Package: WaverideR (via r-universe)

October 19, 2026

Type Package

Title Extracting Signals from Wavelet Spectra

Version 0.6.0

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

Depends R (>= 3.5.0)

Imports astrochron, graphics, grDevices, parallel, stats, utils

Suggests knitr, magick, rmarkdown, scico, shiny, testthat (>= 3.0.0)

VignetteBuilder knitr

Description Tools for extracting and analyzing cyclic signals from time series: continuous wavelet, superlet and synchrosqueezed transforms, significance tests, tracking of astronomical cycles and sedimentation rates, age models with uncertainty and de-chirped spectra. When using the package please cite both the paper that introduced it, Arts et al. (2024) <doi:10.3389/feart.2024.1357751>, and the package itself (Arts 2023, <https://CRAN.R-project.org/package=WaverideR>); see citation(``WaverideR'').

License GPL (>= 2)

URL https://github.com/stratigraphy/WaverideR

Encoding UTF-8

LazyData true

NeedsCompilation no

Config/testthat/edition 3

Config/roxygen2/version 8.1.1

RoxygenNote 7.3.3

Repository https://stratigraphy.r-universe.dev

Date/Publication 2026-10-19 13:49:51 UTC

RemoteUrl https://github.com/stratigraphy/waverider

RemoteRef HEAD

RemoteSha dead88c45c1508257c5d54f32ed72b92b04d92eb

Contents

WaverideR-package . . . 6
add_wavelet . . . 10
add_wavelet_avg . . . 15
age_model_zeeden . . . 19
amtm_prewhiten . . . 20
amtm_spectrum . . . 21
analyse_Awavelet . . . 24
analyse_chirp_eha . . . 28
analyse_chirp_wavelet . . . 31
analyse_chirplet_sst . . . 33
analyse_conceft . . . 35
analyse_dechirped . . . 37
analyse_superlet . . . 39
analyse_superlet_sst . . . 43
analyse_wavelet . . . 46
analyse_wavelet_coherence . . . 50
analyse_wavelet_sst . . . 53
analyse_Xsuperlet . . . 56
analyse_Xwavelet . . . 60
anchor_points_Bisciaro_al . . . 64
anchor_points_grey . . . 64
anchor2time . . . 65
as_wr_track . . . 66
astro_anchor . . . 67
astro_anchor_app . . . 74
astro_anchor_interactive . . . 76
astro_ctest . . . 79
astro_from_peaks . . . 80
astro_from_periods . . . 81
astro_invert . . . 82
astro_mc . . . 83
astro_obliquity . . . 84
astro_periods_tracking . . . 84
astro_term_search . . . 85
astro_terms . . . 86
astrogeo22 . . . 86
astroMCBchron . . . 87
astrosignal_example . . . 91
bayes_changepoint . . . 92
bayes_changepoint_r2 . . . 94
bayes_spectrum . . . 96
Bisciaro_al_wt_track . . . 97
Bisciaro_ca_wt_track . . . 98
Bisciaro_Mg_wt_track . . . 98
Bisciaro_Mn_wt_track . . . 99
Bisciaro_sial_wt_track . . . 99

Bisciaro_XRF . . . 100
bispectral_analysis . . . 100
bispectral_surrogate . . . 104
bundling_ratios . . . 105
calculate_orbit9 . . . 106
chaos_scenario . . . 106
check_agemodel . . . 107
chirp_eha_significance . . . 110
coco_adaptive . . . 111
coco_adaptive_evaluate . . . 112
coco_classic . . . 113
coco_fixed_target_weights . . . 114
coco_prepare_regular_data . . . 114
coco_target_fixed . . . 115
coco_target_from_periods . . . 116
coco_target_from_series . . . 117
compare_published . . . 118
completed_series . . . 118
ctest . . . 120
ctest_combine . . . 121
ctest_earthmoon . . . 122
ctest_input . . . 123
curve2sedrate . . . 124
curve2time . . . 125
curve2time_unc . . . 127
curve2time_unc_anchor . . . 133
curve2tune . . . 143
cvcoco . . . 145
cycle_band . . . 147
cycles_to_k . . . 148
days_per_year . . . 149
delpts_tracked_period_wt . . . 149
depth_rank_example . . . 151
detect_transients . . . 151
dur_gaps . . . 153
dynamic_extraction . . . 155
dynamic_filter . . . 157
dynamic_filter_controls . . . 159
dynot . . . 161
earth_moment_of_inertia . . . 163
earth_omega . . . 164
earthmoon_angular_momentum . . . 164
earthmoon_from_precession . . . 165
earthmoon_state . . . 165
ecoco_adaptive . . . 167
ecoco_best_per_window . . . 168
ecoco_blocked . . . 168
ecoco_classic . . . 171

