Package ‘rbacon’

May 18, 2025

Type Package

Title Age-Depth Modelling using Bayesian Statistics

Version 3.5.2

Author Maarten Blaauw [aut, cre] (ORCID:
<https://orcid.org/0000-0002-5680-1515>),
J. Andres Christen [aut, ctb, cph] (ORCID:
<https://orcid.org/0000-0002-5795-4345>),
Marco A. Aquino Lopez [aut] (ORCID:
<https://orcid.org/0000-0002-5076-7205>),
Judith Esquivel Vazquez [ctb],
Oscar M. Gonzalez V. [ctb],
Ted Belding [cph],
James Theiler [cph],
Brian Gough [cph],
Charles Karney [cph]

Description An approach to age-depth modelling that uses Bayesian statistics to reconstruct accumulation histories for deposits, through combining radiocarbon and other dates with prior information on accumulation rates and their variability. See Blaauw & Christen (2011).

Encoding UTF-8

Repository CRAN

License GPL (>= 2)

Language en-GB

NeedsCompilation yes

Imports coda (>= 0.19-1), graphics, grDevices, Rcpp (>= 0.12.12), stats, utils, data.table, rintcal (>= 1.1.3), rice (>= 1.1.1)

Depends R (>= 3.5.0)

LinkingTo Rcpp

RoxygenNote 7.3.2

Suggests knitr, rmarkdown, utf8

VignetteBuilder knitr

Maintainer Maarten Blaauw <`maarten.blaauw@qub.ac.uk`>

Date/Publication 2025-05-18 17:50:02 UTC

Contents

`A.modelled` *Calculate modelled 210Pb*

Description

Calculate modelled 210Pb values of a sample slice, based on the parameters of the age-model (i.e., time passed since deposition of the bottom and top of the slice), supported and influx

Usage

```
A.modelled(
  d.top,
  d.bottom,
  dens,
  set = get("info"),
  phi = set$phi,
  sup = set$ps
)
```

Arguments

`d.top`	top depth of the slice
`d.bottom`	bottom depth of the slice
`dens`	Density of the slice (in g/cm3)
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`phi`	The modelled values of the 210Pb influx
`sup`	The modelled values of the supported 210Pb

Value

a list of modelled values of A

Author(s)

Maarten Blaauw

`accrate.age` *Obtain estimated accumulation rates for any age of a core.*

Description

Obtain accumulation rates (in years per cm, so actually sedimentation times) as estimated by the MCMC iterations for any age of a core.

Usage

```
accrate.age(
  age,
  set = get("info"),
  cmyr = FALSE,
  ages = c(),
  BCAD = set$BCAD,
  silent = TRUE,
  na.rm = FALSE
)
```

Arguments

`age`	The age for which the accumulation rates need to be returned.
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`cmyr`	Accumulation rates can be calculated in cm/year or year/cm. By default `cmyr=FALSE` and accumulation rates are calculated in year per cm.
`ages`	The ages of the age-depth model. Not provided by default, but can be provided to speed things up if the function is called repeatedly
`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`silent`	Warn when ages are outside the core's range. Default `silent=TRUE`.
`na.rm`	Remove NA entries. These are NOT removed by default, ensuring that always the same amount of iterations is returned.

Details

Considering accumulation rates is crucial for age-depth modelling, and even more so if they are subsequently used for calculating proxy influx values, or interpreted as proxy for environmental change such as carbon accumulation. See also `accrate.age.ghost`, `accrate.depth` and `accrate.depth.ghost`. Bacon deals explicitly with accumulation rate and its variability through defining prior distributions. This function obtains accumulation rates (in years per cm, so actually sedimentation times) as estimated by the MCMC iterations for any age of a core. Deals with only 1 age at a time. See also `accrate.depth`.

Value

all MCMC estimates of accumulation rate of the chosen age.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(yr.res=50, d.res=50, d.by=10)
  accrate.a5000 <- accrate.age(5000)
  plot(accrate.a5000, pch='.')
  hist(accrate.a5000)

## End(Not run)
```

`accrate.age.ghost` *Plot a core's accumulation rates against calendar time.*

Description

Plot a grey-scale representation of a core's estimated accumulation rates against time.

Usage

```
accrate.age.ghost(
  set = get("info"),
  age.lim = c(),
  age.lab = c(),
  kcal = FALSE,
  age.res = 400,
  acc.res = 200,
  cutoff = 0.001,
  dark = 1,
  rgb.scale = c(0, 0, 0),
  rgb.res = 100,
  prob = 0.95,
  plot.range = TRUE,
  range.col = grey(0.5),
  range.lty = 2,
  plot.mean = TRUE,
  mean.col = "red",
  mean.lty = 2,
  plot.median = TRUE,
  median.col = "blue",
  median.lty = 2,
  acc.lim = c(),
  acc.lab = c(),
  BCAD = set$BCAD,
  cmyr = FALSE,
  rotate.axes = FALSE,
  rev.age = FALSE,
  rev.acc = FALSE,
  use.raster = FALSE,
  flip.acc = FALSE,
  flip.age = FALSE,
  xaxs = "i",
  yaxs = "i",
  bty = "l"
)
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable info.
`age.lim`	Minimum and maximum calendar age ranges, calculated automatically by default (`age.lim=c()`).
`age.lab`	The labels for the calendar axis (default `age.lab="cal BP"` or `"BC/AD"` if `BCAD=TRUE`).
`kcal`	Use kcal BP. Default is `kcal=FALSE`.
`age.res`	Resolution or amount of greyscale pixels to cover the age scale of the plot. Default `age.res=400`.
`acc.res`	Resolution or amount of greyscale pixels to cover the accumulation rate scale plot. Default `age.res=400`.
`cutoff`	Point below which colours will no longer be printed. Default `cutoff=0.001`.
`dark`	The darkest grey value is dark=1 by default; lower values will result in lighter grey but values >1 are not advised.
`rgb.scale`	The function to produce a coloured representation of all age-models. Needs 3 values for the intensity of red, green and blue. Defaults to grey-scales: `rgb.scale=c(0,0,0)`, but could also be, say, scales of red (`rgb.scale=c(1,0,0)`).
`rgb.res`	Resolution of the colour spectrum depicting the age-depth model. Default `rgb.res=100`.
`prob`	Probability ranges. Defaults to `prob=0.95`.
`plot.range`	If `plot.range=TRUE`, the confidence ranges (two-tailed; half of the probability at each side) are plotted.
`range.col`	Colour of the confidence ranges.
`range.lty`	Line type of the confidence ranges.
`plot.mean`	If `plot.mean=TRUE`, the means are plotted.
`mean.col`	Colour of the mean accumulation rates.
`mean.lty`	Type of the mean lines.
`plot.median`	If `plot.mean=TRUE`, the medians are plotted.
`median.col`	Colour of the median accumulation rates.
`median.lty`	Type of the median lines.
`acc.lim`	Axis limits for the accumulation rates.
`acc.lab`	Axis label for the accumulation rate.
`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`cmyr`	Accumulation rates can be calculated in cm/year or year/cm. By default `cmyr=FALSE` and accumulation rates are calculated in year per cm. Axis limits are difficult to calculate when `cmyr=TRUE`, so a manual adaptation of `acc.lim` might be a good idea.
`rotate.axes`	The default is to plot the calendar age horizontally and accumulation rates vertically. Change to `rotate.axes=TRUE` value to rotate axes.
`rev.age`	The direction of the age axis, which can be reversed using `rev.age=TRUE`.

`rev.acc`	The direction of the accumulation rate axis, which can be reversed (`rev.acc=TRUE`.
`use.raster`	Rasters can be aligned or not in the underlying image function. Setting `use.raster=FALSE` takes a bit longer to draw and sometimes causes strange lines owing to anti-aliasing. Therefore, `use.raster=TRUE` is the default, however on some devices (e.g., OSX quartz) this causes greyscales to 'flip'. If this is the case, use 'flip.acc=TRUE'.
`flip.acc`	When using `use.raster=TRUE`, sometimes greyscales are flipped. If this is the case, see if setting `flip.acc=TRUE` solves this.
`flip.age`	When using `use.raster=TRUE`, sometimes greyscales are flipped. If this is the case, see if setting `flip.age=TRUE` solves this.
`xaxs`	Extension of the x-axis. White space can be added to the vertical axis using `xaxs="r"`.
`yaxs`	Extension of the y-axis. White space can be added to the vertical axis using `yaxs="r"`.
`bty`	Type of box to be drawn around the plot (`"n"` for none, and `"l"` (default), `"7"`, `"c"`, `"u"`, or `"o"` for correspondingly shaped boxes).

Details

Calculating accumulation rates against calendar age will take some time to calculate, and might show unexpected rates around the core's maximum ages (only a few of all age-model iterations will reach such ages and they will tend to have modelled accumulation rates for the lower depths much lower than the other iterations). Axis limits for accumulation rates are estimated automatically, however upper limits can be very variable (and thus hard to predict) if calculated in `cm/yr`. Therefore you might want to manually adapt the axis limits after plotting with default settings (e.g., `acc.lim=c(0,1)`). See also `accrate.depth.ghost`, `accrate.depth` and `accrate.age`. The greyscale reconstruction around the oldest ages of any reconstruction often indicates very low accumulation rates. This is due to only some MCMC iterations reaching those old ages, and these iterations will have modelled very slow accumulation rates. Currently does not deal well with hiatuses, so do not interpret accumulation rates close to depths with inferred hiatuses.

Value

A greyscale plot of accumulation rate against calendar age, and (invisibly) the list of ages and their accumulation rates (ranges, medians, means).

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(age.res=20, d.res=20, d.by=10)
  layout(1)
  tmp <- accrate.age.ghost(age.res=200, acc.res=100)
  head(tmp)
```

```
## End(Not run)
```

`accrate.age.summary` *Provide a summary of the estimated accumulation rates for any age of a core.*

Description

Obtain a summary (95% range, 68% range, median, mean) of the accumulation rates (in years per cm, so actually sedimentation times) as estimated by the MCMC iterations for any age of a core.

Usage

```
accrate.age.summary(
  age,
  set = get("info"),
  cmyr = FALSE,
  na.rm = TRUE,
  probs = c(0.025, 0.16, 0.84, 0.975, 0.5)
)
```

Arguments

`age`	The age for which accumulation rates need to be returned.
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`cmyr`	Accumulation rates can be calculated in cm/year or year/cm. By default `cmyr=FALSE` and accumulation rates are calculated in year per cm.
`na.rm`	Remove NA entries. These are NOT removed by default, so that always the same amount of iterations is returned.
`probs`	The probability ranges to be returned. Defaults to the minima and maxima of the 95% and 68% ranges, as well as the median: `probs=c(.025, .16, .84, .975, .5)`.

Value

A summary of the estimated accumulation rate of the chosen depth: minimum of the 95% interval, minimum of the 68% interval, maximum of the 68% interval, maximum of the 95% interval, median (i.e., 50%) and mean.

Author(s)

Maarten Blaauw

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(yr.res=50, d.res=50, d.by=10)
  accrate.age.summary(5000)

## End(Not run)
```

`accrate.depth` *Obtain estimated accumulation rates as for any depth of a core.*

Description

Obtain accumulation rates (in years per cm, so actually sedimentation times) as estimated by the MCMC iterations for any depth of a core.

Usage

```
accrate.depth(
  d,
  set = get("info"),
  cmyr = FALSE,
  na.rm = FALSE,
  inversion.threshold = 1e-06
)
```

Arguments

`d`	The depth for which accumulation rates need to be returned.
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`cmyr`	Accumulation rates can be calculated in cm/year or year/cm. By default `cmyr=FALSE` and accumulation rates are calculated in year per cm.
`na.rm`	Remove NA entries. These are NOT removed by default, ensuring that always the same amount of iterations is returned.
`inversion.threshold`	Very small accumulation rate values will become very large when their inverse is calculated. By default, any accumulation rate smaller than 1e-6 is set to 1e-6.

Details

should take into account hiatuses

Considering accumulation rates is crucial for age-depth modelling, and even more so if they are subsequently used for calculating proxy influx values, or interpreted as proxy for environmental change such as carbon accumulation. Bacon deals explicitly with accumulation rate and its variability through defining prior distributions. This function obtains accumulation rates (in years per cm, so actually sedimentation times) as estimated by the MCMC iterations for any depth of a core. Deals with only 1 depth at a time. See also `accrate.age`.

