Package ‘Bchron’

February 19, 2015

Type Package

Title Radiocarbon dating, age-depth modelling, relative sea level rate estimation, and non-parametric phase modelling

Version 4.1.1

Date 2014-04-22

Depends inline

Imports hdrcde, utils, MASS, ellipse, coda, mclust

Author Andrew Parnell

Maintainer Andrew Parnell <Andrew.Parnell@ucd.ie>

Description This package enables quick calibration of radiocarbon dates under various calibration curves (including user generated ones); Age-depth modelling as per the algorithm of Haslett and Parnell (2008); Relative sea level rate estimation incorporating time uncertainty in polynomial regression models; and non-parametric phase modelling via Gaussian mixtures as a means to determine the activity of a site (and as an alternative to the Oxcal function SUM).

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-04-22 18:52:19

R topics documented:

  Bchron-package .................................................. 2
  BchronCalibrate ............................................... 3
  BchronDensity .................................................. 5
  BchronDensityFast ............................................. 6
  Bchronology .................................................... 8
  BchronRSL ...................................................... 11
  Glendalough ................................................... 12
  intcal13 ....................................................... 13
  marine13 .................................................... 14
  normal ........................................................ 14
**Description**

This package enables quick calibration of radiocarbon dates under various calibration curves (including user generated ones); Age-depth modelling as per the algorithm of Haslett and Parnell (2008); Relative sea level rate estimation incorporating time uncertainty in polynomial regression models; and non-parametric phase modelling via Gaussian mixtures as a means to determine the activity of a site (and as an alternative to the Oxcal function SUM).

**Details**

<table>
<thead>
<tr>
<th>Package</th>
<th>Type</th>
<th>Version</th>
<th>Date</th>
<th>License</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bchron</td>
<td>Package</td>
<td>4.0</td>
<td>2014-03-26</td>
<td>GPL (&gt;= 2)</td>
</tr>
</tbody>
</table>

Most important functions are `BchronCalibrate` to calibrate radiocarbon (and non-radiocarbon) dates, `Bchronology` for the age-depth model of Haslett and Parnell (2008), `BchronRSL` to get rate estimates for relative sea level data, `BchronDensity` and `BchronDensityFast` for non-parametric phase modelling of age data. See the help files for these functions for examples

**Author(s)**

Andrew Parnell <andrew.parnell@ucd.ie>

**References**

See individual functions for references and examples
**BchronCalibrate**  

**Fast radiocarbon calibration**

**Description**

A fast function for calibrating large numbers of radiocarbon dates involving multiple calibration curves.

**Usage**

```r
BchronCalibrate(ages, ageSds, calCurves, ids = NULL, positions = NULL, pathToCalCurves = system.file('data', package = "Bchron"), eps = 1e-05, dfs = rep(100, length(ages)))
```

**Arguments**

- `ages`: A vector of ages (most likely 14C).
- `ageSds`: A vector of 1-sigma values for the ages given above.
- `calCurves`: A vector of values containing either 'intcal13', 'shcal13', 'marine13', or 'normal'. Should be the same length the number of ages supplied. Non-standard calibration curves can be used provided they are supplied in the same format as those previously mentioned and are placed in the same directory. Normal indicates a normally-distributed (non-14C) age.
- `ids`: (optional) ID names for each age.
- `positions`: (optional) Position values (e.g. depths) for each age.
- `pathToCalCurves`: (optional) File path to where the calibration curves are located. Defaults to the system directory where the 3 standard calibration curves are stored.
- `eps`: (optional) Cut-off point for density calculation. A value of eps>0 removes ages from the output which have negligible probability density.
- `dfs`: (optional) Degrees-of-freedom values for the t-distribution associated with the calibration calculation. A large value indicates Gaussian distributions assumed for the 14C ages.

