with the (over)dispersion the corresponding $100(1-\alpha/2)\%$ confidence interval.

mswd the Mean Square of the Weighted Deviates (a.k.a. 'reduced Chi-square' statistic)

p.value the p-value of a Chi-square test with n-1 degrees of freedom, testing the null hypothesis that the underlying population is not overdispersed.

df the degrees of freedom for the Chi-square test

tfact the $100(1-\alpha/2)$ percentile of a t-distribution with `df` degrees of freedom

valid vector of logical flags indicating which steps are included into the weighted mean calculation

plotpar list of plot parameters for the weighted mean diagram

See Also

`central`

Examples

```
ages <- c(251.9,251.59,251.47,251.35,251.1,251.04,250.79,250.73,251.22,228.43)
errs <- c(0.28,0.28,0.63,0.34,0.28,0.63,0.28,0.4,0.28,0.33)
weightedmean(cbind(ages,errs))
data(examples)
weightedmean(examples$ArAr)
```

`york` *Linear regression of X,Y-variables with correlated errors*

Description

Implements the unified regression algorithm of York et al. (2004) which, although based on least squares, yields results that are consistent with maximum likelihood estimates of Titterington and Halliday (1979)

Usage

```
york(x, alpha = 0.05)
```

Arguments

`x` — a 5-column matrix with the X-values, the analytical uncertainties of the X-values, the Y-values, the analytical uncertainties of the Y-values, and the correlation coefficients of the X- and Y-values.

`alpha` — cutoff value for confidence intervals

Value

a four-element list of vectors containing:

a the intercept of the straight line fit and its standard error

b the slope of the fit and its standard error

cov.ab the covariance of the slope and intercept

mswd the mean square of the residuals (a.k.a 'reduced Chi-square') statistic

df degrees of freedom of the linear fit $(2n-2)$

p.value p-value of a Chi-square value with `df` degrees of freedom

References

Titterington, D.M. and Halliday, A.N., 1979. On the fitting of parallel isochrons and the method of maximum likelihood. Chemical Geology, 26(3), pp.183-195.

York, Derek, et al. "Unified equations for the slope, intercept, and standard errors of the best straight line." American Journal of Physics 72.3 (2004): 367-375.

Examples

```
X <- c(1.550,12.395,20.445,20.435,20.610,24.900,
       28.530,50.540,51.595,86.51,106.40,157.35)
Y <- c(.7268,.7849,.8200,.8156,.8160,.8322,
       .8642,.9584,.9617,1.135,1.230,1.490)
n <- length(X)
sX <- X*0.01
sY <- Y*0.005
rXY <- rep(0.8,n)
dat <- cbind(X,sX,Y,sY,rXY)
fit <- york(dat)
covmat <- matrix(0,2,2)
plot(range(X),fit$a[1]+fit$b[1]*range(X),type='l',ylim=range(Y))
for (i in 1:n){
    covmat[1,1] <- sX[i]^2
    covmat[2,2] <- sY[i]^2
    covmat[1,2] <- rXY[i]*sX[i]*sY[i]
    covmat[2,1] <- covmat[1,2]
    ell <- ellipse(X[i],Y[i],covmat,alpha=0.05)
    polygon(ell)
}
```

Index