Value

all MCMC estimates of accumulation rate of the chosen depth.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(yr.res=50, d.res=50, d.by=10)
  d20 <- accrate.depth(20)
  hist(d20)
  d20 <- accrate.depth(20, cmyr=TRUE) # to calculate accumulation rates in cm/yr
  mean(d20)

## End(Not run)
```

`accrate.depth.ghost` *Plot modelled accumulation rates against the depths of a core.*

Description

Plot grey-scale representation of modelled accumulation rates over a core's depth. Each section of the core (see Bacon's option `"thick"`) will have modelled accumulation rates.

Usage

```
accrate.depth.ghost(
  set = get("info"),
  d = set$elbows,
  d.lim = c(),
  acc.lim = c(),
  d.lab = c(),
  cmyr = FALSE,
  acc.lab = c(),
  dark = 1,
  cutoff = 0.001,
  rgb.scale = c(0, 0, 0),
  rgb.res = 100,
  prob = 0.95,
  plot.range = TRUE,
  range.col = grey(0.5),
  range.lty = 2,
  plot.mean = TRUE,
  mean.col = "red",
  mean.lty = 2,
```

```
  plot.median = TRUE,
  median.col = "blue",
  median.lty = 2,
  rotate.axes = FALSE,
  rev.d = FALSE,
  rev.acc = FALSE,
  xaxs = "r",
  yaxs = "r",
  bty = "l",
  remove.laststep = TRUE,
  use.raster = FALSE,
  flip.acc = FALSE
)
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable info.
`d`	The depths for which the accumulation rates are to be calculated. Default to the entire core.
`d.lim`	Axis limits for the depths.
`acc.lim`	Axis limits for the accumulation rates.
`d.lab`	Label for the depth axis.
`cmyr`	Accumulation rates can be calculated in cm/year or year/cm. By default `cmyr=FALSE` and accumulation rates are calculated in year per cm. Axis limits are difficult to calculate when `cmyr=TRUE`, so a manual adaptation of `acc.lim` might be a good idea.
`acc.lab`	Axis label for the accumulation rate.
`dark`	The darkest grey value is dark=1 by default; lower values will result in lighter grey but values >1 are not advised.
`cutoff`	Point below which colours will no longer be printed. Default `cutoff=0.001`.
`rgb.scale`	The function to produce a coloured representation of all age-models. Needs 3 values for the intensity of red, green and blue. Defaults to grey-scales: `rgb.scale=c(0,0,0)`, but could also be, say, scales of red (`rgb.scale=c(1,0,0)`).
`rgb.res`	Resolution of the colour spectrum depicting the age-depth model. Default `rgb.res=100`.
`prob`	Probability ranges. Defaults to `prob=0.95`.
`plot.range`	If `plot.range=TRUE`, the confidence ranges (two-tailed; half of the probability at each side) are plotted.
`range.col`	Colour of the confidence ranges.
`range.lty`	Line type of the confidence ranges.
`plot.mean`	If `plot.mean=TRUE`, the means are plotted.
`mean.col`	Colour of the mean accumulation rates.
`mean.lty`	Type of the mean lines.

`plot.median`	If `plot.mean=TRUE`, the medians are plotted.
`median.col`	Colour of the median accumulation rates.
`median.lty`	Type of the median lines.
`rotate.axes`	The default is to plot the accumulation rates horizontally and the depth vertically (`rotate.axes=FALSE`). Change rotate.axes value to rotate axes.
`rev.d`	The direction of the depth axis can be reversed from the default (`rev.d=TRUE`.
`rev.acc`	The direction of the accumulation rate axis can be reversed from the default (`rev.acc=TRUE`).
`xaxs`	Extension of x-axis. By default, add some extra white-space at both extremes (`xaxs="r"`). See ?par for other options.
`yaxs`	Extension of y-axis. By default, add no extra white-space at both extremes (`yaxs="i"`). See ?par for other options.
`bty`	Type of box to be drawn around the plot (`"n"` for none, and `"l"` (default), `"7"`, `"c"`, `"u"`, or `"o"` for correspondingly shaped boxes).
`remove.laststep`	Add a white line to remove spurious lines at the extreme of the graph. Defaults to TRUE.
`use.raster`	Rasters can be aligned or not in the underlying image function. Setting `use.raster=FALSE, default` takes a bit longer to draw and sometimes causes strange lines owing to anti-aliasing. However, the alternative of `use.raster=TRUE` causes greyscales on some devices (e.g., OSX quartz) to 'flip'. If this is the case, use 'flip.acc=TRUE'.
`flip.acc`	When using `use.raster=TRUE`, sometimes greyscales are flipped. If this is the case, see if setting `flip.acc=TRUE` solves this.

Details

This plot shows the modelled accumulation rates in grey-scales, where darker grey indicates more likely accumulation rates. Axis limits for accumulation rates are estimated automatically, however upper limits can be very variable (and thus hard to predict) if calculated in cm/yr; therefore you might want to manually adapt the axis limits after plotting with default settings (e.g., `acc.lim=c(0,1)`). See also `accrate.age.ghost`, `accrate.depth` and `accrate.age`.

Value

A grey-scale plot of accumulation rate against core depth, and (invisibly) the list of depths and their accumulation rates (ranges, medians, means).

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(yr.res=50, d.res=50, d.by=10)
```

```
  layout(1)
  tmp <- accrate.depth.ghost()
  head(tmp)

## End(Not run)
```

`accrate.depth.summary` *Provide a summary of the estimated accumulation rates for any depth of a core.*

Description

Obtain a summary (95% range, 68% range, median, mean) of the accumulation rates (in years per cm, so actually sedimentation times) as estimated by the MCMC iterations for any depth of a core.

Usage

```
accrate.depth.summary(
  d,
  set = get("info"),
  cmyr = FALSE,
  na.rm = FALSE,
  probs = c(0.025, 0.16, 0.84, 0.975, 0.5)
)
```

Arguments

`d`	The depth for which accumulation rates need to be returned.
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`cmyr`	Accumulation rates can be calculated in cm/year or year/cm. By default `cmyr=FALSE` and accumulation rates are calculated in year per cm.
`na.rm`	Remove NA entries. These are NOT removed by default, so that always the same amount of iterations is returned.
`probs`	The probability ranges to be returned. Defaults to the minima and maxima of the 95% and 68% ranges, as well as the median: `probs=c(.025, .16, .84, .975, .5)`.

Value

A summary of the estimated accumulation rate of the chosen depth: minimum of the 95% interval, minimum of the 68% interval, maximum of the 68% interval, maximum of the 95% interval, median (i.e., 50%) and mean.

Author(s)

Maarten Blaauw

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(yr.res=50, d.res=50, d.by=10)
  accrate.depth.summary(20)

## End(Not run)
```

`accrates.core` *Provide a summary of the estimated accumulation rates for a range of core depths*

Description

Obtain a summary (95% range, 68% range, median, mean) of the accumulation rates (in years per cm, so actually sedimentation times) as estimated by the MCMC iterations for a range of depths of a core, and optionally write this as a file to the core directory (ending in '_accrates.txt').

Usage

```
accrates.core(
  dseq = c(),
  set = get("info"),
  cmyr = FALSE,
  na.rm = FALSE,
  probs = c(0.025, 0.16, 0.84, 0.975, 0.5),
  round = 2,
  write = TRUE,
  sep = "\t"
)
```

Arguments

`dseq`	The sequence of depths for which accumulation rates need to be returned. Defaults to whatever info$dseq is, which most often is a sequence from the top to the bottom of the core at 1 cm increments.
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`cmyr`	Accumulation rates can be calculated in cm/year or year/cm. By default `cmyr=FALSE` and accumulation rates are calculated in year per cm.
`na.rm`	Remove NA entries. These are NOT removed by default, so that always the same amount of iterations is returned.
`probs`	The probability ranges to be returned. Defaults to the minima and maxima of the 95% and 68% ranges, as well as the median: `probs=c(.025, .16, .84, .975, .5)`.
`round`	The number of decimals to report. Defaults to `round=2`.

`write`	Whether or not to write the summary to a file, in the core's directory and ending in '_accrates.txt'.
`sep`	Character to separate the entries within the file. Defaults to a tab, `sep="\t"`.

Value

A summary of the estimated accumulation rate for all selected depths: minimum of the 95% interval, minimum of the 68% interval, maximum of the 68% interval, maximum of the 95% interval, median (i.e., 50%) and mean. This is optionally written to a file in the core directory.

Author(s)

Maarten Blaauw

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(yr.res=50, d.res=50, d.by=10)
  myaccrates <- accrates.core()

## End(Not run)
```

`add.dates` *Add dates to age-depth plots*

Description

Add dated depths to plots, e.g. to show dates that weren't used in the age-depth model

Usage

```
add.dates(
  mn,
  sdev,
  depth,
  cc = 1,
  set = get("info"),
  above = 1e-06,
  postbomb = 0,
  normal = TRUE,
  delta.R = 0,
  delta.STD = 0,
  t.a = set$t.a,
  t.b = set$t.b,
  date.res = 100,
  height = 1,
  calheight = 1,
```

```
  agesteps = 1,
  cutoff = 0.005,
  col = rgb(1, 0, 0, 0.5),
  border = rgb(1, 0, 0, 0.5),
  rotate.axes = FALSE,
  mirror = TRUE,
  up = TRUE,
  BCAD = FALSE,
  pch = 4,
  cc.dir = c()
)
```

Arguments

`mn`	Reported mean of the date. Can be multiple dates. Negative numbers indicate postbomb dates (if cc > 0).
`sdev`	Reported error of the date. Can be multiple dates.
`depth`	Depth of the date.
`cc`	The calibration curve to use: `cc=1` for IntCal20 (northern hemisphere terrestrial), `cc=2` for Marine20 (marine), `cc=0` for none (dates that are already on the cal BP scale).
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`above`	Threshold for plotting of probability values. Defaults to `above=1e-3`.
`postbomb`	Use a postbomb curve for negative (i.e. postbomb) 14C ages. `0 = none, 1 = NH1, 2 = NH2, 3 = NH3, 4 = SH1-2, 5 = SH3`
`normal`	By default, Bacon uses the t-distribution (Christen and Perez 2009) to treat the dates. Use `normal=TRUE` to use the normal/Gaussian distribution. This will generally give higher weight to the dates.
`delta.R`	Mean of core-wide age offsets (e.g., regional marine offsets).
`delta.STD`	Error of core-wide age offsets (e.g., regional marine offsets).
`t.a`	The dates are treated using the t distribution by default (`normal=FALSE`). The t model has two parameters, t.a and t.b, set at 3 and 4 by default (see Christen and Perez, 2010). If you want to assign narrower error distributions (more closely resembling the normal distribution), set t.a and t.b at for example 33 and 34 respectively (e.g., for specific dates in your .csv file). For symmetry reasons, t.a must always be equal to t.b-1.
`t.b`	The dates are treated using the t distribution by default (`normal=FALSE`). The t-distribution has two parameters, t.a and t.b, set at 3 and 4 by default (see Christen and Perez, 2010). If you want to assign narrower error distributions (more closely resembling the normal distribution), set t.a and t.b at for example 33 and 34 respectively (e.g., for specific dates in your .csv file). For symmetry reasons, t.a must always be equal to t.b-1.
`date.res`	Resolution of the date's distribution. Defaults to `date.res=100`.
`height`	The heights of the distributions of the dates. See also `normalise.dists`.

`calheight`	Multiplier for the heights of the distributions of dates on the calendar scale. Defaults to `calheight=1`.
`agesteps`	Step size for age units of the distribution. Default `agesteps=1`.
`cutoff`	Avoid plotting very low probabilities of date distributions (default `cutoff=0.005`).
`col`	The colour of the ranges of the date. Default is semi-transparent red: `col=rgb(1,0,0,.5)`.
`border`	The colours of the borders of the date. Default is semi-transparent red: `border=rgb(1,0,0,0.5)`.
`rotate.axes`	The default of plotting age on the horizontal axis and event probability on the vertical one can be changed with `rotate.axes=TRUE`.
`mirror`	Plot the dates as 'blobs'. Set to `mirror=FALSE` to plot simple distributions.
`up`	Directions of distributions if they are plotted non-mirrored. Default `up=TRUE`.
`BCAD`	The calendar scale of graphs is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`pch`	The shape of any marker to be added to the date. Defaults to a cross, `pch=4`. To leave empty, use `pch=NA`.
`cc.dir`	Directory where the calibration curves for C14 dates `cc` are located. By default `cc.dir=c()`.

Details

Sometimes it is useful to add additional dating information to age-depth plots, e.g., to show outliers or how dates calibrate with different estimated offsets. Calls rice's draw.dates function.

Value

A date's distribution, added to an age-depth plot.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
Bacon(run=FALSE, coredir=tempfile())
agedepth()
add.dates(5000, 100, 60)
```

`agedepth` *Plot an age-depth model*

Description

Plot the age-depth model of a core.

