Package ‘WaverideR’

April 20, 2023

Type Package

Title Extracting Signals from Wavelet Spectra

Version 0.2.0

Author Michiel Arts

Maintainer Michiel <michiel.arts@stratigraphy.eu>

Depends R (>= 3.5.0)

Imports DecomposeR, DescTools, Hmisc, Matrix,utils, colorednoise, doSNOW, fANCOVA, foreach,stats,tcltk, matrixStats,reshape2,truncnorm,grDevices,graphics,parallel

Description The continuous wavelet transform enables the observation of transient/non-stationary cyclicity in time-series. The goal of cyclcostratigraphic studies is to define frequency/period in the depth/time domain. By conducting the continuous wavelet transform on cyclostratigraphic data series one can observe and extract cyclic signals/signatures from signals. These results can then be visualized and interpreted enabling one to identify/interpret cyclicity in the geological record, which can be used to construct astrochronological age-models and identify and interpret cyclicity in past en present climate systems.

License GPL (>= 2)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

R topics documented:

`achor2time` *Convert a proxy record to the time domain using anchor points*

Description

Convert a proxy record to the time domain using anchor points made using the `astro_anchor` function.

Usage

```
achor2time(anchor_points = NULL, data = NULL, genplot = TRUE)
```

Arguments

`anchor_points`	Anchor points made using the `astro_anchor` function or a matrix in which the first column is depth and the second column is time.
`data`	Data set which needs to be converted from the depth to time domain using set anchor points. The data set should consist of a matrix with 2 column the first column should be depth and the second column should be a proxy value.
`genplot`	If `genplot=TRUE` then 3 plots stacked on top of each other will be plotted. Plot 1: the original dataset Plot 2: the depth time plot Plot 3: the dataset in the time domain

Value

The output is a matrix with 2 columns. The first column is time. The second column sedimentation proxy value.

If genplot=TRUE then 3 plots stacked on top of each other will be plotted. Plot 1: the original dataset. Plot 2: the depth time plot. Plot 3: the dataset in the time domain.

Examples

```
# Use the \code{age_model_zeeden} example anchor points of Zeeden et al., (2013) \cr
#to anchor the \code{grey} data set of Zeeden et al., (2013) in the time domain.

achored2time <- achor2time(anchor_points=age_model_zeeden,
data=grey,
genplot=TRUE)
```

---

age_model_zeeden *Age model of Zeeden et al., (2013) for the (154-174m) interval of the IODP 926 grey scale record*

Description

Age model (anchor points) of the IODP 926 grey scale (154-174m) record of Zeeden et al., (2013) Anchored to the eccentricity-tilt-precession model p-0.5t of la 2004.

Details

Column 1: Depth (meters)
Column 2: Age (kyr)

References

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, <doi:10.1016/j.palaeo.2012.11.009>

Laskar, Jacques, Philippe Robutel, Frédéric Joutel, Mickael Gastineau, Alexandre CM Correia, and Benjamin Levrard. "A long-term numerical solution for the insolation quantities of the Earth. " Astronomy & Astrophysics 428, no. 1 (2004): 261-285. https://www.aanda.org/articles/aa/pdf/2004/46/aa1335.pdf

analyze_wavelet *Computes the wavelet power spectrum of a time series/signal*

Description

Compute the continuous wavelet transform (CWT) using a Morlet wavelet

Usage

```
analyze_wavelet(
  data = NULL,
  dj = 1/20,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = TRUE,
  omega_nr = 6
)
```

Arguments

data	Input data, should be a matrix or data frame in which the first column is depth or time and the second column is proxy record.
dj	Spacing between successive scales Default=1/200.
lowerPeriod	Lowest period to be analyzed Default=2, scaling is done using power 2 so for the best plotting results select a value to the power or 2.
upperPeriod	Upper period to be analyzed Default=1024, scaling is done using power 2 so for the best plotting results select a value to the power or 2.
verbose	Print text Default=TRUE.
omega_nr	Number of cycles contained within the wavelet.

Value

The output is a list (wavelet object) with the results of the continuous wavelet transform (CWT).

Author(s)

Code based on the R package WaveletComp-package and biwavelet-package which are based on the wavelet MATLAB code written by Christopher Torrence and Gibert P. Compo.

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78.

Examples

```
#Example 1. Using the Total Solar Irradiance data set of Steinhilver et al., (2012)
TSI_wt <-
 analyze_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = TRUE,
   omega_nr = 6
 )


#Example 2. Using the magnetic susceptibility data set of De pas et al., (2018)
mag_wt <-
analyze_wavelet(
data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10
)

#Example 3. Using the greyscale data set of Zeeden et al., (2013)
grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = TRUE,
   omega_nr = 8
 )
```

astro_anchor — *Anchor proxy record to an astronomical solution*

Description

Anchor the extracted signal to an astronomical solution using a GUI. The `astro_anchor` function allows one to tie minima or maxima in the proxy record to minima or maxima in an astronomical solution. By tying the proxy record to an astronomical solution one will generate tie-points which can be used to generate a astrochronological age-model As minima or maxima in the proxy record are tied to minima or maxima in an astronomical solution it is important to provide input which has clearly definable minima and maxima. As such input should be of a "sinusoidal" nature otherwise the `extract_astrosolution=TRUE` and/or `extract_proxy_signal=TRUE` options need to be set to TRUE to create sinusoidal signals.

Astronomical solutions option are:

- La2004 Eccentricity solution available via the getLaskar function or downloadable via `http://vo.imcce.fr/insola/earth/online/earth/earth.html`

- La2004 Obliquity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2004 Precession solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010a Eccentricity solution available via the getLaskarfunction or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010a Obliquity solution downloadable via the http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010a Precession solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010b Eccentricity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010b Obliquity solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010b Precession solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010c Eccentricity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010c Obliquity solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010c Precession solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010d Eccentricity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010d Obliquity solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2010d Precession solution downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- La2011 Eccentricity solution available via the getLaskar function or downloadable via http://vo.imcce.fr/insola/earth/online/earth/earth.html
- ZB17a Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17a Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17b Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17b Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17c Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17c Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17d Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17d Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html

- ZB17e Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17e Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17f Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17f Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17h Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17h Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17i Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17i Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17j Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17j Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17k Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17k Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17p Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB17p Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB18a Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB18a Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20a Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20a Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20b Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20b Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20c Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20c Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- ZB20d Eccentricity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html

- ZB20d Obliquity solution downloadable via https://www.soest.hawaii.edu/oceanography/faculty/zeebe_files/Astro.html
- 405kyr eccentricity 405 metronome can be generated using the formula:
  e405=0.027558-0.010739*cos(0.0118+2(pi)*(t/405000)) (laskar et al., 2004 & laskar 2020)
- 173kyr obliquity metronome can be generated using using the formula:
  es3-s6(t) = 0.144*cos(1.961+2(pi)*(t/172800) (laskar et al., 2004 & laskar 2020)
- An etp model using the etp function of the astrochron-package package

Usage

```
astro_anchor(
  astro_solution = NULL,
  proxy_signal = NULL,
  proxy_min_or_max = "max",
  clip_astrosolution = FALSE,
  astrosolution_min_or_max = "max",
  clip_high = NULL,
  clip_low = NULL,
  extract_astrosolution = FALSE,
  astro_period_up = 1.2,
  astro_period_down = 0.8,
  astro_period_cycle = NULL,
  extract_proxy_signal = FALSE,
  proxy_period_up = 1.2,
  proxy_period_down = 0.8,
  proxy_period_cycle = NULL
)
```

Arguments

`astro_solution` Input is an astronomical solution which the proxy record will be anchored to, the input should be a matrix or data frame with the first column being age and the second column should be a insolation/angle/value

`proxy_signal` Input is the proxy data set which will be anchored to an astronomical solution, the input should be a matrix or data frame with the first column being depth/time and the second column should be a proxy value. For the best results either the astronomical components need to be pre-extracted before anchoring. This means that a filtering/cycle extracting need to be applied to the input data or the extract_proxy_signal option needs to be set to TRUE.