**Details**

This function provides a direct numerical integration strategy for computing calibrated radiocarbon ages. The steps for each 14C age are approximately as follows: 1) Create a grid of ages covering the range of the calibration curve 2) Calculate the probability of each age according to the 14C age, the standard deviation supplied and the calibration curve 3) Normalise the probabilities so that they sum to 1 4) Remove any probabilities that are less than the value given for eps Multiple calibration curves can be specified so that each 14C age can have a different curve. For ages that are not 14C, use the ‘normal’ calibration curve which treats the ages as normally distributed with given standard deviation.
Value

A list of lists where each element corresponds to a single age. Each element contains:

- **ages** The original age supplied
- **ageSDs** The original age standard deviation supplied
- **positions** The position of the age (usually the depth)
- **calCurves** The calibration curve used for that age
- **ageGrid** A grid of age values over which the density was created
- **densities** A vector of probability values indicating the probability value for each element in ageGrid
- **ageLab** The label given to the age variable
- **positionLab** The label given to the position variable

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References

Forthcoming!

See Also

`Bchronology, BchronRSL, BchronDensity, BchronDensityFast`

Examples

```r
# Calibrate a single age
ages1 = BchronCalibrate(ages=11553, ageSDs=230, calCurves='intcal13', ids='Date-1')
summary(ages1)
plot(ages1)

# Calibrate multiple ages with different calibration curves
ages2 = BchronCalibrate(ages=c(3445,11553,7456), ageSDs=c(50,230,110),
                          calCurves=c('intcal13','intcal13','shcal13'))
summary(ages2)
plot(ages2)

# Calibrate multiple ages with multiple calibration curves and including depth
ages3 = BchronCalibrate(ages=c(3445,11553), ageSDs=c(50,230), positions=c(100,150),
                          calCurves=c('intcal13','normal'))
summary(ages3)
plot(ages3, withDepths=TRUE)
```
This function runs a non-parametric phase model on 14C and non-14C ages via Gaussian Mixture density estimation.

**Usage**

```r
BchronDensity(ages, ageSds, calCurves, pathToCalCurves = system.file('data', package = "Bchron"), dfs = rep(100, length(ages)), numMix = 30, iterations = 10000, burn = 2000, thin = 8, updateAges = FALSE)
```

**Arguments**

- `ages`: A vector of ages (most likely 14C)
- `ageSds`: A vector of 1-sigma values for the ages given above
- `calCurves`: A vector of values containing either 'intcal13', 'shcal13', 'marine13', or 'normal'. Should be the same length the number of ages supplied. Non-standard calibration curves can be used provided they are supplied in the same format as those previously mentioned and are placed in the same directory. Normal indicates a normally-distributed (non-14C) age.
- `pathToCalCurves`: (optional) File path to where the calibration curves are located. Defaults to the system directory where the 3 standard calibration curves are stored.
- `dfs`: (optional) Degrees-of-freedom values for the t-distribution associated with the calibration calculation. A large value indicates Gaussian distributions assumed for the 14C ages
- `numMix`: (optional) The number of mixture components in the phase model. Might need to be increased if the data set is large and the phase behaviour is very complex
- `iterations`: The number of iterations to run for
- `burn`: The number of starting iterations to discard
- `thin`: The step size of iterations to keep
- `updateAges`: Whether or not to update ages as part of the MCMC run. Default is FALSE. Changing this to TRUE will improve performance but will fit a slightly invalid model

**Details**

This model places a Gaussian mixture prior distribution on the calibrated ages and so estimates the density of the overall set of radiocarbon ages. It is designed to be a probabilistic version of the Oxcal SUM command which takes calibrated ages and sums the probability distributions with the aim of estimating activity through age as a proxy.
Value

output = list(theta = thetaStore, p = pStore, mu = mu, calAges = xSmall, G = G) class(output) = "BchronDensityRun" An object of class BchronDensityRun with the following elements:

- `theta`: The posterior samples of the restricted ages
- `p`: Posterior samples of the mixture proportions
- `mu`: Values of the means of each Gaussian mixture
- `calAges`: The calibrated ages from `BchronCalibrate`
- `G`: The number of mixture components. Equal to `numMix`

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References

Forthcoming!

See Also

`Bchronology`, `BchronRSL`, `BchronDensityFast` for a faster approximate version of this function

Examples

```r
## Not run: # Read in some data from Sluggan Moss
data(sluggan)

# Run the model
SlugDens = BchronDensity(ages=sluggan$ages, ageSds=sluggan$ageSds, calCurves=sluggan$calCurves, numMix=50)

# plot it
plot(SlugDens)
## End(Not run)
```

---

**BchronDensityFast**

Non-parametric phase model (faster version)

Description

This function runs a non-parametric phase model on 14C and non-14C ages via Gaussian Mixture density estimation through the mclust package.