ecoco_interleaved . . . 172
ecoco_track . . . 174
ecoco_track_ridge . . . 177
eha_log2 . . . 177
em_state . . . 179
example_pridoli . . . 180
expSedRamp . . . 180
extract_amplitude . . . 181
extract_power . . . 183
extract_power_stable . . . 186
extract_signal . . . 187
extract_signal_stable . . . 190
extract_signal_stable_V2 . . . 191
extract_signal_standard_deviation . . . 193
fft_frequency_uncertainty . . . 197
fft_uncertainty . . . 198
flmw . . . 199
flmw_path . . . 203
gaussian_bandpass . . . 204
geo_col . . . 206
geo_loc . . . 210
geo_mid . . . 214
grey . . . 219
grey_track . . . 219
GTS_info . . . 220
k_from_bundling . . . 220
k_from_cycles . . . 221
k_from_period . . . 223
k_from_ratio . . . 223
k_hypotheses . . . 224
k_scan . . . 225
la2004_extrap . . . 226
lag_1 . . . 226
lithlog_disc . . . 228
lod_from_moon_distance . . . 229
lod_hours . . . 230
loess_auto . . . 230
mag . . . 232
mag_track_solution . . . 232
max_detect . . . 233
mesa_average . . . 234
mesa_evolutive . . . 235
mesa_spectrum . . . 237
min_detect . . . 239
minimal_tuning . . . 240
modBchron . . . 242
modBchron_predict . . . 246
model_red_noise_wt . . . 247

months_per_year . . . 248
moon_distance_from_lod . . . 249
moon_distance_from_months . . . 249
omega_from_moon_distance . . . 250
orbit9 . . . 250
orbit9_at_age . . . 251
orbital_periods . . . 251
pda . . . 252
pda_legacy . . . 253
percentile_from_red_noise . . . 255
plot.astroMCBchron . . . 256
plot.coco_adaptive . . . 257
plot.coco_classic . . . 257
plot.ctest . . . 258
plot.ecoco_adaptive . . . 258
plot.ecoco_classic . . . 259
plot_agemodel_check . . . 259
plot_amtm_spectrum . . . 260
plot_astro . . . 261
plot_astro_anchor . . . 262
plot_astro_evidence . . . 265
plot_avg_wavelet . . . 266
plot_Awavelet . . . 268
plot_bayes_changepoint . . . 273
plot_bispectral . . . 274
plot_chirp_eha . . . 276
plot_cvcoco . . . 277
plot_dechirped . . . 278
plot_dynamic_filter . . . 279
plot_dynot . . . 280
plot_ecoco_cv . . . 281
plot_eha_log2 . . . 282
plot_mesa_evolutive . . . 287
plot_mesa_spectrum . . . 289
plot_sed_model . . . 290
plot_spectral_moments . . . 292
plot_superlet . . . 294
plot_test_k . . . 298
plot_track . . . 299
plot_wavelet . . . 300
plot_wavelet_coherence . . . 306
plot_win_fft . . . 310
plot_win_timeOpt . . . 312
plot_Xsuperlet . . . 314
plot_Xwavelet . . . 318
precession_frequency . . . 322
predict.astroMCBchron . . . 323
print.astro_from_peaks . . . 324