`proxy_min_or_max` Tune proxy maxima or minima to the astronomical solution `Default="max"`.

`clip_astrosolution` Clip the astronomical solution `Default=FALSE`.

`astrosolution_min_or_max` Tune to maximum or minimum values of the astronomical solution `Default="max"`

`clip_high` Upper value to clip to.

`clip_low` Lower value to clip to.

`extract_astrosolution` Extract a certain astronomical cycle/component from a astronomical solution prior to anchoring `Default=FALSE`.

astro_period_up
: Upper bound of the astronomical cycle which is extracted from an astronomical solution. The upper bound value is a factor with which the astronomical component is multiplied by. Default=1.2

astro_period_down
: Lower bound of the astronomical cycle which is extracted from an astronomical solution. The upper bound value is a factor with which the astronomical component is multiplied by. Default=0.8

astro_period_cycle
: Period (in kyr) of the to be extracted astronomical component from the astronomical solution.

extract_proxy_signal
: Extract a certain astronomical cycle/component from a proxy signal Default=FALSE.

proxy_period_up
: Upper bound of the astronomical cycle to be extracted from the proxy record. The upper bound value is a factor with which the astronomical component is multiplied by. Default=1.2.

proxy_period_down
: Upper bound of the astronomical cycle to be extracted from the proxy record. The lower bound value is a factor with which the astronomical component is multiplied by. Default=0.8.

proxy_period_cycle
: Period in kyr of the astronomical cycle/component which is extracted from the proxy record.

Value

The output is a matrix with the 4 columns. The first column is the depth/time of the proxy tie-point. The second column is the time value of the astronomical solution tie-point. The third column is the proxy value of the proxy tie-point. The fourth column is the proxy/insolation value of the astronomical solution tie-point.

References

J. Laskar, P. Robutel, F. Joutel, M. Gastineau, A.C.M. Correia, and B. Levrard, B., 2004, A long term numerical solution for the insolation quantities of the Earth: Astron. Astrophys., Volume 428, 261-285.
Laskar, J., Fienga, A., Gastineau, M., Manche, H., 2011a, La2010: A new orbital solution for the long-term motion of the Earth: Astron. Astrophys., Volume 532, A89
Laskar, J., Gastineau, M., Delisle, J.-B., Farres, A., Fienga, A.: 2011b, Strong chaos induced by close encounters with Ceres and Vesta, Astron: Astrophys., Volume 532, L4.
J. Laskar,Chapter 4 - Astrochronology,Editor(s): Felix M. Gradstein, James G. Ogg, Mark D. Schmitz, Gabi M. Ogg,Geologic Time Scale 2020,Elsevier,2020,Pages 139-158,ISBN 9780128243602, <doi:10.1016/B978-0-12-824360-2.00004-8> or https://www.sciencedirect.com/science/article/pii/B9780128243602000048
Zeebe, R. E. and Lourens, L. J. Geologically constrained astronomical solutions for the Cenozoic era, Earth and Planetary Science Letters, 2022
Zeebe, R. E. and Lourens, L. J. Solar system chaos and the Paleocene-Eocene boundary age constrained by geology and astronomy. Science, <doi:10.1126/science.aax0612>
Zeebe, R. E. Numerical Solutions for the orbital motion of the Solar System over the Past 100 Myr: Limits and new results. The Astronomical Journal, 2017

Examples

```
## Not run:
# Use the \code{grey_track} example tracking points to anchor the grey scale data set \cr
# of Zeeden et al., (2013) to the p-0.5t la2004 solution

grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = TRUE,
   omega_nr = 8
 )

grey_track <- completed_series(
 wavelet = grey_wt,
 tracked_curve  = grey_track,
 period_up  = 1.25,
 period_down  = 0.75,
 extrapolate = TRUE,
genplot = FALSE
)
# Extract precession, obliquity and eccentricity to create a synthetic insolation curve

grey_prec <- extract_signal(
tracked_cycle_curve = grey_track[,c(1,2)],
wavelet = grey_wt,
period_up = 1.2,
period_down = 0.8,
add_mean = FALSE,
tracked_cycle_period = 22,
extract_cycle = 22,
tune = FALSE,
plot_residual = FALSE
)

grey_obl <- extract_signal(
 tracked_cycle_curve = grey_track[,c(1,2)],
 wavelet = grey_wt,
 period_up = 1.2,
 period_down = 0.8,
 add_mean = FALSE,
 tracked_cycle_period = 22,
 extract_cycle = 110,
 tune = FALSE,
 plot_residual = FALSE
)

grey_ecc <- extract_signal(
 tracked_cycle_curve = grey_track[,c(1,2)],
 wavelet = grey_wt,
 period_up = 1.25,
 period_down = 0.75,
 add_mean = FALSE,
 tracked_cycle_period = 22,
```

```
 extract_cycle = 40.8,
 tune = FALSE,
 plot_residual = FALSE
)

insolation_extract <- cbind(grey_ecc[,1],grey_prec[,2]+grey_obl[,2]+grey_ecc[,2]+mean(grey[,2]))
insolation_extract <- as.data.frame(insolation_extract)
insolation_extract_mins <- min_detect(insolation_extract)

#download ETP solution (p-0.5t la2004 solution) from the astrochron package \cr
#to create a insolation curve.
#install.packages("astrochron") if not install package.
#library("astrochron") load package to download p-0.5t la2004 solution.

astrosignal=astrochron::etp(tmin=5000,tmax=6000,pWt=1,oWt=-0.5,eWt=0,genplot=FALSE,verbose=FALSE)
astrosignal[,2] <- -1*astrosignal[,2]
astrosignal <- as.data.frame(astrosignal)

#anchor the synthetic insolation curve extracted from the greyscale record to the insolation curve.

anchor_points <- astro_anchor(
astro_solution = astrosignal,
proxy_signal = insolation_extract,
proxy_min_or_max = "min",
clip_astrosolution = FALSE,
astrosolution_min_or_max = "min",
clip_high = NULL,
clip_low = NULL,
extract_astrosolution  = FALSE,
astro_period_up  = NULL,
astro_period_down  = NULL,
astro_period_cycle  = NULL,
extract_proxy_signal  = FALSE,
proxy_period_up  = NULL,
proxy_period_down  = NULL,
proxy_period_cycle  = NULL
)
## End(Not run)
```

completed_series *Complete the tracking of cycle in a wavelet spectra*

Description

Use the traced series and the existing wavelet spectra to complete the tracking of a cycle of the wavelet spectra. The selected points using the `track_period_wavelet` function form a incomplete line unless every point is tracked. However clicking every individual point along a wavelet ridge is time intensive and error prone. To avoid errors and save time the `completed_series` function can be used to complete the tracing of a cycle in a wavelet spectra.The `completed_series` function interpolates the data points selected using the `track_period_wavelet`. A a search a algorithm then looks up and replaces the interpolated curve values with the values of the nearest spectral peak in the wavelet spectra.

Usage

```
completed_series(
  wavelet = NULL,
  tracked_curve = NULL,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = TRUE
)
```

Arguments

wavelet	Wavelet object created using the analyze_wavelet function.
tracked_curve	Traced period result from the track_period_wavelet function.
period_up	The upper period parameter is a factor of the to be completed curve Default=1.2, which puts a limit on how far the codecompleted_series function can "look up" to find the next spectral peak. if no spectral peak is found data is interpolated between spectral peaks.
period_down	The lower period parameter is a factor of the to be completer curve Default=0.8, , #'which puts a limit on how far the codecompleted_series function can "look down" to find the next spectral peak. if no spectral peak is found data is interpolated between spectral peaks.
extrapolate	Extrapolate the completed curve when through parts where no spectral peaks could be traced Default=TRUE.
genplot	Generate a plot Default=TRUE. The red curve is the completed curve, the black curve is the original curve.