Usage

```r
BchronDensityFast(ages, ageSds, calCurves, pathToCalCurves = system.file('data', package = "Bchron"), dfs = rep(100, length(ages)), samples = 2000, G = 30)
```
**BchronDensityFast**

### Arguments

- **ages**: A vector of ages (most likely 14C)
- **ageSds**: A vector of 1-sigma values for the ages given above
- **calCurves**: A vector of values containing either 'intcal13', 'shcal13', 'marine13', or 'normal'. Should be the same length the number of ages supplied. Non-standard calibration curves can be used provided they are supplied in the same format as those previously mentioned and are placed in the same directory. Normal indicates a normally-distributed (non-14C) age.
- **pathToCalCurves** *(optional)*: File path to where the calibration curves are located. Defaults to the system directory where the 3 standard calibration curves are stored.
- **dfs** *(optional)*: Degrees-of-freedom values for the t-distribution associated with the calibration calculation. A large value indicates Gaussian distributions assumed for the 14C ages
- **samples** *(optional)*: Number of samples of calibrated dates required
- **G** *(optional)*: Number of Gaussian mixture components

### Details

This is a faster approximate version of `BchronDensity` that uses the `densityMclust` function to compute the Gaussian mixtures for a set of calibrated ages. The method is an approximation as it does not fit a fully Bayesian model as `BchronDensity` does. It is designed to be a probabilistic version of the Oxcal SUM command which takes calibrated ages and sums the probability distributions with the aim of estimating activity through age as a proxy.

### Value

An object of class `BchronDensityRunFast` with the following components:

- **out**: The output from the run of `densityMclust` with the given number of mixture components
- **calAges**: The calibrated ages from the `BchronDensity` function

### Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

### References

See the mclust package for references about this method.

### See Also

`Bchronology, BchronCalibrate, BchronRSL, BchronDensity` for a slower exact version of this function
Bchronology

Runs the Compound Poisson-Gamma chronology model of Haslett and Parnell (2008)

Description

Fits a non-parametric chronology model to age/position data according to the Compound Poisson-Gamma model defined by Haslett and Parnell (2008). This version uses a slightly modified Markov chain Monte Carlo fitting algorithm which aims to converge quicker and requires fewer iterations. It also a slightly modified procedure for identifying outliers.

Usage

Bchronology(ages, ageSds, positions, positionThicknesses = rep(0, length(ages)), calCurves = rep("intcal13", length(ages)), ids = NULL, outlierProbs = rep(0.01, length(ages)), predictPositions = seq(min(positions), max(positions), length = 100), pathToCalCurves = system.file("data", package = "Bchron"), iterations = 10000, burn = 2000, thin = 8, extractDate = 1950 - as.numeric(format(Sys.time(), "%Y")), maxExtrap = 500, thetaMhSd = 0.5, muMhSd = 0.1, psiMhSd = 0.1, ageScaleVal = 1000, positionScaleVal = 100)

Arguments

- **ages**
  A vector of ages (most likely 14C)
- **ageSds**
  A vector of 1-sigma values for the ages given above
- **positions**
  Position values (e.g. depths) for each age
- **positionThicknesses**
  (optional) Thickness values for each of the positions. By default set to zero
- **calCurves**
  A vector of values containing either 'intcal13', 'shcal13', 'marine13', or 'normal'. Should be the same length the number of ages supplied. Non-standard calibration curves can be used provided they are supplied in the same format as those previously mentioned and are placed in the same directory. Normal indicates a normally-distributed (non-14C) age.

Examples

```r
## Not run: # Read in some data from Sluggan Moss
data(Sluggan)

# Run the model
SlugDensFast = BchronDensityFast(ages=Sluggan$ages, ageSds=Sluggan$ageSds, calCurves=Sluggan$calCurves)

# plot it
plot(SlugDensFast)
## End(Not run)
```
The `Bchronology` function fits a compound Poisson-Gamma distribution to the increments between the dated levels. This involves a stochastic linear interpolation step where the age gaps are Gamma distributed, and the position gaps are Exponential. Radiocarbon and non-radiocarbon dates (including outliers) are updated within the function also by MCMC.