print.ctest . . . 324
rate_gram . . . 325
resonance_state . . . 326
retrack_wt_MC . . . 326
rho1_evolutive . . . 335
sedrate2tune . . . 337
spectral_moments . . . 339
sst_significance . . . 342
sum_power_sedrate . . . 344
swa_background . . . 347
swa_spectrum . . . 348
term_sensitivity . . . 351
test_tuned . . . 352
test_tuned_check . . . 354
track_astro_auto . . . 355
track_astro_hmm . . . 357
track_astro_iterative . . . 362
track_chirp_hmm . . . 364
track_ecoco . . . 366
track_mesa . . . 368
track_period . . . 370
track_period_wavelet . . . 372
track_ridge_hmm . . . 375
TSI . . . 378
upb_grain_loglik . . . 378
wavelet_band_significance . . . 380
wavelet_significance . . . 382
wavelet_uncertainty . . . 384
WaverideR_Datasets . . . 388
win_fft . . . 390
win_timeOpt . . . 392
wr_coloured_noise . . . 395
zeeden2023_table1 . . . 396

Index **397**

`WaverideR-package` *Extracting Signals from Wavelet Spectra*

Description

WaverideR is an R package for advanced cyclostratigraphic analysis of stratigraphic data sets. It provides a comprehensive suite of spectral tools for detecting, visualising, and tracking non-stationary astronomical cycles, including continuous wavelet and superlet transform (CWT), the superlet transform, windowed FFT, and evolutionary harmonic analysis (EHA) (wrapper). These methods allow both manual and automated tracking of orbital cycles in spectra and scalograms, even

in records affected by large changes in sedimentation rate.Building on this spectral framework, WaverideR supports multi-proxy integration and Monte Carlo–based uncertainty propagation to construct statistically robust floating and absolute astrochronological age models. The package includes dedicated tools to quantify analytical wavelet uncertainty, estimate the duration of stratigraphic gaps and hiatuses, and integrate external radioisotopic age constraints. Designed for complex and incomplete stratigraphic records, WaverideR enables investigation of the imprint of astronomical forcing even in suboptimal datasets.

Details

Package: 'WaverideR'

Type: R package

Version: 0.6.0 (2026)

License: GPL (= 2)

Note

If you want to use this package for publication or research purposes, please cite both the paper that introduced the package and the package itself (a double citation, e.g. "using the WaverideR R package (Arts et al., 2024; Arts, 2023)"):

Arts, M., Corradini, C., Pondrelli, M., Pas, D., Da Silva, A.-C. (2024). Age and orbital forcing in the upper Silurian Cellon section (Carnic Alps, Austria) uncovered using the WaverideR R package. Frontiers in Earth Science 12, 1357751. doi:10.3389/feart.2024.1357751

Arts, M.C.M. (2023). WaverideR: Extracting Signals from Wavelet Spectra. https://CRAN.R-project.org/package=WaverideR

`citation("WaverideR")` gives both entries.

Author(s)

Michiel Arts

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

References

The 'WaverideR' package builds upon existing literature and existing codebase. The following list of articles is relevant for the 'WaverideR' R package and its functions. Individual articles are also cited in the descriptions of function when relative for set function. The articles in the list below can be grouped in four subjects: (1) Cyclostratigraphic data analysis, (2) example data sets, (3) the (continuous) wavelet transform and (4) astronomical solutions). For each of these categories the relevance of set articles will be explained in the framework of the 'WaverideR' R package.

1. Cyclostratigraphic data analysis

Meyers, S. R. (2019). Cyclostratigraphy and the problem of astrochronologic testing. Earth-Science Reviews, 190, 190–223. doi:10.1016/j.earscirev.2018.11.015

Meyers, S. R. (2012). Seeing red in cyclic stratigraphy: Spectral noise estimation for astrochronology. Paleoceanography, 27, PA3228. doi:10.1029/2012PA002307

Li, M., Hinnov, L. A., and Kump, L. R. (2019). Acycle: Time-series analysis software for paleoclimate research and education. Computers and Geosciences, 127, 12–22. doi:10.1016/j.cageo.2019.02.011

Li, M., Kump, L. R., Hinnov, L. A., and Mann, M. E. (2018). Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing. Earth and Planetary Science Letters, 501, 165–179. doi:10.1016/j.epsl.2018.08.041

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

Wouters, S., Da Silva, A.-C., Boulvain, F., and Devleeschouwer, X. (2021). StratigrapheR: Concepts for litholog generation in R. The R Journal, 13. doi:10.32614/RJ2021039