Usage

```
agedepth(
  set = get("info"),
  BCAD = set$BCAD,
  depth.unit = set$depth.unit,
  age.unit = "yr",
  unit = depth.unit,
  d.lab = c(),
  age.lab = c(),
  yr.lab = age.lab,
  kcal = FALSE,
  acc.lab = c(),
  mem.lab = c(),
  d.min = c(),
  d.max = c(),
  d.by = c(),
  depths = set$depths,
  depths.file = FALSE,
  accordion = c(),
  plotatthesedepths = c(),
  age.min = c(),
  yr.min = age.min,
  age.max = c(),
  yr.max = age.max,
  hiatus.option = 1,
  dark = c(),
  prob = set$prob,
  rounded = c(),
  d.res = 400,
  age.res = 400,
  yr.res = age.res,
  date.res = 100,
  rotate.axes = FALSE,
  rev.age = FALSE,
  rev.yr = rev.age,
  rev.d = FALSE,
  use.raster = FALSE,
  flip.age = FALSE,
  flip.d = FALSE,
```

```
  maxcalc = 500,
  height = 1,
  calheight = 1,
  ex = 1,
  mirror = TRUE,
  up = TRUE,
  cutoff = 0.1,
  plot.range = TRUE,
  range.col = grey(0.5),
  range.lty = "12",
  range.lwd = 1,
  mn.col = "red",
  mn.lty = "12",
  mn.lwd = 1,
  med.col = NA,
  med.lty = "12",
  med.lwd = 1,
  C14.col = rgb(0, 0, 1, 0.35),
  C14.border = rgb(0, 0, 1, 0.5),
  cal.col = rgb(0, 0.5, 0.5, 0.35),
  cal.border = rgb(0, 0.5, 0.5, 0.5),
  dates.col = c(),
  pb.background = 0.5,
  pbmodelled.col = function(x) rgb(0, 0, 1, 0.5 * x),
  pbmeasured.col = "blue",
  pb.lim = c(),
  supp.col = rgb(0.5, 0, 0.5, 0.5),
  remove.tail = TRUE,
  MCMC.resample = TRUE,
  hiatus.col = grey(0.5),
  hiatus.lty = "12",
  rgb.scale = c(0, 0, 0),
  rgb.res = 100,
  slump.col = grey(0.8),
  normalise.dists = TRUE,
  same.heights = FALSE,
  cc = set$cc,
  title = set$core,
  title.location = "topleft",
  title.size = 1.5,
  plot.labels = FALSE,
  labels = c(),
  label.age = 1,
  label.size = 0.8,
  label.col = "black",
  label.offset = c(0, 0),
  label.adj = c(0.5, 0),
  label.rot = 0,
```

```
  after = set$after,
  bty = "l",
  mar.left = c(3, 3, 1, 0.5),
  mar.middle = c(3, 0, 1, 0.5),
  mar.right = c(3, 0, 1, 0.5),
  mar.main = c(3, 3, 1, 1),
  righthand = 3,
  mgp = c(1.7, 0.7, 0),
  xaxs = "r",
  yaxs = "i",
  MCMC.col = grey(0.4),
  post.col = grey(0.8),
  post.border = grey(0.4),
  prior.col = 3,
  prior.lwd = 2,
  prior.fontcol = 2,
  prior.ticks = "n",
  prior.fontsize = 0.9,
  toppanel.fontsize = 0.9,
  mainpanel.tickfontsize = 1,
  mainpanel.labelfontsize = 1,
  acc.xlim = c(),
  acc.ylim = c(),
  mem.xlim = c(),
  mem.ylim = c(),
  hiatus.xlim = c(),
  hiatus.ylim = c(),
  phi.xlim = c(),
  phi.ylim = c(),
  supp.xlim = c(),
  supp.ylim = c(),
  xaxt = "s",
  yaxt = "s",
  plot.pb = TRUE,
  pb.lty = 1,
  plot.pdf = FALSE,
  dates.only = FALSE,
  model.only = FALSE,
  verbose = TRUE,
  roundby = 2,
  save.info = set$save.info,
  ssize = 4000
)
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable `info`.

`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`depth.unit`	Units of the depths. Defaults to the one provided in the Bacon() command, `depth.unit=set$depth.unit`.
`age.unit`	Units of the ages. Defaults to `age.unit="yr"`.
`unit`	Deprecated and replaced by `depth.unit`.
`d.lab`	The labels for the depth axis. Default `d.lab="Depth (cm)"`. See also `depth.unit`.
`age.lab`	The labels for the calendar axis (default `age.lab="cal BP"` or `"BC/AD"` if `BCAD=TRUE`).
`yr.lab`	Deprecated - use age.lab instead
`kcal`	Use kcal BP. Default is `kcal=FALSE`.
`acc.lab`	The labels for the accumulation rate plot (top middle). Default `d.lab="Acc. rate (yr/cm)"` (or whatever units you're using).
`mem.lab`	The labels for the memory plot (top right). Default `d.lab="Memory"`.
`d.min`	Minimum depth of age-depth model (use this to extrapolate to depths higher than the top dated depth).
`d.max`	Maximum depth of age-depth model (use this to extrapolate to depths below the bottom dated depth).
`d.by`	Depth intervals at which ages are calculated. Default 1. Alternative depth intervals can be provided using, e.g., d.`d.by=0.5`.
`depths`	By default, Bacon will calculate the ages for the depths `d.min` to `d.max` in steps of `d.by`. Alternative depths can be provided as, e.g., `depths=seq(0, 100, length=500)` or as a file, e.g., `depths=read.table("CoreDepths.txt"`. See also `depths.file`.
`depths.file`	By default, Bacon will calculate the ages for the depths `d.min` to `d.max` in steps of `d.by`. If `depths.file=TRUE`, Bacon will read a file containing the depths for which you require ages. This file, containing the depths in a single column without a header, should be stored within `coredir`, and its name should start with the core's name and end with '_depths.txt'. Then specify `depths.file=TRUE` (default `FALSE`). See also `depths`.
`accordion`	An experimental option to squeeze and stretch depths below a boundary. Needs 2 parameters: boundary depth and compression ratio, e.g., `accordion=c(25,20)`. Defaults to inactive, `accordion=c()`.
`plotatthesedepths`	An option to plot ages at different depths (e.g., if a core has been compressed during a run). Use with extreme caution!
`age.min`	Minimum age of the age-depth plot.
`yr.min`	Deprecated - use age.min instead.
`age.max`	Maximum age of the age-depth plot.
`yr.max`	Deprecated - use age.min instead.
`hiatus.option`	How to calculate accumulation rates and ages for sections with hiatuses. Either extrapolate from surrounding sections (default, `hiatus.option=1`), use a w-weighted mix between the prior and posterior values for depths below the hiatus and prior information only for above the hiatus (`hiatus.option=2`), or use the originally calculated slopes (`hiatus.option=0`).

`dark`	Darkness of the greyscale age-depth model. By default, the darkest grey value is calculated as 10 times the height of the lowest-precision age estimate `dark=c()`. Lower values will result in lighter grey but values >1 are not allowed.
`prob`	Confidence interval to report (between 0 and 1, default 0.95 or 95%).
`rounded`	Rounding of years. Default is to round to single years (1 digit for plum models).
`d.res`	Resolution or amount of greyscale pixels to cover the depth scale of the age-model plot. Default `d.res=200`.
`age.res`	Resolution or amount of greyscale pixels to cover the age scale of the age-model plot. Default `yr.res=200`.
`yr.res`	Deprecated - use age.res instead.
`date.res`	Date distributions are plotted using `date.res=100` points by default.
`rotate.axes`	By default, the age-depth model is plotted with the depths on the horizontal axis and ages on the vertical axis. This can be changed with `rotate.axes=TRUE`.
`rev.age`	The direction of the age axis, which can be reversed using `rev.age=TRUE`.
`rev.yr`	Deprecated - use rev.age instead.
`rev.d`	The direction of the depth axis, which can be reversed using `rev.d=TRUE`.
`use.raster`	Rasters can be aligned or not in the underlying image function. Setting `use.raster=FALSE, default` takes a bit longer to draw and sometimes causes strange lines owing to anti-aliasing. However, the alternative of `use.raster=TRUE` causes greyscales on some devices (e.g., OSX quartz) to 'flip'. If this is the case, use 'flip.acc=TRUE'.
`flip.age`	When using `use.raster=TRUE`, sometimes greyscales are flipped. If this is the case, see if setting `flip.age=TRUE` can flip the ages back again.
`flip.d`	When using `use.raster=TRUE`, sometimes greyscales are flipped. If this is the case, see if setting `flip.d=TRUE` can flip the depths back again.
`maxcalc`	Number of depths to calculate ages for. If this is more than `maxcalc=500`, a warning will be shown that calculations will take time.
`height`	The maximum heights of the distributions of the dates on the plot. See also `normalise.dists`.
`calheight`	Multiplier for the heights of the distributions of dates on the calendar scale. Defaults to `calheight=1`.
`ex`	As an alternative to 'height' and 'calheight', the distribution heights can be set as either a single value (e.g., `ex=1`) or for each date (for example, `myex=rep(1, nrow(info$dets)); myex[1:5] <- 10; agedepth(ex=myex)`).
`mirror`	Plot the dates as 'blobs'. Set to `mirror=FALSE` to plot simple distributions.
`up`	Directions of distributions if they are plotted non-mirrored. Default `up=TRUE`.
`cutoff`	Avoid plotting very low probabilities of date distributions (default `cutoff=0.1`).
`plot.range`	Whether or not to plot the curves showing the confidence ranges of the age-model. Defaults to (`plot.range=TRUE`).
`range.col`	The colour of the curves showing the confidence ranges of the age-model. Defaults to medium grey (`range.col=grey(0.5)`).
`range.lty`	The line type of the curves showing the confidence ranges of the age-model. Defaults to `range.lty=12`.

`range.lwd`	Widths of the lines of the ranges of the age-depth model. Default `range.lwd=1`.
`mn.col`	The colour of the mean age-depth model: default `mn.col="red"`.
`mn.lty`	The line type of the mean age-depth model. Default `mn.lty=12`.
`mn.lwd`	Width of the line of the mean age-depth model. Default `mn.lwd=1`.
`med.col`	The colour of the median age-depth model: not drawn by default `med.col=NA`.
`med.lty`	The line type of the median age-depth model. Default `med.lty=12`.
`med.lwd`	Width of the line of the median age-depth model. Default `med.lwd=1`.
`C14.col`	The colour of the calibrated ranges of the dates. Default is semi-transparent blue: `C14.col=rgb(0,0,1,.35)`.
`C14.border`	The colours of the borders of calibrated 14C dates. Default is semi-transparent dark blue: `C14.border=rgb(0, 0, 1, 0.5)`.
`cal.col`	The colour of the non-14C dates. Default is semi-transparent blue-green: `cal.col=rgb(0,.5,.5,.35)`.
`cal.border`	The colour of the border of non-14C dates in the age-depth plot: default semi-transparent dark blue-green: `cal.border=rgb(0,.5,.5,.5)`. Not used by default.
`dates.col`	As an alternative to colouring dates based on whether they are 14C or not, sets of dates can be coloured as, e.g., `dates.col=colours()[2:100]`.
`pb.background`	Probability at which total Pb values are considered to have reached background values, or in other words, that their modelled values are at or below supported + detection limit (Al)). Setting this at 0.5 means that any depth with a Pb measurement, where at least half of the iterations model Pb values reaching background values, is flagged as having reached background. The age-model is not extended to any Pb measurements that have reached background.
`pbmodelled.col`	Colour of the modelled 210Pb values. Defaults to shades of blue: `pbmodelled.col=function(x) rgb(0,0,1,x)`.
`pbmeasured.col`	Colour of the measured 210Pb values (default `pbmeasured.col="blue"`). Draws rectangles of the upper and lower depths as well as the Pb values with 95 percent error ranges.
`pb.lim`	Axis limits for the Pb-210 data. Calculated automatically by default (`pblim=c()`).
`supp.col`	Colour of supported Pb-210. Defaults to semi-transparent purple, because why not.
`remove.tail`	Whether or not to remove the tail measurements when plotting. Sometimes automated removal might go wrong, so then this option can be used to avoid removing the tail measurements.
`MCMC.resample`	After the MCMC run, if there are more MCMC iterations than requested, only the last 'ssize' iterations will be retained. Defaults to TRUE.
`hiatus.col`	The colour of the depths of any hiatuses. Default `hiatus.col=grey(0.5)`.
`hiatus.lty`	The line type of the depths of any hiatuses. Default `hiatus.lty=12`.
`rgb.scale`	The function to produce a coloured representation of all age-models. Needs 3 values for the intensity of red, green and blue. Defaults to grey-scales: `rgb.scale=c(0,0,0)`, but could also be, say, scales of red (`rgb.scale=c(1,0,0)`).
`rgb.res`	Resolution of the colour spectrum depicting the age-depth model. Default `rgb.res=100`.

`slump.col`	Colour of slumps. Defaults to `slump.col=grey(0.8)`.
`normalise.dists`	By default, the distributions of more precise dates will cover less time and will thus peak higher than less precise dates. This can be avoided by specifying `normalise.dists=FALSE`.
`same.heights`	Plot the distributions of the dates all at the same maximum height (default `same.height=FALSE`).
`cc`	Calibration curve for 14C dates: `cc=1` for IntCal20 (northern hemisphere terrestrial), `cc=2` for Marine20 (marine), `cc=3` for SHCal20 (southern hemisphere terrestrial). For dates that are already on the cal BP scale use `cc=0`.
`title`	The title of the age-depth model is plotted on the main panel. By default this is the core's name. To leave empty: `title=""`.
`title.location`	Location of the title. Default `title.location='topleft'`.
`title.size`	Size of the title font. Defaults to `title.size=1.5`.
`plot.labels`	Whether or not to plot labels next to the dated depths. Defaults to `FALSE`.
`labels`	Add labels to the dates (as given by the first column of the .csv file). `FALSE` by default.
`label.age`	Position on the age axis of the date labels. By default draws them before the youngest age (1), but can also draw them after the oldest age (2), or above its mean (3).
`label.size`	Size of labels.
`label.col`	Colour of the labels. Defaults to the colour given to the borders of the dates.
`label.offset`	Offsets of the positions of the labels, giving the x and y offsets. Defaults to c(0,0).
`label.adj`	Justification of the labels. Follows R's adj option: A value of '0' produces left-justified text, '0.5' (the default) centered text and '1' right-justified text.
`label.rot`	Rotation of the label. 0 by default (horizontal).
`after`	Sets a short section above and below hiatus.depths within which to calculate ages. For internal calculations - do not change.
`bty`	Type of box to be drawn around plots (`"n"` for none, and `"l"` (default), `"7"`, `"c"`, `"u"`, or `"o"` for correspondingly shaped boxes).
`mar.left`	Plot margins for the topleft panel (amount of white space along edges of axes 1-4). Default `mar.left=c(3,3,1,1)`.
`mar.middle`	Plot margins for the middle panel(s) at the top (amount of white space along edges of axes 1-4). Default `mar.middle=c(3,3,1,1)`.
`mar.right`	Plot margins for the topright panel (amount of white space along edges of axes 1-4). Default `mar.right=c(3,3,1,1)`.
`mar.main`	Plot margins for the main panel (amount of white space along edges of axes 1-4). Default `mar.main=c(3,3,1,1)`.
`righthand`	Adapt the righthand margins by a certain amount (default 2) to allow a righthand axis to be plotted (for plum)

`mgp`	Axis text margins (where should titles, labels and tick marks be plotted). Defaults to `mgp=c(1.7, 0.7, 0.0)`.
`xaxs`	Extension of x-axis. By default, add some extra white-space at both extremes (`xaxs="r"`). See ?par for other options.
`yaxs`	Extension of y-axis. By default, add no extra white-space at both extremes (`yaxs="i"`). See ?par for other options.
`MCMC.col`	Colour of the MCMC output. Defaults to `post.col=grey(0.4))`.
`post.col`	Colour of the posterior histogram. Defaults to `post.col=grey(0.8))`.
`post.border`	Colour of the posterior border. Defaults to `post.border=grey(0.4))`.
`prior.col`	Colour of the prior curve. Defaults to light green, `prior.col=3)`.
`prior.lwd`	Line width of the prior curve. Defaults to `prior.lwd=2)`.
`prior.fontcol`	Colour of the font accompanying the posterior histograms. Defaults to red, `prior.fontcol=2)`.
`prior.ticks`	Plot tickmarks and values on the vertical axes for the prior and posterior distributions. Defaults to no tick marks (`prior.ticks="n"`). Set to `prior.ticks="s"` to plot the tick marks. Note that these values are of little practical use, as they correspond poorly to, e.g., the mean and strength values. All that matters is that the areas of both the prior and the posterior distributions sum to 1; wider distributions tend to give lower peaks, and narrower distributions higher peaks.
`prior.fontsize`	Font size of the prior, relative to R's standard size. Defaults to `prior.fontsize=0.9`.
`toppanel.fontsize`	Font size of the top panels, relative to R's standard size. Defaults to `prior.fontsize=0.9`.
`mainpanel.tickfontsize`	Font size of values at the tick marks in the main panel, relative to R's standard size. Defaults to `mainpanel.tickfontsize=1`.
`mainpanel.labelfontsize`	Font size of axis labels in the main panel, relative to R's standard size. Defaults to `mainpanel.labelsize=1`.
`acc.xlim`	Horizontal axis limits of the accumulation rate panel. Calculated automatically by default.
`acc.ylim`	Vertical axis limits of the accumulation rate panel. Calculated automatically by default.
`mem.xlim`	Horizontal axis limits of the memory panel. Calculated automatically by default.
`mem.ylim`	Vertical axis limits of the memory panel. Calculated automatically by default.
`hiatus.xlim`	Horizontal axis limits of the hiatus size panel. Calculated automatically by default.
`hiatus.ylim`	Vertical axis limits of the hiatus size panel. Calculated automatically by default.
`phi.xlim`	Horizontal axis limits of the phi panel. Calculated automatically by default.
`phi.ylim`	Vertical axis limits of the phi panel. Calculated automatically by default.
`supp.xlim`	Horizontal axis limits of the supported-Pb panel. Calculated automatically by default.

`supp.ylim`	Vertical axis limits of the supported-Pb panel. Calculated automatically by default.
`xaxt`	Whether or not to plot the x-axis. Can be used to adapt axes after a plot. See ?par for other options.
`yaxt`	Whether or not to plot the y-axis. Can be used to adapt axes after a plot. See ?par for other options.
`plot.pb`	Plot the 210Pb data (if present). Defaults to `plot.pb=TRUE`.
`pb.lty`	Line type of measured Pb-210 data.
`plot.pdf`	Produce a pdf file of the age-depth plot.
`dates.only`	By default, the age-depth model is plotted on top of the dates. This can be avoided by supplying `dates.only=TRUE`.
`model.only`	By default, panels showing the MCMC iterations and the priors and posteriors for accumulation rate and memory are plotted above the main age-depth model panel. This can be avoided by supplying `model.only=TRUE`. Note however that this removes relevant information to evaluate the age-depth model, so we do recommend to present age-models together with these upper panels.
`verbose`	Provide a summary of the age ranges after producing the age-depth model graph; default `verbose=FALSE`.
`roundby`	Rounding of the values reported at the end of the function. Defaults to 2 decimals.
`save.info`	By default, a variable called 'info' with relevant information about the run (e.g., core name, priors, settings, ages, output) is saved into the working directory. Note that this will overwrite any existing variable with the same name - as an alternative, one could run, e.g., `myvar <- Bacon()`, followed by supplying the variable `myvar` in any subsequent commands.
`ssize`	Number of MCMC iterations to use.

Details

After loading a previous run, or after running either the scissors or thinner command, plot the age-model again using the command `agedepth()`.

Value

A plot of the age-depth model, and estimated ages incl. confidence ranges for each depth.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
Bacon(ask=FALSE, coredir=tempfile())
agedepth()
```

