Package ‘IsoplotR’

June 6, 2021

Title  Statistical Toolbox for Radiometric Geochronology

Version 4.0

Date 2021-06-06


Author  Pieter Vermeesch [aut, cre]

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

Depends R (>= 3.0.0)

Imports MASS, methods

License GPL-3


LazyData true

RoxygenNote 7.1.1

Encoding UTF-8

NeedsCompilation no

Repository CRAN

Date/Publication 2021-06-06 21:30:02 UTC
R topics documented:

age ................................................................. 2
age2ratio .......................................................... 8
agespectrum ....................................................... 10
cad ............................................................... 13
central ........................................................... 18
classes ........................................................... 21
concordia ......................................................... 23
data2york ......................................................... 27
discfilter .......................................................... 29
diseq .............................................................. 31
ellipse ............................................................ 33
evolution .......................................................... 33
examples .......................................................... 36
helioplot ........................................................... 39
isochron .......................................................... 42
IsoplotR .......................................................... 54
dk ................................................................. 55
ludwig ............................................................. 64
mclean ........................................................... 66
mds ............................................................... 67
Pb0corr ........................................................... 70
peakfit ............................................................ 72
radialplot ......................................................... 77
read.data .......................................................... 87
scatterplot ....................................................... 91
set.zeta ........................................................... 94
settings ........................................................... 95
titterington ...................................................... 99
weightedmean .................................................. 100
york ............................................................. 110

Index 112

age

Calculate isotopic ages

Description

Calculates U-Pb, Pb-Pb, Th-Pb, Ar-Ar, K-Ca, Re-Os, Sm-Nd, Rb-Sr, Lu-Hf, U-Th-He, Th-U and fission track ages and propagates their analytical uncertainties. Includes options for single grain, isochron and concordia ages.
Usage

age(x, ...)

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

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

## S3 method for class 'PbPb'
age(
  x,
  isochron = TRUE,
  common.Pb = 2,
  exterr = FALSE,
  i = NA,
  sigdig = NA,
  projerr = FALSE,
  ...
)

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

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

## S3 method for class 'UThHe'
age(x, isochron = FALSE, 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 = FALSE,
i = NA,
sigdig = NA,
detritus = 0,
...)

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

## S3 method for class 'ReOs'
age(x,
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,
• a four element vector containing Ar40Ar39, s[Ar40Ar39], J, s[J],
• a two element vector containing K40Ca40 and s[K40Ca40],
• 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 two element vector containing Sr87Rb87 and s[Sr87Rb87]
• a two element vector containing Os187Re187 and s[Os187Re187]
• a two element vector containing Nd143Sm147 and s[Nd144Sm147]
• a two element vector containing Hf176Lu176 and s[Hf176Lu176]
• a five element vector containing Th230U238, s[Th230/U238], U234U238, s[U234U238] and cov[Th230U238, U234U238]

OR
• an object of class U-Pb, Pb-Pb, Th-Pb, Ar-Ar, K-Ca, Th-U, Rb-Sr, Sm-Nd, Re-Os, Lu-Hf, U-Th-He or fissiontracks.

... additional arguments
method one of either 'U238-Pb206', 'U235-Pb207', 'Pb207-Pb206', 'Th232-Pb208',
  'Ar-Ar', 'K-Ca', '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.
d an object of class diseq.
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 fitted through all the U-Pb analyses using the max-
      imum 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 fitted ignoring the analytical uncertainties.
  5: a discordia line should be fitted using a modified maximum likelihood al-
     gorithm that accounts for overdispersion by adding a geological (co)variance
     term.
i index of a particular aliquot
sigdig number of significant digits for the uncertainty estimate (only used if type=1,
  isochron=FALSE and central=FALSE).
common.Pb common lead correction:
0: none
1: use the Pb-composition stored in
settings('iratio', 'Pb207Pb206') (if x has class UPb and x$format<4);
settings('iratio', 'Pb206Pb204') and settings('iratio', 'Pb207Pb204')
(if x has class PbPb or x has class UPb and 3<x$format<7); or
settings('iratio', 'Pb208Pb206') and settings('iratio', 'Pb208Pb207')
(if x has class UPb and x$format=7 or 8).
2: use the isochron intercept as the initial Pb-composition
3: use the Stacey-Kramer two-stage model to infer the initial Pb-composition

discordance discordance calculator. This is an object of class discfilter, or a two element
list containing:
option: one of
1 or 't' (absolute age filter);
2 or 'r' (relative age filter);
3 or 'sk' (Stacey-Kramers common Pb filter);
4 or 'a' (perpendicular Aitchison distance);
5 or 'c' (concordia distance);
6 or 'p' (p-value of concordance); or
NA (omit the discordance from the output)
before: logical flag indicating whether the discordance should be calculated
before (TRUE) or after (FALSE) the common-Pb correction.

isochron logical flag indicating whether each analysis should be considered separately
(isochron=FALSE) or an isochron age should be calculated from all analyses
together (isochron=TRUE).

projerr logical. If TRUE, propagates the uncertainty of the non-radiogenic isotope cor-
rection (the ‘projection error’) into the age uncertainty. Note that the resulting
single grain age uncertainties may be strongly correlated with each other, but
these error correlations are not reported in the output.

i2i 'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'com-
mon') $^{40}\text{Ar}/^{36}\text{Ar}, ^{40}\text{Ca}/^{44}\text{Ca}, ^{87}\text{Sr}/^{86}\text{Sr}, ^{143}\text{Nd}/^{144}\text{Nd}, ^{187}\text{Os}/^{188}\text{Os}, ^{176}\text{Hf}/^{177}\text{Hf}
or $^{204}\text{Pb}/^{208}\text{Pb}$ ratio from an isochron fit. Setting i2i to FALSE uses the default
values stored in settings('iratio',...). When applied to data of class ThU,
setting i2i to TRUE applies a detrital Th-correction.

central logical flag indicating whether each analysis should be considered separately
(central=FALSE) or a central age should be calculated from all analyses to-
gether (central=TRUE).

detritus detrital $^{230}\text{Th}$ correction (only applicable when x$format = 1 or 2).
0: no correction
1: project the data along an isochron fit
2: correct the data using an assumed initial $^{230}\text{Th}/^{232}\text{Th}$-ratio for the detritus.
3: correct the data using the measured present day $^{230}\text{Th}/^{238}\text{U}, ^{232}\text{Th}/^{238}\text{U}$ and
$^{234}\text{U}/^{238}\text{U}$-ratios in the detritus.
age2ratio

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.82, err[t.82]), t.conc, err[t.conc], (disc) or err[p.conc], containing the \( { }^{207} \text{Pb}/^{235} \text{U} \)-age and standard error, the \( { }^{206} \text{Pb}/^{238} \text{U} \)-age and standard error, the \( { }^{207} \text{Pb}/^{206} \text{Pb} \)-age and standard error, the \( { }^{208} \text{Pb}/^{232} \text{Th} \)-age and standard error, the single grain concordia age and standard error, (and the % discordance or p-value for concordance,) respectively.
3. if x has class UPb and type=2, 3, 4 or 5, returns the output of the concordia function.
4. if x has class PbPb, ThPb, ArAr, KCa, RbSr, SmNd, ReOs, LuHf, ThU or UThHe and isochron=FALSE, returns a table of Pb-Pb, Th-Pb, Ar-Ar, K-Ca, Rb-Sr, Sm-Nd, Re-Os, Lu-Hf, Th-U or U-Th-He 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 \( { }^{234} \text{U}/^{238} \text{U} \)-ratios, their standard errors, and the correlation coefficient between the ages and the initial ratios.
6. if x has class PbPb, ThPb, ArAr, KCa, RbSr, SmNd, ReOs, LuHf, UThHe 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

```
attach(examples)
tUPb <- age(UPb, type=1)
tconc <- age(UPb, type=2)
tdisc <- age(UPb, type=3)
tArAr <- age(ArAr)
tiso <- age(ArAr, isochron=TRUE, i2i=TRUE)
tcentral <- age(FT1, central=TRUE)
```

Description

Groups a set of functions that take one (or more) ages (and their uncertainties) as input and produces the U–Pb, Th–Pb, Pb–Pb, Ar–Ar, K–Ca, Rb–Sr, Sm–Nd, Lu–Hf, Re–Os, concordia or Stacey-Kramers ratios as output.
age2ratio

Usage

age2ratio(tt, st = 0, ratio = "Pb206U238", exterr = FALSE, d = diseq(), J, sJ)

Arguments

**tt**
a scalar or (except when ratio = 'Wetherill', 'Tera-Wasserburg' or 'U-Th-Pb')
vector of ages.

**st**
a scalar or (except when ratio = 'Wetherill', 'Tera-Wasserburg' or 'U-Th-Pb')
vector of ages. Not used when ratio = 'Stacey-Kramers'.

**ratio**
one of 'Pb207U235', 'U238Pb206', 'Pb207Pb206', 'Pb208Th232', 'Wetherill',
'Tera-Wasserburg', 'U-Th-Pb', 'Ar40Ar39', 'Ca40K40', 'Hf176Lu176', 'Sr87Rb87',
'Os187Re187', 'Nd143Sm147' or 'Stacey-Kramers'.

**exterr**
logical. If TRUE, propagates decay constant uncertainties into st. Not used when
ratio = 'Stacey-Kramers'.

**d**
an object of class diseq.

**J**
the J-factor of the Ar–Ar system (only used if ratio is 'Ar40Ar39').

**sJ**
the standard error of J (only used if ratio is 'Ar40Ar39').

Value

If **ratio** is 'Pb207U235', 'U238Pb206', 'Pb207Pb206', 'Pb208Th232', 'Ar40Ar39', 'Ca40K40',
'Hf176Lu176', 'Sr87Rb87', 'Os187Re187' or 'Nd143Sm147': either a two-element vector or a
two-column matrix with the predicted isotopic ratio(s) and its/their standard error(s).

If **ratio** is 'Wetherill', 'Tera-Wasserburg' or 'U-Th-Pb': a two-element list containing
x: the concordia ratios
cov: the covariance matrix of the concordia ratios

If **ratio** is 'Stacey-Kramers': a three-column matrix with predicted $^{206}\text{Pb}/^{204}\text{Pb}$, $^{207}\text{Pb}/^{204}\text{Pb}$
and $^{208}\text{Pb}/^{204}\text{Pb}$ ratios.

Examples

```r
ratios <- c('Pb207U235','U238Pb206','Pb207Pb206','Pb208Th232',
'Wetherill','Tera-Wasserburg','U-Th-Pb','Ar40Ar39',
'Ca40K40','Hf176Lu176','Sr87Rb87','Os187Re187',
'Nd143Sm147','Stacey-Kramers')
for (ratio in ratios){
  r <- age2ratio(tt=1000, st=10, ratio=ratio, J=1, sJ=0.1)
  print(r)
}
```
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 variable), and whose heights express the analytical uncertainties. Only propagates the analytical uncertainty associated with decay constants and J-factors *after* computing the plateau composition.

**Usage**

```r
agespectrum(x, ...)  
## Default S3 method:  
agespectrum(
  x,
  alpha = 0.05,
  plateau = TRUE,
  random.effects = FALSE,
  levels = NA,
  clabel = "",
  plateau.col = c("#00FF0080", "#FF000080"),
  non.plateau.col = "#00FFFF80",
  sigdig = 2,
  line.col = "red",
  lwd = 2,
  xlab = "cumulative fraction",
  ylab = "age [Ma]",
  hide = NULL,
  omit = NULL,
  ...
)

## S3 method for class 'ArAr'  
agespectrum(
  x,
  alpha = 0.05,
  plateau = TRUE,
  random.effects = FALSE,
  levels = NA,
  clabel = "",
  plateau.col = c("#00FF0080", "#FF000080"),
  non.plateau.col = "#00FFFF80",
  sigdig = 2,
  exterr = TRUE,
  line.col = "red",
  lwd = 2,
)```

---

*agespectrum*
Arguments

\texttt{x} \hspace{2cm} 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

\texttt{an object of class \texttt{ArAr}}

\ldots \hspace{2cm} \texttt{optional parameters to the generic \texttt{plot} function}

\texttt{alpha} \hspace{2cm} the confidence level of the error bars/boxes and confidence intervals.

\texttt{plateau} \hspace{2cm} \texttt{logical flag indicating whether a plateau age should be calculated. If \texttt{plateau=TRUE}, the function computes the weighted mean of the largest succession of steps that pass the Chi-square test for age homogeneity. If \texttt{TRUE}, it returns a list with plateau parameters.}

\texttt{random.effec\texttt{s}} \hspace{2cm} \texttt{if TRUE, computes the weighted mean using a random effects model with two parameters: the mean and the dispersion. This is akin to a ‘model-3’ isochron regression.}

\hspace{2cm} \texttt{if FALSE, attributes any excess dispersion to an underestimation of the analytical uncertainties. This akin to a ‘model-1’ isochron regression.}

\texttt{levels} \hspace{2cm} \texttt{a vector with additional values to be displayed as different background colours of the plot symbols.}

\texttt{clabel} \hspace{2cm} \texttt{label of the colour legend}

\texttt{plateau.col} \hspace{2cm} \texttt{Fill colours of the rectangles used to mark the steps belonging to the age plateau. This can either be a single colour or multiple colours to form a colour ramp (to be used if \texttt{levels!}=\texttt{NA}):}

\hspace{2cm} \texttt{a single colour: rgb(0,1,0,0.5), '#FF000080', 'white', etc.;}

\hspace{2cm} \texttt{multiple colours: c(rgb(1,0,0,0.5), rgb(0,1,0,0.5)), c('#FF000080', '#00FF0080'), c('blue', 'red'), c('blue', 'yellow', 'red'), etc.}

\hspace{2cm} \texttt{a colour palette: rainbow(n=100), topo.colors(n=100, alpha=0.5), etc.; or a reversed palette: rev(topo.colors(n=100, alpha=0.5)), etc.}

\hspace{2cm} \texttt{For empty boxes, set \texttt{plateau.col=}\texttt{NA}}

\texttt{non.plateau.col} \hspace{2cm} \texttt{if \texttt{plateau=}\texttt{TRUE}, the steps that do NOT belong to the plateau are given a different colour.}

\texttt{sigdig} \hspace{2cm} \texttt{the number of significant digits of the numerical values reported in the title of the graphical output.}

\texttt{line.col} \hspace{2cm} \texttt{colour of the average age line}

\texttt{lwd} \hspace{2cm} \texttt{width of the average age line}

\texttt{xlab} \hspace{2cm} \texttt{x-axis label}

\texttt{ylab} \hspace{2cm} \texttt{y-axis label}
hide vector with indices of aliquots that should be removed from the plot.
omit vector with indices of aliquots that should be plotted but omitted from age plateau calculation
exterr propagate the external (decay constant and calibration factor) uncertainties?
i2i 'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') $^{39}$Ar/$^{36}$Ar ratio from an isochron fit. Setting i2i to FALSE uses the default values stored in settings('iratio',...)

Details

IsoplotR defines the ‘plateau age’ as the weighted mean age (using a random effects model with two sources of dispersion) 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. It is possible (and likely) that the plateau steps exhibit significant overdispersion. This overdispersion can be manually reduced by removing individual heating steps with the optional omit argument.

Value

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

mean a 3-element vector with:
  t: the plateau mean
  s[t]: the standard error of x
  ci[t]: the width of a 100(1-α)% confidence interval of t

disp a 3-element vector with:
  w: the overdispersion, i.e. the standard deviation of the Normal distribution that is assumed to describe the true ages.
  ll: the width of the lower half of a 100(1-α)% confidence interval for the overdispersion
  ul: the width of the upper half of a 100(1-α)% confidence interval for the overdispersion

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

plotpar plot parameters for the weighted mean (see weightedmean)

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

See Also

weightedmean
Examples

attach(examples)
par(mfrow=c(2,1))
agespectrum(ArAr)
# removing the first 6 steps yields the longest plateau
# that passes the chi-square test for homogeneity
agespectrum(ArAr, omit=1:6)

---

cad

Plot continuous data as cumulative age distributions

Description

Plot a dataset as a Cumulative Age Distribution (CAD), also known as a ‘empirical cumulative distribution function’.

Usage

cad(x, ...)

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

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

## S3 method for class 'UPb'
cad(
  x,
  pch = NA,
  verticals = TRUE,
```r
xlab = "age [Ma]",
col = "black",
type = 4,
cutoff.76 = 1100,
cutoff.disc = discfilter(),
common.Pb = 0,
hide = NULL,
...
)

## S3 method for class 'PbPb'
cad(
  x,
pch = NA,
  verticals = TRUE,
xlab = "age [Ma]",
col = "black",
  common.Pb = 1,
  hide = NULL,
...
)