A list of class `BchronologyRun` which include elements:

- `theta` The posterior estimated values of the ages
- `phi` The posterior estimated outlier values (1=outlier, 2=not outlier). The means of this parameter give the posterior estimated outlier probabilities

Details

The `Bchronology` function fits a compound Poisson-Gamma distribution to the increments between the dated levels. This involves a stochastic linear interpolation step where the age gaps are Gamma distributed, and the position gaps are Exponential. Radiocarbon and non-radiocarbon dates (including outliers) are updated within the function also by MCMC.
The posterior values of the Compound Poisson-Gamma mean

The posterior values of the Compound Poisson-Gamma scale

The posterior estimated ages for each of the values in predictPosition

The positions at which estimated ages were required

The calibrated ages as output from `BchronCalibrate`

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References


See Also

`BchronCalibrate`, `BchronRSL`, `BchronDensity`, `BchronDensityFast`

Examples

```r
## Not run:
# Data from Glendalough
data(Glendalough)

# Run in Bchronology - all but first age uses intcal13
GlenOut = Bchronology(ages=Glendalough$ages, ageSds=Glendalough$ageSds, calCurves=Glendalough$calCurves, positions=Glendalough$position, positionThicknesses=Glendalough$thickness, ids=Glendalough$id, predictPositions=seq(0, 1500, by=10))

# Summarise it a few different ways
summary(GlenOut) # Default is for quantiles of ages at predictPosition values
summary(GlenOut, type='convergence') # Check model convergence
summary(GlenOut, type='outliers') # Look at outlier probabilities

# Predict for some new positions
predictAges = predict(GlenOut, newPositions = c(150, 725, 1500), newPositionThicknesses=c(5, 0, 20))

# Plot the output
plot(GlenOut, main="Glendalough", xlab='Age (cal years BP)', ylab='Depth (cm)', las=1)
## End(Not run)
```
Description

This function takes output from a `Bchronology` run, some RSL mean and standard deviations, and fits an errors-in-variables polynomial regression model which takes account of the age uncertainty. It then estimates rates and accelerations in RSL with appropriate uncertainty quantification.

Usage

```r
BchronRSL(BchronologyRun, RSLmean, RSLsd, degree = 1, iterations = 10000, burn = 2000, thin = 8)
```

Arguments

- `BchronologyRun`: Output from a run of `Bchronology`
- `RSLmean`: A vector of RSL mean estimates of the same length as the number of predictions given to the `Bchronology` function
- `RSLsd`: A vector RSL standard deviations of the same length as the number of predictions given to the `Bchronology` function
- `degree`: (optional) The degree of the polynomial regression: linear=1 (default), quadratic=2, etc. Supports up to degree 5, though this will depend on the data given
- `iterations`: (optional) The number of MCMC iterations to run
- `burn`: (optional) The number of starting iterations to discard
- `thin`: (optional) The step size of iterations to discard

Details

This function fits an errors-in-variables regression model to relative sea level (RSL) data. An errors-in-variables regression model allows for uncertainty in the explanatory variable, here the age of sea level data point. The algorithm is more fully defined in the reference below.

Value

An object of class `BchronRSLRun` with elements

- `BchronologyRun`: The output from the run of `Bchronology`
- `samples`: The posterior samples of the regression parameters
- `degree`: The degree of the polynomial regression
- `RSLmean`: The RSL mean values given to the function
- `RSLsd`: The RSL standard deviations as given to the function
- `const`: The mean of the predicted age values. Used to standardise the design matrix and avoid computational issues
Author(s)
Andrew Parnell <andrew.parnell@ucd.ie>

References
level reconstructions from salt marsh sediments’ In: I. Shennan, B.P. Horton, and A.J. Long (eds).
Handbook of Sea Level Research. Chichester: Wiley