`agemodel.it` *Extract one age-model iteration*

Description

For one MCMC iteration (it), extract the corresponding age-depth model.

Usage

```
agemodel.it(it, set = get("info"), BCAD = set$BCAD, save.info = set$save.info)
```

Arguments

`it`	The MCMC iteration of which the age-model should be calculated.
`set`	Detailed information of the current run, stored within this session's memory as variable info.
`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`save.info`	If TRUE, a variable called 'info' with relevant information about the run (e.g., core name, priors, settings, ages, output) is saved into the working directory. Note that this will overwrite any existing variable with the same name.

Value

A variable with two columns - depth and the age-depth model of a single iteration.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(age.res=50, d.res=50, d.by=10)
  lines(agemodel.it(5), col="red")

## End(Not run)
```

`ageranges` *save age ranges of depth(s)*

Description

Calculates an age summary (min/max probability range, median and mean) of a depth or a series of depths. The depths are returned and can be saved to a file.

Usage

```
ageranges(
  d = c(),
  file = c(),
  sep = "\t",
  set = get("info"),
  BCAD = set$BCAD,
  na.rm = FALSE,
  prob = 0.95,
  d.by = 1,
  roundby = 1,
  show.progress = TRUE
)
```

Arguments

`d`	Depth or depths for which age summaries are to be calculated. If left empty, a sequence of depths is calculated from the top to the bottom of the core, see 'd.by'.
`file`	The name of the file where the age summary has to be stored. If empty (default), the ages are returned.
`sep`	Separator for the fields, if saving to a file (defaults to a tab, "\t").
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`BCAD`	The calendar scale of graphs and age output-files is in cal BP (calendar or calibrated years before the present, where the present is AD 1950) by default, but can be changed to BC/AD using `BCAD=TRUE`.
`na.rm`	Whether or not NAs are to be removed. Defaults to `na.rm=FALSE`. NAs will still be returned for depths outside of the core's depth range.
`prob`	Probability range. Half of the range is taken away from both sides of the distribution (e.g., 2.5% for the default of `prob=0.95`).
`d.by`	Steps for calculation of depths, if 'd' is left empty. Defaults to steps of 1.
`roundby`	Rounding for the age estimates. Defaults to 1 decimal.
`show.progress`	Show a progress bar. Defaults to TRUE.

Value

The minimum and maximum of the probability range (default 2.5

`AgesOfEvents` *Event probabilities against calendar age*

Description

Plot probability curves for events in the core, expressed against calendar age.

Usage

```
AgesOfEvents(
  window,
  move,
  set = get("info"),
  plot.steps = FALSE,
  BCAD = set$BCAD,
  age.lab = c(),
  yr.lab = age.lab,
  age.lim = c(),
  yr.lim = age.lim,
  prob.lab = "probability",
  prob.lim = c(),
  rotate.axes = FALSE,
  rev.age = TRUE,
  rev.yr = rev.age,
  yaxs = "i",
  bty = "l"
)
```

Arguments

`window`	Width of the window.
`move`	Step size with which the window moves.
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`plot.steps`	Plot probability values step-wise (defaults to `plot.steps=FALSE`, which plots smooth curves instead).
`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`age.lab`	The labels for the calendar axis (default `age.lab="cal BP"` or `"BC/AD"` if `BCAD=TRUE`).
`yr.lab`	Deprecated - use age.lab instead
`age.lim`	Minimum and maximum calendar age ranges, calculated automatically by default (`age.lim=c()`).

`yr.lim`	Deprecated - use age.lim instead
`prob.lab`	Label of the probability axis (default `prob.lab="probability"`).
`prob.lim`	Limits of the probability axis (calculated automatically by default).
`rotate.axes`	The default of plotting age on the horizontal axis and event probability on the vertical one can be changed with `rotate.axes=TRUE`.
`rev.age`	The direction of the age axis, which can be reversed using `rev.age=TRUE`.
`rev.yr`	Deprecated - use rev.age instead
`yaxs`	Extension of the y-axis. Defaults to the exact ranges of the probability values. White space can be added to the vertical axis using `yaxs="r"`.
`bty`	Type of box to be drawn around plots. Draw a box around the graph (`"n"` for none, and `"l"`, `"7"`, `"c"`, `"u"`, "]" or `"o"` for correspondingly shaped boxes).

Details

Probabilities of depths with 'events' in an age-modelled core can be plotted against time, taking into account chronological uncertainties (Blaauw et al. 2007). Such events could be for example core depths at which proxies indicate changes toward wetter local conditions. This can be expressed as values between 0 (no event) and 1 (event at 100% probability) for each depth.

Blaauw et al. 2010 propose to estimate probabilities of events by finding specific proxy features such as increasing curves. Probabilities are then estimated through resampling from the proxy values, where low to modest rises of proxy curves result in low event probabilities, and clear proxy rises in high probabilities. A smooth spline can be applied to adapt the balance of short-term vs long-term events. To calculate the event probabilities, produce a file with two columns (depth and corresponding proxy-derived probabilities, separated by white spaces). Do not provide headers at the file's first line, and save the file with extension "_events.txt" within the core's Bacon folder. See Cores/MSB2K/MSB2K_events.txt (or Bacon_runs/MSB2K/MSB2K_events.txt) for an example. Events are calculated as the probability that an event took place within specific time windows - or more specifically, that the Bacon age-depth model puts depths with assigned event probabilities in that time window.

does not yet deal correctly with hiatuses.

Value

The resulting probabilities are plotted and saved within the core's folder (file names ending with the window width and "_probs.txt").

Author(s)

Maarten Blaauw, J. Andres Christen

References

Blaauw, M., Christen, J.A., Mauquoy, D., van der Plicht, J., Bennett, K.D. (2007) Testing the timing of radiocarbon-dated events between proxy archives. _The Holocene_, *17*, 283-288. Blaauw, M., Wohlfarth, B., Christen, J.A., Ampel, L., Veres, D., Hughen, K.A., Preusser, F., Svensson, A. (2010) Were last glacial climate events simultaneous between Greenland and France? A quantitative comparison using non-tuned chronologies. _Journal of Quaternary Science_ *25*, 387-394.

Examples

```
Bacon(run=FALSE, coredir=tempfile())
agedepth(yr.res=50)
AgesOfEvents(100, 10)
```

`background`	*calculate probabilities that Pb-210 data have reached background levels*

Description

Checks which of the Pb-210 data most likely have reached background levels and thus are below the detection limit Al (probabilities between 0 and 1)

Usage

```
background(set = get("info"), Al = set$Al)
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`Al`	The detection limit. Default `Al=0.1`.

Value

a list of probabilities for each Pb-210 data point

Author(s)

Maarten Blaauw

`Bacon`	*Main age-depth modelling function*

Description

This is the main age-depth modelling function of the rbacon package.