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

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

## S3 method for class 'ThPb'
cad(
```
## S3 method for class 'ThU'

cad(
  x,
  pch = NA,
  verticals = TRUE,
  xlab = "age [Ma]",
  col = "black",
  i2i = TRUE,
  hide = NULL,
  ...  
)

## S3 method for class 'ThPb'

cad(
  x,
  pch = NA,
  verticals = TRUE,
  xlab = "age [ka]",
  col = "black",
  i2i = FALSE,
  detritus = 0,
  hide = NULL,
  ... 
)

## S3 method for class 'ReOs'

cad(
  x,
  pch = NA,
  verticals = TRUE,
  xlab = "age [Ma]",
  col = "black",
  i2i = TRUE,
  hide = NULL,
  ...  
)

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

## S3 method for class 'RbSr'

## S3 method for class 'LuHf'

## S3 method for class 'UTHe'

## S3 method for class 'fissiontracks'

x,
pch = NA,
verticals = TRUE,
xlab = "age [Ma]",
col = "black",
hide = NULL,
...)

Arguments

- **x**: a numerical vector OR an object of class UPb, PbPb, ThPb, ArAr, KCa, 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
- **col**: either the name of one of R’s built-in colour palettes (e.g., ‘heat.colors’, ‘terrain.colors’, ‘topo.colors’, ‘cm.colors’) (if `x` has class detritals) OR the name or code for a colour to give to single sample datasets.
- **hide**: vector with indices of aliquots that should be removed from the plot.
- **type**: scalar indicating whether to plot the \(^{207}\text{Pb}/^{235}\text{U}\) age (`type`=1), the \(^{206}\text{Pb}/^{238}\text{U}\) age (`type`=2), the \(^{207}\text{Pb}/^{206}\text{Pb}\) age (`type`=3), the \(^{207}\text{Pb}/^{206}\text{Pb}-^{206}\text{Pb}/^{238}\text{U}\) age (`type`=4), the concordia age (`type`=5), or the \(^{208}\text{Pb}/^{232}\text{Th}\) age (`type`=6).
- **cutoff.76**: the age (in Ma) below which the \(^{206}\text{Pb}/^{238}\text{U}\)-age and above which the \(^{207}\text{Pb}/^{206}\text{Pb}\)-age is used. This parameter is only used if `type`=4.
- **cutoff.disc**: discordance cutoff filter. This is an object of class `discfilter`.
- **common.Pb**: common lead correction:
  0: none
  1: use the Pb-composition stored in
    settings('iratio', 'Pb207Pb206') (if `x` has class UPb and x$format<4);
    settings('iratio', 'Pb206Pb204') and settings('iratio', 'Pb207Pb204')
    (if `x` has class PbPb or `x` has class UPb and 3<x$format<7); or
    settings('iratio', 'Pb208Pb206') and settings('iratio', 'Pb208Pb207')
    (if `x` has class UPb and x$format=7 or 8).
  2: use the isochron intercept as the initial Pb-composition
  3: use the Stacey-Kramers two-stage model to infer the initial Pb-composition
    (only applicable if `x` has class UPb)
- **i2i**: ‘isochron to intercept’: calculates the initial (aka ‘inherited’, ‘excess’, or ‘common’) \(^{40}\text{Ar}/^{36}\text{Ar}, ^{40}\text{Ca}/^{44}\text{Ca}, ^{207}\text{Pb}/^{204}\text{Pb}, ^{87}\text{Sr}/^{86}\text{Sr}, ^{143}\text{Nd}/^{144}\text{Nd}, ^{187}\text{Os}/^{188}\text{Os}, ^{230}\text{Th}/^{232}\text{Th}, ^{176}\text{Hf}/^{177}\text{Hf} or ^{204}\text{Pb}/^{208}\text{Pb}\) ratio from an isochron fit. Setting `i2i` to FALSE uses the default values stored in settings('iratio',...).
Central detrital $^{230}$Th correction (only applicable when x$\text{format}=1$ or $2$).

0: no correction
1: project the data along an isochron fit
2: correct the data using an assumed initial $^{230}$Th/$^{232}$Th-ratio for the detritus.
3: correct the data using the measured present day $^{230}$Th/$^{238}$U, $^{232}$Th/$^{238}$U and $^{234}$U/$^{238}$U-ratios in the detritus.

Details

Empirical cumulative distribution functions or cumulative age distributions 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 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 $1(\ast) = 1$ if $\ast$ is true and $1(\ast) = 0$ if $\ast$ is false. CADs have two desirable properties (Vermeesch, 2007). 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.

References


See Also

kde, radialplot

Examples

attach(examples)

cad(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 continuous mixture of fission track or U-Th-He data and returns the corresponding age and fitting parameters. Only propagates the systematic uncertainty associated with decay constants and calibration factors after computing the weighted mean isotopic composition. Does not propagate the uncertainty of any initial daughter correction, because this is neither a purely random or purely systematic uncertainty.
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, alpha = 0.05, exterr = FALSE, ...)

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      if the scatter between the data points is solely caused by the analytical uncertainty, then the MSWD value should be approximately equal to one. There are three strategies to deal with the case where MSWD>1.choose one of the following statistical models:
            1: assume that the analytical uncertainties have been underestimated by a factor \(\sqrt{MSWD}\).
            2: ignore the analytical uncertainties.
            3: attribute any excess dispersion to the presence of geological uncertainty, which manifests itself as an added (co)variance term.
exterr     include the zeta or decay constant uncertainty into the error propagation for the central age?

Details

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 and uncertainties are assumed to follow a logistic normal distribution.
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.

The uncertainty budget of the central age does not include the uncertainty of the initial daughter correction (if any), for the same reasons as discussed under the weightedmean function.
Value

If x has class UThHe, returns a list containing the following items:

- **uvw** (if the input data table contains Sm) or **uv** (if it does not): the 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. \( mswd = \frac{SS}{df} \), 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.)
- **age** a three- or four-element vector with:
  - **t**: the central age.
  - **s[t]**: the standard error of t.
  - **ci[t]**: the width of a 100(1 - \( \alpha \))% confidence interval for t.
  - **disp[t]**: the studentised 100(1 - \( \alpha \))% 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 three-element vector with the standard deviation of the (assumedly) Normal dispersion and the lower and upper half-widths of its 100(1 - \( \alpha \))% confidence interval. \( w=0 \) if model<3.

OR, otherwise:

- **age** a three-element vector with:
  - **t**: the central age.
  - **s[t]**: the standard error of t.
  - **ci[t]**: the width of a 100(1 - \( \alpha \))% confidence interval for t.
- **disp** a three-element vector with the overdispersion (standard deviation) of the excess scatter, and the upper and lower half-widths of its 100(1 - \( \alpha \))% confidence interval.
- **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


See Also

weightedmean, radialplot, helioplot
Classes

Examples

```r
attach(examples)
prompt(central(UTbHe)$age)
```

<table>
<thead>
<tr>
<th>classes</th>
<th>Geochronological data classes</th>
</tr>
</thead>
</table>

Description

S3 classes to store geochronological data generated by `read.data` or `diseq`.

Usage

```r
is.UPb(x)
```

```r
is.PbPb(x)
```

```r
is.ThPb(x)
```

```r
is.ArAr(x)
```

```r
is.KCa(x)
```

```r
is.PD(x)
```

```r
is.RbSr(x)
```

```r
is.SmNd(x)
```

```r
is.LuHf(x)
```

```r
is.ReOs(x)
```

```r
is.ThU(x)
```

```r
is.UTbHe(x)
```

```r
is.fissiontracks(x)
```

```r
is.detritals(x)
```

```r
is.diseq(x)
```

Arguments

```r
x
```
a data object returned by `read.data` or `diseq`.
Details

IsoplotR uses the following S3 classes to store geochronological data: UPb, PbPb, ThPb, KCa, UThHe, fissiontracks, detritals and PD, where the latter is the parent class of the simple parent-daughter chronometers RbSr, SmNd, LuHf and ReOs. All these classes have overloaded versions of the generic `length()` function and `[` subsetting method.

- **UPb**: a list containing:
  - a matrix containing the isotopic measurements
  - format a number between 1 and 8
  - d an object of class diseq, i.e. the output of the `diseq` function
- **ArAr**: a list containing
  - x a matrix containing the isotopic measurements
  - J a two-element vector with the J-factor and its uncertainty
  - format a number between 1 and 3
- **ThU**: a list containing
  - x a matrix containing the isotopic measurements
  - format a number between 1 and 4
  - Th02 a two element vector with the assumed initial $^{230}\text{Th}/^{232}\text{Th}$-ratio of Th-bearing detritus (for formats 1 and 2) or rock (for formats 3 and 4) and its standard error
  - Th02U48 a 9-element vector with the measured composition of Th-bearing detritus
- **PbPb, ThPb, KCa, PD, RbSr, SmNd, LuHf, or ReOs**: a list containing
  - x a matrix containing the isotopic measurements
  - format a number between 1 and 3
- **UThHe**: a matrix of He, U, Th (and Sm) measurements
- **fissiontracks**: a list containing
  - format a number between 1 and 3
  - x a matrix of spontaneous and induced fission track counts (only included if `format==1`)
  - rhoD the track density of the dosimeter glass, extracted from the input data (only included if `format==1`)
  - zeta the zeta calibration constant extracted from the input data (only included if `format<3`)
  - Ns a list containing the spontaneous fission track counts (only included if `format>1`)
  - U a list of lists containing the U-concentration or U/Ca-ratio measurements for each of the analysed grains (only included if `format>1`)
  - sU a list of lists containing the standard errors of the U-concentration or U/Ca-ratio measurements for each of the analysed grains (only include if `format>1`)
  - spotSize the laser ablation spot size (only included if `format>1`)
- **detritals**: a list of named vectors, one for each detrital sample.
- **diseq**: is a class that contains the output of the `diseq` function, which stores initial disequilibrium data for U–Pb geochronology.

**Value**

logical
concordia

See Also

read.data diseq

Examples

attach(examples)
ns <- length(UPb)
concordia(UPb[-ns,])
if (is.PD(RbSr)) print('RbSr has class PD')

concordia  Concordia diagram

Description

Plots U-Pb data on Wetherill, Tera-Wasserburg or U-Th-Pb concordia diagrams, calculates concordia ages and compositions, evaluates the equivalence of multiple \( \frac{206}{238} \text{Pb} \), \( \frac{207}{235} \text{U} \), \( \frac{207}{206} \text{Pb} \), \( \frac{208}{232} \text{Th} \), \( \frac{206}{238} \text{U} \), or \( \frac{207}{206} \text{Pb} \) 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 \( \frac{207}{206} \text{Pb} \) intercept (for Tera-Wasserburg), taking into account error correlations and decay constant uncertainties.

Usage

concordia(
x = NULL,
tlim = NULL,
alpha = 0.05,
type = 1,
show.numbers = FALSE,
levels = NA,
clabel = "",
ellipse.fill = c("#00FF0080", "#FF000080"),
ellipse.stroke = "black",
concordia.col = "darksalmon",
esterr = FALSE,
show.age = 0,
sigdig = 2,
common.Pb = 0,
ticks = 5,
anchor = 0,
hide = NULL,
omit = NULL,
omit.fill = NA,
omit.stroke = "grey",
...
)

Arguments

x          an object of class UPb
tlim       age limits of the concordia line
alpha      probability cutoff for the error ellipses and confidence intervals
type       one of
            1: Wetherill – \(^{206}\text{Pb}/^{238}\text{U}\) vs. \(^{207}\text{Pb}/^{235}\text{U}\)
            2: Tera-Wasserburg – \(^{207}\text{Pb}/^{206}\text{Pb}\) vs. \(^{238}\text{U}/^{206}\text{Pb}\)
            3: U-Th-Pb concordia – \(^{208}\text{Pb}/^{232}\text{Th}\) vs. \(^{206}\text{Pb}/^{238}\text{U}\) (only available if x$format=7 or 8)
show.numbers logical flag (TRUE to show grain numbers)
levels     a vector with length(x) values to be displayed as different background colours within the error ellipses.
clabel     label for the colour legend (only used if levels is not NA).
ellipse.fill Fill colour for the error ellipses. This can either be a single colour or multiple
colours to form a colour ramp. Examples:
            a single colour: rgb(0,1,0,0.5), '#FF000080', 'white', etc.;
            multiple colours: c(rgb(1,0,0,0.5),rgb(0,1,0,0.5)), c('#FF000080','#00FF0080'),
                         c('blue','red'), c('blue','yellow','red'), etc.;
            a colour palette: rainbow(n=100), topo.colors(n=100, alpha=0.5), etc.; or
            a reversed palette: rev(topo.colors(n=100, alpha=0.5)), etc.
            For empty ellipses, set ellipse.fill=NA
ellipse.stroke the stroke colour for the error ellipses. Follows the same formatting guidelines
            as ellipse.fill
concordia.col colour of the concordia line
exterr     show decay constant uncertainties?
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 \(^{207}\text{Pb}/^{206}\text{Pb}\)-ratio intercept (for Tera-Wasserburg). If mswd>0, then
               the analytical uncertainties are augmented by a factor \(\sqrt{\text{mswd}}\).
            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  common lead correction:
0: none
1: use the Pb-composition stored in
  settings('iratio', 'Pb207Pb206') (if x$format<4);
  settings('iratio', 'Pb206Pb204') and settings('iratio', 'Pb207Pb204')
  (if 3<x$format<7); or
  settings('iratio', 'Pb208Pb206') and settings('iratio', 'Pb208Pb207')
  (if x$format>6).
2: use the isochron intercept as the initial Pb-composition
3: use the Stacey-Kramers two-stage model to infer the initial Pb-composition.

ticks  either a scalar indicating the desired number of age ticks to be placed along the
       concordia line, OR a vector of tick ages.

anchor  control parameters to fix the intercept age or common Pb composition of the
        isochron fit. This can be a scalar or a vector.
        If anchor[1]=0: do not anchor the isochron.
        If anchor[1]=1: fix the common Pb composition at the values stored in settings('iratio', ...).
        If anchor[1]=2: force the isochron line to intersect the concordia line at an age
                      equal to anchor[2].

hide  vector with indices of aliquots that should be removed from the concordia dia-
       gram

omit  vector with indices of aliquots that should be plotted but omitted from concordia
       or discordia age calculation

omit.fill  fill colour that should be used for the omitted aliquots.

omit.stroke  stroke colour that should be used for the omitted aliquots.

...  optional arguments to the generic plot function

Details

The concordia diagram is a graphical means of assessing the internal consistency of U-Pb data. It
sets out the measured $^{206}\text{Pb}/^{238}\text{U}$- and $^{207}\text{Pb}/^{235}\text{U}$-ratios against each other (‘Wetherill’ diagram); or,
equivalently, the $^{207}\text{Pb}/^{206}\text{Pb}$- and $^{206}\text{Pb}/^{238}\text{U}$-ratios (‘Tera-Wasserburg’ diagram). Alternatively, for data format 7 and 8, it is also possible to plot $^{208}\text{Pb}/^{232}\text{Th}$ against the $^{206}\text{Pb}/^{238}\text{U}$. 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.

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 (weighted 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) 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** a three-element vector with the number of degrees of freedom used for the mswd calculation.

**age** a 4-element vector with:
- t: the concordia age (in Ma)
- s[t]: the estimated uncertainty of t
- ci[t]: the studentised 100(1 − α)% confidence interval of t for the appropriate degrees of freedom
- disp[t]: the studentised 100(1 − α)% confidence interval for t augmented by $\sqrt{\text{mswd}}$ to account for overdispersed datasets.

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

**model** the fitting model (=show.age-1).

**par** a vector with the upper and lower intercept ages (if type=1) or the lower intercept age and common Pb intercept(s) (if type=2). If show.age=4, includes an overdispersion term as well.

**cov** the covariance matrix of the elements in par.

**logpar** the logarithm of par

**logcov** the logarithm of cov

**err** a matrix with the following rows:
- s: the estimated standard deviation for x
- ci: the studentised 100(1 − α)% confidence interval of x for the appropriate degrees of freedom
- disp[t]: the studentised 100(1 − α)% confidence interval for x augmented by $\sqrt{\text{mswd}}$ to account for overdispersed datasets (only reported if show.age=2).

**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 show.age=3).

**fact** the $(1 − \alpha/2)$-percentile of a t-distribution with df degrees of freedom.

**n** the number of aliquots in the dataset

**References**

Examples

```r
attach(examples)
concordia(UPb, show.age=2)

dev.new()
concordia(UPb, type=1, xlim=c(24.9, 25.4),
         ylim=c(0.0508, 0.0518), ticks=249:254, exterr=TRUE)

dev.new()
concordia(UPb, type=2, show.age=2, anchor=list(TRUE, 0))
```

---

**data2york**

Prepare geochronological data for York regression

**Description**

Takes geochronology data as input and produces a five-column table with the variables, their uncertainties and error correlations as output. These can subsequently be used for York regression.

**Usage**