Huang, N. E., Wu, Z., Long, S. R., Arnold, K. C., Chen, X., and Blank, K. (2009).On instantaneous frequency. Advances in Adaptive Data Analysis, 1, 177–229. doi:10.1142/S1793536909000096

Cleveland, W. S. (1979). Robust locally weighted regression and smoothing scatterplots. Journal of the American Statistical Association, 74, 829–836. doi:10.1080/01621459.1979.10481038

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: Series B, 60, 271–293. doi:10.1111/14679868.00125

Golub, G. H., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. Technometrics, 21, 215–224. doi:10.2307/1268518

#2. Example data sets

Pas, D., Hinnov, L. A., Day, J. E., Kodama, K., Sinnesael, M., and Liu, W. (2018). Cyclostratigraphic calibration of the Famennian Stage (Late Devonian, Illinois Basin, USA). Earth and Planetary Science Letters, 488, 102–114. doi:10.1016/j.epsl.2018.02.010

Steinhilber, F., Abreu, J., Beer, J., Brunner, I., Christl, M., Fischer, H., Heikkilä, U., Kubik, P. W., Mann, M. E., McCracken, K. G., Miller, H., Miyahara, H., Oerter, H., and Wilhelms, F. (2012). 9400 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, 5967–5971. doi:10.1073/pnas.1118965109

Zeeden, C., Hilgen, F. J., Westerhold, T., Lourens, L. J., Röhl, U., and Bickert, T. (2013). Revised Miocene splice, astronomical tuning and calcareous plankton biochronology of ODP Site

926 between 5 and 14.4 Ma. Palaeogeography, Palaeoclimatology, Palaeoecology, 369, 430–451. doi:10.1016/j.palaeo.2012.11.009

#3. Continuous wavelet and superlet transform

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., and Mureșan, R. C. (2021). Time-frequency super-resolution with superlets. Nature Communications, 12, 337. doi:10.1038/s41467020205399

Morlet, J., Arens, G., Fourgeau, E., and Giard, D. (1982a). Wave propagation and sampling theory. Part I: Complex signal and scattering in multilayered media. Geophysics, 47, 203–221.

Morlet, J., Arens, G., Fourgeau, E., and Giard, D. (1982b). Wave propagation and sampling theory. Part II: Sampling theory and complex waves. Geophysics, 47, 222–236.

Torrence, C., and Compo, G. P. (1998). A practical guide to wavelet analysis. Bulletin of the American Meteorological Society, 79, 61–78.

Gouhier, T. C., Grinsted, A., and Simko, V. (2021). biwavelet: Conduct univariate and bivariate wavelet analyses. R package version 0.20.21.

Roesch, A., and Schmidbauer, H. (2018). WaveletComp: Computational wavelet analysis. R package version 1.1.

Gabor, D. (1946). Theory of communication. Part I: The analysis of information. Journal of the Institution of Electrical Engineers, Part III, 93, 429–441.

#4. Astronomical solutions

Laskar, J., Robutel, P., Joutel, F., Gastineau, M., Correia, A. C. M., and Levrard, B. (2004). A long-term numerical solution for the insolation quantities of the Earth. Astronomy and Astrophysics, 428, 261–285. doi:10.1051/00046361:20041335

Laskar, J., Fienga, A., Gastineau, M., and Manche, H. (2011a). La2010: A new orbital solution for the long-term motion of the Earth. Astronomy and Astrophysics, 532, A89. doi:10.1051/0004-6361/201116836

Zeebe, R. E., and Lourens, L. J. (2019). Solar System chaos and the Paleocene–Eocene boundary age constrained by geology and astronomy. Science, 365, 926–929. doi:10.1126/science.aax0612

See Also

Useful links:

- https://github.com/stratigraphy/WaverideR

add_wavelet *Add a wavelet plot*

Description

Generates a plot of a wavelet scalogram which can be integrated into a larger composite plot