Value

Returns a matrix with 2 columns The first column is the depth axis The second column is the completed tracking of the period a cycle of the wavelet spectra

Examples

```
#Use the \code{grey_track} example points to complete the tracking of the \cr
# precession cycle in the wavelet spectra of the greyscale data set \cr
# of Zeeden et al., (2013).



grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = TRUE,
   omega_nr = 8
 )

#The ~22kyr precession cycle is between 0.25 and 1m The \code{grey_track} data \cr
#set is a pre-loaded uncompleted tracking of the precession cycle
```

```
#grey_track <- track_period_wavelet(
#astro_cycle = 22,
#wavelet = NULL,
#n.levels = 100,
#periodlab = "Period (metres)",
#x_lab = "depth (metres)"
#)


grey_track <- completed_series(
 wavelet = grey_wt,
 tracked_curve  = grey_track,
 period_up  = 1.25,
 period_down  = 0.75,
 extrapolate = TRUE,
genplot = TRUE
)
```

`curve2sedrate` *Converts a tracked tracked to a sedimentation rate curve*

Description

Converts the period of a tracked cycle to a sedimentation rate curve by assiging a duration (in kyr) to the period of a tracked cycle

Usage

```
curve2sedrate(tracked_cycle_curve = NULL, tracked_cycle_period = NULL)
```

Arguments

`tracked_cycle_curve`
: A cycle tracked using the `track_period_wavelet` function
  Any input (matrix or dataframe) in which the first column is depth in meters and the second column is period in meters

`tracked_cycle_period`
: Period of the tracked cycle (in kyr).

Value

The output is a matrix with 2 columns The first column is depth The second column sedimentation rate in cm/kyr

Examples

```
#Conversion of the period (in meters) of a 405 kyr eccentricity cycle tracked \cr
#in a wavelet spectra by assigning a duration of  405 kyr to the tracked cycle.
# perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
```

```
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                  wavelet=mag_wt,
#                                  n.levels = 100,
#                                  periodlab = "Period (metres)",
#                                  x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} is used \cr
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = TRUE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = TRUE, print_span = TRUE)

#convert period in meters to sedrate in cm/kyr
mag_track_sedrate <- curve2sedrate(tracked_cycle_curve=mag_track_complete,
tracked_cycle_period=405)
```

---

curve2time *Converts the tracked curve to a depth time space*

Description

Converts the tracked curve to a depth time space.

Usage

```
curve2time(
  tracked_cycle_curve = NULL,
  tracked_cycle_period = NULL,
  genplot = TRUE
)
```

Arguments

tracked_cycle_curve
: Curve of the cycle tracked using the track_period_wavelet function
Any input (matrix or data frame) in which the first column is depth in meters and the second column is period in meters can be used.

tracked_cycle_period
: Period of the tracked curve in kyr.

genplot
: Generates a plot with depth vs time Default=TRUE.

Value

The output is a matrix with 2 columns. The first column is depth. The second column sedimentation rate in cm/kyr. If genplot=TRUE then a depth vs time plot will be plotted.

Author(s)

Based on the the sedrate2time function of the astrochron-package.

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis doi:<doi:10.1016/j.earscirev.2018.11.015>.

Examples

```
#Convert a tracked curve to a depth time space. The examples uses the \cr
#magnetic susceptibility data set of De pas et al., (2018).\cr

#'# perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                  wavelet=mag_wt,
#                                  n.levels = 100,
#                                  periodlab = "Period (metres)",
#                                  x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} is used \cr
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = TRUE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = TRUE, print_span = TRUE)

#convert period in meters to sedrate depth vs time
```

```
mag_track_time<- curve2time(tracked_cycle_curve=mag_track_complete,
tracked_cycle_period=405,
genplot=TRUE)
```

---

curve2tune *Converts data from the depth to the time domain*

Description

Converts a data set from the depth to the time domain using a tracked curve/cycle to depth domain an assigning a duration (in kyr) set tracked curve/cycle.

Usage

```
curve2tune(
  data = NULL,
  tracked_cycle_curve = NULL,
  tracked_cycle_period = NULL,
  genplot = TRUE
)
```

Arguments

`data` Data set (matrix with 2 columns 1st column depth 2nd column proxy value) which was used as input for the `analyze_wavelet` function.
That result was then used to tracked a cycle using the `track_period_wavelet` function

`tracked_cycle_curve` Tracking result of a cycle tracked using the `track_period_wavelet` function Any input (matrix or data frame) in which the first column is depth in meters and the second column is period in meters can be used.

`tracked_cycle_period` Period of the tracked curve (in kyr).

`genplot` If `genplot=TRUE` 3 plots stacked on top of each other will be plotted. Plot 1: the original data set. Plot 2: the depth time plot. Plot 3: the data set in the time domain.

Value

The output is a matrix with 2 columns. The first column is time. The second column sedimentation proxy value.

If `genplot=TRUE` then 3 plots stacked on top of each other will be plotted. Plot 1: the original dataset. Plot 2: the depth time plot. Plot 3: the dataset in the time domain.

Author(s)

Based on the the sedrate2time function of the astrochron-package

References

Routines for astrochronologic testing, astronomical time scale construction, and time series analysis doi:10.1016/j.earscirev.2018.11.015

Examples

```
#The example uses the magnetic susceptibility data set of De pas et al., (2018).
# perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                    wavelet=mag_wt,
#                                    n.levels = 100,
#                                    periodlab = "Period (metres)",
#                                    x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} is used
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = TRUE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = TRUE, print_span = TRUE)

mag_track_time<- curve2tune(data=mag,
                           tracked_cycle_curve=mag_track_complete,
                           tracked_cycle_period=405,
                           genplot=TRUE)
```

---

extract_power *Extract power from a wavelet spectra*

Description

Extracts the the spectral power from a wavelet spectra in the depth domain using a traced period and boundaries surround the traced period. The extraction of spectral is useful for cyclostratigraphic studies because the spectral power of an astronomical cycle is modulated by higher order astronomical cycles. The spectral power record from an astronomical cycle can thus be used as a proxy for amplitude modulating cycles The traced period result from the track_period_wavelet function with boundaries is used to extract spectral power in the depth domain from a wavelet spectra.

Usage

```
extract_power(
  completed_series = NULL,
  wavelet = NULL,
  period_up = 1.2,
  period_down = 0.8,
  tracked_cycle_period = NULL,
  extract_cycle_power = NULL
)
```

Arguments

- `completed_series` Traced period result from the track_period_wavelet function completed using the completed_series. The input can be pre-smoothed using the the loess_auto function.
- `wavelet` Wavelet object created using the analyze_wavelet function.
- `period_up` Upper period as a factor of the to be extracted power `Default=1.2`.
- `period_down` Lower period as a factor of the to be extracted power `Default=0.8`.
- `tracked_cycle_period` Period of the tracked cycle (make sure that `tracked_cycle_period`) and `extract_cycle_power`) are of the same unit (either depth or time domain).
- `extract_cycle_power` Period of the cycle for which the power will be extracted (make sure that `extract_cycle_power`) and `tracked_cycle_period`) are of the same unit (either depth or time domain).

Value

Returns a matrix with 3 columns. The first column is depth/time. The second column is extracted power. The third column is extracted power/total power.