```r
data2york(x, ...)
```

## Default S3 method:
```r
data2york(x, format = 1, ...)
```

## S3 method for class 'UPb'
```r
data2york(x, option = 1, tt = 0, ...)
```

## S3 method for class 'ArAr'
```r
data2york(x, inverse = TRUE, ...)
```

## S3 method for class 'ThPb'
```r
data2york(x, inverse = FALSE, ...)
```

## S3 method for class 'KCa'
```r
data2york(x, inverse = FALSE, ...)
```

## S3 method for class 'PbPb'
```r
data2york(x, inverse = TRUE, ...)
```

## S3 method for class 'PD'
```r
data2york(x, exterr = FALSE, inverse = FALSE, ...)
```

## S3 method for class 'UThHe'
```r
data2york(x, ...)
```
## S3 method for class 'ThU'

`data2york(x, type = 2, generic = TRUE, ...)`

### Arguments

- **x**
  - A five or six column matrix OR an object of class UPb, PbPb, ThPb, ArAr, ThU, UTHHe, or PD (which includes objects of class RbSr, SmNd, LuHf and ReOs), generated by the `read.data(...)` function.

- **...**
  - Optional arguments.

- **format**
  - One of:
    1. `1` or `2`: $X = 07/35$, $s[X] = s[07/35]$, $Y = 06/38$, $s[Y] = s[06/38]$, rho = $r_{XY}$.
    2. `3`: $X = 38/06$, $s[X] = s[38/06]$, $Y = 07/06$, $s[Y] = s[07/06]$, rho = $r_{XY}$.
  - For which the error correlations are automatically computed from the redundancy of the three ratios.

- **option**
  - One of:
    1. `1`: Wetherill concordia ratios: $X = 07/35$, $s[X]$ = $s[07/35]$, $Y = 06/38$, $s[Y]$ = $s[06/38]$, $r_{XY}$.
    2. `2`: Tera-Wasserburg ratios: $X = 38/06$, $s[X]$ = $s[38/06]$, $Y = 07/06$, $s[Y]$ = $s[07/06]$, rho = $r_{XY}$.
    3. `3`: $X = 35/07$, $s[X]$ = $s[35/07]$, $Y = 08/32$, $s[Y]$ = $s[08/32]$, $r_{XY}$.
    4. `4`: $X = 35/07$, $s[X]$ = $s[35/07]$, $Y = 04/07$, $s[Y]$ = $s[04/07]$, rho = $r_{XY}$.
    5. `5`: U-Th-Pb concordia ratios: $X = 06/38$, $s[X]$ = $s[06/38]$, $Y = 08/32$, $s[Y]$ = $s[08/32]$, $r_{XY}$.
    6. `6`: $X = 35/07$, $s[X]$ = $s[35/07]$, $Y = 08/07$, $s[Y]$ = $s[08/07]$, $r_{XY}$.
    7. `7`: $X = 32/08$, $s[X]$ = $s[32/08]$, $Y = 08/32$, $s[Y]$ = $s[08/32]$, $r_{XY}$.
    8. `8`: $X = 32/08$, $s[X]$ = $s[32/08]$, $Y = 07/08$, $s[Y]$ = $s[07/08]$, $r_{XY}$.
    9. `9`: $X = 32/08$, $s[X]$ = $s[32/08]$, $Y = 08/07$, $s[Y]$ = $s[08/07]$, $r_{XY}$.

- **tt**
  - The age of the sample. This is only used if x$format = 7 or 8, in order to calculate the inherited $^{208}$Pb/$^{232}$Th ratio.

- **inverse**
  - Toggles between normal and inverse isochron ratios. `data2york` returns five columns $X$, $s[X]$, $Y$, $s[Y]$ and $r[X,Y]$.

If `inverse = TRUE`, then $X = ^{204}$Pb/$^{206}$Pb and $Y = ^{207}$Pb/$^{206}$Pb (if x has class PbPb), or $X = ^{232}$Th/$^{208}$Pb and $Y = ^{204}$Pb/$^{208}$Pb (if x has class ThPb), or $X = ^{39}$Ar/$^{40}$Ar and $Y = ^{36}$Ar/$^{40}$Ar (if x has class ArAr), or $X = ^{40}$K/$^{40}$Ca and $Y = ^{44}$Ca/$^{40}$Ca (if x has class KCa), or $X = ^{87}$Rb/$^{87}$Sr and $Y = ^{86}$Sr/$^{87}$Sr (if x has class RbSr), or $X = ^{147}$Sm/$^{143}$Nd and $Y = ^{144}$Nd/$^{143}$Nd (if x has class SmNd), or $X = ^{187}$Re/$^{187}$Os and $Y = ^{188}$Os/$^{187}$Os (if x has class ReOs), or $X = ^{176}$Lu/$^{176}$Hf and $Y = ^{177}$Hf/$^{176}$Hf (if x has class LuHf).
If `inverse=FALSE`, then $X = \frac{206}{204}Pb/\frac{204}{204}Pb$ (if x has class PbPb), or $X = \frac{232}{204}Th/\frac{204}{204}Pb$ and $Y = \frac{208}{204}Pb/\frac{204}{204}Pb$ (if x has class ThPb), or $X = \frac{39}{46}Ar/\frac{46}{46}Ar$ and $Y = \frac{40}{46}Ar/\frac{46}{46}Ar$ (if x has class ArAr), or $X = \frac{40}{44}K/\frac{44}{44}Ca$ and $Y = \frac{40}{44}Ca/\frac{44}{44}Ca$ (if x has class KCa), or $X = \frac{87}{86}Rb/\frac{86}{86}Sr$ and $Y = \frac{87}{86}Sr/\frac{86}{86}Sr$ (if x has class RbSr), or $X = \frac{147}{144}Sm/\frac{144}{144}Nd$ and $Y = \frac{143}{144}Nd/\frac{144}{144}Nd$ (if x has class SmNd), or $X = \frac{187}{188}Re/\frac{188}{188}Os$ and $Y = \frac{187}{188}Os/\frac{188}{188}Os$ (if x has class ReOs), or $X = \frac{176}{177}Lu/\frac{177}{177}Hf$ and $Y = \frac{176}{177}Hf/\frac{177}{177}Hf$ (if x has class LuHf).

- `exterr` If TRUE, propagates the external uncertainties (e.g. decay constants) into the output errors.
- `type` Return 'Rosholt' or 'Osmond' ratios?
  - Rosholt (type=1) returns $X=8/2$, $sX=s[8/2]$, $Y=0/2$, $sY=s[0/2]$, $rXY$.
  - Osmond (type=2) returns $X=2/8$, $sX=s[2/8]$, $Y=0/8$, $sY=s[0/8]$, $rXY$.
- `generic` If TRUE, uses the following column headers: X, sX, Y, sY, rXY.
  - If FALSE and type=1, uses U238/Th232, errU238/Th232, Th230/Th232, errTh230/Th232, rho
  - or if FALSE and type=2, uses Th232/U238, errTh232/U238, Th230/U238, errTh230/U238, rho.

**Value**

A five-column table that can be used as input for `york-regression`.

**See Also**

`york`

**Examples**

```r
f <- system.file("RbSr1.csv", package="IsoplotR")
dat <- read.csv(f)
yorkdat <- data2york(dat)
fit <- york(yorkdat)
```

**Description**

Define a discordance cutoff to filter U–Pb data.

**Usage**

```r
discfilter(option = 0, before = TRUE, cutoff)
```

---

**discfilter**

Set up a discordance filter

**Description**

Define a discordance cutoff to filter U–Pb data.

**Usage**

```r
 discfilter(option = 0, before = TRUE, cutoff)
```
discfilter

Arguments

<table>
<thead>
<tr>
<th>option</th>
</tr>
</thead>
<tbody>
<tr>
<td>option</td>
</tr>
<tr>
<td>one of five options:</td>
</tr>
<tr>
<td>0: do not apply a discordance filter</td>
</tr>
<tr>
<td>1 or 't': the absolute age difference (Ma) between the ( {^{206}\text{Pb}}/{^{238}\text{U}} ) and ( {^{207}\text{Pb}}/{^{206}\text{Pb}} ) ages.</td>
</tr>
<tr>
<td>2 or 'r': the relative age difference (( {^{206}\text{Pb}}/{^{238}\text{U}} ) and ( {^{207}\text{Pb}}/{^{206}\text{Pb}} ) ages.</td>
</tr>
<tr>
<td>3 or 'sk': percentage of common Pb measured along a mixing line connecting the measured composition and the Stacey-Kramers mantle composition in Tera-Wasserburg space.</td>
</tr>
<tr>
<td>4 or 'a': logratio distance (perpendicular line connecting Tera-Wasserburg concordia and the measured composition.</td>
</tr>
<tr>
<td>5 or 'c': logratio distance (connecting the measured composition and the corresponding single grain concordia age composition.</td>
</tr>
</tbody>
</table>


before logical flag indicating whether the discordance filter should be applied before (TRUE) or after (FALSE) the common-Pb correction.

cutoff a two-element vector with the minimum (negative) and maximum (positive) allowed discordance. Default values vary between the different options. To view them, enter discfilter(option) at the R command line.

Details

The most reliable U–Pb age constraints are obtained from (zircon) grains whose \( {^{206}\text{Pb}}/{^{238}\text{U}} \) and \( {^{207}\text{Pb}}/{^{206}\text{Pb}} \) ages are statistically indistinguishable from each other. U–Pb compositions that fulfil this requirements are called ‘concordant’. Those that violate it are called ‘discordant’. The discordance of the \( {^{206}\text{Pb}}/{^{238}\text{U}} \) and \( {^{207}\text{Pb}}/{^{206}\text{Pb}} \) systems can be defined in five different ways. By setting a cutoff for any of these criteria, U–Pb data can be filtered for data quality.

Value

a list with the input parameters. Default values for cutoff are

c(-50,140) if option=='t';
c(-5,15) if option=='r';
c(-0.3,1) if option=='sk';
c(-2,6) if option=='a'; and

c(-2,7) if option=='c'.

References


See Also
cad, kde, radialplot
Examples

dscf <- discfilter(option='c',before=TRUE,cutoff=c(-1,1))
weightedmean(x=examples$UPb,exterr=FALSE,sigdig=2,
cutoff.disc=dscf,common.Pb=3)

diseq

Set up U-series disequilibrium correction for U-Pb geochronology

Description

The U-Pb method conventionally assumes initial secular equilibrium of all the intermediate daughters of the $^{238}$U-$^{206}$Pb and $^{235}$U-$^{207}$Pb decay chains. Violation of this assumption may produce inaccurate results. diseq sets up initial disequilibrium parameters that are subsequently passed on to the read.data function for incorporation in other functions.

Usage

diseq(
  U48 = list(x = 1, sx = 0, option = 0),
  ThU = list(x = 1, sx = 0, option = 0),
  RaU = list(x = 1, sx = 0, option = 0),
  PaU = list(x = 1, sx = 0, option = 0)
)

Arguments

U48 a list containing three items (x, sx and option) specifying the $^{234}$U/$^{238}$U disequilibrium.
If option=0, then x and sx are ignored and no disequilibrium correction is applied.
If option=1, then x contains the initial $^{234}$U/$^{238}$U ratio and sx its standard error.
If option=2, then x contains the measured $^{234}$U/$^{238}$U ratio and sx its standard error.

ThU a list containing three items (x, sx and option) specifying the $^{230}$Th/$^{238}$U disequilibrium.
If option=0, then x and sx are ignored and no disequilibrium correction is applied.
If option=1, then x contains the initial $^{230}$Th/$^{238}$U ratio and sx its standard error.
If option=2, then x contains the measured $^{230}$Th/$^{238}$U ratio and sx its standard error.
If option=3, then x contains the measured Th/U ratio of the magma (assumed or determined from the whole rock or volcanic glass) and sx its standard error. This only applies for Th-bearing U-Pb data formats 7 and 8.
RaU  a list containing three items (x, sx and option) specifying the $^{226}$Ra/$^{238}$U disequilibrium.
  If option=0, then x and sx are ignored and no disequilibrium correction is applied.
  If option=1, then x contains the initial $^{226}$Ra/$^{238}$U ratio and sx its standard error.

PaU  a list containing three items (x, sx and option) specifying the $^{231}$Pa/$^{235}$U disequilibrium.
  If option=0, then x and sx are ignored and no disequilibrium correction is applied.
  If option=1, then x contains the initial $^{231}$Pa/$^{235}$U ratio and sx its standard error.

Details

There are three ways to correct for the initial disequilibrium between the activity of $^{238}$U, $^{234}$U, $^{230}$Th, and $^{226}$Ra; or between $^{235}$U and $^{231}$Pa:

1. Specify the assumed initial activity ratios and calculate how much excess $^{206}$Pb and $^{207}$Pb these would have produced.
2. Measure the current activity ratios to infer the initial ratios. This approach only works for young samples.
3. The initial $^{230}$Th/$^{238}$U activity ratio can also be estimated by providing the Th/U-ratio of the magma.

Value

a list with the following items:

U48, ThU, RaU, PaU  the same as the corresponding input arguments
equilibrium  a boolean flag indicating whether option=TRUE and/or x=1 for all activity ratios
Q  the eigenvectors of the disequilibrium matrix exponential
Qinv  the inverse of Q
L  a named vector of all the relevant decay constants
n0  the initial atomic abundances of all the parent and daughter isotopes (used by mclean)

See Also

mclean, concordia, ludwig

Examples

d <- diseq(U48=list(x=0,option=1),ThU=list(x=2,option=1),
  RaU=list(x=2,option=1),PaU=list(x=2,option=1))
fn <- system.file("diseq.csv",package="IsoplotR")
UPb <- read.data(fn,method="U-Pb",format=2,d=d)
concordia(UPb,type=2,xlim=c(0,5000),ylim=c(0.047,0.057))
ellipse

Description

Constructs an error ellipse at a given confidence level from its centre and covariance matrix.

Usage

\texttt{ellipse(x, y, covmat, alpha = 0.05, n = 50)}

Arguments

\begin{itemize}
  \item \texttt{x} \, \text{x-coordinate (scalar) for the centre of the ellipse}
  \item \texttt{y} \, \text{y-coordinate (scalar) for the centre of the ellipse}
  \item \texttt{covmat} \, \text{the} \, [2x2] \, \text{covariance matrix of the x-y coordinates}
  \item \texttt{alpha} \, \text{the probability cutoff for the error ellipses}
  \item \texttt{n} \, \text{the resolution (number of segments) of the error ellipses}
\end{itemize}

Value

\text{an} \, [n \times 2] \, \text{matrix of plot coordinates}

Examples

\begin{verbatim}
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')
\end{verbatim}

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

```r
definition

Arguments

x
xlim
ylim
alpha
transform
detritus
show.numbers
levels
clabel
ellipse.fill
eclipse.stroke
line.col
isochron
model
exterr
sigdig
hide
omit
omit.fill
omit.stroke
...
multiple colours: c(rgb(1,0,0,.5), rgb(0,1,0,.5)), c('#FF000080', '#00FF0080'), c('blue', 'red'), c('blue', 'yellow', 'red'), etc.;
a colour palette: rainbow(n=100), topo.colors(n=100, alpha=0.5), etc.; or
a reversed palette: rev(topo.colors(n=100, alpha=0.5)), etc.
For empty ellipses, set ellipse.fill=NA

ellips.stroke the stroke colour for the error ellipses. Follows the same formatting guidelines as ellipse.fill

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 this 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 \(\sqrt{MSWD}\).
2: ordinary least squares regression: a second way to deal with over- or under-dispersed 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
hide vector with indices of aliquots that should be removed from the plot.
omit vector with indices of aliquots that should be plotted but omitted from the isochron age calculation.

omit.fill fill colour that should be used for the omitted aliquots.
omit.stroke stroke colour that should be used for the omitted aliquots.

Details

Similar to the concordia diagram (for U-Pb data) and the helioplot diagram (for U-Th-He data), the evolution diagram simultaneously displays the isotopic composition and age of U-series data. For carbonate data (Th-U formats 1 and 2), the Th-U evolution diagram consists of a scatter plot that sets out the \(^{234}\text{U}/^{238}\text{U}\)-activity ratios against the \(^{230}\text{Th}/^{238}\text{U}\)-activity ratios as error ellipses, and displays the initial \(^{234}\text{U}/^{238}\text{U}\)-activity ratios and ages as a set of intersecting lines. Alternatively, the \(^{234}\text{U}/^{238}\text{U}\)-ratios can also be set out against the \(^{230}\text{Th}/^{234}\text{U}/^{238}\text{U}\)-ages. In both types of evolution diagrams, IsoplotR provides the option to project the raw measurements along the best fitting
isochron line and thereby remove the detrital $^{230}$Th-component. This procedure allows a visual assessment of the degree of homogeneity within a dataset, as is quantified by the MSWD.

Neither the U-series evolution diagram, nor the $^{234}$U/$^{238}$U vs. age plot is applicable to igneous datasets (Th-U formats 3 and 4), in which $^{234}$U and $^{238}$U are in secular equilibrium. For such datasets, IsoplotR produces an Osmond-style regression plot that is decorated with a fanning set of isochron lines.

References


See Also

isochron

Examples

attach(examples)
evolution(ThU)

dev.new()
evolution(ThU,transform=TRUE,
isochron=TRUE,model=1)

Description

Built-in datasets for U-Pb, Pb-Pb, Ar-Ar, K-Ca, Re-Os, Sm-Nd, Rb-Sr, Lu-Hf, U-Th-He, Th-U, fission track and detrital geochronology.

examples is 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 (2003)'s Isoplot program.
PbPb: an object of class PbPb containing a Pb-Pb dataset from Connelly et al. (2017).
ThPb: an object of class ThPb containing the Th-Pb data for allanite sample MF482 of Janots and Rubatto (2014).
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).
examples

**KCa**: an object of class KCa containing a $^{40}$K/$^{40}$Ca dataset for sample 140025 grain h spot 5 of Harrison et al. (2010).

**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).

**MountTom**: an object of class other containing a collection of zircon fission track ages and errors from Brandon (1992).