Usage

```
add_wavelet(
  wavelet = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  lower_depth_time = NULL,
  upper_depth_time = NULL,
  n.levels = 100,
  plot.COI = TRUE,
  color_brewer = "grDevices",
  palette_name = "rainbow",
  plot_dir = FALSE,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  plot_horizontal = TRUE,
  period_ticks = 1,
  periodlab = "period (m)",
  main = NULL,
  yaxt = "s",
  xaxt = "s",
  depth_time_lab = "depth (m)",
  colour_brewer = NULL
)
```

Arguments

wavelet	wavelet object created using the analyse_wavelet function.
lowerPeriod	Lowest period value which will be plotted
upperPeriod	Highest period value which will be plotted
lower_depth_time	lowest depth/time value which will be plotted
upper_depth_time	Highest depth/time value which will be plotted
n.levels	Number of color levels Default=100.
plot.COI	Option to plot the cone of influence Default=TRUE.

color_brewer Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices

palette_name Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

plot_dir The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE plot_dir=TRUE

add_lines Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL

add_points Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values Default=NULL

add_abline_h Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL

add_abline_v Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL

plot_horizontal

plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE

period_ticks tick mark spacing 1 is all tickmarks and higher value removes tick marks by the fraction of the tick mark spacing value, the opposite is true for value lower than 1 which will add aditional tickmarks

periodlab lable for the the period column

main main title

yaxt turn on of off the yaxis "s" is on "n" is off Default="s"

xaxt turn on of off the xaxis "s" is on "n" is off Default="s"

depth_time_lab lable for the the depth/time column

colour_brewer British spelling of color_brewer; if given it overrides color_brewer.

Value

returns a plot of a wavelet scalogram

Author(s)

Code based on the "analyze.wavelet" and "wt.image" functions of the 'WaveletComp' R package and the "wt" function of the 'biwavelet' R package which are based on the wavelet MATLAB code written by Christopher Torrence and Gibert P. Compo (1998). The MTM analysis is from the astrochron R package of Meyers et al., (2012)

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#generate a plot for the magnetic susceptibility data set of Pas et al., (2018)

plot.new()
layout.matrix <- matrix(c(rep(0, 2), 1, 0,0,seq(2, 6, by = 1)),
                        nrow = 2,
                       ncol = 5 ,
                        byrow = TRUE)
graphics::layout(mat = layout.matrix,
                heights = c(0.25, 1),
                # Heights of the two rows
                widths = c(rep(c(1, 2, 4,2,2), 2)))

par(mar = c(0, 0.5, 1, 0.5))


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

```
  omega_nr = 10
 )

 add_wavelet_avg(
 wavelet = mag_wt,
 plot_horizontal = TRUE,
 add_abline_h = NULL,
 add_abline_v = NULL,
 lowerPeriod = 0.15,
 upperPeriod = 80
)

par(mar = c(4, 4, 0, 0.5))


plot(
 x = c(0, 1),
 y = c(max(mag[, 1]), min(mag[, 1])),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(max(mag[, 1]), min(mag[, 1])))
)            # Draw empty plot


polygon(
 x = c(0, 1, 1, 0),
 y = c(max(mag[, 1]), max(mag[, 1]), min(mag[, 1]), min(mag[, 1])),
 col = geo_col("Famennian")
)

text(
 0.5,
 (max(mag[, 1]) - min(mag[, 1])) / 2,
 "Fammenian",
 cex = 1,
 col = "black",
 srt = 90
)
par(mar = c(4, 0.5, 0, 0.5))


plot(
 mag[, 2],
 mag[, 1],
 type = "l",
 ylim = rev(c(max(mag[, 1]), min(mag[, 1]))),
 yaxs = "i",
 yaxt = "n",
 xlab = "Mag. suc.",
```

```
 ylab = ""
)


add_wavelet(
 wavelet = mag_wt,
 lowerPeriod = 0.15,
 upperPeriod = 80,
 lower_depth_time = NULL,
 upper_depth_time = NULL,
 n.levels = 100,
 plot.COI = TRUE,
 color_brewer = "grDevices",
 palette_name = "rainbow",
 plot_dir = FALSE,
 add_lines = NULL,
 add_points = NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 plot_horizontal = FALSE,
 period_ticks = 1,
 periodlab = "period (m)",
 main = NULL,
 yaxt = "n",
 xaxt = "s",
 depth_time_lab = ""
)


lines(log2(mag_track_solution[,2]),mag_track_solution[,1],lwd=4,lty=4)

mag_405 <- extract_signal(
 tracked_cycle_curve = mag_track_solution,
 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
)

plot(mag_405[,2],mag_405[,1],type="l",
    yaxt="n", yaxs = "i",
    xlab="405-kyr ecc")


mag_110 <- extract_signal(
 tracked_cycle_curve = mag_track_solution,
 wavelet = mag_wt,
 period_up = 1.25,
 period_down = 0.75,
```

```
 add_mean = TRUE,
 tracked_cycle_period = 405,
 extract_cycle = 110,
 tune = FALSE,
 plot_residual = FALSE
)

# amplitude of the ~110-kyr cycle from the wavelet: the tracked 405-kyr curve scaled to 110 kyr
track_110 <- cbind(mag_track_solution[, 1], mag_track_solution[, 2] * 110 / 405)
mag_110_amp <- extract_amplitude(mag_110, cycle = track_110, band = c(0.75, 1.33))
mag_110_mean <- mean(mag_110[, 2])

plot(mag_110[,2],mag_110[,1],type="l",
    yaxt="n", yaxs = "i",
    xlab="110-kyr ecc")

lines(mag_110_mean + mag_110_amp[, 2], mag_110_amp[, 1], col = "red")
lines(mag_110_mean - mag_110_amp[, 2], mag_110_amp[, 1], col = "red")
```