Examples

```
#Extract the power of the 405 kyr eccentricity cycle from the the magnetic \cr
# susceptibility data set of De pas et al., (2018) \cr
#Perform the CWT on the magnetic susceptibility data set of De pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (metres)",
```

```
#                                    x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} \cr
#is used
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = TRUE
)

#Smooth the completed tracking of the 405 kyr eccentricity cycle in the wavelet spectra

mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = TRUE, print_span = TRUE)

#extract the spectral power of the 405 kyr eccentricity cycle
mag_power <- extract_power(
completed_series = mag_track_complete,
wavelet = mag_wt,
period_up = 1.2,
period_down = 0.8,
tracked_cycle_period = 405,
extract_cycle_power = 405
)
```

---

extract_power_stable *Extract power from a wavelet spectra by using a constant period/duration*

Description

Extract spectral power from the wavelet using a constant period/duration and boundaries as selection criteria. The extraction of spectral is useful for cyclostratigraphic studies because the spectral power of an astronomical cycle is modulated by higher order astronomical cycles. The spectral power record from an astronomical cycle can thus be used as a proxy for amplitude modulating cycles. The spectral power is extracted from a wavelet spectra which was created using the analyze_wavelet function for a given, cycle, period_up and period_down

Usage

```
extract_power_stable(
  wavelet = NULL,
  cycle = NULL,
  period_up = 1.2,
  period_down = 0.8
)
```

Arguments

wavelet	Wavelet object created using the analyze_wavelet function.
cycle	Period of cycle for which the power will be extracted from the record.
period_up	Upper period as a factor of the to be extracted power Default=1.2.
period_down	Lower period as a factor of the to be extracted power Default=0.8.

Value

Returns a matrix with 3 columns. The first column is depth/time. The second column is extracted power. The third column is extracted power/total power.

Examples

```
#Extract the spectral power of the 210 yr de Vries cycle from the Total Solar \cr
#Irradiance data set of Steinhilver et al., (2012).

TSI_wt <-
 analyze_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = TRUE,
   omega_nr = 6
 )
TSI_wt_pwr_de_Vries_cycle <-  extract_power_stable(
 wavelet = TSI_wt,
 cycle = 210,
 period_up = 1.2,
 period_down = 0.8
)
```

---

extract_signal *Extract signal from a wavelet spectra using a traced period curve*

Description

Extract signal power from the wavelet in the depth domain using the traced period.

Usage

```
extract_signal(
  tracked_cycle_curve = NULL,
  wavelet = NULL,
  period_up = 1.2,
  period_down = 0.8,
  add_mean = TRUE,
  tracked_cycle_period = NULL,
  extract_cycle = NULL,
  tune = FALSE,
```

```
  plot_residual = FALSE
)
```

Arguments

tracked_cycle_curve
: Traced period result from the track_period_wavelet function completed using the completed_series. The input can be pre-smoothed using the the loess_auto function.

wavelet
: wavelet object created using the analyze_wavelet function.

period_up
: Upper period as a factor of the to be extracted cycle Default=1.2.

period_down
: Lower period as a factor of the to be extracted cycle Default=0.8.

add_mean
: Add mean to the extracted cycle Default=TRUE.

tracked_cycle_period
: Period in time of the traced cycle.

extract_cycle
: Period of the to be extracted cycle.

tune
: Convert record from the depth to the time domain using the traced period Default=FALSE.

plot_residual
: Plot the residual signal after extraction of set cycle Default=FALSE.

Value

Returns a matrix with 2 columns The first column is depth/time The second column is extracted signal

Examples

```
#Extract the 405 kyr eccentricity cycle from the the magnetic susceptibility \cr
#record of the Sullivan core and use the Gabor uncertainty principle to define \cr
#the mathematical uncertainty of the analysis and use a factor of that standard \cr
#deviation to define boundaries.

#Perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (metres)",
#                                   x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} is used \cr
mag_track <- mag_track_solution


mag_track_complete <- completed_series(
  wavelet = mag_wt,
```

```
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = TRUE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = TRUE, print_span = TRUE)

# extract the 405 kyr eccentricty cycle from the wavelet spectrum and use the \cr
# tracked cycle curve and set factors of the extracted cycle as boundaries

mag_405_ecc  <- extract_signal(
tracked_cycle_curve = mag_track_complete,
wavelet = mag_wt,
period_up = 1.2,
period_down = 0.8,
add_mean = TRUE,
tracked_cycle_period = 405,
extract_cycle = 405,
tune = FALSE,
plot_residual = FALSE
)
```

---

extract_signal_stable *Extract a signal/cycle from a wavelet spectra using a set period and boundaries*

Description

Extracts a cycle from the wavelet object created using the analyze_wavelet function using a fixed period and fixed period boundaries defined as factors of the original cycle

Usage

```
extract_signal_stable(
  wavelet = NULL,
  cycle = NULL,
  period_up = 1.2,
  period_down = 0.8,
  add_mean = TRUE,
  plot_residual = FALSE
)
```

Arguments

wavelet	Wavelet object created using the analyze_wavelet function.
cycle	Period of the cycle which needs to be extracted.
period_up	Upper period as a factor of the to be extracted cycle Default=1.2.
period_down	Lower period as a factor of the to be extracted cycle Default=0.8.

add_mean Add mean to the extracted cycle `Default=TRUE`.

plot_residual plot the residual signal after extraction of set cycle `Default=FALSE`.

Value

#'Returns a matrix with 2 columns. The first column is time/depth. The second column is the extracted signal/cycle.

Examples

```
#Example in which the ~210yr de Vries cycle is extracted from the Total Solar
#Irradiance data set of Steinhilver et al., (2012)\cr

#Perform the CWT
TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = TRUE,
   omega_nr = 6
 )

#Extract the 210 yr de Vries cycle from the wavelet spectra
de_Vries_cycle <- extract_signal_stable(wavelet=TSI_wt,
cycle=210,
period_up =1.25,
period_down = 0.75,
add_mean=TRUE,
plot_residual=FALSE)
```

---

extract_signal_stable_V2 *Extract signal from a wavelet spectrum using a upper and lower period boundary*

Description

Extract a signal from the wavelet using a upper and lower period boundary

Usage

```
extract_signal_stable_V2(
  wavelet = NULL,
  period_max = NULL,
  period_min = NULL,
  add_mean = TRUE,
  plot_residual = FALSE
)
```

Arguments

wavelet	wavelet object created using the analyze_wavelet function.
period_max	Maximum period (upper boundary) to be used to extract a cycle.
period_min	Minimum period (lower boundary) to be used to extract a cycle.
add_mean	Add mean to the extracted cycle Default=TRUE.
plot_residual	Plot the signal from which the extracted cycle is subtracted Default=FALSE.

Value

Signal extracted from the wavelet spectra. Output is a matrix with the first column being depth/time and the second column is the cycle extracted from the proxy record.

Examples

```
#Example in which the ~210yr de Vries cycle is extracted from the Total Solar \cr
# Irradiance data set of Steinhilver et al., (2012)\cr

TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = TRUE,
   omega_nr = 6
 )

de_Vries_cycle <- extract_signal_stable_V2(wavelet=TSI_wt,
period_max = 240,
period_min = 180,
add_mean=TRUE,
plot_residual=FALSE)
```

---

extract_signal_standard_deviation

Extract a signal using standard deviation

Description

Extract signal from a wavelet spectra in the depth domain using a the standard deviation of the omega (number of cycles) as boundaries. The uncertainty is based on the Gabor uncertainty principle applied to the continuous wavelet transform using a Morlet wavelet. The calculated uncertainty is the underlying analytical uncertainty which is the result of applying the Gabor uncertainty principle to the continuous wavelet transform using a Morlet wavelet.

Usage

```
extract_signal_standard_deviation(
  wavelet = NULL,
  tracked_cycle_curve = NULL,
  multi = 1,
  extract_cycle = NULL,
  tracked_cycle_period = NULL,
  add_mean = TRUE,
  tune = FALSE,
  genplot_uncertainty_wt = TRUE,
  genplot_extracted = TRUE
)
```

Arguments

`wavelet` Wavelet object created using the `analyze_wavelet` function.