See Also
BchronCalibrate, Bchronology, BchronDensity, BchronDensityFast

Examples
```r
## Not run: # Load in data
data(TestChronData)
data(TestRSLData)

# Run through Bchronology
RSLrun = Bchronology(ages=TestChronData$ages, ageSds=TestChronData$ageSds,
  positions=TestChronData$position, positionThicknesses=TestChronData$thickness,
  ids=TestChronData$id, calCurves=TestChronData$calCurves,
  predictPositions=TestRSLData$Depth)

# Now run through BchronRSL
RSLrun2 = BchronRSL(RSLrun, RSLmean=TestRSLData$RSL, RSLsd=TestRSLData$Sigma, degree=3)

# Summarise it
summary(RSLrun2)

# Plot it
plot(RSLrun2)
## End(Not run)
```

Glendalough

Glendalough data

Description
Chronology data for Glendalough data set

Usage
data(Glendalough)
Format

A data frame with 6 observations on the following 6 variables.

id  ID of each age
ages Age in (14C) years BP
agesds Age standard deviations
position Depths in cm
thickness Thicknesses in cm
calCurves Calibration curve for each age

Details

This Glendalough data has been used with Bchronology or BchronDensity

References


Description

Northern hemisphere 2013 calibration curve

Usage

data(intcal13)

Format

A data frame with 5141 observations on 5 variables.

Details

For full details and reference see http://www.radiocarbon.org/IntCal13.htm. For usage details see BchronCalibrate
### marine13

*Marine 2013 calibration curve*

**Description**

Marine 2013 calibration curve

**Usage**

```r
data(marine13)
```

**Format**

A data frame with 4801 observations on 5 variables.

**Details**


### normal

*Data for dummy calibration of normally distributed ages*

**Description**

Data for dummy calibration of normally distributed ages

**Usage**

```r
data(normal)
```

**Format**

A data frame with 2 observations on 3 variables.

**Details**

This is dummy data so that `BchronCalibrate` can calibrate normally distributed dates.
Description

Plots calibrated radiocarbon dates from a `BchronCalibrate` run. Has options to plot on a position (usually depth) scale if supplied with the original run.

Usage

```r
## S3 method for class 'BchronCalibratedDates'
plot(x, withPositions = FALSE, ...)
```

Arguments

- `x` Output from `BchronCalibrate`
- `withPositions` Whether to plot with positions (i.e. using the position values as the y axis). Default is `FALSE` in which case it will produce a sequence of plots, one for each calibrated age
- `...` Other arguments to plot, see `par`.

Details

These plots are intended to be pretty basic and used simply for quick information. Users are encouraged to learn the R plotting features to produce publication quality graphics.

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

References

Forthcoming!

See Also

`BchronCalibrate`, `Bchronology`, `BchronRSL`, `BchronDensity`, `BchronDensityFast`

Examples

```r
# Examples in \code{\link{BchronCalibrate}}
```
plot.BchronDensityRunFast

Description

Plot output from BchronDensity

Usage

## S3 method for class 'BchronDensityRun'
plot(x, plotDates = TRUE, plotSum = FALSE, ...)

Arguments

x
plotDates (optional) Whether to plot the individual calibrated dates (default TRUE)
plotSum (optional) Whether to plot the sum of the probability distributions (default FALSE)
...

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

See Also

Bchronology, BchronRSL, BchronDensity, BchronDensityFast for a faster approximate version of this function

Examples

# Examples in \code{\link{BchronDensity}}

plot.BchronDensityRunFast

Description

Plot run from BchronDensityFast

Usage

## S3 method for class 'BchronDensityRunFast'
plot(x, plotDates = TRUE, plotSum = FALSE, ...)

Examples

# Examples in \code{\link{BchronDensityFast}}
Arguments

x  Output from BchronDensityFast
plotDates (optional) Whether to include individual age pdfs (default TRUE)
plotSum (optional) Whether to include sum of age pdfs (default FALSE)
... Other graphical parameters, see par

Details

Creates a basic plot of output for a run of BchronDensityFast

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

See Also

Bchronology, BchronRSL, BchronDensity, BchronDensityFast for a faster approximate version of this function

Examples

# Examples in \code{\link{BchronDensityFast}}

plot.BchronologyRun  Plot output from Bchronology

Description

Plots output from a run of Bchronology

Usage

## S3 method for class 'BchronologyRun'
plot(x, ...)

Arguments

x  The object created by Bchronology
... Other graphical parameters as detailed in par