**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**


**Examples**

```r
attach(examples)
concordia(UPb)
agespectrum(ArAr)
iscochron(ReOs)
radialplot(FT1)
helioplot(UTHHe)
evolution(ThU)
kde(DZ)
radialplot(LudwigMixture)
agespectrum(LudwigSpectrum)
weightedmean(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,
  clabel = "",
  ellipse.fill = c("#00FF0080", "#0000FF80"),
  ellipse.stroke = "black",
  sigdig = 2,
  xlim = NA,
  ylim = NA,
  fact = NA,
  hide = NULL,
  omit = NULL,
  omit.fill = NA,
  omit.stroke = "grey",
  ...
)

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] diagram, or a U-Th-He ternary diagram.
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 $\sqrt{\text{MSWD}}$.
2: unweighted mean. A second way to deal with over- or underdispersed datasets is to simply ignore the analytical uncertainties.
helioplot

3: weighted mean with overdispersion: instead of attributing any overdispersion (MSWD > 1) to underestimated analytical uncertainties (model 1), it can also be attributed 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.
clabel label of the colour scale
ellipse.fill Fill colour for the error ellipses. This can either be a single colour or multiple colours to form a colour ramp. Examples:
   a single colour: rgb(0,1,0,0.5), '#FF000080', 'white', etc.;
   multiple colours: c(rgb(1,0,0,0.5),rgb(0,1,0,0.5)), c('#FF000080', '#00FF0080'), c('blue','red'), c('blue','yellow','red'), etc.;
   a colour palette: rainbow(n=100), topo.colors(n=100, alpha=0.5), etc.; or a reversed palette: rev(topo.colors(n=100, alpha=0.5)), etc.;
   For empty ellipses, set ellipse.fill=NA
ellipse.stroke the stroke colour for the error ellipses. Follows the same formatting guidelines as ellipse.fill
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 scaling factors of the ternary diagram if fact=NA, these will be determined automatically
hide vector with indices of aliquots that should be removed from the plot.
omit vector with indices of aliquots that should be plotted but omitted from the central age calculation.
omit.fill fill colour that should be used for the omitted aliquots.
omit.stroke stroke colour that should be used for the omitted aliquots.
...
optional arguments to the generic plot function

Details
U, Th, Sm and He are compositional data. This means that it is not so much the absolute concentrations of these elements that bear the chronological information, but rather their relative proportions. The space of all possible U-Th-He compositions fits within the constraints of a ternary diagram
or ‘helioplot’ (Vermeesch, 2008, 2010). If Sm is included as well, then this expands to a three-dimensional tetrahedral space (Vermeesch, 2008). Data that fit within these constrained spaces must be subjected to a logratio transformation prior to statistical analysis (Aitchison, 1986). In the case of the U-Th-He-(Sm)-He system, this is achieved by first defining two (or three) new variables:
\[ u \equiv \ln[U/He] \quad v \equiv \ln[Th/He] \quad w \equiv \ln[Sm/He] \]
and then performing the desired statistical analysis (averaging, uncertainty propagation, ...) on the transformed data. Upon completion of the mathematical operations, the results can then be mapped back to U-Th-(Sm)-He space using an inverse logratio transformation:
\[
\begin{align*}
[He] &= 1/[e^u + e^v + (e^w + 1)] \\
[U] &= e^u/[e^u + e^v + (e^w + 1)] \\
[Th] &= e^v/[e^u + e^v + (e^w + 1)] \\
[Sm] &= e^w/[e^u + e^v + (e^w + 1)]
\end{align*}
\]
where \([He] + [U] + [Th](+|Sm|) = 1\). In the context of U-Th-(Sm)-He dating, the central age is defined as the age that corresponds to the arithmetic mean composition in logratio space, which is equivalent to the geometric mean in compositional dataspace (Vermeesch, 2008). IsoplotR’s helioplot function performs this calculation using the same algorithm that is used to obtain the weighted mean U-Pb composition for the concordia age calculation. Overdispersion is treated similarly as in a regression context (see isochron). Thus, there are options to augment the uncertainties with a factor \(\sqrt{MSWD}\) (model 1); to ignore the analytical uncertainties altogether (model 2); or to add a constant overdispersion term to the analytical uncertainties (model 3). The helioplot function visualises U-Th-(Sm)-He data on either a ternary diagram or a bivariate \(\ln[Th/U] \) vs. \(\ln[U/He]\) contour plot. These diagrams provide a convenient way to simultaneously display the isotopic composition of samples as well as their chronological meaning. In this respect, they fulfil the same purpose as the U-Pb concordia diagram and the U-series evolution plot.

References


See Also

radialplot

Examples

attach(examples)
helioplot(UThHe)
dev.new()
helioplot(UThHe, logratio=FALSE)
isochron

Calculate and plot isochrons

Description

Plots cogenetic U-Pb, Ar-Ar, K-Ca, Pb-Pb, Th-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, titterington or ludwig function, and computes the corresponding isochron age, including decay constant uncertainties.

Usage

isochron(x, ...)

## Default S3 method:

isochron(
  x,
  alpha = 0.05,
  sigdig = 2,
  show.numbers = FALSE,
  levels = NA,
  clabel = "",
  xlab = "x",
  ylab = "y",
  ellipse.fill = c("#00FF0080", "#FF000080"),
  ellipse.stroke = "black",
  ci.col = "gray80",
  line.col = "black",
  lwd = 1,
  plot = TRUE,
  title = TRUE,
  model = 1,
  show.ellipses = 1 * (model != 2),
  hide = NULL,
  omit = NULL,
  omit.fill = NA,
  omit.stroke = "grey",
  ...
)

## S3 method for class 'UPb'

isochron(
  x,
  alpha = 0.05,
  sigdig = 2,
  show.numbers = FALSE,
  levels = NA,
isochron

clabel = "", ellipse.fill = c("#00FF0080", "#FF000080"), ellipse.stroke = "black", type = 1, 
  ci.col = "gray80", line.col = "black", lwd = 1, plot = TRUE, exterr = FALSE, 
  model = 1, show.ellipses = 1 * (model != 2), anchor = 0, hide = NULL, 
  omit = NULL, omit.fill = NA, omit.stroke = "grey", 
  ... 
)

## S3 method for class 'PbPb'
isochron(
x, alpha = 0.05, sigdig = 2, show.numbers = FALSE, 
  levels = NA, clabel = "", ellipse.fill = c("#00FF0080", "#FF000080"), ellipse.stroke = "black", 
  inverse = TRUE, ci.col = "gray80", line.col = "black", lwd = 1, plot = TRUE, exterr = TRUE, 
  model = 1, growth = FALSE, show.ellipses = 1 * (model != 2), hide = NULL, 
  omit = NULL, omit.fill = NA, omit.stroke = "grey", 
  ...
)

## S3 method for class 'ArAr'
isochron(
x,
alpha = 0.05,
sigdig = 2,
show.numbers = FALSE,
levels = NA,
clabel = "",
ellipse.fill = c("#00FF0080", "#FF000080"),
ellipse.stroke = "black",
inverse = TRUE,
clabel = "gray80",
line.col = "black",
lwd = 1,
plot = TRUE,
exterr = TRUE,
model = 1,
show.ellipses = 1 * (model != 2),
hide = NULL,
omit = NULL,
omit.fill = NA,
omit.stroke = "grey",
...
)

## S3 method for class 'ThPb'

isochron(
  x,
  alpha = 0.05,
  sigdig = 2,
  show.numbers = FALSE,
  levels = NA,
  clabel = "",
  ellipse.fill = c("#00FF0080", "#FF000080"),
  ellipse.stroke = "black",
  inverse = TRUE,
  clabel = "gray80",
  line.col = "black",
  lwd = 1,
  plot = TRUE,
  exterr = TRUE,
  model = 1,
  show.ellipses = 1 * (model != 2),
  hide = NULL,
  omit = NULL,
  omit.fill = NA,
  omit.stroke = "grey",
  ...
)

## S3 method for class 'KCa'
isochron

isochron(
  x,
  alpha = 0.05,
  sigdig = 2,
  show.numbers = FALSE,
  levels = NA,
  clabel = "",
  inverse = FALSE,
  ci.col = "gray80",
  ellipse.fill = c("#00FF0080", "#FF000080"),
  ellipse.stroke = "black",
  line.col = "black",
  lwd = 1,
  plot = TRUE,
  exterr = TRUE,
  model = 1,
  show.ellipses = 1 * (model != 2),
  hide = NULL,
  omit = NULL,
  omit.fill = NA,
  omit.stroke = "grey",
  ...
)

## S3 method for class 'RbSr'
isochron(
  x,
  alpha = 0.05,
  sigdig = 2,
  show.numbers = FALSE,
  levels = NA,
  clabel = "",
  ellipse.fill = c("#00FF0080", "#FF000080"),
  ellipse.stroke = "black",
  inverse = FALSE,
  ci.col = "gray80",
  line.col = "black",
  lwd = 1,
  plot = TRUE,
  exterr = TRUE,
  model = 1,
  show.ellipses = 1 * (model != 2),
  hide = NULL,
  omit = NULL,
  omit.fill = NA,
  omit.stroke = "grey",
  ...
)
## S3 method for class 'ReOs'

```r
isochron(
  x,
  alpha = 0.05,
  sigdig = 2,
  show.numbers = FALSE,
  levels = NA,
  clabel = "",
  ellipse.fill = c("#00FF0080", "#FF000080"),
  ellipse.stroke = "black",
  inverse = FALSE,
  ci.col = "gray80",
  line.col = "black",
  lwd = 1,
  plot = TRUE,
  exterr = TRUE,
  model = 1,
  show.ellipses = 1 * (model != 2),
  hide = NULL,
  omit = NULL,
  omit.fill = NA,
  omit.stroke = "grey",
```

## S3 method for class 'SmNd'

```r
isochron(
  x,
  alpha = 0.05,
  sigdig = 2,
  show.numbers = FALSE,
  levels = NA,
  clabel = "",
  ellipse.fill = c("#00FF0080", "#FF000080"),
  ellipse.stroke = "black",
  inverse = FALSE,
  ci.col = "gray80",
  line.col = "black",
  lwd = 1,
  plot = TRUE,
  exterr = TRUE,
  model = 1,
  show.ellipses = 1 * (model != 2),
  hide = NULL,
  omit = NULL,
  omit.fill = NA,
  omit.stroke = "grey",
```
isochron

## S3 method for class 'LuHf'
isochron(
  x,
  alpha = 0.05,
  sigdig = 2,
  show.numbers = FALSE,
  levels = NA,
  clabel = "",
  ellipse.fill = c("#00FF0080","#FF000080"),
  ellipse.stroke = "black",
  inverse = FALSE,
  ci.col = "gray80",
  line.col = "black",
  lwd = 1,
  plot = TRUE,
  exterr = TRUE,
  model = 1,
  show.ellipses = 1 * (model != 2),
  hide = NULL,
  omit = NULL,
  omit.fill = NA,
  omit.stroke = "grey",
  ...
)

## S3 method for class 'ThU'
isochron(
  x,
  type = 2,
  alpha = 0.05,
  sigdig = 2,
  show.numbers = FALSE,
  levels = NA,
  clabel = "",
  ellipse.fill = c("#00FF0080","#FF000080"),
  ellipse.stroke = "black",
  ci.col = "gray80",
  line.col = "black",
  lwd = 1,
  plot = TRUE,
  exterr = TRUE,
  model = 1,
  show.ellipses = 1 * (model != 2),
  hide = NULL,
  omit = NULL,
omit.fill = NA,
omit.stroke = "grey",
...
)

## S3 method for class 'UThHe'
isochron(
x,
alpha = 0.05,
sigdig = 2,
show.numbers = FALSE,
levels = NA,
clabel = "",
ellipse.fill = c("#00FF0080", "#FF000080"),
ellipse.stroke = "black",
ci.col = "gray80",
line.col = "black",
lwd = 1,
plot = TRUE,
model = 1,
show.ellipses = 2 * (model != 2),
hide = NULL,
omit = NULL,
omit.fill = NA,
omit.stroke = "grey",
...
)

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, KCa, PbPb, ThPb, ReOs, RbSr, SmNd, LuHf, UThHe or ThU.

... optional arguments to be passed on to the generic plot function if model=2

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.
clabel | label for the colour scale
xlab | text label for the horizontal plot axis
ylab | text label for the vertical plot axis
ellipse.fill | Fill colour for the error ellipses. This can either be a single colour or multiple colours to form a colour ramp. Examples:
a single colour: rgb(0,1,0,0.5), '#FF000080', 'white', etc.;
multiple colours: c(rgb(1,0,0,0.5),rgb(0,1,0,0.5)), c('#FF000080','#00FF0080'),
c('blue','red'), c('blue','yellow','red'), etc.;
a colour palette: rainbow(n=100), topo.colors(n=100,alpha=0.5), etc.; or a reversed palette: rev(topo.colors(n=100,alpha=0.5)), etc.

For empty ellipses, set ellipse.col=NA

ellipse.stroke | the stroke colour for the error ellipses. Follows the same formatting guidelines as ellipse.fill

.ci.col | the fill colour for the confidence interval of the intercept and slope.

line.col | colour of the isochron line

lwd | line width

plot | if FALSE, suppresses the graphical output

.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

show.ellipses | show the data as:
0: points
1: error ellipses
2: error crosses

hide | vector with indices of aliquots that should be removed from the plot.

omit | vector with indices of aliquots that should be plotted but omitted from the isochron age calculation.

omit.fill | fill colour that should be used for the omitted aliquots.

omit.stroke | stroke colour that should be used for the omitted aliquots.

type | if x has class UPb and x$format=4, 5 or 6:
1: $^{204}$Pb/$^{206}$Pb vs. $^{238}$U/$^{206}$Pb
2: $^{204}$Pb/$^{207}$Pb vs. $^{235}$U/$^{207}$Pb
if x has class UPb and x$format=7 or 8:
1: $^{208}$Pb,/$^{206}$Pb vs. $^{238}$U/$^{206}$Pb
2: $^{208}$Pb,/$^{207}$Pb vs. $^{235}$U/$^{207}$Pb
3: $^{206}$Pb,/$^{208}$Pb vs. $^{232}$Th/$^{208}$Pb
4: $^{207}$Pb,/$^{208}$Pb vs. $^{232}$Th/$^{208}$Pb
if x has class ThU, and 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

---

Details

Given several aliquots from a single sample, isochrons allow the non-radiogenic component of the daughter nuclide to be quantified and separated from the radiogenic component. In its simplest form, an isochron is obtained by setting out the amount of radiogenic daughter against the amount of radioactive parent, both normalised to a non-radiogenic isotope of the daughter element, and fitting a straight line through these points by least squares regression (Nicolaysen, 1961). The slope and intercept then yield the radiogenic daughter-parent ratio and the non-radiogenic daughter composition, respectively. There are several ways to fit an isochron. The easiest of these is ordinary least squares regression, which weighs all data points equally. In the presence of quantifiable analytical uncertainty, it is equally straightforward to use the inverse of the y-errors as weights. It is significantly more difficult to take into account uncertainties in both the x- and the y-variable (York, 1966). IsoplotR does so for its U-Th-He isochron calculations. The York (1966) method assumes that the analytical uncertainties of the x- and y-variables are independent from each other. This assumption is rarely met in geochronology. York (1968) addresses this issue with a bivariate error weighted linear least squares algorithm that accounts for covariant errors in both variables. This algorithm was further improved by York et al. (2004) to ensure consistency with the maximum likelihood approach of Titterington and Halliday (1979).
IsoplotR uses the York et al. (2004) algorithm for its Ar-Ar, K-Ca, Pb-Pb, Th-Pb, Rb-Sr, Sm-Nd, Re-Os and Lu-Hf isochrons. The maximum likelihood algorithm of Titterington and Halliday (1979) was generalised from two to three dimensions by Ludwig and Titterington (1994) for U-series disequilibrium dating. Also this algorithm is implemented in IsoplotR. Finally, the constrained maximum likelihood algorithm of Ludwig (1998) is used for isochron regression of U-Pb data. The extent to which the observed scatter in the data can be explained by the analytical uncertainties can be assessed using the Mean Square of the Weighted Deviates (MSWD, McIntyre et al., 1966), which is defined as:

$$MSWD = ([X - \hat{X}]\Sigma_X^{-1}[X - \hat{X}]^T)/df$$

where $X$ are the data, $\hat{X}$ are the fitted values, and $\Sigma_X$ is the covariance matrix of $X$, and $df = k(n - 1)$ are the degrees of freedom, where $k$ is the dimensionality of the linear fit. MSWD values that are far smaller or greater than 1 indicate under- or overdispersed measurements, respectively. Underdispersion can be attributed to overestimated analytical uncertainties. IsoplotR provides three alternative strategies to deal with overdispersed data:

1. Attribute the overdispersion to an underestimation of the analytical uncertainties. In this case, the excess scatter can be accounted for by inflating those uncertainties by a factor $\sqrt{MSWD}$.
2. Ignore the analytical uncertainties and perform an ordinary least squares regression.
3. Attribute the overdispersion to the presence of ‘geological scatter’. In this case, the excess scatter can be accounted for by adding an overdispersion term that lowers the MSWD to unity.

Value

If x has class PbPb, ThPb, ArAr, KCa, 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 $^{40}\text{Ar}/^{36}\text{Ar}$ or initial $^{40}\text{Ca}/^{44}\text{Ca}$, $^{187}\text{Os}/^{188}\text{Os}$, $^{87}\text{Sr}/^{87}\text{Rb}$, $^{143}\text{Nd}/^{144}\text{Nd}$, $^{176}\text{Hf}/^{177}\text{Hf}$ or $^{208}\text{Pb}/^{204}\text{Pb}$ ratio.
  - s[y]: the propagated uncertainty of y
  - ci[y]: the studentised $100(1 - \alpha)\%$ confidence interval for y.
  - disp[y]: the studentised $100(1 - \alpha)\%$ confidence interval for y enhanced by $\sqrt{mswd}$ (only applicable if model=1).
- age a four-element list containing:
  - t: the $^{207}\text{Pb}/^{206}\text{Pb}$, $^{208}\text{Pb}/^{232}\text{Th}$, $^{40}\text{Ar}/^{39}\text{Ar}$, $^{40}\text{K}/^{40}\text{Ca}$, $^{187}\text{Os}/^{187}\text{Re}$, $^{87}\text{Sr}/^{87}\text{Rb}$, $^{143}\text{Nd}/^{144}\text{Nd}$ or $^{176}\text{Hf}/^{177}\text{Hf}$ age.
  - s[t]: the propagated uncertainty of t
  - ci[t]: the studentised $100(1 - \alpha)\%$ confidence interval for t.
  - disp[t]: the studentised $100(1 - \alpha)\%$ confidence interval for t enhanced by $\sqrt{mswd}$ (only applicable 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 (omitted if model=2)

w  the overdispersion term, i.e. a three-element vector with the standard deviation of the (assumedly) Normally distributed geological scatter that underlies the measurements, and the lower and upper half-widths of its 100(1−α)% confidence interval (only returned if model=3).

ski  (only reported if x has class PbPb and growth is TRUE) the intercept(s) of the isochron with the Stacey-Kramers mantle evolution curve.

OR, if x has class ThU:

par if x$type=1 or x$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 x$type=2 or x$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.

fact  the 100(1 − α/2)% multiplier for the confidence intervals.

y0  a four-element vector containing:
y: the initial $^{234}$U/$^{238}$U-ratio
s[y]: the propagated uncertainty of y
ci[y]: the studentised 100(1 − α)% confidence interval for y.
disp[y]: the studentised 100(1 − α)% confidence interval for y enhanced by $\sqrt{mswd}$.

age  a three (or four) element vector containing:
t: the initial $^{234}$U/$^{238}$U-ratio
s[t]: the propagated uncertainty of t
ci[t]: the studentised 100(1 − α)% confidence interval for t
disp[t]: the studentised 100(1 − α)% confidence interval for t enhanced by $\sqrt{mswd}$ (only reported if model=1).

w  the overdispersion term, i.e. a three-element vector with the standard deviation of the (assumedly) Normally distributed geological scatter that underlies the measurements, and the lower and upper half-width of its 100(1−α)% confidence interval (only returned if model=3).

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
OR if \( x \) has class UPb:

\[
\text{par} \quad \text{if model=1 or 2, a three element vector containing the isochron age and the common Pb isotope ratios. If model=3, adds a fourth element with the overdispersion parameter } w.
\]

\[
\text{cov} \quad \text{the covariance matrix of par}
\]

\[
\text{logpar} \quad \text{the logarithm of par}
\]

\[
\text{logcov} \quad \text{the logarithm of cov}
\]

\[
n \quad \text{the number of analyses in the dataset}
\]

\[
df \quad \text{the degrees of freedom for the linear fit, i.e. } 2n - 3
\]

\[
a \quad \text{the y-intercept and its standard error}
\]

\[
b \quad \text{the isochron slope and its standard error}
\]

\[
\text{cov.ab} \quad \text{the covariance between } a \text{ and } b.
\]

\[
\text{mswd} \quad \text{the mean square of the residuals (a.k.a 'reduced Chi-square') statistic.}
\]

\[
\text{p.value} \quad \text{the p-value of a Chi-square test for linearity.}
\]

\[
\text{fact} \quad \text{the } 100(1 - \alpha/2)\% \text{ multiplier for the confidence intervals.}
\]

\[
y0 \quad \text{a three or four-element vector containing:}
\]

\[
y: \quad \text{the initial } ^{206}\text{Pb}^{204}\text{Pb-ratio (if type=1 and } x$\text{format}=4,5 \text{ or } 6); ^{207}\text{Pb}^{204}\text{Pb-ratio (if type=2 and } x$\text{format}=4,5 \text{ or } 6); ^{208}\text{Pb}^{204}\text{Pb-ratio (if type=1 and } x$\text{format}=7 \text{ or } 8); ^{208}\text{Pb}^{207}\text{Pb ratio (if type=2 and } x$\text{format}=7 \text{ or } 8); ^{206}\text{Pb}^{207}\text{Pb-ratio (if type=3 and } x$\text{format}=7 \text{ or } 8); \text{ or } ^{207}\text{Pb}^{206}\text{Pb-ratio (if type=4 and } x$\text{format}=7 \text{ or } 8).
\]

\[
s[y]: \quad \text{the propagated uncertainty of } y
\]

\[
ci[y]: \quad \text{the studentised } 100(1 - \alpha)\% \text{ confidence interval for } y.
\]

\[
disp[y]: \quad \text{the studentised } 100(1 - \alpha)\% \text{ confidence interval for } y \text{ enhanced by } \sqrt{mswd} \text{ (only returned if model=1)}
\]

\[
y0\text{label} \quad \text{the y-axis label of the isochron plot}
\]

\[
age \quad \text{a three (or four) element vector containing:}
\]

\[
t: \quad \text{the isochron age}
\]

\[
s[t]: \quad \text{the propagated uncertainty of } t
\]

\[
ci[t]: \quad \text{the studentised } 100(1 - \alpha)\% \text{ confidence interval for } t
\]

\[
disp[t]: \quad \text{the studentised } 100(1 - \alpha)\% \text{ confidence interval for } t \text{ enhanced by } \sqrt{mswd} \text{ (only reported if model=1).}
\]

\[
xlab \quad \text{the x-label of the isochron plot}
\]

\[
ylab \quad \text{the y-label of the isochron plot}
\]

References


See Also

york, titterington, ludwig

Examples

attach(examples)

isochron(RbSr)

fit <- isochron(ArAr,inverse=FALSE,plot=FALSE)

dev.new()

isochron(ThU,type=4)

Description

A list of documented functions may be viewed by typing help(package='IsoplotR'). Detailed instructions are provided at https://www.ucl.ac.uk/~ucfbpve/isoplotr/. Further details about the theoretical background are provided by Vermeesch (2018).

Author(s)

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

References


See Also

Useful links:

- https://www.ucl.ac.uk/~ucfbpve/isoplotr/
- https://github.com/pvermees/IsoplotR/
**kde**

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

**Description**


**Usage**