`tracked_cycle_curve` Curve of the cycle tracked using the `track_period_wavelet` function. Any input (matrix or dataframe) in which the first column is depth or time and the second column is period should work.

`multi` multiple of the standard deviation to be used as boundaries for the cycle extraction `Default=1`.

`extract_cycle` Period of the cycle to be extracted.

`tracked_cycle_period` Period of the tracked cycle.

`add_mean` Add mean to the extracted cycle `Default=TRUE`.

`tune` Tune data set using the `Default=tracked_cycle_curve` curve `Default=FALSE`.

`genplot_uncertainty_wt` Generate a wavelet spectra plot with the tracked curve and its analytical uncertainty based the Gabor uncertainty principle applied continuous wavelet transform using a Morlet wavelet on superimposed on top of it. In the plot the red curve and blue curves are the upper and lower bounds based on the `multi` parameter which x-times the standard deviation of uncertainty. The black curve is the `Default=tracked_cycle_curve` curve.

`genplot_extracted` Generates a plot with the data set and the extracted cycle on top `Default=TRUE` of it.

Value

Signal extracted from the wavelet spectra. Output is a matrix with the first column being depth/time and the second column is the astronomical cycle extracted from the proxy record

If `genplot_uncertainty_wt=TRUE` then a wavelet spectra will be plotted with the uncertainty superimposed on top of it. In the plot the red curve and blue curves are the upper and lower bounds based on the `multi` parameter.The black curve is the `Default=tracked_cycle_curve` curve. If `genplot_extracted=TRUE` plot with the data set and the extracted cycle on top of it will be plotted.

References

Gabor, Dennis. "Theory of communication. Part 1: The analysis of information." Journal of the Institution of Electrical Engineers-part III: radio and communication engineering 93, no. 26 (1946): 429-441.

Russell, Brian, and Jiajun Han. "Jean Morlet and the continuous wavelet transform. " CREWES Res. Rep 28 (2016): 115.

Examples

```
#Extract the 405 kyr eccentricity cycle from the the magnetic susceptibility \cr
#record of the Sullivan core and use the Gabor uncertainty principle to define \cr
# the mathematical uncertainty of the analysis and use a factor of that standard \cr
#  deviation to define boundaries

# perform the CWT
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                   wavelet=mag_wt,
#                                   n.levels = 100,
#                                   periodlab = "Period (metres)",
#                                   x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} is used \cr
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = TRUE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = TRUE, print_span = TRUE)

# extract the 405 kyr eccentricty cycle from the wavelet spectrum and use \cr
# the Gabor uncertainty principle to define the mathematical uncertainty of \cr
# the analysis and use a multiple of the derived standard deviation to define boundaries

mag_405_ecc <- extract_signal_standard_deviation(
wavelet = mag_wt,
tracked_cycle_curve = mag_track_complete,
multi = 1,
extract_cycle = 405,
tracked_cycle_period = 405,
```

```
add_mean = TRUE,
tune = FALSE,
genplot_uncertainty_wt = TRUE,
genplot_extracted = TRUE
)
```

`grey` *Grey scale record IODP 926 of Zeeden et al., (2013)*

Description

IODP 926 grey scale record of Zeeden et al., (2013) for the (154-174m) interval. The (154-174m) interval spans the Miocene.

Details

Column 1: depth (meters)
Column 2: greyscale value

References

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, <doi:10.1016/j.palaeo.2012.11.009>

`grey_track` *Tracking points of the precession (22 kyr cycle) IODP 926 grey scale (154-174m) record of Zeeden et al., (2013)*

Description

Example data which consists of tracking points of the precession (22 kyr cycle) in the wavelet spectra of the IODP 926 grey scale (154-174m) record of Zeeden et al., (2013)

Details

Column 1: Depth (meters)
Column 2: period (meters)

References

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, <doi:10.1016/j.palaeo.2012.11.009>

`Hilbert_transform` *Perform a Hilbert transform on a signal*

Description

Extract the amplitude modulation using the Hilbert transform.

Usage

```
Hilbert_transform(data = NULL, demean = TRUE)
```

Arguments

data	Input is a time series with the first column being depth or time and the second column being a proxy.
demean	Remove the mean from the time series.

Value

Returns a matrix with 2 columns. The first column is depth/time. The second column is the hilbert transform of the signal.

Author(s)

Based on the the inst.pulse function of the DecomposeR

References

Wouters, S., Crucifix, M., Sinnesael, M., Da Silva, A.C., Zeeden, C., Zivanovic, M., Boulvain, F., Devleeschouwer, X., 2022, "A decomposition approach to cyclostratigraphic signal processing". Earth-Science Reviews 225 (103894).<doi:10.1016/j.earscirev.2021.103894>

Huang, Norden E., Zhaohua Wu, Steven R. Long, Kenneth C. Arnold, Xianyao Chen, and Karin Blank. 2009. "On Instantaneous Frequency". Advances in Adaptive Data Analysis 01 (02): 177–229. <doi:10.1142/S1793536909000096>

Examples

```
#Example in which the Hilbert transform (eg. amplitude modulation) of the ~210yr \cr
#de Vries cycle is extracted from the Total Solar Irradiance data set of \cr
#Steinhilber et al., (2012)

#Perform the CWT
TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = TRUE,
   omega_nr = 6
 )
```

```
#Extract the 210 yr de Vries cycle from the wavelet spectra
de_Vries_cycle <- extract_signal_stable(wavelet=TSI_wt,
cycle=210,
period_up =1.25,
period_down = 0.75,
add_mean=TRUE,
plot_residual=FALSE)

#Perform the Hilbert transform on the amplitude record of the 210 yr de Vries \cr
# cycle which was extracted from the wavelet spectra

de_Vries_cycle_hilbert <- Hilbert_transform(data=de_Vries_cycle,demean=TRUE)
```

`loess_auto` *Perform an automatically loess based smoothing of a timeseries*

Description

Perform an automatically loess based smoothing of a timeseries. The local polynomial regression with automatic smoothing parameter selection is based on an optimization using the aicc bias-corrected AIC criterion and the gcv generalized cross-validation criterion.

Usage

```
loess_auto(time_series = NULL, genplot = TRUE, print_span = TRUE)
```

Arguments

`time_series`	Input is a time series with the first column being depth or time and the second column being a proxy
`genplot`	Option to generate plot `Default=TRUE`. The plot will consist of the original signal in blue, the smoothed plot is displayed in black and the + and - 1 sd bounds of the smoothing are displayed in red.
`print_span`	Print span length as a fraction of the total length of the record.

Value

A matrix with 3 columns. The first column is depth/time. The second column is the smoothed curve. The third column is difference between the original curve and the smoothed curve.

Author(s)

Based on the the `loess.as` function of the fANCOVA package.

References

Cleveland, W. S. (1979) Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association. 74, 829–836. Hurvich, C.M., Simonoff, J.S., and Tsai, C.L. (1998), Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion. Journal of the Royal Statistical Society B. 60, 271–293 Golub, G., Heath, M. and Wahba, G. (1979). Generalized cross validation as a method for choosing a good ridge parameter. Technometrics. 21, 215–224.

Examples

```
#'smooth the period curve of the 405 kyr eccentricity cycle extracted from \cr
# the magnetic susceptibility data set of De pas et al., (2018) \cr
#perform the CWT on the magnetic susceptibility data set of De pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                    wavelet=mag_wt,
#                                    n.levels = 100,
#                                    periodlab = "Period (metres)",
#                                    x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} is used \cr
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = TRUE
)

#Smooth the completed tracking of the 405 kyr eccentricity cycle as tracked in the wavelet spectra
mag_track_complete <- loess_auto(time_series = mag_track_complete,
genplot = TRUE, print_span = TRUE)
```

---

mag — *Magnetic susceptibility data of the Sullivan core of De Pas et al., (2018)*

Description

The magnetic susceptibility data set consists of the magnetic susceptibility measurements of De Pas et al., (2018), which measured the magnetic susceptibility on the Sullivan core which is of Famennian age.