Details

Creates a simple plot of the chronology output. More detailed plots can be created by manipulating the Bchronology object as required
Author(s)
Andrew Parnell <andrew.parnell@ucd.ie>

See Also
BchronCalibrate, Bchronology, BchronRSL, BchronDensity, BchronDensityFast

Examples
# Examples in main function \code{\link{Bchronology}}

plot.BchronRSLRun

---

Description
Plot output from the BchronRSL function

Usage
```r
## S3 method for class 'BchronRSLRun'
plot(x, xlab = "Age (cal BP)", ylab = "Depth (m)", ...)
```

Arguments
- `x` An object created by BchronRSL
- `xlab` (optional) Label for the x-axis of the plot
- `ylab` (optional) Label for the y-axis of the plot
- `...` Other arguments to plot, see \code{par}

Author(s)
Andrew Parnell <andrew.parnell@ucd.ie>

See Also
BchronCalibrate, Bchronology, BchronRSL, BchronDensity, BchronDensityFast

Examples
# Examples in \code{\link{BchronRSL}}
predict.BchronologyRun

Predict ages of other positions for a BchronologyRun object

Description

This function will predict the ages of new positions (usually depths) based on a previous run of the function Bchronology. It will also allow for thickness uncertainties to be included in the resulting ages, for example when the age of a particular event is desired.

Usage

## S3 method for class 'BchronologyRun'
predict(object, newPositions, newPositionThicknesses = NULL, maxExtrap = 500, ...)

Arguments

- **object**: Output from a run of Bchronology
- **newPositions**: A vector of new positions at which to find ages
- **newPositionThicknesses**: A vector of thicknesses for the above positions. Must be the same length as newPositions
- **maxExtrap**: The maximum number of extrapolation attempts. It might be worth increasing this if you are extrapolating a long way from the other dated positions
- **...**: Other arguments to predict (not currently supported)

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

See Also

BchronCalibrate, Bchronology, BchronRSL, BchronDensity, BchronDensityFast

Examples

# Examples in main function 
\code{\link{Bchronology}}
---

**shcal13**  
*Southern hemisphere 2013 calibration curve*

**Description**

Southern hemisphere 2013 calibration curve

**Usage**

```r
data(shcal13)
```

**Format**

A data frame with 5141 observations on 5 variables.

**Details**

For full details and reference see [http://www.radiocarbon.org/IntCal13.htm](http://www.radiocarbon.org/IntCal13.htm). For usage details see *BchronCalibrate*

---

**Sluggan**  
*Sluggan Moss data*

**Description**

Chronology data for Sluggan Moss data set

**Usage**

```r
data(Sluggan)
```

**Format**

A data frame with 31 observations on the following 6 variables.

<table>
<thead>
<tr>
<th>id</th>
<th>ID of each age</th>
</tr>
</thead>
<tbody>
<tr>
<td>ages</td>
<td>Age in (14C) years BP</td>
</tr>
<tr>
<td>ageSds</td>
<td>Age standard deviations</td>
</tr>
<tr>
<td>position</td>
<td>Depths in cm</td>
</tr>
<tr>
<td>thickness</td>
<td>Thicknesses in cm</td>
</tr>
<tr>
<td>calCurves</td>
<td>Calibration curve for each age</td>
</tr>
</tbody>
</table>

**Details**

This Sluggan Moss data can be downloaded from the European Pollen Database: [www.europeanpollendatabase.net](http://www.europeanpollendatabase.net). For usage see *Bchronology* or *BchronDensity*
**summary.BchronCalibratedDates**

### References


---

**summary.BchronCalibratedDates**

*Summarise a BchronCalibrate object*

---

### Description

Produces summary output from a `BchronCalibrate` run, including the highest density regions for the calibrated ages for given probability levels.

### Usage

```r
## S3 method for class 'BchronCalibratedDates'
summary(object, prob = c(50, 95, 99), ...,
         digits = max(3,getOption("digits") - 3))
```

### Arguments

- `object` The output of a run of `BchronCalibrate`
- `prob` A vector of percentage values (between 0 and 100) at which the highest density regions for each age are calculated
- `...` Further arguments (not currently supported)
- `digits` Significant digits to display (not currently supported)

### Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

### References

Forthcoming!