Usage

```
Bacon(
  core = "MSB2K",
  thick = 5,
  coredir = "",
  prob = 0.95,
  d.min = NA,
  d.max = NA,
  add.bottom = TRUE,
  d.by = 1,
  seed = NA,
  depths.file = FALSE,
  depths = c(),
  depth.unit = "cm",
  age.unit = "yr",
  unit = depth.unit,
  acc.shape = 1.5,
  acc.mean = 20,
  mem.strength = 10,
  mem.mean = 0.5,
  boundary = NA,
  hiatus.depths = NA,
  hiatus.max = 10000,
  add = c(),
  after = 1e-04/thick,
  cc = 1,
  cc1 = "IntCal20",
  cc2 = "Marine20",
  cc3 = "SHCal20",
  cc4 = "ConstCal",
  cc.dir = c(),
  postbomb = 0,
  F14C = c(),
  pMC = c(),
  delta.R = 0,
  delta.STD = 0,
  t.a = 3,
  t.b = 4,
  normal = FALSE,
  suggest = TRUE,
  accept.suggestions = FALSE,
  reswarn = c(10, 200),
  remember = TRUE,
  ask = TRUE,
  run = TRUE,
  defaults = "defaultBacon_settings.txt",
  sep = ",",
  dec = ".",
```

```
  runname = "",
  slump = c(),
  remove = FALSE,
  BCAD = FALSE,
  ssize = 4000,
  th0 = c(),
  burnin = min(500, ssize),
  youngest.age = c(),
  oldest.age = c(),
  MinAge = c(),
  MaxAge = c(),
  cutoff = 0.01,
  plot.pdf = TRUE,
  dark = 1,
  date.res = 100,
  age.res = 200,
  yr.res = age.res,
  close.connections = TRUE,
  save.info = TRUE,
  older.than = c(),
  younger.than = c(),
  save.elbowages = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

`core`	Name of the core, given using quotes. Defaults to one of the cores provided with rbacon, `core="MSB2K"`. An alternative core provided with this package is RLGH3 (Jones et al., 1989). To run your own core, produce a .csv file with the dates as outlined in the manual, add a folder with the core's name to the default directory for cores (see `coredir`), and save the .csv file there. For example, the file's location and name could be `Bacon_runs/MyCore/MyCore.csv`. Then run Bacon as follows: `Bacon("MyCore")`
`thick`	Bacon will divide the core into sections of equal thickness specified by thick (default `thick=5`).
`coredir`	Folder where the core's files `core` are and/or will be located. This will be a folder with the core's name, within either the folder `coredir='Bacon_runs/'`, or the folder Cores/ if it already exists within R's working directory, or a custom-built folder. For example, use `coredir="."` to place the core's folder within the current working directory, or `coredir="F:"` if you want to put the core's folder and files on a USB drive loaded under F:. Thinner (and thus more) sections will result in smoother age-models, but too many sections can cause 'run-away' models.
`prob`	Confidence interval to report. This should lie between 0 and 1, default 0.95 (95 %).

`d.min`	Minimum depth of age-depth model (use this to extrapolate to depths higher than the top dated depth).
`d.max`	Maximum depth of age-depth model (use this to extrapolate to depths below the bottom dated depth).
`add.bottom`	Add a model section at the bottom of the core, in order to ensure the bottommost date is taken into account. Default `add.bottom=TRUE`. This is a new option and can cause age-models to differ from previous version. Please re-run the model if in doubt.
`d.by`	Depth intervals at which ages are calculated. Defaults to `d.by=1`. Please ensure that the value of d.by is smaller than that of ’thick’, otherwise plots might turn out wrong.
`seed`	Seed used for C++ executions. If it is not assigned (`seed=NA`; default) then the seed is set by system.
`depths.file`	By default, Bacon will calculate the ages for the depths `d.min` to `d.max` in steps of `d.by`. If `depths.file=TRUE`, Bacon will read a file containing the depths for which you require ages. This file, containing the depths in a single column without a header, should be stored within `coredir`, and its name should start with the core’s name and end with ‘_depths.txt’. Then specify `depths.file=TRUE` (default `FALSE`). See also `depths`.
`depths`	By default, Bacon will calculate the ages for the depths `d.min` to `d.max` in steps of `d.by`. Alternative depths can be provided as, e.g., `depths=seq(0, 100, length=500)` or as a file, e.g., `depths=read.table("CoreDepths.txt"`. See also `depths.file`.
`depth.unit`	Units of the depths. Defaults to `depth.unit="cm"`.
`age.unit`	Units of the ages. Defaults to `age.unit="yr"`.
`unit`	Deprecated and replaced by `depth.unit`.
`acc.shape`	The prior for the accumulation rate consists of a gamma distribution with two parameters. Its shape is set by acc.shape (default `acc.shape=1.5`; higher values result in more peaked shapes).
`acc.mean`	The accumulation rate prior consists of a gamma distribution with two parameters. Its mean is set by acc.mean (default `acc.mean=20` yr/cm (or whatever age or depth units are chosen), which can be changed to, e.g., 5, 10 or 50 for different kinds of deposits). Multiple values can be given in case of hiatuses or boundaries, e.g., Bacon(hiatus.depths=23, acc.mean=c(5,20))
`mem.strength`	The prior for the memory (dependence of accumulation rate between neighbouring depths) is a beta distribution, which looks much like the gamma distribution. but its values are always between 0 (no assumed memory) and 1 (100% memory). Its default settings of `mem.strength=10` (higher values result in more peaked shapes) allow for a large range of posterior memory values. Please note that the default memory prior has been updated from rbacon version 2.5.1 on, to repair a bug.
`mem.mean`	The prior for the memory is a beta distribution, which looks much like the gamma distribution but its values are always between 0 (no assumed memory) and 1 (100% memory). Its default settings of `mem.mean=0.5` allow for a large range of posterior memory values. Please note that the default memory prior has been updated from rbacon version 2.5.1. on, to repair a bug.

`boundary`	The assumed depths of any boundary, which divides sections of different accumulation rate regimes (e.g., as indicated by major change in the stratigraphy). No hiatus is assumed between these sections, and memory is reset crossing the boundary. Different accumulation priors can be set for the sections above and below the boundary, e.g., `acc.mean=c(5, 20)`. See also `hiatus.depths`, `mem.mean`, `acc.mean` and `acc.shape`. Setting many boundaries might not work, and having more than one boundary per model section (see `'thick'`) might not work either.
`hiatus.depths`	The assumed depths for any hiatus should be provided as, e.g., `hiatus.depths=20` for one at 20cm depth, and `hiatus.depths=c(20,40)` for two hiatuses at 20 and 40 cm depth.
`hiatus.max`	The prior for the maximum length of the hiatus. Hiatus length is a uniform distribution, with equal probabilities between 0 and `hiatus.max` yr (or whatever other `age.unit` is chosen).
`add`	Add a value to the maximum hiatus length if a boundary is chosen. Defaults to 100 yr (or whatever other age unit is chosen). Can be adapted if Bacon complains that the parameters are out of support.
`after`	Sets a short section above and below hiatus.depths within which to calculate ages. For internal calculations - do not change.
`cc`	Calibration curve for C-14 dates: `cc=1` for IntCal20 (northern hemisphere terrestrial), `cc=2` for Marine20 (marine), `cc=3` for SHCal20 (southern hemisphere terrestrial). For dates that are already on the cal BP scale use `cc=0`.
`cc1`	For northern hemisphere terrestrial 14C dates (IntCal20).
`cc2`	For marine 14C dates (Marine20).
`cc3`	For southern hemisphere 14C dates (SHCal20).
`cc4`	Provide the name of an alternative curve (3 columns: cal BP, 14C age, error, separated by white spaces and saved as a plain-text file). It is important here to first produce a tailor-made folder for your and the default calibration curves to live in. See `cc.dir`. Defaults to `cc4="mixed.14C"`.
`cc.dir`	Directory where the calibration curves for C14 dates `cc` are located. By default uses the location of the rintcal package which provides the calibration curves. If you want to use custom-made calibration curves, first set up a new folder using the function new.ccdir() in the rintcal package, e.g., `new.ccdir="MyCurves"`, then place the custom curve in that folder using `rintcal::mix.ccurves(, cc.dir="MyCurves", save=TRUE)`.
`postbomb`	Use a postbomb curve for negative (i.e. postbomb) 14C ages. `0 = none, 1 = NH1, 2 = NH2, 3 = NH3, 4 = SH1-2, 5 = SH3`
`F14C`	Radiocarbon ages can be provided as F14C values. If doing so, please indicate here which dates were entered as F14C (e.g., if the first 4 dates are in F14C, write `F14C=1:4`). The F14C values in your .csv file will then be replaced by their corresponding C14 ages.
`pMC`	Radiocarbon ages can be provided as pMC values. If doing so, please indicate here which dates were entered as pMC (e.g., if the first 4 dates are in pMC, write `pMC=1:4`). The pMC values in your .csv file will then be replaced by their corresponding C14 ages.

`delta.R`	Mean of core-wide age offsets (e.g., regional marine offsets).
`delta.STD`	Error of core-wide age offsets (e.g., regional marine offsets).
`t.a`	The dates are treated using the t distribution (Christen and Perez 2009) by default (`normal=FALSE`). This t-distribution has two parameters, t.a and t.b, set at 3 and 4 by default (see Christen and Perez, 2010). If you want to assign narrower error distributions (more closely resembling the normal distribution), set t.a and t.b at for example 33 and 34 respectively (e.g., for specific dates in your .csv file). For symmetry reasons, t.a must always be equal to t.b-1.
`t.b`	The dates are treated using t distribution by default (`normal=FALSE`). The t-distribution has two parameters, t.a and t.b, set at 3 and 4 by default (see Christen and Perez, 2009). If you want to assign narrower error distributions (more closely resembling the normal distribution), set t.a and t.b at for example 33 and 34 respectively (e.g., for specific dates in your .csv file). For symmetry reasons, t.a must always be equal to t.b-1.
`normal`	By default, Bacon uses the t-distribution to treat the dates. Use `normal=TRUE` to use the normal/Gaussian distribution. This will generally give higher weight to the dates.
`suggest`	If initial analysis of the data indicates abnormally slow or fast accumulation rates, Bacon will suggest to change the prior.
`accept.suggestions`	Automatically accept the suggested values. Use with care. Default `accept.suggestions=FALSE`. Also, if the length of the core would cause too few or too many sections with the default settings, Bacon will suggest an alternative section thickness `thick`. Accept these suggested alternative settings by typing "y" (or "yes please" if you prefer to be polite), or leave as is by typing "n" (or anything else, really). To get rid of these suggestions, use `suggest=FALSE`.
`reswarn`	Bacon will warn you if the number of sections lies outside the safe range (default between 10 and 200 sections; `reswarn=c(10,200)`). Too few sections could lead to an 'elbowy' model while with too many sections the modelling process can get lost, resulting in age-models far away from the dated depths.
`remember`	Bacon will try to remember which settings you have applied to your cores (default `remember=TRUE`). If you run into inconsistencies or other problems, try running your core again with `remember=FALSE`, or, start cleanly by typing `Bacon.cleanup()`.
`ask`	By default Bacon will ask you to confirm that you want to run the core with the provided settings. Disable this using `ask=FALSE` (e.g., for batch runs).
`run`	In order to load an existing Bacon run instead of producing a new one, you can use `run=FALSE`.
`defaults`	Name of the file containing settings for the core. For internal use only - do not change.
`sep`	Separator between the fields of the plain text file containing the dating information. Default `sep=","`.
`dec`	Character for decimal points. Default to `dec="."`.
`runname`	Text to add to the corename for specific runs, e.g., `runname="MyCore_Test1"`.

`slump`	Upper and lower depths of any sections of assumed abrupt accumulation, that require excising before age-modelling (and adding after age-modelling). Requires pairs of depths, e.g., `slump=c(10,15,60,67)` for slumps at 67-60 and 15-10 cm core depth.
`remove`	Whether or not to remove depths within slumps. Defaults to `remove=FALSE`.
`BCAD`	The calendar scale of graphs and age output-files is in cal BP (calendar or calibrated years before the present, where the present is AD 1950) by default, but can be changed to BC/AD using `BCAD=TRUE`.
`ssize`	The amount of iterations to store at the end of the MCMC run. Default 4000; decrease for faster (but less reliable) runs or increase for cores where the MCMC mixing (panel at upper-left corner of age-model graph) appears problematic.
`th0`	Starting years for the MCMC iterations. These are randomly chosen by default.
`burnin`	Amount of initial, likely sub-optimal MCMC iterations that will be removed.
`youngest.age`	Minimum age limit for Bacon runs, default at current year in cal BP. To set plot limits, use `age.min` instead.
`oldest.age`	Maximum age limit for Bacon runs, default at 1,000,000 cal BP. To set plot limits, use `age.max` instead.
`MinAge`	Deprecated - use youngest.age instead.
`MaxAge`	Deprecated - use oldest.age instead.
`cutoff`	Avoid plotting very low probabilities of date distributions (default `cutoff=0.001`).
`plot.pdf`	Produce a pdf file of the age-depth plot. Defaults to `plot.pdf=TRUE` after a Bacon run.
`dark`	Darkness of the greyscale age-depth model. The darkest grey value is `dark=1` by default. Lower values will result in lighter grey but values >1 are not allowed.
`date.res`	Date distributions are plotted using `date.res=100` segments by default.
`age.res`	Resolution or amount of greyscale pixels to cover the age scale of the age-model plot. Default `yr.res=200`.
`yr.res`	Deprecated - use age.res instead
`close.connections`	Internal option to close connections after a run. Default `close.connections=TRUE`.
`save.info`	By default, a variable called 'info' with relevant information about the run (e.g., core name, priors, settings, ages, output) is saved into the working directory. Note that this will overwrite any existing variable with the same name - as an alternative, one could run, e.g., `myvar <- Bacon()`, followed by supplying the variable `myvar` in any subsequent commands.
`older.than`	an option to enable dates at the limit of C-14 dating. If there are older.than dates, they tell us that the core should be older than a certain age at that depth. For example, if the 7th and 8th dates in the core's .csv file are older-than dates, use as `older.than=c(7,8)`. The MCMC run could be problematic if the older-than ages do not fit with the other information.
`younger.than`	an option to provide younger-than ages, for example a historical pollen marker. If there are younger-than dates, they tell us that the core should be younger than a certain age at that depth. For example, if the 7th and 8th dates in the core's

	.csv file are younger.than dates, use as `younger.than=c(7,8)`. The MCMC run could be problematic if the younger.than ages do not fit with the other information.
`save.elbowages`	If you want to have a file with the MCMC-derived ages for all the age-depth model's elbows, set `save.elbowages=TRUE` and a file with the ages will be saved in the core's folder, starting with the core name, followed by its number of sections, d.min, and section thickness, and ending in "_elbowages.txt".
`verbose`	Provide feedback on what is happening (default `verbose=TRUE`).
`...`	options for the age-depth graph. See agedepth and calib.plot

Details

Bacon is an approach to age-depth modelling that uses Bayesian statistics in order to reconstruct Bayesian accumulation histories for deposits, through combining radiocarbon and other dates with prior information ('Blaauw' and 'Christen', 2011).

Bacon divides a core into many thin vertical sections (by default of `thick=5` cm thickness), and through millions of Markov Chain Monte Carlo (MCMC) iterations estimates the accumulation rate (in years/cm; so more correctly, sedimentation times) for each of these sections. Combined with an estimated starting date for the first section, these accumulation rates then form the age-depth model. The accumulation rates are constrained by prior information on the accumulation rate (`acc.mean, acc.shape`) and its variability between neighbouring depths, or "memory" (`mem.mean, mem.strength`). Hiatuses can be introduced as well, also constrained by prior information (`hiatus.max`).

Although Bacon works with any kind of absolute dates (e.g., OSL, tephra or other dates on a calendar scale), it is often used to age-model 14C-dated sequences. Radiocarbon dates should be calibrated using either IntCal20 (for terrestrial northern hemisphere material; Reimer et al., 2020), Marine20 (for marine dates; Hughen et al., 2020), SHCal20 (for southern hemisphere dates; Hogg et al., 2020) or any other calibration curve (see below), while modern 14C dates are calibrated using one of the post-bomb calibration curves (NH1, NH2 or NH3 for the northern hemisphere, SH1-2 or SH3 for the southern hemisphere; Hua et al., 2022). See `http://calib.org/CALIBomb/` if you are unsure which postbomb curve you need. If Bacon finds postbomb dates (negative 14C ages) and you haven't specified a postbomb curve, you will be prompted. Provide postbomb curves as, e.g., `postbomb=1` for the NH1 postbomb curve (2 for NH2, 3 for NH3, 4 for SH1-2, 5 for SH3). For calendar dates, i.e. dates that are already on the calendar scale and thus should not be calibrated, `setcc=0`.

Since version 3.1.0, rbacon can also handle younger-than and older-than ages, with the model aiming to either go 'above' or 'below' such dates as requested. If the resulting combination of parameters becomes problematic (e.g., no initial combination of parameters can be found that obeys the priors or is in chronological order), then the output will often be wrong. If so, using the function set.initvals could help.

By default, the initial MCMC values of the Bacon age-depth model (upper ages and accumulation rate for each model section) are estimated randomly. Since version 3.1.0, these starting values can also be provided in a file with extension _bacon.init, placed within the core's folder. This file will need to have two rows, each for one of the two initial sets of parameters required (the t-walk requires two starting estimates for all MCMC parameters). If such a file is found (and correctly formatted), Bacon will use the values within this file as starting points for the MCMC run. See function set.initvals for more information.

From version 2.5.1 on (i.e., since February 2021), the default memory prior has changed to `mem.mean=0.5` and `mem.strength=10`. Previously used c++ code contained a bug which caused the prior information for the memory not to be taken into account correctly. Now that this bug has been repaired, the default memory prior has been updated such that it should work for most types of cores, and should result in similar output to previous versions of Bacon. There is no need to re-do previous runs. However, it is considered good practice to test the impact of different settings on a site's age-depth model (e.g., thick, acc.mean, acc.shape, mem.mean, acc.strength).

Value

An age-depth model graph, its age estimates, and a summary.

Author(s)

Maarten Blaauw, J. Andres Christen

References

Blaauw, M. and Christen, J.A., 2011. Flexible paleoclimate age-depth models using an autoregressive gamma process. Bayesian Anal. 6(3): 457-474.

Christen, J.A., Perez E., S., 2010. A new robust statistical model for radiocarbon data. Radiocarbon 51: 1047-1059.

Reimer et al., 2020. The IntCal20 Northern Hemisphere radiocarbon age calibration curve (0–55 cal kBP). Radiocarbon 62(4): 725-757. doi:10.1017/RDC.2020.41

Hogg et al. 2020 SHCal20 Southern Hemisphere calibration, 0-55,000 years cal BP. Radiocarbon 62(4): 759-778. doi:10.1017/RDC.2020.59

Hughen et al. 2020 Marine20-the marine radiocarbon age calibration curve (0-55,000 cal BP). Radiocarbon 62(4): 779-820. doi:10.1017/RDC.2020.68

Hua et al. 2022 Atmospheric radiocarbon for the period 1950-2019. Radiocarbon 64(4), 723-745, doi:10.1017/RDC.2021.95

Jones, V.J., Stevenson, A.C., Battarbee, R.W., 1989. Acidification of lakes in Galloway, south west Scotland - a diatom and pollen study of the post-glacial history of the Round Loch of Glenhead. Journal of Ecology 77: 1-23.

Examples