Details

Column 1: depth value (meters depoth)
Column 2: magnetic susceptibility value

References

Damien Pas, Linda Hinnov, James E. (Jed) Day, Kenneth Kodama, Matthias Sinnesael, Wei Liu, Cyclostratigraphic calibration of the Famennian stage (Late Devonian, Illinois Basin, USA), Earth and Planetary Science Letters, Volume 488, 2018, Pages 102-114, ISSN 0012-821X, <doi:1016/j.epsl.2018.02.010>

`mag_track_solution` *Period of the 405 kyr ecc cycle in the magnetic susceptibility record of the Sullivan core*

Description

Data points which give the period (in meters) of the 405 kyr eccentricity cycle tracked in the wavelet spectra of the magnetic susceptibility record of the Sullivan core
The period was tracked using the `track_period_wavelet` function
The tracking is based on the original age model of De Pas et al., (2018)

Details

Column 1: Depth (meters)
Column 2: tracked period of 405 kyr eccentricity cycle (meters)

References

Damien Pas, Linda Hinnov, James E. (Jed) Day, Kenneth Kodama, Matthias Sinnesael, Wei Liu, Cyclostratigraphic calibration of the Famennian stage (Late Devonian, Illinois Basin, USA), Earth and Planetary Science Letters, Volume 488, 2018, Pages 102-114, ISSN 0012-821X, <doi:10.1016/j.epsl.2018.02.010>

`max_detect` *Detect and filter out all maxima in a signal*

Description

The `max_detect` function is used to detect and filter out local maxima in a sinusoidal signal.

Usage

```
max_detect(data = NULL)
```

Arguments

`data`	Matrix or data frame with the first column being depth or time and the second column being a proxy

Value

#Returns a matrix with 2 columns first column is depth/time the second column are local maxima values

Examples

```
#Example in which the ~210yr de Vries cycle is extracted from the Total Solar
#Irradiance data set of Steinhilver et al., (2012)\cr
#after which all maxima are extracted

TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = TRUE,
   omega_nr = 6
 )

de_Vries_cycle <- extract_signal_stable(wavelet=TSI_wt,
cycle=210,
period_up =1.25,
period_down = 0.75,
add_mean=TRUE,
plot_residual=FALSE)


min_de_Vries_cycle <- min_detect(de_Vries_cycle)
```

---

min_detect *Detect and filter out all minima in a signal*

Description

The min_detect function is used to detect and filter out local minima in a sinusoidal signal

Usage

```
min_detect(data)
```

Arguments

data	Matrix or data frame with first column being depth or time and the second column being a proxy

Value

#Returns a matrix with 2 columns first column is depth/time the second column are local minima values

Examples

```
#Example in which the ~210yr de Vries cycle is extracted from the Total Solar \cr
#Irradiance data set of Steinhilber et al., (2012)\cr
#after which all minima are extracted
```

```
TSI_wt <-
analyze_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = TRUE,
   omega_nr = 6
 )

de_Vries_cycle <- extract_signal_stable(wavelet=TSI_wt,
cycle=210,
period_up =1.25,
period_down = 0.75,
add_mean=TRUE,
plot_residual=FALSE)


min_de_Vries_cycle <- min_detect(de_Vries_cycle)
```

`model_red_noise_wt` *Models average spectral power based curves based on a red-noise signal generated using the characteristics of an input signal.*

Description

The model_red_noise_wt function is used to generate average spectral power curves based on and input signal and set wavelet settings.

Usage

```
model_red_noise_wt(wavelet = NULL, n_simulations = NULL, verbose = TRUE)
```

Arguments

wavelet	Wavelet object created using the analyze_wavelet function.
n_simulations	Number of red noise simulations.
verbose	Print text Default=TRUE.

Value

Returns a matrix in which each column represents the average spectral power resulting from a red-noise run.

Examples

```
## Not run:
#'#generate average spectral power curves based on red noise curves which are\cr
# based on the magnetic susceptibility record of the Sullivan core of De pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
```

```
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#increase n_simulations to better define the red noise spectral power curve
mag_wt_red_noise <- model_red_noise_wt(wavelet=mag_wt,
n_simulations=100,
verbose=FALSE)

## End(Not run)
```

---

percentile_from_red_noise

Calculate average spectral power from red noise curves for a given percentile

---

Description

The percentile_from_red_noise function is used to generate and average spectral power curve based on a set percentile based. To generate the percentile curve the results of the model_red_noise_wt function are used.

Usage

```
percentile_from_red_noise(red_noise = NULL, wavelet = NULL, percentile = NULL)
```

Arguments

red_noise	Red noise curves generated using the model_red_noise_wt function.
wavelet	Wavelet object created using the analyze_wavelet function.
percentile	Percentile value (0-1).

Value

Returns a matrix with 2 columns.
The first column is the period (m).
The second column is the spectral power at percentile x based on
the red noise modelling runs.

Examples

```
## Not run:
#'#generate red noise curves based on the magnetic susceptibility record of \cr
#the Sullivan core of De pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
```

```
verbose = FALSE,
omega_nr = 10)

mag_wt_red_noise <- model_red_noise_wt(data=NULL,
n_simulations =1000,
verbose=FALSE)

prob_curve <- percentile_from_red_noise(
red_noise = mag_wt_red_noise,
wavelet = mag_wt,
percentile = 0.9
)
## End(Not run)
```

`plot_avg_wavelet` *Plot the average spectral power of a wavelet spectra*

Description

Plot the average spectral power of a wavelet spectra using the results of the `analyze_wavelet` function.

Usage

```
plot_avg_wavelet(wavelet = NULL, y_lab = "Power", x_lab = "period (metres)")
```

Arguments

`wavelet`	Wavelet object created using the `analyze_wavelet` function.
`y_lab`	Label for the y-axis `Default="Power"`.
`x_lab`	Label for the x-axis `Default="depth (metres)"`.

Value

The output is a plot of the average spectral power of a wavelet spectra

Examples

```
#Example 1. Plot the average spectral power of the wavelet spectra of \cr
# the Total Solar Irradiance data set of Steinhilver et al., (2012)
TSI_wt <-
 analyze_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = TRUE,
   omega_nr = 6
 )

plot_avg_wavelet(wavelet=TSI_wt,
                 y_lab= "power",
```

```
                 x_lab="period (years)")


#Example 2. Plot the average spectral power of the wavelet spectra of \cr
# the magnetic susceptibility data set of De pas et al., (2018)
mag_wt <-
analyze_wavelet(
data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10
)
plot_avg_wavelet(wavelet=mag_wt,
                 y_lab= "power",
                 x_lab="period (metres)")



#Example 3. Plot the average spectral power of the wavelet spectra of \cr
#the greyscale data set of Zeeden et al., (2013)
grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = TRUE,
   omega_nr = 8
 )

plot_avg_wavelet(wavelet=grey_wt,
                 y_lab= "power",
                 x_lab="period (metres)")
```

---

plot_wavelet *Plots a wavelet power spectra*

Description

Plot wavelet spectra using the outcome of the analyze_wavelet function.