```r
kde(x, ...)```

## Default S3 method:

```r
kde(
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
  rug = TRUE,
  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,
  hide = NULL,
  ...
)
```

## S3 method for class 'UPb'

```r
kde(
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
  rug = TRUE,
  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,
type = 4,
cutoff.76 = 1100,
cutoff.disc = discfilter(),
common.Pb = 0,
hide = NULL,
...
)

## S3 method for class 'detritals'
kde(
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
  rug = FALSE,
  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,
  hide = NULL,
  ...
)

## S3 method for class 'PbPb'
kde(
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
rug = TRUE,
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,
common.Pb = 2,
hide = NULL,
...
)

## S3 method for class 'ArAr'
kde(  
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
  rug = TRUE,
  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,
  i2i = FALSE,
  hide = NULL,
  ...
)

## S3 method for class 'KCa'
kde(  
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
  rug = TRUE,
  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,
  i2i = FALSE,
  hide = NULL,
  ...
ylab = "", kde.col = rgb(1, 0, 1, 0.6), hist.col = rgb(0, 1, 0, 0.2), show.hist = TRUE, bty = "n", binwidth = NA, i2i = FALSE, hide = NULL, ... )

## S3 method for class 'ThPb'
kde(
x, from = NA, to = NA, bw = NA, adaptive = TRUE, log = FALSE, n = 512, plot = TRUE, rug = TRUE, 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, i2i = FALSE, hide = NULL, ...
 )

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

hist.col = rgb(0, 1, 0, 0.2),
show.hist = TRUE,
bty = "n",
binwidth = NA,
i2i = FALSE,
detritus = 0,
hide = NULL,
...
)

## S3 method for class 'ReOs'
kde(
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
  rug = TRUE,
  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,
i2i = TRUE,
  hide = NULL,
  ...
)

## S3 method for class 'SmNd'
kde(
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
  rug = TRUE,
  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,
i2i = TRUE,
hide = NULL,
...
)

## S3 method for class 'RbSr'
kde(  
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
  rug = TRUE,
  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,
  i2i = TRUE,
  hide = NULL,
  ...
)

## S3 method for class 'LuHf'
kde(  
  x,
  from = NA,
  to = NA,
  bw = NA,
  adaptive = TRUE,
  log = FALSE,
  n = 512,
  plot = TRUE,
  rug = TRUE,
  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,
  i2i = TRUE,
  hide = NULL,
  ...
kde

binwidth = NA,
i2i = TRUE,
hide = NULL,
...
)

## S3 method for class 'UThHe'
kde(
x,
from = NA,
to = NA,
bw = NA,
adaptive = TRUE,
log = FALSE,
n = 512,
plot = TRUE,
rug = TRUE,
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,
hide = NULL,
...
)

## S3 method for class 'fissiontracks'
kde(
x,
from = NA,
to = NA,
bw = NA,
adaptive = TRUE,
log = FALSE,
n = 512,
plot = TRUE,
rug = TRUE,
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,
hide = NULL,
...
Arguments

x  
a vector of numbers OR an object of class UPb, PbPb, ThPb, ArAr, KCa, ReOs, SmNd, RbSr, UThHe, fissiontracks, ThU or detrital

...  
optional arguments to be passed on to R’s density function.

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 the algorithm by Botev et al. (2010).

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 (i.e., the number of segments) of the density estimate.

plot  
show the KDE as a plot

rug  
add a rug plot?

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 log = FALSE, or as a fractional value if log = TRUE. Sturges’ Rule \((\log_2[n] + 1)\) is used if binwidth = NA

hide  
vector with indices of aliquots that should be removed from the plot.

type  
scalar indicating whether to plot the \(^{207}\text{Pb}/^{235}\text{U}\) age (type=1), the \(^{206}\text{Pb}/^{238}\text{U}\) age (type=2), the \(^{207}\text{Pb}/^{206}\text{Pb}\) age (type=3), the \(^{207}\text{Pb}/^{206}\text{Pb}-^{206}\text{Pb}/^{238}\text{U}\) age (type=4), the concordia age (type=5), or the \(^{208}\text{Pb}/^{232}\text{Th}\) age (type=6).

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

cutoff.disc  
discordance cutoff filter. This is an object of class discfilter.

common.Pb  
common lead correction:
0: none
1: use the Pb-composition stored in
settings('iratio', 'Pb207Pb206') (if x has class UPb and x$format<4);
settings('iratio', 'Pb206Pb204') and settings('iratio', 'Pb207Pb204')
(if x has class PbPb or x has class UPb and 3<x$format<7); or
settings('iratio', 'Pb208Pb206') and settings('iratio', 'Pb204Pb207')
(if x has class UPb and x$format=7,8).
2: use the isochron intercept as the initial Pb-composition
3: use the Stacey-Kramers two-stage model to infer the initial Pb-composition
   (only valid if x has class UPb).

ncol scalar value indicating the number of columns over which the KDEs should be divided.
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') 40Ar/36Ar, 40Ca/44Ca, 207Pb/204Pb, 87Sr/86Sr, 143Nd/144Nd, 187Os/188Os,
   230Th/232Th, 176Hf/177Hf or 204Pb/208Pb ratio from an isochron fit. Setting i2i to FALSE uses the default values stored in settings('iratio',...).
detritus detrital 230Th correction (only applicable when x$format=1 or 2).
   0: no correction
   1: project the data along an isochron fit
   2: correct the data using an assumed initial 230Th/232Th-ratio for the detritus.
   3: correct the data using the measured present day 230Th/238U, 232Th/238U and
      234U/238U-ratios in the detritus.

Details

Given a set of n age estimates \{t_1, t_2, ..., t_n\}, histograms and KDEs are probability density estimators that display age distributions by smoothing. Histograms do this by grouping the data into a number of regularly spaced bins. Alternatively, kernel density estimates (KDEs; Vermeesch, 2012) smooth data by applying a (Gaussian) kernel:

\[
KDE(t) = \frac{1}{n} \sum_{i=1}^{n} N(t|\mu = t_i, \sigma = h[t])/n
\]

where \(N(t|\mu, \sigma)\) is the probability of observing a value \(t\) under a Normal distribution with mean \(\mu\) and standard deviation \(\sigma\). \(h[t]\) is the smoothing parameter or 'bandwidth' of the kernel density estimate, which may or may not depend on the age \(t\). If \(h[t]\) depends on \(t\), then \(KDE(t)\) is known as an 'adaptive' KDE. The default bandwidth used by IsoplotR is calculated using the algorithm of Botev et al. (2010) and modulated by the adaptive smoothing approach of Abramson (1982). The rationale behind adaptive kernel density estimation is to use a narrower bandwidth near the peaks of the sampling distribution (where the ordered dates are closely spaced in time), and a wider bandwidth in the distribution's sparsely sampled troughs. Thus, the resolution of the density estimate is optimised according to data availability.

Value

If x has class UPb, PbPb, ArAr, KCa, ReOs, SmNd, RbSr, UThHe, fissiontracks or ThU, returns an object of class KDE, i.e. a list containing the following items:

x horizontal plot coordinates
vertical plot coordinates
bw the base bandwidth of the density estimate
ages the data values from the input to the kde function
log copied from the input

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

See Also
radialplot, cad

Examples
kde(examples$UPb)

dev.new()
kde(examples$FT1, log=TRUE)

dev.new()
kde(examples$DZ, from=1, to=3000, kernel="epanechnikov")

ludwig

Linear regression of U-Pb data with correlated errors, taking into account decay constant uncertainties.

Description
Implements the maximum likelihood algorithm for Total-Pb/U isochron regression of Ludwig (1998) and extends the underlying methodology to accommodate U-Th-Pb data and initial U-series disequilibrium.
ludwig

Usage

ludwig(x, ...)

## Default S3 method:
ludwig(x, exterr = FALSE, alpha = 0.05, model = 1, anchor = 0, ...)

Arguments

x an object of class UPb
...
optional arguments
exterr propagate external sources of uncertainty (i.e. decay constants)?
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 $^{207}\text{Pb}/^{206}\text{Pb}$-ratio intercept (for Tera-Wasserburg). If the p-value for the chi-square test is less than alpha, 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.
anchor control parameters to fix the intercept age or common Pb composition of the isochron fit. This can be a scalar or a vector.
If anchor[1]=0: do not anchor the isochron.
If anchor[1]=1: fix the common Pb composition at the values stored in settings('iratio', ...).
If anchor[1]=2: force the isochron line to intersect the concordia line at an age equal to anchor[2].

Details

The 3-dimensional regression algorithm of Ludwig and Titterington (1994) was modified by Ludwig (1998) to fit so-called ‘Total Pb-U isochrons’. These are constrained to a radiogenic endmember composition that falls on the concordia line. In its most sophisticated form, this algorithm does not only allow for correlated errors between variables, but also between aliquots. IsoplotR currently uses this algorithm to propagate decay constant uncertainties in the total Pb-U isochron ages.

Value

LL the log likelihood of the discordia fit
par a vector with the lower concordia intercept, the common Pb ratios and (if model=3) the dispersion parameter
cov the covariance matrix of par
logpar the logarithms of par
logcov  the logarithms of cov
df  the degrees of freedom of the model fit \((n - 2\) if \(x\text{format}<4\) or \(2n - 3\) if \(x\text{format}>3\), 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


See Also

concordia, titterington, isochron

Examples

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

mclean

Predict disequilibrium concordia compositions

Description

Returns the predicted \(^{206}\text{Pb}/^{238}\text{U}\) and \(^{207}\text{Pb}/^{235}\text{U}\) ratios for any given time with or without initial U-series disequilibrium.

Usage

mclean(tt = 0, d = diseq(), exterr = FALSE, i = 1)

Arguments

tt  the age of the sample
d  an object of class diseq
exterr  propagate the uncertainties associated with decay constants and the \(^{238}\text{U}/^{235}\text{U}\)-ratio.
i  aliquot to be used, if d$ThU$option=3.
Details

U decays to Pb in 14 (for \(^{238}\text{U}\)) or 11/12 (for \(^{235}\text{U}\)) steps. Conventional U-Pb geochronology assumes that secular equilibrium between all the short lived intermediate daughters was established at the time of isotopic closure. Under this assumption, the relative abundances of those intermediate daughters can be neglected and the age equation reduces to a simple function of the measured Pb/U ratios. In reality, however, the assumption of initial secular equilibrium is rarely met. Accounting for disequilibrium requires a more complex set of age equations, which are based on a coupled system of differential equations. The solution to this system of equations is given by a matrix exponential. IsoplotR solves this matrix exponential for any given time, using either the assumed initial activity ratios, or (for young samples) the measured activity ratios of the longest lived intermediate daughters. Based on a Matlab script by Noah McLean.

Value

a list containing the predicted \(^{206}\text{Pb}/^{238}\text{U}\), \(^{207}\text{Pb}/^{235}\text{U}\) and \(^{207}\text{Pb}/^{206}\text{Pb}\) ratios at time \(t\); the derivatives of the \(^{206}\text{Pb}/^{238}\text{U}\), \(^{207}\text{Pb}/^{235}\text{U}\) and \(^{207}\text{Pb}/^{206}\text{Pb}\) ratios with respect to time; and the derivatives of the \(^{206}\text{Pb}/^{238}\text{U}\), \(^{207}\text{Pb}/^{235}\text{U}\) and \(^{207}\text{Pb}/^{206}\text{Pb}\) ratios with respect to the intermediate decay constants and \(^{238}\text{U}/^{235}\text{U}\)-ratio.

Author(s)

Noah McLean (algorithm) and Pieter Vermeesch (code)

See Also
diseq

Examples

d <- diseq(U48=list(x=0,option=1),ThU=list(x=2,option=1),
          RaU=list(x=2,option=1),PaU=list(x=2,option=1))
mclean(tt=2,d=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 = NA,
ylab = NA,
...
)

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

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 par('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
Multidimensional Scaling (MDS) is a dimension-reducting technique that takes a matrix of pairwise ‘dissimilarities’ between objects (e.g., age distributions) as input and produces a configuration of two (or higher-) dimensional coordinates as output, so that the Euclidean distances between these coordinates approximate the dissimilarities of the input matrix. Thus, an MDS-configuration serves as a ‘map’ in which similar samples cluster closely together and dissimilar samples plot far apart. In the context of detrital geochronology, the dissimilarity between samples is given by the statistical distance between age distributions. There are many ways to define this statistical distance. IsoplotR uses the Kolmogorov-Smirnov (KS) statistic due to its simplicity and the fact that it behaves like a true distance in the mathematical sense of the word (Vermeesch, 2013). The KS-distance is given by the maximum vertical distance between two \textit{cad} step functions. Thus, the KS-distance takes on values between zero (perfect match between two age distributions) and one (no overlap between two distributions). Calculating the KS-distance between samples two at a time populates a symmetric dissimilarity matrix with positive values and a zero diagonal. IsoplotR implements two algorithms to convert this matrix into a configuration. The first (‘classical’) approach uses a sequence of basic matrix manipulations developed by Young and Householder (1938) and Torgerson (1952) to achieve a linear fit between the KS-distances and the fitted distances on the MDS configuration. The second, more sophisticated (‘nonmetric’) approach subjects the input distances to a transformation $f$ prior to fitting a configuration:

$$\delta_{i,j} = f(KS_{i,j})$$

where $KS_{i,j}$ is the KS-distance between samples $i$ and $j$ (for $1 \leq i \neq j \leq n$) and $\delta_{i,j}$ is the ‘disparity’ (Kruskal, 1964). Fitting an MDS configuration then involves finding the disparity transformation that maximises the goodness of fit (or minimises the ‘stress’) between the disparities and the fitted distances. The latter two quantities can also be plotted against each other as a ‘Shepard plot’.

Returns an object of class \texttt{MDS}, i.e. a list containing the following items:

- \texttt{points} a two-column vector of the fitted configuration
- \texttt{classical} a logical flag indicating whether the MDS configuration was obtained by classical (\texttt{TRUE}) or nonmetric (\texttt{FALSE}) MDS
- \texttt{diss} the dissimilarity matrix used for the MDS analysis
- \texttt{stress} (only if \texttt{classical}=\texttt{TRUE}) the final stress achieved (in percent)

References


Pb0corr


See Also
cad, kde

Examples

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

Pb0corr Common Pb correction

Description

Applies a common-Pb correction to a U-Pb dataset using either the Stacey-Kramers mantle evolution model, isochron regression, or any nominal initial Pb isotope composition.

Usage

Pb0corr(x, option = 3, omit4c = NULL)

Arguments

x an object of class UPb
option one of either
1: nominal common Pb isotope composition
2: isochron regression
3: Stacey-Kramers correction
omit4c vector with indices of aliquots that should be omitted from the isochron regression (only used if option=2)

Details

IsoplotR implements nine different methods to correct for the presence of non-radiogenic (‘common’) lead. This includes three strategies tailored to datasets that include $^{204}$Pb measurements, three strategies tailored to datasets that include $^{208}$Pb measurements, and a further three strategies for datasets that only include $^{206}$Pb and $^{207}$Pb.

$^{204}$Pb is the only one of lead’s four stable isotopes that does not have a naturally occurring radioactive parent. This makes it very useful for common-Pb correction:
\[
\left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_r = \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_m - \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_o
\]
where \( \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_r \) marks the radiogenic \(^{206}\text{Pb} or \(^{207}\text{Pb} component; \( \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_m \) is the measured ratio; and \( \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_o \) is the non-radiogenic component.

IsoplotR offers three different ways to determine \( \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_o \). The first and easiest option is to simply use a nominal value such as the \( \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_m \)-ratio of a cogenetic feldspar, assuming that this is representative for the common-Pb composition of the entire sample. A second method is to determine the non-radiogenic isotope composition by fitting an isochron line through multiple aliquots of the same sample, using the 3-dimensional regression algorithm of Ludwig (1998).

Unfortunately, neither of these two methods is applicable to detrital samples, which generally lack identifiable cogenetic minerals and aliquots. For such samples, IsoplotR infers the common-Pb composition from the two-stage crustal evolution model of Stacey and Kramers (1975). The second stage of this model is described by:

\[
\left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_o = \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_m - \left[ \frac{^{238}_{\text{U}}}{^{204}_{\text{Pb}}} \right]_{sk} \times (e^{3.7Ga \times 238 - e^{3.7Ga \times 238}})
\]