```
  Bacon(ask=FALSE, coredir=tempfile())
  Bacon(cc=2, delta.R=80, delta.STD=40, coredir=tempfile())
```

`Bacon.Age.d` *Output all ages for a single depth.*

Description

Output all MCMC-derived age estimates for a given depth.

Usage

```
Bacon.Age.d(
  d,
  set = get("info"),
  its = set$output,
  BCAD = set$BCAD,
  na.rm = FALSE
)
```

Arguments

`d`	The depth of which Bacon age estimates are to be returned. Has to be a single depth.
`set`	Detailed information of the current run, stored within this session's memory as variable info.
`its`	The set of MCMC iterations to be used. Defaults to the entire MCMC output, `its=set$output`.
`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`na.rm`	Whether or not to remove NA values (ages within slumps)

Details

Obtaining an age-depth model is often only a step towards a goal, e.g., plotting a core's fossil series ('proxies') against calendar time. Bacon.Age.d can be used to list all MCMC-derived age estimates for a given (single) depth, for example to calculate mean ages for a depth. See also Bacon.d.Age which calculates the depths of a single age estimate.

Value

Outputs all MCMC-derived ages for a given depth.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(age.res=50, d.res=50, d.by=10)
  ages.d20 = Bacon.Age.d(20)
  mean(ages.d20)

## End(Not run)
```

`Bacon.cleanup` *Remove files made to produce the current core's age-depth model.*

Description

Remove files ending in .bacon, .plum (if it exists), .out, .pdf, _ages.txt, and _settings.txt of current core.

Usage

```
Bacon.cleanup(set = get("info"))
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable `info`.

Details

If cores behave badly, you can try cleaning up previous runs and settings, by removing files *.bacon, *.plum, *.out, *.pdf, *_ages.txt, and *_settings.txt of current core.

Value

A message stating that the files and settings of this run have been deleted.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
  Bacon(run=FALSE, coredir=tempfile())
  Bacon.cleanup()
```

`Bacon.d.Age` *Output all depths for a single age.*

Description

Output all MCMC-derived depth estimates for a single given age.

Usage

```
Bacon.d.Age(
  age,
  set = get("info"),
  BCAD = set$BCAD,
  its = set$output,
  na.rm = FALSE
)
```

Arguments

`age`	The age estimate for which depths are to be returned. Has to be a single age.
`set`	Detailed information of the current run, stored within this session's memory as variable info.
`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`its`	The set of MCMC iterations to be used. Defaults to the entire MCMC output, `its=set$output`.
`na.rm`	Whether or not to remove NA values (ages within slumps)

Details

Obtaining an age-depth model is often only a step towards a goal, e.g., plotting a core's fossil series ('proxies') against calendar time. Bacon.d.Age can be used to list all MCMC-derived depths belonging to a given (single) age, for example to calculate mean depths belonging to a modelled depth. This function was kindly written and provided by Timon Netzel (Bonn University). See also Bacon.Age.d, which calculates the ages for a single depth.

Value

Outputs all MCMC-derived depths for a given age.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(age.res=50, d.res=50, d.by=10)
  ages.d20 = Bacon.Age.d(20)
  mean(ages.d20)

## End(Not run)
```

`Bacon.hist` *Calculate age distributions of depths.*

Description

Calculate the distribution of age estimates of single or multiple depths.

Usage

```
Bacon.hist(
  d,
  set = get("info"),
  BCAD = set$BCAD,
  age.lab = c(),
  age.lim = c(),
  hist.lab = "Frequency",
  calc.range = TRUE,
  hist.lim = c(),
  draw = TRUE,
  prob = set$prob,
  hist.col = grey(0.5),
  hist.border = grey(0.2),
  range.col = "blue",
  med.col = "green",
  mean.col = "red",
  verbose = TRUE,
  save.info = set$save.info
)
```

Arguments

`d`	The depth or depths for which a histogram and age ranges should be provided. If multiple depths are given, then just the age ranges, median and means (no graphs) are provided for each depth.
`set`	Detailed information of the current run, stored within this session's memory as variable info.
`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.

`age.lab`	The labels for the calendar axis (default `age.lab="cal BP"` or `"BC/AD"` if `BCAD=TRUE`).
`age.lim`	Minimum and maximum calendar age ranges, calculated automatically by default (`age.lim=c()`).
`hist.lab`	The y-axis is labelled `ylab="Frequency"` by default.
`calc.range`	Calculate ranges? Takes time so can be left out
`hist.lim`	Limits of the y-axis.
`draw`	Draw a plot or not. Defaults to `draw=TRUE`, however no plots are made if more than one depth `d` is provided. If `draw=FALSE`, then the age ranges, median and mean are given for each depth (as four columns).
`prob`	Age ranges are given as quantiles, e.g., 2.5% and 97.5% for the default of 95% confidence limits (`prob=0.95`)).
`hist.col`	Colour of the histogram. Default grey, `hist.col=grey(0.5)`.
`hist.border`	Colour of the histogram's outline. Default dark grey, `hist.border=grey(0.2)`.
`range.col`	Colour of confidence ranges. Defaults to `range.col="blue"`.
`med.col`	Colour of the median. Defaults to `med.col="green"`.
`mean.col`	Colour of the mean. Defaults to `mn.col="red"`.
`verbose`	Provide feedback on what is happening (default `verbose=TRUE`).
`save.info`	A variable called 'info' with relevant information about the run (e.g., core name, priors, settings, ages, output) can be saved into the working directory. Note that this will overwrite any existing variable with the same name - as an alternative, one could run, e.g., `myvar <- Bacon()`, followed by supplying the variable `myvar` in any subsequent commands.

Details

Age estimates of specific depths can also be plotted.

Value

A local variable called 'hists', and a plot with the histogram and the age ranges, median and mean, or just the age ranges, medians and means if more than one depth `d` is given.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(age.res=50, d.res=50, d.by=10)
  Bacon.hist(20)
  Bacon.hist(20:30)

## End(Not run)
```

`bacon2clam` *Translate Bacon .csv files to clam .csv files.*

Description

Reads a Bacon .csv file containing the dates, and transforms it into a clam .csv file.

Usage

```
bacon2clam(core, bacondir = "Bacon_runs", clamdir = "clam_runs", sep = ",")
```

Arguments

`core`	The name of the core for which a Bacon .csv.file needs to be translated into a clam .csv file
`bacondir`	The directory where the Bacon runs reside. Defaults to `coredir="Bacon_runs"`.
`clamdir`	The directory where the clam runs reside. Defaults to `coredir="clam_runs"`.
`sep`	The separator for the .csv files. Defaults to `sep=","`.

Details

Assumes that Bacon .csv files with 4 columns indicate 14C dates. Please make sure this is correct.

Value

A clam .csv file

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
{

 tmpfl <- tempfile()
  Bacon(run=FALSE, ask=FALSE, coredir=tmpfl)
  bacon2clam("MSB2K", bacondir=tmpfl, clamdir=tmpfl)

}
```

`Baconvergence` *Test to identify poorly mixed MCMC runs.*

Description

Test how well-mixed and converged the MCMC runs are with the chosen core and settings, by running the core several times and comparing the different runs using the Gelman and Rubin Reduction factor (Brooks and Gelman, 1998).

Usage

```
Baconvergence(core = "MSB2K", runs = 5, suggest = FALSE, verbose = TRUE, ...)
```

Arguments

`core`	Name of the core, given using quotes. Defaults to one of the cores provided with rbacon, `core="MSB2K"`.
`runs`	Amount of runs to test for mixing. Default `runs=5`.
`suggest`	If initial analysis of the data indicates abnormally slow or fast accumulation rates, Bacon will suggest to change the prior.
`verbose`	Provide feedback on what is happening (default `verbose=TRUE`).
`...`	additional options that can be given to the Bacon function.

Details

Generally Bacon will perform millions of MCMC iterations for each age-model run, although only a fraction of these will be stored. In most cases the remaining MCMC iterations will be well mixed (the upper left panel of the fit of the iterations shows no strange features such as sudden systematic drops or rises). However if the iterations seem not well mixed, or if too few remain (say less than a few hundred), then you could check the Gelman and Rubin Reduction Factor. Too high differences (high Factors) between runs indicate poor MCMC mixing. Robust MCMC mixing is indicated by a Gelman and Rubin Reduction factor (Brooks and Gelman, 1998) below the 1.05 safety threshold.

Value

NA

Author(s)

Maarten Blaauw, J. Andres Christen

References

Brooks, SP. and Gelman, A. (1998) General methods for monitoring convergence of iterative simulations. _Journal of Computational and Graphical Statistics_, *7*, 434-455.

Examples

```
Baconvergence(runs=2, ssize=100, coredir=tempfile()) # a quick-and-dirty toy example
```

---

calib.plot *Plot the dates*

Description

Produce a plot of the dated depths and their dates

Usage