Usage

```
plot_wavelet(
  wavelet = NULL,
  plot.COI = TRUE,
  n.levels = 100,
  color.palette = "rainbow(n.levels, start = 0, end = 0.7)",
  useRaster = TRUE,
  periodlab = "Period (metres)",
```

```
  x_lab = "depth (metres)"
)
```

Arguments

wavelet	wavelet object created using the analyze_wavelet function.
plot.COI	Option to plot the cone of influence Default=TRUE.
n.levels	Number of color levels Default=100.
color.palette	Definition of color palette Default="rainbow(n.levels, start = 0, end = 0.7)".
useRaster	plot as a raster or vector image Default=TRUE. WARNING plotting as a vector image is computationally intensive.
periodlab	label for the y-axis Default="Period (metres)".
x_lab	label for the x-axis Default="depth (metres)".

Value

The output is a plot of a wavelet spectra.

Author(s)

Code based on the R packages :WaveletComp and biwavelet which are based on the wavelet MATLAB program written by Christopher Torrence and Gibert P. Compo.

References

Angi Roesch and Harald Schmidbauer (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Gouhier TC, Grinsted A, Simko V (2021). R package biwavelet: Conduct Univariate and Bivariate Wavelet Analyses. (Version 0.20.21), https://github.com/tgouhier/biwavelet

Torrence, C., and G. P. Compo. 1998. A Practical Guide to Wavelet Analysis. Bulletin of the American Meteorological Society 79:61-78.

Examples

```
#Example 1. A plot of a wavelet spectra using the Total Solar Irradiance \cr
# data set of Steinhilver et al., (2012)
TSI_wt <-
 analyze_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = TRUE,
   omega_nr = 6
 )

plot_wavelet(
 wavelet = TSI_wt,
plot.COI = TRUE,
 n.levels = 100,
 color.palette = "rainbow(n.levels, start = 0, end = 0.7)",
 useRaster = TRUE,
 periodlab = "Period (years)",
```

```
x_lab = "years (before present)"
)

#Example 2. A plot of a wavelet spectra using the magnetic susceptibility \cr
#data set of De pas et al., (2018)
mag_wt <-
analyze_wavelet(
data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10
)
plot_wavelet(
 wavelet = mag_wt,
 plot.COI = TRUE,
 n.levels = 100,
 color.palette = "rainbow(n.levels, start = 0, end = 0.7)",
 useRaster = TRUE,
 periodlab = "Period (metres)",
 x_lab = "depth (metres)"
)

#Example 3. A plot of a wavelet spectra using the greyscale \cr
# data set of Zeeden et al., (2013)
grey_wt <-
 analyze_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = TRUE,
   omega_nr = 8
 )
plot_wavelet(
 wavelet = grey_wt,
 plot.COI = TRUE,
 n.levels = 100,
 color.palette = "rainbow(n.levels, start = 0, end = 0.7)",
 useRaster = TRUE,
 periodlab = "Period (metres)",
 x_lab = "depth (metres)"
)
```

sum_power_sedrate — *Calculate sum of maximum spectral power for sedimentation rates based for a wavelet spectra*

Description

The sum_power_sedrate function is used calculate the sum of maximum spectral power for a list of astronomical cycles from a wavelet spectra. The data is first normalized using the average spectral power curves for a given percentile based on results of the model_red_noise_wt function

Usage

```
sum_power_sedrate(
  red_noise = NULL,
  wavelet = NULL,
  percentile = NULL,
  sedrate_low = NULL,
  sedrate_high = NULL,
  spacing = NULL,
  cycles = c(NULL),
  x_lab = "depth",
  y_lab = "sedrate",
  genplot = TRUE
)
```

Arguments

red_noise	Red noise curves generated using the model_red_noise_wt function
wavelet	Wavelet object created using the analyze_wavelet function
percentile	Percentile value (0-1) of the rednoise runs which is used to normalize the data for. To account for the distribution/distortion of the spectral power distribution based on the analytical technique and random red-noise the data is normalized against a percentile based red-noise curve which is the results of the 'model_red_noise_wt modelling runs.
sedrate_low	Minimum sedimentation rate (cm/kyr)for which the sum of maximum spectral power is calculated for.
sedrate_high	Maximum sedimentation rate (cm/kyr) for which the sum of maximum spectral power is calculated for.
spacing	Spacing (cm/kyr) between sedimentation rates
cycles	Astronomical cycles (in kyr) for which the combined sum of maximum spectral power is calculated for
x_lab	label for the y-axis Default="depth"
y_lab	label for the y-axis Default="sedrate"
genplot	Generate plot Default="TRUE"

Value

Returns a matrix with sum of maximum spectral power for a given sedimentation rates and a given depths.

If Default="TRUE" a plot is created with 3 subplots. Subplot 1 is plot in which the the sum of maximum spectral power for a given sedimentation rate is plotted for each depth given depth. Subplot 2 is a plot in which the average sum of maximum spectral power is plotted fro each sedimentation Subplot 3 is a color scale for subplot 1.

Examples

```
## Not run:
#estimate sedimentation rate for the the magnetic susceptibility record \cr
# of the Sullivan core of De pas et al., (2018).

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10)

#increase n_simulations to better define the red noise spectral power curve
mag_wt_red_noise <- model_red_noise_wt(wavelet=mag_wt,
n_simulations=100,
verbose=FALSE)

sedrates <- sum_power_sedrate(red_noise=mag_wt_red_noise,
wavelet=mag_wt,
percentile=0.75,
sedrate_low = 0.5,
sedrate_high = 4,
spacing = 0.05,
cycles = c(2376,1600,1180,696,406,110),
x_lab="depth",
y_lab="sedrate",
genplot = TRUE)

## End(Not run)
```

`track_period_wavelet` *Track the period of a cycle in a wavelet spectra*

Description

Interactively select points in a wavelet spectra to trace a period in a wavelet spectra. The `track_period_wavelet` function plots a wavelet spectra in which spectral peaks can selected allowing one to track a ridge hence one can track the a cycle with a changing period.

Usage

```
track_period_wavelet(
  astro_cycle = 405,
  wavelet = NULL,
  n.levels = 100,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)"
)
```

Arguments

astro_cycle	Duration (in kyr) of the cycle which traced.
wavelet	Wavelet object created using the analyze_wavelet function.
n.levels	Number of color levels Default=100.
periodlab	label for the y-axis Default="Period (metres)".
x_lab	label for the x-axis Default="depth (metres)".

Value

Results of the tracking of a cycle in the wavelet spectra is a matrix with 3 columns. The first column is depth/time The second column is the period of the tracked cycle The third column is the sedimentation rate based on the duration (in time) of the tracked cycle

Examples

```
## Not run:
#Track the 405kyr eccentricity cycle in the magnetic susceptibility record \cr
# of the Sullivan core of De pas et al., (2018)

mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10)

mag_track <- track_period_wavelet(astro_cycle = 405,
                                  wavelet=mag_wt,
                                  n.levels = 100,
                                  periodlab = "Period (metres)",
                                  x_lab = "depth (metres)")

## End(Not run)
```

TSI — *Total solar irradiation data (0-9400ka) of steinhilber et al., (2012)*

Description

The Total solar irradiation data set consists of the TSI values of steinhilber et al., (2012)

Details

Column 1: Age (kyr)
Column 2: Total solar Irradiation (TSI)

References

Steinhilber, Friedhelm & Abreu, Jacksiel & Beer, Juerg & Brunner, Irene & Christl, Marcus & Fischer, Hubertus & Heikkilä, U. & Kubik, Peter & Mann, Mathias & Mccracken, K. & Miller, Heinrich & Miyahara, Hiroko & Oerter, Hans & Wilhelms, Frank. (2012). 9,400 Years of cosmic radiation and solar activity from ice cores and tree rings. Proceedings of the National Academy of Sciences of the United States of America. 109. 5967-71. 10.1073/pnas.1118965109. <doi:10.1073/pnas.1118965109>

`wavelet_uncertainty` *Calculate the uncertainty associated with the wavelet analysis based on the Gabor uncertainty principle*

Description

The `wavelet_uncertainty` function is used to calculate uncertainties associated with the wavelet analysis based on the Gabor uncertainty principle applied to the continuous wavelet transform using a Mortlet wavelet. The calculated uncertainty is the underlying analytical uncertainty which is the result of applying the Gabor uncertainty principle to the continuous wavelet transform using a Mortlet wavelet.