where \( \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_m = 11.152 \) and \( \left[ \frac{^{238}_{\text{U}}}{^{204}_{\text{Pb}}} \right]_{sk} = 9.74 \). These Equations can be solved for \( t \) and \( \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_o \) using the method of maximum likelihood. The \( \left[ \frac{^{207}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_o \)-ratio is corrected in exactly the same way, using \( \left[ \frac{^{207}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_m = 12.998 \).

In the absence of \(^{204}\text{Pb} measurements, a \(^{208}\text{Pb-based common lead correction can be used:}

\[
\left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_{co} = \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_{co} - \left[ \frac{^{207}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_{co}
\]
where \( \left[ \frac{^{206}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_{co} \) marks the non-radiogenic \(^{208}\text{Pb}-component, which is obtained by removing the radiogenic component for any given age.

If neither \(^{204}\text{Pb} nor \(^{208}\text{Pb} were measured, then a \(^{207}\text{Pb-based common lead correction can be used:}

\[
\left[ \frac{^{207}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_{co} = f \left[ \frac{^{207}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_{co} + (1 - f) \left[ \frac{^{207}_{\text{Pb}}}{^{204}_{\text{Pb}}} \right]_{co}
\]
where \( f \) is the fraction of common lead, and \( \left[ \frac{^{207}_{\text{Pb}}}{^{206}_{\text{Pb}}} \right]_{co} \) is obtained by projecting the U-Pb measurements on the concordia line in Tera-Wasserburg space. Like before, the initial lead composition \( \left[ \frac{^{207}_{\text{Pb}}}{^{206}_{\text{Pb}}} \right]_{co} \) can be obtained in three possible ways: by analysing a cogenetic mineral, by isochron regression through multiple aliquots, or from the Stacey and Kramers (1975) model.

Besides the common-Pb problem, a second reason for U-Pb discordance is radiogenic Pb-loss during igneous and metamorphic activity. This moves the data away from the concordia line along a linear array, forming an isochron or ‘discordia’ line. IsoplotR fits this line using the Ludwig (1998) algorithm. If the data are plotted on a Wetherill concordia diagram, the program will not only report the usual lower intercept with the concordia line, but the upper intercept as well. Both values are geologically meaningful as they constrain both the initial igneous age as well as the timing of the partial resetting event.

Value
Returns a list in which \( x.raw \) contains the original data and \( x \) the common Pb-corrected compositions. All other items in the list are inherited from the input data.
References


Examples

attach(examples)
corrected <- Pb8corr(UPb,option=2)
concordia(corrected)
# produces identical results as:
dev.new()
concordia(UPb,common.Pb=2)

Description

Implements the discrete mixture modelling algorithms of Galbraith and Laslett (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 = discfilter(),
  common.Pb = 0,
  exterr = TRUE,
  sigdig = 2,
  log = TRUE,
  alpha = 0.05,
  ...
)
## S3 method for class 'PbPb'
peakfit(
  x,
  k = 1,
  exterr = TRUE,
  sigdig = 2,
  log = TRUE,
  common.Pb = 0,
  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 'ThPb'
peakfit(
  x,
  k = 1,
  exterr = TRUE,
  sigdig = 2,
  log = TRUE,
  i2i = FALSE,
  alpha = 0.05,
  ...
)

## S3 method for class 'KCa'
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,
  detritus = 0,
  ...
)

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

**Arguments**

- `x` either an \([nx2]\) matrix with measurements and their standard errors, or an object of class `fissiontracks, UPb, PbPb, ThPb, ArAr, KCa, 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 \(^{207}\text{Pb}/^{235}\text{U}\) age (type=1), the \(^{206}\text{Pb}/^{238}\text{U}\) age (type=2), the \(^{207}\text{Pb}/^{206}\text{Pb}\) age (type=3), the \(^{207}\text{Pb}/^{206}\text{Pb} \times \text{Th}^{238}\text{U}\) age (type=4), the concordia age (type=5), or the \(^{208}\text{Pb}/^{232}\text{Th}\) age (type=6).
- `cutoff.76` the age (in Ma) below which the \(^{206}\text{Pb}/^{238}\text{U}\) and above which the \(^{207}\text{Pb}/^{206}\text{Pb}\) age is used. This parameter is only used if type=4.
- `cutoff.disc` discordance cutoff filter. This is an object of class `discfilter`.
- `common.Pb` common lead correction: 0: none 1: use the Pb-composition stored in `settings('iratio','Pb206Pb204')` (if `x` has class UPb and `x$format<4`); `settings('iratio','Pb206Pb204')` and `settings('iratio','Pb207Pb204')` (if `x` has class PbPb or `x` has class UPb and `3<x$format<7`); or `settings('iratio','Pb208Pb206')` and `settings('iratio','Pb208Pb207')` (if `x` has class UPb and `x$format=7 or 8`).
peakfit

2: use the isochron intercept as the initial Pb-composition
3: use the Stacey-Kramers two-stage model to infer the initial Pb-composition (only applicable if x has class UPb)

'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') 
$^{40}$Ar/$^{36}$Ar, $^{40}$Ca/$^{44}$Ca, $^{207}$Pb/$^{204}$Pb, $^{87}$Sr/$^{86}$Sr, $^{143}$Nd/$^{144}$Nd, $^{187}$Os/$^{188}$Os, 
$^{230}$Th/$^{232}$Th, $^{176}$Hf/$^{177}$Hf or $^{204}$Pb/$^{208}$Pb ratio from an isochron fit. Setting i2i to FALSE uses the default values stored in settings('iratio',...).

detritus detrital $^{230}$Th correction (only applicable when x$format=1$ or 2).

0: no correction
1: project the data along an isochron fit
2: correct the data using an assumed initial $^{230}$Th/$^{232}$Th-ratio for the detritus.
3: correct the data using the measured present day $^{230}$Th/$^{238}$U, $^{232}$Th/$^{238}$U and $^{234}$U/$^{238}$U-ratios in the detritus.

Details

Consider a dataset of n dates \{t_1, t_2, ..., t_n\} with analytical uncertainties \{s[t_1], s[t_2], ..., s[t_n]\}. Define \( z_i = \log(t_i) \) and \( s[z_i] = s[t_i] / t_i \). Suppose that these n values are derived from a mixture of k > 2 populations with means \{\mu_1, ..., \mu_k\}. Such a discrete mixture may be mathematically described by \( P(z_i | \mu, \omega) = \sum_{j=1}^{k} \pi_j N(z_i | \mu_j, s[z_j]^2) \) where \( \pi_j \) is the proportion of the population that belongs to the j\textsuperscript{th} component, and \( \pi_k = 1 - \sum_{j=1}^{k-1} \pi_j \). This equation can be solved by the method of maximum likelihood (Galbraith and Laslett, 1993). IsoplotR implements the Bayes Information Criterion (BIC) as a means of automatically choosing k. This option should be used with caution, as the number of peaks steadily rises with sample size (n). If one is mainly interested in the youngest age component, then it is more productive to use an alternative parameterisation, in which all grains are assumed to come from one of two components, whereby the first component is a single discrete age peak (exp(m), say) and the second component is a continuous distribution (as described by the central age model), but truncated at this discrete value (Van der Touw et al., 1997).

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 widths of approximate 100(1 - \alpha)\% confidence intervals for t

props a 2 x k matrix with the following rows:
   p: the proportions of the k peaks
   s[p]: the estimated uncertainties (standard errors) of p

L the log-likelihood of the fit

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

References


See Also
radialplot, central

Examples
attach(examples)
peakfit(FT1,k=2)

peakfit(LudwigMixture,k='min')

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. Serves as a vehicle to display finite and continuous mixture models.

Usage
radialplot(x, ...)

## Default S3 method:
radialplot(
x,
from = NA,
to = NA,
z0 = NA,
transformation = "log",
sigdig = 2,
show.numbers = FALSE,
pch = 21,
levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
k = 0,
markers = NULL,
alpha = 0.05,
units = "",
hide = NA,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'fissiontracks'
radialplot(
  x,
  from = NA,
  to = NA,
  z0 = NA,
  transformation = "arcsin",
sigdig = 2,
  show.numbers = FALSE,
pch = 21,
  levels = NA,
  clabel = "",
  bg = c("yellow", "red"),
  col = "black",
  markers = NULL,
  k = 0,
  exterr = TRUE,
  alpha = 0.05,
  hide = NULL,
  omit = NULL,
  omit.col = NA,
  ...
)

## S3 method for class 'UPb'
radialplot(
  x,
  from = NA,
  to = NA,
  z0 = NA,
  transformation = "log",
  type = 4,
  cutoff.76 = 1100,
  cutoff.disc = discfilter(),
  show.numbers = FALSE,
pch = 21,
sigdig = 2,
  levels = NA,
  clabel = "",
  bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
exterr = TRUE,
common.Pb = 0,
alpha = 0.05,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'PbPb'
radialplot(
  x,
  from = NA,
  to = NA,
  z0 = NA,
  sigdig = 2,
  transformation = "log",
  show.numbers = FALSE,
  pch = 21,
  levels = NA,
  clabel = "",
  bg = c("yellow", "red"),
  col = "black",
  markers = NULL,
  k = 0,
  exterr = TRUE,
  common.Pb = 2,
  alpha = 0.05,
  hide = NULL,
  omit = NULL,
  omit.col = NA,
  ...
)

## S3 method for class 'ArAr'
radialplot(
  x,
  from = NA,
  to = NA,
  z0 = NA,
  sigdig = 2,
  transformation = "log",
  show.numbers = FALSE,
  pch = 21,
  levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
extrerr = TRUE,
i2i = FALSE,
alpha = 0.05,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'KCa'
radialplot(
x,
from = NA,
to = NA,
z0 = NA,
sigdig = 2,
transformation = "log",
show.numbers = FALSE,
pch = 21,
levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
extrerr = TRUE,
i2i = FALSE,
alpha = 0.05,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'ThPb'
radialplot(
x,
from = NA,
to = NA,
z0 = NA,
sigdig = 2,
transformation = "log",
show.numbers = FALSE,
pch = 21,
levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
exterr = TRUE,
i2i = TRUE,
alpha = 0.05,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'UTHHe'
radialplot(
x,
from = NA,
to = NA,
z0 = NA,
sigdig = 2,
transformation = "log",
show.numbers = FALSE,
pch = 21,
levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
alpha = 0.05,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'ReOs'
radialplot(
x,
from = NA,
to = NA,
z0 = NA,
sigdig = 2,
transformation = "log",
show.numbers = FALSE,
pch = 21,
levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
exterr = TRUE,
i2i = TRUE,
alpha = 0.05,
hide = NULL,
omit = NULL,
omit.col = NA,
...)

## S3 method for class 'SmNd'
radialplot(
x,
from = NA,
to = NA,
z0 = NA,
sigdig = 2,
transformation = "log",
show.numbers = FALSE,
pch = 21,
levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
exterr = TRUE,
i2i = TRUE,
alpha = 0.05,
hide = NULL,
omit = NULL,
omit.col = NA,
...)

## S3 method for class 'RbSr'
radialplot(
x,
from = NA,
to = NA,
z0 = NA,
sigdig = 2,
transformation = "log",
show.numbers = FALSE,
pch = 21,
levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
exterr = TRUE,
i2i = TRUE,
alpha = 0.05,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'LuHf'
radialplot(
  x,
  from = NA,
  to = NA,
  z0 = NA,
sigdig = 2,
transformation = "log",
show.numbers = FALSE,
pch = 21,
levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
exterr = TRUE,
i2i = TRUE,
alpha = 0.05,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'ThU'
radialplot(
  x,
  from = NA,
  to = NA,
z0 = NA,
sigdig = 2,
transformation = "log",
show.numbers = FALSE,
pch = 21,
levels = NA,
clabel = "",
bg = c("yellow", "red"),
col = "black",
markers = NULL,
k = 0,
i2i = FALSE,
alpha = 0.05,
detritus = 0,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

Arguments

x Either an \([nx2]\) matrix of (transformed) values \(z\) and their standard errors \(s\)

OR

and object of class fissiontracks, UThHe, ArAr, KCa, ReOs, SmNd, RbSr, LuHf, ThU, PbPb, ThPb 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

z0 central value

transformation one of either log, linear, sqrt or arcsin (if \(x\) has class fissiontracks and fissiontracks$format \(\neq 1\)).
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.

clabel label of the colour legend

bg Fill colour for the plot symbols. This can either be a single colour or multiple colours to form a colour ramp (to be used if levels! = NA):
a single colour: rgb(0,1,0,0.5), '#FF000080', 'white', etc.;
multiple colours: c(rgb(1,0,0,0.5),rgb(0,1,0,0.5)), c('#FF000080', '#00FF0080'), c('blue','red'), c('blue','yellow','red'), etc.;
a colour palette: rainbow(n=100), topo.colors(n=100,alpha=0.5), etc.; or
a reversed palette: `rev(topo.colors(n=100, alpha=0.5))`, etc. for plot symbols, set bg=NA

`col` text colour to be used if `show.numbers=TRUE`

`k` number of peaks to fit using the finite mixture models of Galbraith and Laslett (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

`units` measurement units to be displayed in the legend.

`hide` vector with indices of aliquots that should be removed from the radial plot.

`omit` vector with indices of aliquots that should be plotted but omitted from the central age calculation or mixture models.

`omit.col` colour that should be used for the omitted aliquots.

`extern` include the external sources of uncertainty into the error propagation for the central age and mixture models?

`type` scalar indicating whether to plot the $^{207}\text{Pb}/^{235}\text{U}$ age (type=1), the $^{206}\text{Pb}/^{238}\text{U}$ age (type=2), the $^{207}\text{Pb}/^{206}\text{Pb}$ age (type=3), the $^{207}\text{Pb}/^{206}\text{Pb},^{208}\text{Pb}/^{238}\text{U}$ age (type=4), the concordia age (type=5), or the $^{208}\text{Pb}/^{232}\text{Th}$ age (type=6).

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

`cutoff.disc` discordance cutoff filter. This is an object of class `discfilter`.

`common.Pb` common lead correction:

0: none

1: use the Pb-composition stored in `settings('iratio', 'Pb207Pb206')` (if `x` has class `UPb` and `x$format<4`); `settings('iratio', 'Pb206Pb204')` and `settings('iratio', 'Pb207Pb204')` (if `x` has class `PbPb` or `x` has class `UPb` and $3<x$`$format<7$); or `settings('iratio', 'Pb208Pb206')` and `settings('iratio', 'Pb208Pb207')` (if `x` has class `UPb` and `x$format=7` or 8).

2: remove the common Pb by projecting the data along an inverse isochron. Note: choosing this option introduces a degree of circularity in the central age calculation. In this case the radial plot just serves as a way to visualise the residuals of the data around the isochron, and one should be careful not to over-interpret the numerical output.

3: use the Stacey-Kramers two-stage model to infer the initial Pb-composition ‘isochron to intercept’: calculates the initial (aka ‘inherited’, ‘excess’, or ‘common’) $^{40}\text{Ar}/^{36}\text{Ar}$, $^{40}\text{Ca}/^{44}\text{Ca}$, $^{207}\text{Pb}/^{204}\text{Pb}$, $^{87}\text{Sr}/^{86}\text{Sr}$, $^{143}\text{Nd}/^{144}\text{Nd}$, $^{187}\text{Os}/^{188}\text{Os}$, $^{230}\text{Th}/^{232}\text{Th}$, $^{176}\text{Hf}/^{177}\text{Hf}$ or $^{204}\text{Pb}/^{208}\text{Pb}$ ratio from an isochron fit. Setting `i2i` to FALSE uses the default values stored in `settings('iratio',...)`.

Note that choosing this option introduces a degree of circularity in the central age calculation. In this case the radial plot just serves as a way to visualise the residuals of the data around the isochron, and one should be careful not to over-interpret the numerical output.
detritus detrital $^{230}$Th correction (only applicable when x$\$format == 1 or 2).
0: no correction
1: project the data along an isochron fit
2: correct the data using an assumed initial $^{230}$Th/$^{232}$Th-ratio for the detritus.
3: correct the data using the measured present day $^{230}$Th/$^{238}$U, $^{232}$Th/$^{238}$U and $^{234}$U/$^{238}$U-ratios in the detritus.

Details

The radial plot (Galbraith, 1988, 1990) is a graphical device that was specifically designed to display heteroscedastic data, and is constructed as follows. Consider a set of dates \( \{t_1, ..., t_i, ..., t_n\} \) and uncertainties \( \{s[t_1], ..., s[t_i], ..., s[t_n]\} \). Define \( z_i = z[t_i] \) to be a transformation of \( t_i \) (e.g., \( z_i = \log[t_i] \)), and let \( s[z_i] \) be its propagated analytical uncertainty (i.e., \( s[z_i] = s[t_i]/t_i \) in the case of a logarithmic transformation). Create a scatter plot of \( (x_i, y_i) \) values, where \( x_i = 1/s[z_i] \) and \( y_i = (z_i - z_o)/s[z_i] \), where \( z_o \) is some reference value such as the mean. The slope of a line connecting the origin of this scatter plot with any of the \( (x_i, y_i) \)s is proportional to \( z_i \) and, hence, the date \( t_i \).

These dates can be more easily visualised by drawing a radial scale at some convenient distance from the origin and annotating it with labelled ticks at the appropriate angles. While the angular position of each data point represents the date, its horizontal distance from the origin is proportional to the precision. Imprecise measurements plot on the left hand side of the radial plot, whereas precise age determinations are found further towards the right. Thus, radial plots allow the observer to assess both the magnitude and the precision of quantitative data in one glance.

Value

does not produce any numerical output, but does report the central age and the results of any mixture modelling in the title. An asterisk is added to the plot title if the initial daughter correction is based on an isochron regression, to highlight the circularity of using an isochron to compute a central age, and to indicate that the reported uncertainties do not include the uncertainty of the initial daughter correction. This is because this uncertainty is neither purely random nor purely systematic.

References


See Also

peakfit, central
read.data

Examples