```
calib.plot(
  set = get("info"),
  dets = set$dets,
  accordion = c(),
  BCAD = set$BCAD,
  cc = set$cc,
  rotate.axes = FALSE,
  rev.d = FALSE,
  rev.age = FALSE,
  rev.yr = rev.age,
  age.lim = c(),
  yr.lim = age.lim,
  date.res = 100,
  d.lab = c(),
  age.lab = c(),
  yr.lab = age.lab,
  height = 1,
  calheight = 1,
  ex = 1,
  mirror = TRUE,
  up = TRUE,
  cutoff = 0.1,
  C14.col = rgb(0, 0, 1, 0.5),
  C14.border = rgb(0, 0, 1, 0.75),
  cal.col = rgb(0, 0.5, 0.5, 0.5),
  cal.border = rgb(0, 0.5, 0.5, 0.75),
  dates.col = c(),
  slump.col = grey(0.8),
  new.plot = TRUE,
  plot.dists = TRUE,
  same.heights = FALSE
)
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`dets`	The set of determinations to be plotted.
`accordion`	If depths have to be squeezed/stretched, the parameters can be set here. Defaults to being empty, but requires 2 parameters if active, e.g., `accordion=c(10,20)`.
`BCAD`	The calendar scale of graphs is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`cc`	Calibration curve to be used (defaults to info$cc)
`rotate.axes`	The default of plotting age on the horizontal axis and event probability on the vertical one can be changed with `rotate.axes=TRUE`.
`rev.d`	The direction of the depth axis can be reversed from the default (`rev.d=TRUE`).
`rev.age`	The direction of the calendar age axis can be reversed from the default (`rev.age=TRUE`)
`rev.yr`	Deprecated - use rev.age instead
`age.lim`	Minimum and maximum calendar age ranges, calculated automatically by default (`age.lim=c()`).
`yr.lim`	Deprecated - use age.lim instead
`date.res`	Date distributions are plotted using `date.res=100` points by default.
`d.lab`	The labels for the depth axis. Default `d.lab="Depth (cm)"`.
`age.lab`	The labels for the calendar axis (default `yr.lab="cal BP"` or `"BC/AD"` if `BCAD=TRUE`).
`yr.lab`	Deprecated - use age.lab instead
`height`	The heights of the distributions of the dates. See also `normalise.dists`.
`calheight`	Multiplier for the heights of the distributions of dates on the calendar scale. Defaults to `calheight=1`.
`ex`	Alternative for providing heights of the dates' distributions. Defaults to 1 for all dates, but could also provide individual ex values for each date, e.g., ex=rep(1, nrow(set$dets)); ex[1:4] <- 2
`mirror`	Plot the dates as 'blobs'. Set to `mirror=FALSE` to plot simple distributions.
`up`	Directions of distributions if they are plotted non-mirrored. Default `up=TRUE`.
`cutoff`	Avoid plotting very low probabilities of date distributions (default `cutoff=0.1`).
`C14.col`	Colour of the calibrated distributions of the dates. Default is semi-transparent blue: `rgb(0,0,1,.35)`.
`C14.border`	Colours of the borders of calibrated 14C dates. Default is transparent dark blue: cal.col
`cal.col`	Colour of the non-14C dates in the age-depth plot: default semi-transparent blue-green: `rgb(0,.5,.5,.35)`.
`cal.border`	Colour of the of the border of non-14C dates in the age-depth plot: default semi-transparent dark blue-green: `rgb(0,.5,.5,.5)`.
`dates.col`	As an alternative to colouring dates based on whether they are 14C or not, sets of dates can be coloured as, e.g., `dates.col=colours()[2:100]`.

`slump.col`	Colour of slumps. Defaults to `slump.col=grey(0.8)`.
`new.plot`	Start a new plot (`new.plot=TRUE`) or plot over an existing plot (`new.plot=FALSE`).
`plot.dists`	Plot the distributions of the dates (default `plot.dists=TRUE`).
`same.heights`	Plot the distributions of the dates all at the same maximum height (default `same.height=FALSE`), which instead normalises the distributions (all have an area of 1).

Details

This function is generally called internally to produce the age-depth graph. It can be used to produce custom-built graphs.

Value

NA

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
Bacon(run=FALSE, coredir=tempfile())
calib.plot()
```

`clam2bacon` *Translate clam .csv files to Bacon .csv files.*

Description

Reads a clam .csv file containing the dates, and transforms it into a Bacon .csv file.

Usage

```
clam2bacon(
  core,
  clamdir = "clam_runs",
  bacondir = "Bacon_runs",
  sep = ",",
  cc = 1
)
```

Arguments

`core`	The name of the core for which a clam .csv.file needs to be translated into a Bacon .csv file
`clamdir`	The directory where the clam runs reside. Defaults to `coredir="clam_runs"`.
`bacondir`	The directory where the Bacon runs reside. Defaults to `coredir="Bacon_runs"`.
`sep`	The separator for the .csv files. Defaults to `sep=","`.
`cc`	Calibration curve for C-14 dates: `cc=1` for IntCal20 (northern hemisphere terrestrial), `cc=2` for Marine20 (marine),

Details

Please ensure that if the clam file has offset (d.R) estimates, that errors (d.STD) are provided manually, since these values cannot be determined automatically from the clam .csv file.

Value

A Bacon .csv file

Author(s)

Maarten Blaauw, J. Andres Christen

`draw.pbmeasured` *Plot the 210Pb data*

Description

Produce a plot of the 210Pb data and their depths

Usage

```
draw.pbmeasured(
  set = get("info"),
  rotate.axes = FALSE,
  rev.d = FALSE,
  rev.age = FALSE,
  BCAD = set$BCAD,
  pb.lim = c(),
  age.lim = c(),
  d.lim = c(),
  d.lab = c(),
  pb.lab = c(),
  pbmeasured.col = "blue",
  pbmeasured.lty = 2,
  pb.log = FALSE,
  supp.col = "purple",
```

```
  newplot = TRUE,
  on.agescale = FALSE
)
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`rotate.axes`	The default of plotting age on the horizontal axis and event probability on the vertical one can be changed with `rotate.axes=TRUE`.
`rev.d`	The direction of the depth axis can be reversed from the default (`rev.d=TRUE`).
`rev.age`	The direction of the calendar age axis can be reversed from the default (`rev.age=TRUE`)
`BCAD`	The calendar scale of graphs and age output-files is in cal BP (calendar or calibrated years before the present, where the present is AD 1950) by default, but can be changed to BC/AD using `BCAD=TRUE`.
`pb.lim`	Minimum and maximum of the 210Pb axis ranges, calculated automatically by default (`pb.lim=c()`).
`age.lim`	Minimum and maximum of the age ranges to be used to plot 210Pb values. Calculated automatically by default (`age.lim=c()`).
`d.lim`	Minimum and maximum depths to plot; calculated automatically by default (`d.lim=c()`).
`d.lab`	The labels for the depth axis. Default `d.lab="Depth (cm)"`.
`pb.lab`	The label for the 210Pb axis (default `pb.lab="210Pb (Bq/kg)"` or `"210Pb (dpm/g)"`).
`pbmeasured.col`	The label for the measured 210Pb data. `pbmeasured.col="blue"`.
`pbmeasured.lty`	Line type of the measured 210Pb data. Defaults to continuous lines.
`pb.log`	Use a log scale for the 210Pb-axis (default `pb.log=FALSE`).
`supp.col`	Colour of the supported 210Pb data. Defaults to red: `supp.col="red"`.
`newplot`	make new plot (default TRUE)
`on.agescale`	Plot the Pb-210 on the cal BP scale. Defaults to FALSE.

Details

This function is generally called internally to produce the age-depth graph. It can be used to produce custom-built graphs.

Value

A plot of the measured 210Pb values

Author(s)

Maarten Blaauw, J. Andres Christen, Marco Aquino-Lopez

`draw.pbmodelled` *Plot the 210Pb data*

Description

Produce a plot of the 210Pb data and their depths

Usage

```
draw.pbmodelled(
  set = get("info"),
  BCAD = set$BCAD,
  rotate.axes = FALSE,
  rev.d = FALSE,
  rev.age = FALSE,
  pb.lim = c(),
  d.lim = c(),
  d.lab = c(),
  pb.lab = c(),
  pbmodelled.col = function(x) rgb(0, 0, 1, x),
  pbmeasured.col = "blue",
  supp.col = "purple",
  plot.measured = TRUE,
  age.lim = c(),
  mgp = mgp,
  pb.lty = 1
)
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`BCAD`	The calendar scale of graphs is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`rotate.axes`	The default of plotting age on the horizontal axis and event probability on the vertical one can be changed with `rotate.axes=TRUE`.
`rev.d`	The direction of the depth axis can be reversed from the default (`rev.d=TRUE`).
`rev.age`	The direction of the calendar age axis can be reversed from the default (`rev.age=TRUE`)
`pb.lim`	Minimum and maximum of the 210Pb axis ranges, calculated automatically by default (`pb.lim=c()`).
`d.lim`	Minimum and maximum depths to plot; calculated automatically by default (`d.lim=c()`).
`d.lab`	The labels for the depth axis. Default `d.lab="Depth (cm)"`.
`pb.lab`	The label for the 210Pb axis (default `pb.lab="210Pb (Bq/kg)"` or `"210Pb (dpm/g)"`).

`pbmodelled.col`	Colour of the modelled 210Pb values. Defaults to scales of blue: `pbmodelled.col=function(x) rgb(0,0,1,x)`.
`pbmeasured.col`	Colour of the measured 210Pb values. Defaults to blue.
`supp.col`	Colour of the supported 210Pb data. Defaults to red: `supp.col="red"`.
`plot.measured`	Plot the measured 210Pb values (default `plot.measured=TRUE`).
`age.lim`	values of the age axis. Used to calculate where to plot the pb values on the secondary axis
`mgp`	Axis text margins (where should titles, labels and tick marks be plotted). Defaults to `mgp=c(1.7, .7, .0)`.
`pb.lty`	Line type of measured Pb-210 data.

Details

This function is generally called internally to produce the age-depth graph. It can be used to produce custom-built graphs.

Value

A plot of the modelled (and optionally the measured) 210Pb values

Author(s)

Maarten Blaauw, J. Andres Christen, Marco Aquino-Lopez

`flux.age.ghost` *Plot flux rates for proxies.*

Description

Plot grey-scale representation of estimated flux rates for proxies against calendar age.

Usage

```
flux.age.ghost(
  proxy = 1,
  age.lim = c(),
  yr.lim = age.lim,
  age.res = 400,
  yr.res = age.res,
  set = get("info"),
  flux = c(),
  plot.range = TRUE,
  prob = 0.8,
  range.col = grey(0.5),
  range.lty = 2,
  plot.mean = TRUE,
```

```
  mean.col = "red",
  mean.lty = 2,
  plot.median = TRUE,
  median.col = "blue",
  median.lty = 2,
  flux.lim = c(),
  flux.lab = expression("flux (g cm"^-1 * " yr"^-1 * ")"),
  upper = 0.95,
  rgb.scale = c(0, 0, 0),
  rgb.res = 100,
  dark = set$dark,
  cutoff = 0.001,
  BCAD = set$BCAD,
  age.lab = c(),
  yr.lab = age.lab,
  rotate.axes = FALSE,
  rev.flux = FALSE,
  rev.age = FALSE,
  rev.yr = rev.age,
  use.raster = FALSE,
  flip.age = FALSE,
  flip.flux = FALSE
)
```

Arguments

`proxy`	Which proxy to use (counting from the column number in the .csv file after the depths column).
`age.lim`	Minimum and maximum calendar age ranges, calculated automatically by default (`age.lim=c()`).
`yr.lim`	Deprecated - use age.lim instead
`age.res`	Resolution or amount of greyscale pixels to cover the age scale of the plot. Default `age.res=200`.
`yr.res`	Deprecated - use age.res instead
`set`	Detailed information of the current run, stored within this session's memory as variable info.
`flux`	Define a flux variable within the R session (consisting of depths and their proxy concentrations in two columns) and provide the name of this variable, e.g.: `flux.age.ghost(flux=flux1)`. If left empty (`flux=c()`), a flux file is expected (see `proxy`).
`plot.range`	Plot curves that indicate a probability range, at resolution of yr.res.
`prob`	Probability range, defaults to `prob=0.8` (10 % at each side).
`range.col`	Red seems nice.
`range.lty`	Line type of the confidence ranges.
`plot.mean`	Plot the mean fluxes.

`mean.col`	Red seems nice.
`mean.lty`	Line type of the means.
`plot.median`	Plot the median fluxes.
`median.col`	Blue seems nice.
`median.lty`	Line type of the medians.
`flux.lim`	Limits of the flux axes.
`flux.lab`	Axis labels. Defaults to `flux.lab="flux"`.
`upper`	Maximum flux rates to plot. Defaults to the upper 99%; `upper=0.99`.
`rgb.scale`	The function to produce a coloured representation of all age-models. Needs 3 values for the intensity of red, green and blue. Defaults to grey-scales: `rgb.scale=c(0,0,0)`, but could also be, say, scales of red (`rgb.scale=c(1,0,0)`).
`rgb.res`	Resolution of the colour spectrum depicting the age-depth model. Default `rgb.res=100`.
`dark`	The darkest grey value is `dark=1` by default; lower values will result in lighter grey but `values >1` are not allowed.
`cutoff`	Point below which colours will no longer be printed. Default `cutoff=0.001`.
`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`age.lab`	The labels for the calendar axis (default `age.lab="cal BP"` or `"BC/AD"` if `BCAD=TRUE`).
`yr.lab`	Deprecated - use age.lab instead
`rotate.axes`	The default of plotting calendar year on the horizontal axis and fluxes on the vertical one can be changed with `rotate.axes=TRUE`.
`rev.flux`	The flux axis can be reversed with `rev.flux=TRUE`.
`rev.age`	The direction of the age axis can be reversed using `rev.age=TRUE`.
`rev.yr`	Deprecated - use rev.age instead
`use.raster`	Rasters can be aligned or not in the underlying image function. By default, we use `use.raster=FALSE`. This takes a bit longer to draw and sometimes causes strange lines owing to anti-aliasing. Therefore, `use.raster=TRUE` would be preferable, however on some devices this causes greyscales to 'flip'. If this is the case, use 'flip.flux=TRUE' or 'flip.age=TRUE'.
`flip.age`	When using `use.raster=TRUE`, sometimes greyscales are flipped. If this is the case, see if setting `flip.age=TRUE` solves this.
`flip.flux`	When using `use.raster=TRUE`, sometimes greyscales are flipped. If this is the case, see if setting `flip.acc=TRUE` solves this.

Details

To plot flux rates (e.g. pollen grains/cm2/yr) as greyscales, provide a plain text file with headers and the data in columns separated by commas, ending in '_flux.csv' and saved in your core's folder. The first column should contain the depths, and the next columns should contain the proxy concentration values (leaving missing values empty). Then type for example `flux.age.ghost(1)` to plot the flux values for the first proxy in the .csv file. Instead of using a _flux.csv file, a flux variable can also be defined within the R session (consisting of depths and their proxy concentrations in two columns). Then provide the name of this variable, e.g.: `flux.age.ghost(flux=flux1)`. See Bacon_runs/MSB2K/MSB2K_flux.csv for an example.

Value

A plot of flux rates.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
## Not run:
  Bacon(run=FALSE, coredir=tempfile())
  agedepth(yr.res=50)
  flux.age.ghost(1)

## End(Not run)
```

`MCMC.diagnostics` *Test mixing and stationarity of the MCMC run*

Description

Test how well-mixed and stationary the MCMC run is. A good value for the effective sample size ('ess', number of effective independent samples from the MCMC iterations) is >200 (>1000 indicates an excelling mixing). Besides the mixing, stationarity 'z' is also measured (the start of the run is compared with the end). A 'z' below 1.96 (1 standard deviation) indicates no drift, and if it is >2.58 (2 standard deviations) then the MCMC chain is likely drifting.