Usage

```
wavelet_uncertainty(
  tracked_cycle = NULL,
  period_of_tracked_cycle = NULL,
  wavelet = NULL,
  multi = 1,
  verbose = TRUE,
  genplot_time = TRUE,
  genplot_uncertainty = TRUE,
  genplot_uncertainty_wt = TRUE
)
```

Arguments

`tracked_cycle` Curve of the cycle tracked using the `track_period_wavelet` function Any input (matrix or dataframe) in which the first column is depth or time and the second column is period should work

`period_of_tracked_cycle` period of the tracked curve (in kyr).

`wavelet` wavelet object created using the `analyze_wavelet` function.

`multi` multiple of the standard deviation to be used for defining uncertainty `Default=1`.

`verbose` Print text `Default=TRUE`.

`genplot_time` plot time curves with a upper and lower uncertainty based on Gabor uncertainty principle applied to the continuous wavelet transform using a Mortlet wavelet, which uses which uses the omega number (number of cycles in the wavelet) at one standard deviation to define the analytical uncertainty `Default=TRUE`

`genplot_uncertainty`
: Plot period curves with upper and lower uncertainty based on Gabor uncertainty principle applied to the continuous wavelet transform using a Mortlet wavelet, which uses which uses the omega number (number of cycles in the wavelet) to define uncertainty at one standard deviation `Default=TRUE`

`genplot_uncertainty_wt`
: generate a wavelet plot with the uncertainty based on Gabor uncertainty principle applied to the continuous wavelet transform using a Mortlet wavelet superimposed on top of original wavelet plot. The red curve is period of the tracked curve plus the analytical uncertainty. The blue curve is period of the tracked curve min the analytical uncertainty. The blakc black curve is the curve tracked using the '`Default=tracked_cycle_curve` function `Default=TRUE`

Value

Results pertaining to the uncertainty calculated based on the Gabor uncertainty principle.
If the genplot_time is TRUE then a depth time plot will be plotted with 3 lines, the mean age,age plus x times the standard deviation and age minus x times the standard deviation .
If the genplot_uncertainty is TRUE then a curve will be plotted with the mean period, the tracked period plus x times the standard deviation and the tracked period minus x times the standard deviation.
If the genplot_uncertainty_wt is TRUE a wavelet spectra will be plotted with the tracked period, the tracked period plus x times the standard deviation,the tracked period minus x times the standard devation and the area in between will be shaded in grey.

Returns a matrix with 8 columns.
The first column is called "depth" eg. depth
The second column is "period" of the originally tracked period.
The third column is "frequency" of the originally tracked period.
The fourth column "uncertainty in frequency FWHM" is the uncertainty in frequency based on the Gabor uncertainty principle defined as (FWHM) full width at half maximum.
The fifth column "uncertainty in frequency x_times SD" is the uncertainty in frequency based on the Gabor uncertainty principle defined as times x standard deviations.
The sixth column "time mean" is the mean time based on the tracked period.
The seventh column "time plus x_times sd" is the time based on the tracked period plus x times the standard deviation.
The eight column "time min x_times sd" is the time based on the tracked period min x times the standard deviation.

References

Gabor, Dennis. "Theory of communication. Part 1: The analysis of information." Journal of the Institution of Electrical Engineers-part III: radio and communication engineering 93, no. 26 (1946): 429-441.

Russell, Brian, and Jiajun Han. "Jean Morlet and the continuous wavelet transform. " CREWES Res. Rep 28 (2016): 115.

Examples

```
#calculate the gabor uncertainty derived mathematical uncertainty of the \cr
#magnetic susceptibility record of the Sullivan core
```

```
mag_wt <- analyze_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = TRUE,
omega_nr = 10)

#Track the 405 kyr eccentricity cycle in a wavelet spectra

#mag_track <- track_period_wavelet(astro_cycle = 405,
#                                    wavelet=mag_wt,
#                                    n.levels = 100,
#                                    periodlab = "Period (metres)",
#                                    x_lab = "depth (metres)")

#Instead of tracking, the tracked solution data set \code{\link{mag_track_solution}} is used \cr
mag_track <- mag_track_solution

mag_track_complete <- completed_series(
  wavelet = mag_wt,
  tracked_curve = mag_track,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = FALSE,
  genplot = FALSE
)

mag_track_complete <- loess_auto(time_series = mag_track_complete,
 genplot = TRUE, print_span = TRUE)

uncertainty <- wavelet_uncertainty(
  tracked_cycle = mag_track_complete,
  period_of_tracked_cycle = 405,
  wavelet = mag_wt,
  multi=1,
  verbose = TRUE,
  genplot_time = TRUE,
  genplot_uncertainty = TRUE,
  genplot_uncertainty_wt = TRUE
)
```

WaverideR *Extracting Signals from Wavelet Spectra*

Description

The functions of the WaverideR package aid in the manipulation and extraction of cyclic signals using wavelets

Details

Package: WaverideR

Type: R package

Version: 0.2.0 (begin of 2023)

License: GPL (>= 2)

Note

If you want to use this package for publication or research purposes, please cite: to be submitted

Author(s)

Michiel Arts

Maintainer: Michiel Arts <michiel.arts@stratigraphy.eu>

`WaverideR_Datasets` *Example data sets for the WaverideR package*

Description

Data sets for testing the WaverideR package:
The `mag` data set is the magnetic susceptibility record of De Pas et al., (2018)
The `TSI` data set is the Total Solar Irradiance record of Steinhilber et al., (2012)
The `grey` data set is the grey scale record of IODP 926 for the interval (154-174m) which originates from Zeeden et al., (2013)
The `mag_track_solution` is the period of the 405 kyr ecc cycle in the magnetic susceptibility record of rom De Pas et al., (2018)
the `age_model_zeeden` data set is and age model (anchor points) for the IODP 926 grey scale (154-174m) record of Zeeden et al., (2013)
the `grey_track` data set consists of tracking points of the precession (22 kyr cycle) in the IODP 926 grey scale (154-174m) record of Zeeden et al., (2013)

References

Damien Pas, Linda Hinnov, James E. (Jed) Day, Kenneth Kodama, Matthias Sinnesael, Wei Liu, Cyclostratigraphic calibration of the Famennian stage (Late Devonian, Illinois Basin, USA), Earth and Planetary Science Letters, Volume 488,2018,Pages 102-114,ISSN 0012-821X, <doi:10.1016/j.epsl.2018.02.010>

Steinhilber, Friedhelm & Abreu, Jacksiel & Beer, Juerg & Brunner, Irene & Christl, Marcus & Fischer, Hubertus & Heikkilä, U. & Kubik, Peter & Mann, Mathias & Mccracken, K. & Miller, Heinrich & Miyahara, Hiroko & Oerter, Hans & Wilhelms, Frank. (2012). 9,400 Years of cosmic radiation and solar activity from ice cores and tree rings. Proceedings of the National Academy of Sciences of the United States of America. 109. 5967-71. 10.1073/pnas.1118965109. <doi:10.1073/pnas.1118965109>

Christian Zeeden, Frederik Hilgen, Thomas Westerhold, Lucas Lourens, Ursula Röhl, Torsten Bickert, Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site 926 between 5 and 14.4Ma, Palaeogeography, Palaeoclimatology, Palaeoecology,Volume 369,2013,Pages 430-451,ISSN 0031-0182, <doi:10.1016/j.palaeo.2012.11.009>

Index