```
attach(examples)
radialplot(FT1)

dev.new()
radialplot(LudwigMixture,k='min')
```

---

**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, 
  ierr = 1, 
  d = diseq(), 
  Th02 = c(0, 0), 
  Th02U48 = c(0, 0, 1e+06, 0, 0, 0, 0, 0, 0), 
  ... 
)
```

### S3 method for class 'data.frame'

```
read.data(
  x, 
  method = "U-Pb", 
  format = 1, 
  ierr = 1, 
  d = diseq(), 
  Th02 = c(0, 0), 
  Th02U48 = c(0, 0, 1e+06, 0, 0, 0, 0, 0, 0), 
  ... 
)
```

### S3 method for class 'matrix'

```
read.data(
  x, 
  method = "U-Pb", 
  format = 1, 
  ... 
)
```
ierr = 1,
d = diseq(),
Th02 = c(0, 0),
Th02U48 = c(0, 0, 1e+06, 0, 0, 0, 0, 0, 0),
...
)

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', 'Th-Pb', 'Ar-Ar', 'K-Ca', '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. 07/35, err[07/35], 06/38, err[06/38], rho
2. 38/06, err[38/06], 07/06, err[07/06]
3. X=07/35, err[X], Y=06/38, err[Y], Z=07/06, err[Z] (, rho[X,Y]) (, rho[Y,Z])
4. X=07/35, err[X], Y=06/38, err[Y], Z=04/38, rho[X,Y], rho[X,Z], rho[Y,Z]
5. X=38/06, err[X], Y=07/06, err[Y], Z=04/06, err[Z] (, rho[X,Y], rho[X,Z], rho[Y,Z])
6. 07/35, err[07/35], 06/38, err[06/38], 04/38, err[04/38], 07/06, err[07/06],
4/07, err[04/07], 04/06, err[04/06]
7. W=07/35, err[W], X=06/38, err[X], Y=08/06, err[Y], and Z=32/38, err[Z],
   rho[W,X], rho[W,Y], rho[W,Z], rho[X,Y], rho[X,Z], rho[Y,Z]
8. W=38/06, err[W], X=07/06, err[X], Y=08/06, err[Y], and Z=32/38, (err[Z],
   rho[W,X], rho[W,Y], rho[W,Z], rho[X,Y], rho[X,Z], rho[Y,Z])

where optional columns are marked in round brackets

if method='Pb-Pb', then format is one of either:
1. 6/4, err[6/4], 7/4, err[7/4], rho
2. 4/6, err[4/6], 7/6, err[7/6], rho
3. 6/4, err[6/4], 7/4, err[7/4], 6/7, err[6/7]

if method='Th-Pb', then format is one of either:
1. 32/04, err[32/04], 08/04, err[08/04], rho
2. 32/08, err[32/08], 04/08, err[04/08], rho
3. 32/04, err[32/04], 08/04, err[08/04], 32/08, err[32/08]

if method='Ar-Ar', then format is one of either:
1. 9/6, err[9/6], 0/6, err[0/6], rho (, 39)
2. 6/0, err[6/0], 9/0, err[9/0] (, rho) (, 39)
3. 9/0, err[9/0], 6/0, err[6/0], 9/6, err[9/6] (, 39)

if method='K-Ca', then format is one of either:
1. K40/Ca44, err[K40/Ca44], Ca40/Ca44, err[Ca40/Ca44], rho
2. K40/Ca40, err[K40/Ca40], Ca44/Ca40, err[Ca44/Ca40], rho
3. K40/Ca44, err[K40/Ca44], Ca40/Ca44, err[Ca40/Ca44], K40/Ca40, err[K40/Ca40] if method='Rb-Sr', then format is one of either:
   1. Rb87/Sr86, err[Rb87/Sr86], Sr87/Sr86, err[Sr87/Sr86] (, rho)
   2. Rb/Sr, err[Rb/Sr], Sr/Sr, err[Sr/Sr]
   3. Rb, err[Rb], Sr, err[Sr], Sr87/Sr86, err[Sr87/Sr86]

   where Rb and Sr are in ppm

if method='Sm-Nd', then format is one of either:
   1. Sm147/Nd144, err[Sm147/Nd144], Nd143/Nd144, err[Nd143/Nd144] (, rho)
   2. Sm147/Nd143, err[Sm147/Nd143], Nd144/Nd143, err[Nd144/Nd143] (, rho)
   3. Sm, err[Sm], Nd, err[Nd], Nd143/Nd144, err[Nd143/Nd144]

   where Sm and Nd are in ppm

if method='Re-Os', then format is one of either:
   1. Re187/Os188, err[Re187/Os188], Os187/Os188, err[Os187/Os188] (, rho)
   2. Re187/Os187, err[Re187/Os187], Os188/Os187, err[Os188/Os187] (, rho)
   3. Re, err[Re], Os, err[Os], Os187/Os188, err[Os187/Os188]

   where Re and Os are in ppm

if method='Lu-Hf', then format is one of either:
   1. Lu176/Hf177, err[ Lu176/Hf177], Hf176/Hf177, err[ Hf176/Hf177] (, rho)
   2. Lu176/Hf176, err[ Lu176/Hf176], Hf177/Hf176, err[ Hf177/Hf176] (, rho)
   3. Lu, err[ Lu], Hf, err[ Hf], Hf176/Hf177, err[ Hf176/Hf177]

   where Lu and Hf are in ppm

if method='Th-U', then format is one of either:
   1. X=8/2, err[X], Y=4/2, err[Y], Z=0/2, err[Z], rho[X,Y], rho[X,Z], rho[Y,Z]
   2. X=2/8, err[X], Y=4/8, err[Y], Z=0/8, err[Z], rho[X,Y], rho[X,Z], rho[Y,Z]
   3. X=8/2, err[X], Y=0/2, err[Y], rho[X,Y]
   4. X=2/8, err[X], Y=0/8, err[Y], rho[X,Y]

   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-ratios 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-ratios or U-concentration measurements (in ppm) and their analytical uncertainties.
if method = 'other', x is read as a table, unless format is one of either:
  radial or average: X, err[X]
regression: X, err[X], Y, err[Y], rho
spectrum: f, X, err[X]

ierr indicates whether the analytical uncertainties are reported as:
1: 1σ absolute uncertainties.
2: 2σ absolute uncertainties.
3: 1σ relative uncertainties (%).
4: 2σ relative uncertainties (%).

d an object of class diseq.

Thθ2 2-element vector with the assumed initial $^{230}\text{Th}/^{232}\text{Th}$-ratio of the detritus (for formats 1 and 2) or rock (for formats 3 and 4) and its standard error.

Thθ2U48 9-element vector with the measured composition of the detritus, containing X=θ/8, sX, Y=2/8, sY, Z=4/8, sZ, rXY, rXZ, rYZ.

Details

IsoplotR provides the following example input files:

- **U-Pb**: UPb1.csv, UPb2.csv, UPb3.csv, UPb4.csv, UPb5.csv, UPb6.csv, UPb7.csv, UPb8.csv
- **Pb-Pb**: PbPb1.csv, PbPb2.csv, PbPb3.csv
- **Th-Pb**: ThPb1.csv, ThPb2.csv, ThPb3.csv
- **Ar-Ar**: ArAr1.csv, ArAr2.csv, ArAr3.csv
- **K-Ca**: KCa1.csv, KCa2.csv, KCa3.csv
- **Re-Os**: ReOs1.csv, ReOs2.csv, ReOs3.csv
- **Sm-Nd**: SmNd1.csv, SmNd2.csv, SmNd3.csv
- **Rb-Sr**: RbSr1.csv, RbSr2.csv, RbSr3.csv
- **Lu-Hf**: LuHf1.csv, LuHf2.csv, LuHf3.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. For example, to read the ArAr1.csv file:

fname <- system.file('ArAr1.csv', package = 'IsoplotR')
ArAr <- read.data(fname, method = 'Ar-Ar', format = 1)

Value

An object of class UPb, PbPb, ThPb, KCa, RbSr, SmNd, LuHf, ReOs, UThHe, fissiontracks, detritals or PD. See classes for further details.
scatterplot

See Also

examples, settings

Examples

```r
f1 <- system.file("UPb1.csv",package="IsoplotR")
file.show(f1) # inspect the contents of 'UPb1.csv'
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)
```

---

**scatterplot**

Create a scatter plot with error ellipses or crosses

**Description**

Takes bivariate data with (correlated) uncertainties as input and produces a scatter plot with error ellipses or crosses as output. (optionally) displays the linear fit on this diagram, and can show a third variable as a colour scale.
Usage

scatterplot(
  xy,
  alpha = 0.05,
  show.numbers = FALSE,
  show.ellipses = 1,
  levels = NA,
  clabel = "",
  ellipse.fill = c("#00FF0080", "#FF000080"),
  ellipse.stroke = "black",
  fit = "none",
  add = FALSE,
  empty = FALSE,
  ci.col = "gray80",
  line.col = "black",
  lwd = 1,
  hide = NULL,
  omit = NULL,
  omit.fill = NA,
  omit.stroke = "grey",
  addcolourbar = TRUE,
  bg,
  cex,
  xlim = NULL,
  ylim = NULL,
  xlab,
  ylab,
  ...
)

Arguments

xy matrix with columns X, sX, Y, sY(,rXY)
alpha the probability cutoff for the error ellipses
show.numbers logical flag (TRUE to show grain numbers)
show.ellipses show the data as:
  0: points
  1: error ellipses
  2: error crosses
levels a vector with additional values to be displayed as different background colours
         within the error ellipses.
clabel label for the colour scale
ellipse.fill Fill colour for the error ellipses. This can either be a single colour or multiple
colours to form a colour ramp. Examples:
a single colour: rgb(0,1,0,0.5), '#FF000080', 'white', etc.;
multiple colours: c(rbg(1,0,0,0.5), rgb(0,1,0,0.5)), c('#FF000080', '#00FF0080'),
c('blue', 'red'), c('blue', 'yellow', 'red'), etc.;
a colour palette: rainbow(n=100), topo.colors(n=100, alpha=0.5), etc.; or
a reversed palette: rev(topo.colors(n=100, alpha=0.5)), etc.
For empty ellipses, set ellipse.col=NA

Examples

```r
         28.530, 50.540, 51.595, 86.51, 106.40, 157.35)
c Y <- c(0.7268, 0.7809, 0.8200, 0.8116, 0.8160, 0.8302,
         0.8642, 0.9534, 0.9617, 1.105, 1.230, 1.440)
c sX <- X*0.02
c sY <- Y*0.01
c dat <- cbind(X,sX,Y,sY)
c scatterplot(dat, fit=york(dat), show.ellipses=2)
```
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, 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

Details

The fundamental fission track age is given by:

\[ t = \frac{1}{\lambda_{238}} \ln \left( 1 + \frac{\lambda_{238} N_s}{\lambda_f [238U] A_s L} \right) \quad \text{(eq.1)} \]

where \( N_s \) is the number of spontaneous fission tracks measured over an area \( A_s \), \([238U]\) is the \( ^{238}\text{U} \)-concentration in atoms per unit volume, \( \lambda_f \) is the fission decay constant, \( L \) is the etchable fission track length, and the factor 2 is a geometric factor accounting for the fact that etching reveals tracks from both above and below the internal crystal surface. Two analytical approaches are used to measure \([238U]\): neutron activation and LAICPMS. The first approach estimates the \( ^{238}\text{U} \)-concentration indirectly, using the induced fission of neutron-irradiated \( ^{235}\text{U} \) as a proxy for the \( ^{238}\text{U} \). In the most common implementation of this approach, the induced fission tracks are recorded by an external detector made of mica or plastic that is attached to the polished grain surface (Fleischer and Hart, 1972; Hurford and Green, 1983). The fission track age equation then becomes:

\[ t = \frac{1}{\lambda_{238}} \ln \left( 1 + \frac{\lambda_{238} \zeta \rho_d N_i}{2 N_s} \right) \quad \text{(eq.2)} \]

where \( N_i \) is the number of induced fission tracks counted in the external detector over the same area as the spontaneous tracks, \( \zeta \) is a ‘zeta’-calibration factor that incorporates both the fission decay constant and the etchable fission track length, and \( \rho_d \) is the number of induced fission tracks per unit area counted in a co-irradiated glass of known U-concentration. \( \rho_d \) allows the \( \zeta \)-factor to be ‘recycled’ between irradiations.
LAICPMS is an alternative means of determining the $^{238}\text{U}$-content of fission track samples without the need for neutron irradiation. The resulting $\text{U}$-concentrations can be plugged directly into the fundamental age equation (eq.1), but this is limited by the accuracy of the $\text{U}$-concentration measurements, the fission track decay constant and the etching and counting efficiencies. Alternatively, these sources of bias may be removed by normalising to a standard of known fission track age and defining a new ‘zeta’ calibration constant $\zeta_{\text{ICP}}$:

$$t = \frac{1}{\lambda_{^{238}\text{U}}} \ln \left(1 + \frac{\lambda_{^{238}\text{U}} \zeta_{\text{ICP}}}{2} \frac{N_{s}}{[^{238}\text{U}]A_{s}}\right) \quad \text{(eq.3)}$$

where $[^{238}\text{U}]$ may either stand for the $^{238}\text{U}$-concentration (in ppm) or for the U/Ca (for apatite) or U/Si (for zircon) ratio measurement (Vermeesch, 2017).

**Value**

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

**References**


**See Also**

`age`

**Examples**

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

---

**settings**

*Load settings to and from json*

**Description**

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

Usage

settings(setting = NA, ..., fname = NA, reset = FALSE)

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", "Pa231", "Ra226", "Re187", "Sm147", "Rb87", "Lu176", or "K40") PLUS (optionally) the decay constant value and its analytical error. Omitting the latter two numbers simply returns the existing values.

For 'iratio': the isotopic ratio of interest (one of either "Ar40Ar36", "Ar38Ar36", "Ca40Ca44", "Rb85Rb87", "Sr88Sr86", "Sr87Sr86", "Sr85Sr86", "Re185Re187", "Os184Os192", "Os186Os192", "Os187Os192", "Os188Os192", "Os189Os192", "Sm144Sm152", "Sm145Sm152", "Sm146Sm152", "Sm147Sm152", "Sm148Sm152", "Nd144Nd146", "Nd145Nd147", "Nd146Nd148", "Nd147Nd149", "Lu176Lu178", "Hf176Hf178", "Hf177Hf179", "Hf178Hf179", "Hf179Hf181", "U238U235", "Pb207Pb206", "Pb206Pb204", "Pb204Pb203", "Pb203Pb202", "Pb202Pb200") PLUS (optionally) the isotopic ratio and its analytical error. Omitting the latter 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 the latter two numbers simply returns the existing values.

For 'mindens': the mineral of interest (one of either "apatite" or "zircon") PLUS the mineral density. Omitting the latter number simply returns the existing value.

For 'etchfact': the mineral of interest (one of either "apatite" or "zircon") PLUS the etch efficiency factor. Omitting this number simply returns the existing value.