Usage

```
MCMC.diagnostics(set = get("info"), ssize = nrow(set$output))
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`ssize`	Number of MCMC iterations.

Details

Generally Bacon will perform millions of MCMC iterations for each age-model run, although only a fraction of these will be stored. In most cases the remaining MCMC iterations will be well mixed (ess, and also visually check that the upper left panel of the fit of the iterations shows no strange features such as sudden systematic drops or rises).

Value

The 'ess' and 'z' scores, together with an evaluation of the values.

Examples

```
Bacon(ssize=100, coredir=tempfile()) # check the reported warnings
```

`proxy.ghost` *Proxies analysed along the depths of a core can be plotted as 'proxy-ghost' graphs against calendar time while taking into account chronological uncertainties. Here darker grey indicates more likely calendar ages for specific proxy values.*

Description

Proxies analysed along the depths of a core can be plotted as 'proxy-ghost' graphs against calendar time while taking into account chronological uncertainties. Here darker grey indicates more likely calendar ages for specific proxy value.

Usage

```
proxy.ghost(
  proxy = 1,
  proxy.lab = NULL,
  proxy.res = 250,
  age.res = 200,
  yr.res = age.res,
  rgb.scale = c(0, 0, 0),
  rgb.res = 100,
  set = get("info"),
  cutoff = 0.001,
  dark = 1,
  darkest = 1,
  rotate.axes = FALSE,
  rev.proxy = FALSE,
  rev.age = FALSE,
  yr.rev = rev.age,
  plot.mean = FALSE,
  mean.col = "red",
  age.lim = NULL,
  yr.lim = age.lim,
  proxy.lim = NULL,
  sep = ",",
  xaxs = "i",
  yaxs = "i",
  xaxt = "s",
  yaxt = "s",
  bty = "l",
  BCAD = set$BCAD,
  age.lab = ifelse(BCAD, "BC/AD", "cal yr BP"),
```

```
  yr.lab = age.lab,
  verbose = TRUE,
  add = FALSE
)
```

Arguments

`proxy`	Which proxy to use (counting from the column number in the .csv file after the depths column).
`proxy.lab`	Label of the proxy axis. Default names are taken from the csv file.
`proxy.res`	Greyscale pixels are calculated for `proxy.res=250` proxy values by default, as a compromise between image quality and calculation speed. If the output looks very pixel-like (e.g., when choosing to plot only part of the record using proxy.lim), set this option to higher values.
`age.res`	Resolution or amount of greyscale pixels to cover the age scale of the age-model plot. Default `age.res=250` as a compromise between image quality and calculation speed. If the output looks very pixel-like (e.g., when choosing to plot only part of the record using age.lim), set this option to higher values.
`yr.res`	Deprecated - use age.res instead
`rgb.scale`	The function to produce a coloured representation of all age-models. Needs 3 values for the intensity of red, green and blue. Defaults to grey-scales: `rgb.scale=c(0,0,0)`, but could also be, say, scales of red (`rgb.scale=c(1,0,0)`).
`rgb.res`	Resolution of the colour spectrum depicting the age-depth model. Default `rgb.res=100`.
`set`	Detailed information of the current run, stored within this session's memory as variable info.
`cutoff`	Point below which colours will no longer be printed. Default `cutoff=0.001`.
`dark`	By default, the darkest grey value is assigned to the most likely value within the entire core (normalised to 1; `dark=1`). By setting dark to, e.g., `dark=.8`, all values of and above 0.8 will be darkest (and values below that threshold will be lighter grey the lower their probabilities).
`darkest`	Darkness of the most likely value. Is black by default (`darkest=1`); lower values will result in lighter grey.
`rotate.axes`	The default is to plot the calendar horizontally, however the plot can be rotated (`rotate.axes=TRUE`).
`rev.proxy`	The proxy axis can be reversed if `rev.proxy=TRUE`.
`rev.age`	The calendar axis can be reversed using `rev.age=TRUE`.
`yr.rev`	Deprecated - use rev.age instead
`plot.mean`	The mean ages of the proxy values can be added using `plot.mean=TRUE`.
`mean.col`	Colour of the weighted mean ages of the proxy values.
`age.lim`	Minimum and maximum calendar age ranges, calculated automatically by default (`yr.lim=NULL`).
`yr.lim`	Deprecated - use age.lim instead
`proxy.lim`	Ranges of the proxy axis, calculated automatically by default (`proxy.lim=NULL`).

`sep`	Separator between the fields of the plain text file containing the depth and proxy data.
`xaxs`	Extension of x-axis. By default, no white-space will be added at the axis extremes (`xaxs="i"`). See ?par for other options.
`yaxs`	Extension of y-axis. By default, no white-space will be added at the axis extremes (`xaxs="i"`). See ?par for other options.
`xaxt`	The x-axis is plotted by default, but this can be switched off using `xaxt="n"`.
`yaxt`	The y-axis is plotted by default, but this can be switched off using `yaxt="n"`.
`bty`	Type of box to be drawn around the plot (`"n"` for none, and `"l"` (default), `"7"`, `"c"`, `"u"`, or `"o"` for correspondingly shaped boxes).
`BCAD`	The calendar scale of graphs and age output-files is in `cal BP` by default, but can be changed to BC/AD using `BCAD=TRUE`.
`age.lab`	The labels for the calendar axis (default `age.lab="cal BP"` or `"BC/AD"` if `BCAD=TRUE`).
`yr.lab`	Deprecated - use age.lab instead
`verbose`	Provide feedback on what is happening (default `verbose=TRUE`).
`add`	Add to an existing graph (default `add=FALSE`).

Details

Place a csv file with the values of proxies against depth within your core's folder. The values should be in columns separated by commas (default `sep=","`), the first column containing the depths and the first line (header) containing the proxy names. The file name should start with the core's name and end with "_proxies.csv". For an example see `"Bacon_coredir/MSB2K/MSB2K_proxies.csv"` or `"Cores/MSB2K/MSB2K_proxies.csv"`.

Value

A grey-scale graph of the proxy against calendar age.

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
Bacon(ask=FALSE, coredir=tempfile())
layout(1)
proxy.ghost()
```

`scissors` *Remove the first n iterations.*

Description

Removes iterations of the MCMC time series, and then updates the output file.

Usage

```
scissors(burnin, set = get("info"), write = TRUE, save.info = set$save.info)
```

Arguments

`burnin`	Number of iterations to remove of the iterative time series. If this value is higher than the amount of remaining iterations, a warning is given and the iterations are not removed. If the provided number is negative, the iterations will be removed from the end of the run, not from the start. If a range is given, this range of iterations is removed.
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`write`	Whether or not to write the changes to the output file. Defaults to TRUE.
`save.info`	By default, a variable called 'info' with relevant information about the run (e.g., core name, priors, settings, ages, output) is saved into the working directory. Note that this will overwrite any existing variable with the same name - as an alternative, one could run, e.g., `myvar <- Bacon()`, followed by supplying the variable `myvar` in any subsequent commands.

Details

Bacon will perform millions of MCMC iterations for each age-model run by default, although only a fraction of these will be stored. In most cases the remaining MCMC iterations will be well mixed (the upper left panel of the fit of the iterations shows no undesirable features such as trends or sudden systematic drops or rises). If the run has a visible remaining burn-in, scissors can be used to remove them. To remove, e.g., the first 300 iterations, type `scissors(300)`. To remove the last 300 iterations, type `scissors(-300)`. To remove iterations 300 to 600, type `scissors(300:600)`.

Value

NA

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
Bacon(ask=FALSE, coredir=tempfile())
nrow(info$output)
scissors(100)
nrow(info$output)
```

---

set.initvals *Set initial values for the Bacon MCMC run.*

Description

Select initial values th0 and th1 for a Bacon MCMC run and write them into a file that can be read by Bacon.

Usage

```
set.initvals(set = get("info"), core = set$core, values = c(), click = 1)
```

Arguments

`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`core`	The name of the core for which a bacon.init file needs to be made
`values`	use this if you wish to provide the values (2 rows with starting age, accumulation rates for each model section, and memory parameter w).
`click`	use this if you wish to use the cursor to manually select age-depth points from the current graphic device. This is the default option. Right click once you have selected all datapoints, or provide the number of expected datapoints as a value (e.g., `click=5`).

Details

By default, the initial MCMC values th0 and th1 of the Bacon age-depth model (upper ages and accumulation rate for each model section) are estimated randomly. Since version 3.1.0, these starting values can also be provided in a file with extension _bacon.init, placed within the core's folder. This file will need to have two rows, each for one of the two initial sets of parameters required (the t-walk requires two starting estimates for all MCMC parameters). If such a file is found (and correctly formatted), Bacon will use the values within this file as starting points for the MCMC run.

Value

A .bacon.init file

Author(s)

Maarten Blaauw, J. Andres Christen

`squeeze` *Squeeze some depths of a core*

Description

Squeeze or compress depths below a boundary by a certain amount. Accompanies the stretch function; see the stretch function for code on running the accordion

Usage

```
squeeze(d, boundary, times)
```

Arguments

`d`	The depth(s) to be squeezed
`boundary`	The depth below which depths should be squeezed
`times`	The factor by which the depths should be squeezed

Value

The squeezed depth(s)

Author(s)

Maarten Blaauw

Examples

```
squeeze(40,25,20)
```

`stretch` *Stretch some depths of a core*

Description

Stretch squeezed depths e.g., calculate the original depths of depths that were squeezed. Accompanies the squeeze function.

Usage

```
stretch(d, boundary, times)
```

Arguments

`d`	The depth(s) to be stretched
`boundary`	The depth below which depths should be stretched
`times`	The factor by which the depths should be stretched

Value

The stretched depth(s)

Author(s)

Maarten Blaauw

Examples

```
  stretch(25.75,25,20)
## Not run: 
  # To play the accordion, first squeeze an existing core.
  # Let's squeeze the depths below 10 cm core depth 20 times:
  Bacon("accordion", 1)
  dets <- info$dets
  dets[,4] <- squeeze(dets[,4], 10, 20)

  # make a new directory for the squeezed core, and place the dets file there:
  dir.create("Bacon_runs/squeezed")
  write.table(dets, "Bacon_runs/squeezed/squeezed.csv", row.names=FALSE, sep=",")

 # now run that squeezed core, adding a boundary (10cm) and adapting the acc.mean prior (20x):
  Bacon("squeezed", 1, boundary=10, acc.mean=c(5, 20*5))
  # finally, plot while stretching the depths onto the original scale:
  agedepth(accordion=c(10,20))

## End(Not run)
```

`thinner` *Thin iterations.*

Description

Randomly thin iterations by a given proportion, for example if autocorrelation is visible within the MCMC series.

Usage

```
thinner(
  proportion = 0.1,
  set = get("info"),
  write = TRUE,
  save.info = set$save.info
)
```

Arguments

`proportion`	Proportion of iterations to remove. Should be between 0 and 1. Default `proportion=0.1`.
`set`	Detailed information of the current run, stored within this session's memory as variable `info`.
`write`	Whether or not to write the changes to the output file. Defaults to TRUE.
`save.info`	By default, a variable called 'info' with relevant information about the run (e.g., core name, priors, settings, ages, output) is saved into the working directory. Note that this will overwrite any existing variable with the same name - as an alternative, one could run, e.g., `myvar <- Bacon()`, followed by supplying the variable `myvar` in any subsequent commands.

Details

From all iterations, a proportion is removed with to-be-removed iterations sampled randomly among all iterations.

Value

NA

Author(s)

Maarten Blaauw, J. Andres Christen

Examples

```
Bacon(ask=FALSE, coredir=tempfile())
nrow(info$output)
thinner(.2)
nrow(info$output)
```

`tofu` *Bacon for vegans*

Description

Bacon for vegans

Usage

```
tofu(...)
```

Arguments

`...`	options for the Bacon command. See Bacon

Details

A vegan wrapper for Bacon - does everything Bacon does, but without the meat.

Value

A tofu age-model

Index