### See Also

`BchronCalibrate, Bchronology, BchronRSL, BchronDensity, BchronDensityFast`

### Examples

```r
# Examples in \code{\link{BchronCalibrate}}
```
Summary of a Bchronology object

Description

Summarise a Bchronology object

Usage

```r
## S3 method for class 'BchronologyRun'
summary(object, type = c('quantiles', 'outliers', 'convergence'),
probs = c(0.025, 0.1, 0.5, 0.9, 0.975), ..., digits = max(3,getOption("digits") - 3))
```

Arguments

- `object`: Output from a run of Bchronology
- `type`: (optional) Type of output required. The default (quantiles) gives the quantiles of the ages for each depth in predictPositions from Bchronology. The other options provide outlier probabilities or convergence diagnostics.
- `probs`: (optional) Probabilities (between 0 and 1) at which to summarise the predicted chronologies
- `...`: Other arguments (not currently supported)
- `digits`: Number of digits to report values (not currently supported)

Author(s)

Andrew Parnell <andrew.parnell@ucd.ie>

See Also

BchronCalibrate, Bchronology BchronRSL, BchronDensity, BchronDensityFast

Examples

```r
# Examples in main function \code{\link{Bchronology}}
```
summary.BchronRSLRun  

**Summary**

**Description**

Summarise a `BchronRSL` run

**Usage**

```r
## S3 method for class 'BchronRSLRun'
summary(object, ...)  
```

**Arguments**

- `object`: The output from a run of `BchronRSL`
- `...`: (optional) Other arguments to functions (not currently implemented)

**Author(s)**

Andrew Parnell <andrew.parnell@ucd.ie>

**See Also**

`BchronCalibrate`, `Bchronology`, `BchronRSL`, `BchronDensity`, `BchronDensityFast`

**Examples**

```r
# Examples in \code{\link{BchronRSL}}
```

---

TestChronData  

**Example chronology file for use with the BchronRSL function.**

**Description**

Some example chronology data for use with the `BchronRSL` function

**Usage**

```r
data(Glendalough)
```
**Format**

A data frame with 27 observations on the following 6 variables.

- **id**: ID names
- **ages**: Ages in years BP
- **ageSds**: Ages standard deviations in years BP
- **position**: Depths in cm
- **thickness**: Thicknesses in cm
- **calCurves**: Calibration curve for each age

**References**


**Examples**

```R
data(TestChronData)
# See \code{\link{BchronRSL}}
```

---

<table>
<thead>
<tr>
<th>TestRSLData</th>
<th>Relative sea level data</th>
</tr>
</thead>
</table>

**Description**

A set of relative sea level data for use with BchronRSL

**Usage**

```R
data(Glendalough)
```

**Format**

A data frame with 24 observations on the following 3 variables.

- **Depth**: Depth in cm
- **RSL**: Relative sea level in m
- **Sigma**: Standard deviation of RSL measurement

**References**

Examples

```r
data(TestChronData)
# See \code{\link{BchronRSL}}
```
Index

*Topic datasets
  Glendalough, 12
  intcal13, 13
  marine13, 14
  normal, 14
  shcal13, 20
  Sluggan, 20
  TestChronData, 23
  TestRSLData, 24

Bchron (Bchron-package), 2
Bchron-package, 2
BchronCalibrate, 2, 3, 6, 7, 10, 12–15, 18–23
BchronDensity, 2, 4, 5, 7, 10, 12, 13, 15–23
BchronDensityFast, 2, 4, 6, 6, 10, 12, 15–19, 21–23
Bchronology, 2, 4, 6, 7, 8, 11–13, 15–23
BchronRSL, 2, 4, 6, 7, 10, 11, 15–19, 21–24

densityMclust, 7
Glendalough, 12
intcal13, 13
marine13, 14

normal, 14

par, 15–18
plot.BchronCalibratedDates, 15
plot.BchronDensityRun, 16
plot.BchronDensityRunFast, 16
plot.BchronologyRun, 17
plot.BchronRSLRun, 18
predict.BchronologyRun, 19

shcal13, 20
Sluggan, 20
summary.BchronCalibratedDates, 21
summary.BchronologyRun, 22

summary.BchronRSLRun, 23
TestChronData, 23
TestRSLData, 24