For '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
reset logical. If TRUE, restores the default values
settings

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, then settings returns either a scalar with the existing value, or a two-element vector with the value and its uncertainty.

References

1. Decay constants:
   - Ar: Renne, Paul R., et al. "Response to the comment by WH Schwarz et al. on "Joint determination of $^{40}\text{K}$ decay constants and $^{40}\text{Ar}/^{40}\text{K}$ for the Fish Canyon sanidine standard, and improved accuracy for $^{40}\text{Ar}/^{39}\text{Ar}$ geochronology" by PR Renne et al.(2010)." Geochimica et Cosmochimica Acta 75.17 (2011): 5097-5100.

2. Isotopic ratios:


See Also

read.data

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) for linear regression of three dimensional data with correlated uncertainties.

**Usage**

`titterington(x, alpha = 0.05)`

**Arguments**

- `x` an `[nx9]` matrix with the following columns: `X`, `sX`, `Y`, `sY`, `Z`, `sZ`, `rhoXY`, `rhoXZ`, `rhoYZ`.
- `alpha` cutoff value for confidence intervals

**Details**

Ludwig and Titterington (1994)’s 3-dimensional linear regression algorithm for data with correlated uncertainties is an extension of the 2-dimensional algorithm by Titterington and Halliday (1979), which itself is equivalent to the algorithm of York et al. (2004). Given `n` triplets of (approximately) collinear measurements `X_i`, `Y_i` and `Z_i` (for `1 ≤ i ≤ n`), their uncertainties `s[X_i]`, `s[Y_i]` and `s[Z_i]`, and their covariances `cov[X_i, Y_i]`, `cov[X_i, Z_i]` and `cov[Y_i, Z_i]`, the `titterington` function fits two slopes and intercepts with their uncertainties. It computes the MSWD as a measure of under/overdispersion. Overdispersed datasets (MSWD>1) can be dealt with in the same three ways that are described in the documentation of the `isochron` function.

**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` [4x4]-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 (2n-4)
- `tfact` the 100(1 − α/2)% percentile of the t-distribution with `(n − 2k + 1)` degrees of freedom
weightedmean

Calculate the weighted mean age

Description

Averages heteroscedastic data either using the ordinary weighted mean, or using a random effects model with two sources of variance. Computes the MSWD of a normal fit without overdispersion. Implements a modified Chauvenet criterion to detect and reject outliers. Only propagates the systematic uncertainty associated with decay constants and calibration factors after computing the weighted mean isotopic composition. Does not propagate the uncertainty of any initial daughter correction, because this is neither a purely random or purely systematic uncertainty.

Usage

weightedmean(x, ...)

## Default S3 method:
weightedmean(
x,
from = NA,
to = NA,
random.effects = FALSE,
weightedmean

detect.outliers = TRUE,
plot = TRUE,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
ranked = FALSE,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'UPb'
weightedmean(
x,
random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
to = NA,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
type = 4,
cutoff.76 = 1100,
alpha = 0.05,
cutoff.disc = discfilter(),
exterr = TRUE,
ranked = FALSE,
common.Pb = 0,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'PbPb'
weightedmean(
x,
random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
weightedmean

to = NA,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
extrerr = TRUE,
common.Pb = 2,
ranked = FALSE,
hide = NULL,
omit = NULL,
omit.col = NA,
...)

## S3 method for class 'ThU'
weightedmean(
  x,
  random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
to = NA,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
ranked = FALSE,
i2i = TRUE,
detritus = 0,
hide = NULL,
omit = NULL,
omit.col = NA,
...)

## S3 method for class 'ArAr'
weightedmean(
  x,
  random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
to = NA,
levels = NA,
```
weightedmean

clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
exterr = TRUE,
ranked = FALSE,
i2i = FALSE,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'KCa'
weightedmean(
  x,
  random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
to = NA,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
exterr = TRUE,
ranked = FALSE,
i2i = FALSE,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'ThPb'
weightedmean(
  x,
  random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
to = NA,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
```
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
exterr = TRUE,
ranked = FALSE,
i2i = TRUE,
hide = NULL,
omit = NULL,
omit.col = NA,
...}

## S3 method for class 'ReOs'
weightedmean(
  x,
  random.effects = FALSE,
detect.outliers = TRUE,
  plot = TRUE,
  from = NA,
  to = NA,
  levels = NA,
  clabel = "",
  rect.col = c("#00FF0080", "#FF000080"),
  outlier.col = "#00FFFF80",
  sigdig = 2,
  alpha = 0.05,
  exterr = TRUE,
  ranked = FALSE,
i2i = TRUE,
  hide = NULL,
  omit = NULL,
  omit.col = NA,
...}

## S3 method for class 'SmNd'
weightedmean(
  x,
  random.effects = FALSE,
detect.outliers = TRUE,
  plot = TRUE,
  from = NA,
  to = NA,
  levels = NA,
  clabel = "",
  rect.col = c("#00FF0080", "#FF000080"),
  outlier.col = "#00FFFF80",
  sigdig = 2,
alpha = 0.05,
exterr = TRUE,
ranked = FALSE,
i2i = TRUE,
hide = NULL,
omit = NULL,
omit.col = NA,
...)

## S3 method for class 'RbSr'
weightedmean(
x,
random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
to = NA,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
...)

## S3 method for class 'LuHf'
weightedmean(
x,
random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
to = NA,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
...
weightedmean

i2i = TRUE,
ranked = FALSE,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'UTHHe'
weightedmean(
  x,
  random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
to = NA,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
ranked = FALSE,
hide = NULL,
omit = NULL,
omit.col = NA,
...
)

## S3 method for class 'fissiontracks'
weightedmean(
  x,
  random.effects = FALSE,
detect.outliers = TRUE,
plot = TRUE,
from = NA,
to = NA,
levels = NA,
clabel = "",
rect.col = c("#00FF0080", "#FF000080"),
outlier.col = "#00FFFF80",
sigdig = 2,
alpha = 0.05,
exterr = TRUE,
ranked = FALSE,
hide = NULL,
omit = NULL,
omit.col = NA,
Arguments

x a two column matrix of values (first column) and their standard errors (second column) OR an object of class UPb, PbPb, ThPb, ArAr, KCa, ReOs, SmNd, RbSr, LuHf, ThU, fissiontracks or UThHe

... optional arguments

from minimum y-axis limit. Setting from=NA scales the plot automatically.
to maximum y-axis limit. Setting to=NA scales the plot automatically.

random.effects if TRUE, computes the weighted mean using a random effects model with two parameters: the mean and the dispersion. This is akin to a ‘model-3’ isochron regression.
if FALSE, attributes any excess dispersion to an underestimation of the analytical uncertainties. This akin to a ‘model-1’ isochron regression.

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.

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

clabel label of the colour legend

rect.col Fill colour for the measurements or age estimates. This can either be a single colour or multiple colours to form a colour ramp (to be used if levels!=NA):
a single colour: rgb(0,1,0,0.5), 'FF000080', 'white', etc.;
multiple colours: c(rgb(1,0,0,0.5), rgb(0,1,0,0.5)), c('FF000080', '00FF0080'), c('blue', 'red'), c('blue', 'yellow', 'red'), etc.;
a colour palette: rainbow(n=100), topo.colors(n=100, alpha=0.5), etc.; or
a reversed palette: rev(topo.colors(n=100, alpha=0.5)), etc.
For empty boxes, set rect.col=NA

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.

ranked plot the aliquots in order of increasing age?

hide vector with indices of aliquots that should be removed from the weighted mean plot.

omit vector with indices of aliquots that should be plotted but omitted from the weighted mean calculation.

omit.col colour that should be used for the omitted aliquots.
weightedmean

type scalar indicating whether to plot the $^{207}\text{Pb}/^{235}\text{U}$ age (type=1), the $^{206}\text{Pb}/^{238}\text{U}$ age (type=2), the $^{207}\text{Pb}/^{206}\text{Pb}$ age (type=3), the $^{207}\text{Pb}/^{206}\text{Pb-}^{206}\text{Pb}/^{238}\text{U}$ age (type=4), the concordia age (type=5), or the $^{208}\text{Pb}/^{232}\text{Th}$ age (type=6).
cutoff the age (in Ma) below which the $^{206}\text{Pb}/^{238}\text{U}$ age and above which the $^{207}\text{Pb}/^{206}\text{Pb}$ age is used. This parameter is only used if type=4.
cutoff.disc discordance cutoff filter. This is an object of class discfilter
extern propagate decay constant uncertainties?
common.Pb common lead correction:
0: none
1: use the Pb-composition stored in settings('iratio', 'Pb207Pb206') (if x has class UPb and x$\text{format}<4$); settings('iratio', 'Pb206Pb204') and settings('iratio', 'Pb207Pb204') (if x has class PbPb or x has class UPb and 3<x$\text{format}<7$); or settings('iratio', 'Pb208Pb206') and settings('iratio', 'Pb208Pb207') (if x has class UPb and x$\text{format}=7$ or 8).
2: remove the common Pb by projecting the data along an inverse isochron. Note: choosing this option introduces a degree of circularity in the weighted age calculation. In this case the weighted mean plot just serves as a way to visualise the residuals of the data around the isochron, and one should be careful not to over-interpret the numerical output.
3: use the Stacey-Kramers two-stage model to infer the initial Pb-composition (only applicable if x has class UPb)
i2i 'isochron to intercept': calculates the initial (aka 'inherited', 'excess', or 'common') $^{40}\text{Ar}/^{36}\text{Ar}$, $^{40}\text{Ca}/^{44}\text{Ca}$, $^{207}\text{Pb}/^{204}\text{Pb}$, $^{87}\text{Sr}/^{86}\text{Sr}$, $^{143}\text{Nd}/^{144}\text{Nd}$, $^{187}\text{Os}/^{188}\text{Os}$, $^{230}\text{Th}/^{232}\text{Th}$, $^{176}\text{Hf}/^{177}\text{Hf}$ or $^{204}\text{Pb}/^{208}\text{Pb}$ ratio from an isochron fit. Setting i2i to FALSE uses the default values stored in settings('iratio',...).
Note that choosing this option introduces a degree of circularity in the weighted age calculation. In this case the weighted mean plot just serves as a way to visualise the residuals of the data around the isochron, and one should be careful not to over-interpret the numerical output.
detritus detrital $^{230}\text{Th}$ correction (only applicable when x$\text{format}=1$ or 2).
0: no correction
1: project the data along an isochron fit
2: correct the data using an assumed initial $^{230}\text{Th}/^{232}\text{Th}$-ratio for the detritus.
3: correct the data using the measured present day $^{230}\text{Th}/^{238}\text{U}$, $^{232}\text{Th}/^{238}\text{U}$ and $^{234}\text{U}/^{238}\text{U}$-ratios in the detritus.

Details

Let $\{t_1, \ldots, t_n\}$ be a set of n age estimates determined on different aliquots of the same sample, and let $\{s[t_1], \ldots, s[t_n]\}$ be their analytical uncertainties. IsoplotR then calculates the weighted mean of these data using one of two methods:

1. The ordinary error-weighted mean:

   $$\mu = \frac{\sum(t_i/s[t_i]^2) / \sum(1/s[t_i]^2)}$$
2. A random effects model with two sources of variance:
\[
\log(t_i) \sim N(\log(\mu), \sigma^2 = (s[t_i]/t_i)^2 + \omega^2)
\]
where \( \mu \) is the mean, \( \sigma^2 \) is the total variance and \( \omega \) is the 'overdispersion'. This equation can be solved for \( \mu \) and \( \omega \) by the method of maximum likelihood.

IsoplotR uses a modified version of Chauvenet’s criterion for outlier detection:

1. Compute the error-weighted mean \( (\mu) \) of the \( n \) age determinations \( t_i \) using their analytical uncertainties \( s[t_i] \).
2. For each \( t_i \), compute the probability \( p_i \) that that \( |t_i - \mu| > |t_i - \mu| \) for \( t \sim N(\mu, s[t_i]^2 \text{MSWD}) \) (ordinary weighted mean) or \( \log(t) \sim N(\log(\mu), s[t_i]^2 + \omega^2) \) (random effects model).
3. Let \( p_j \equiv \min(p_1, ..., p_n) \). If \( p_j < 0.05/n \), then reject the \( j^{th} \) date, reduce \( n \) by one (i.e., \( n \to n - 1 \)) and repeat steps 1 through 3 until the surviving dates pass the third step.

If the analytical uncertainties are small compared to the scatter between the dates (i.e. if \( \omega \gg s[t] \) for all \( i \)), then this generalised algorithm reduces to the conventional Chauvenet criterion. If the analytical uncertainties are large and the data do not exhibit any overdispersion, then the heuristic outlier detection method is equivalent to Ludwig (2003)’s ‘2-sigma’ method.

The uncertainty budget of the weighted mean does not include the uncertainty of the initial daughter correction (if any). This uncertainty is neither a purely systematic nor a purely random uncertainty and cannot easily be propagated with conventional geochronological data processing algorithms. This caveat is especially pertinent to chronometers whose initial daughter composition is determined by isochron regression. You may note that the uncertainties of the weighted mean are usually much smaller than those of the isochron. In this case the isochron errors are more meaningful, and the weighted mean mean plot should just be used to inspect the residuals of the data around the isochron.

Value

Returns a list with the following items:

- **mean** a three element vector with:
  - \( t \): the weighted mean. An asterisk is added to the plot title if the initial daughter correction is based on an isochron regression, to mark the circularity of using an isochron to compute a weighted mean.
  - \( s[t] \): the standard error of the weighted mean, excluding the uncertainty of the initial daughter correction. This is because this uncertainty is neither purely random nor purely systematic.
  - \( ci[t] \): the \( 100(1 - \alpha)\% \) confidence interval for \( t \)
- **disp** a three-element vector with the (over)dispersion and the lower and upper half-widths of its \( 100(1 - \alpha)\% \) confidence interval.
- **mswd** the Mean Square of the Weighted Deviates (a.k.a. ‘reduced Chi-square’ statistic)
- **df** the number of degrees of freedom of the Chi-square test for homogeneity (\( df = n − 1 \), where \( n \) is the number of samples).
- **p.value** the p-value of a Chi-square test with \( df \) degrees of freedom, testing the null hypothesis that the underlying population is not overdispersed.
- **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, including mean (the mean value), ci (a grey rectangle with the $100(1-\alpha)$% confidence interval ignoring systematic errors), ci.exterr (a grey rectangle with the $100(1-\alpha)$% confidence interval including systematic errors), dash1 and dash2 (lines marking the $100(1-\alpha)$% confidence interval augmented by $\sqrt{mswd}$ overdispersion if random.effcts=FALSE), and marking the $100(1-\alpha)$% confidence limits of a normal distribution whose standard deviation equals the overdispersion parameter if random.effcts=TRUE).

See Also

central

Examples

ages <- c(251.9,251.59,251.47,251.35,251.04,250.79,250.73,251.22,228.43)
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))

attach(examples)
weightedmean(LudwigMean)

---

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 4 or 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 (optionally) the correlation coefficients of the X- and Y-values.

alpha  cutoff value for confidence intervals

Details

Given n pairs of (approximately) collinear measurements $X_i$ and $Y_i$ (for $1 \leq i \leq n$), their uncertainties $s[X_i]$ and $s[Y_i]$, and their covariances $\text{cov}[X_i, Y_i]$, the york function finds the best fitting straight line using the least-squares algorithm of York et al. (2004). This algorithm is modified from an earlier method developed by York (1968) to be consistent with the maximum likelihood approach of Titterington and Halliday (1979). It computes the MSWD as a measure of under/overdispersion. Overdispersed datasets (MSWD>1) can be dealt with in the same three ways that are described in the documentation of the isochron function.
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 \((n - 2)\)
- **p.value** p-value of a Chi-square value with df degrees of freedom

References


See Also

- `data2york, titterington, isochron, ludwig`

Examples

```r
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)
scatterplot(dat, fit=fit)
```
Index

_PACKAGE (IsoplotR), 54

age, 2, 95
age2ratio, 8
agespectrum, 10
ArAr (classes), 21

cad, 13, 30, 64, 69, 70
central, 8, 18, 76, 77, 86, 110
classes, 21, 90
concordia, 8, 23, 32, 35, 41, 65, 66
data2york, 27, 111
detritals (classes), 21
discfilter, 7, 17, 29, 62, 75, 85
diseq, 6, 9, 21, 22, 31, 66, 67, 90

ejellipse, 33
evolution, 33, 41
eexamples, 36, 91
fissiontracks (classes), 21
heliotplot, 20, 35, 39

is.ArAr (classes), 21
is.detritals (classes), 21
is.diseq (classes), 21
is.fissiontracks (classes), 21
is.KCa (classes), 21
is.LuHf (classes), 21
is.PbPb (classes), 21
is.PD (classes), 21
is.RbSr (classes), 21
is.ReOs (classes), 21
is.SmNd (classes), 21
is.ThPb (classes), 21
is.ThU (classes), 21
is.UPb (classes), 21
is.UTHHe (classes), 21

isochron, 8, 36, 41, 42, 66, 99, 100, 110, 111

IsoplotR, 54
IsoplotR-package (IsoplotR), 54
KCa (classes), 21
kde, 18, 30, 55, 70
ludwig, 32, 54, 64, 100, 111
LuHf (classes), 21
mclean, 32, 66
mds, 67
Pb0corr, 70
PbPb (classes), 21
PD (classes), 21
peakfit, 72, 86

radialplot, 18, 20, 30, 41, 64, 77, 77
RbSr (classes), 21
read.data, 21, 87, 98
ReOs (classes), 21
scatterplot, 91
set.zeta, 94
settings, 91, 95
SmNd (classes), 21

ThPb (classes), 21
ThU (classes), 21
titterington, 54, 66, 99, 111

UPb (classes), 21
UTHHe (classes), 21

weightedmean, 12, 19, 20, 100

york, 29, 54, 100, 110