`add_wavelet_avg` *Add a plot of a the average spectral power of a continous wavelet transform*

Description

Generates a plot of a the average spectral power of a continous wavelet transform which can be added to a larger composite plot

Usage

```
add_wavelet_avg(
  wavelet = NULL,
  plot_horizontal = TRUE,
  add_abline_h = NULL,
  add_abline_v = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL
)
```

Arguments

`wavelet`	wavelet object created using the `analyse_wavelet` function.
`plot_horizontal`	plot the wavelet horizontal or vertical eg y axis is depth or y axis power `Default=TRUE`
`add_abline_h`	Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) `Default=NULL`

add_abline_v	Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
lowerPeriod	Lowest period value which will be plotted
upperPeriod	Highest period value which will be plotted

Value

returns a plot of a the average spectral power of a continuous wavelet transform

Author(s)

Code based on the "analyze.wavelet" and "wt.image" functions of the 'WaveletComp' R package and "wt" function of the 'biwavelet' R package which are based on the wavelet MATLAB code written by Christopher Torrence and Gibert P. Compo (1998). The MTM analysis is from the astrochron R package of Meyers et al., (2012)

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#generate a plot for the magnetic susceptibility data set of Pas et al., (2018)

plot.new()
layout.matrix <- matrix(c(rep(0, 2), 1, 0,0,seq(2, 6, by = 1)),
                        nrow = 2,
                       ncol = 5 ,
                        byrow = TRUE)
graphics::layout(mat = layout.matrix,
                heights = c(0.25, 1),
                # Heights of the two rows
                widths = c(rep(c(1, 2, 4,2,2), 2)))

par(mar = c(0, 0.5, 1, 0.5))


mag_wt <-
```

```
 analyse_wavelet(
   data = mag,
   dj = 1 / 100,
  lowerPeriod = 0.1,
   upperPeriod = 254,
   verbose = FALSE,
   omega_nr = 10
 )

add_wavelet_avg(
 wavelet = mag_wt,
 plot_horizontal = TRUE,
 add_abline_h = NULL,
 add_abline_v = NULL,
 lowerPeriod = 0.15,
 upperPeriod = 80
)


par(mar = c(4, 4, 0, 0.5))


plot(
 x = c(0, 1),
 y = c(max(mag[, 1]), min(mag[, 1])),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(max(mag[, 1]), min(mag[, 1])))
)            # Draw empty plot


polygon(
 x = c(0, 1, 1, 0),
 y = c(max(mag[, 1]), max(mag[, 1]), min(mag[, 1]), min(mag[, 1])),
 col = geo_col("Famennian")
)

text(
 0.5,
 (max(mag[, 1]) - min(mag[, 1])) / 2,
 "Fammenian",
 cex = 1,
 col = "black",
 srt = 90
)
par(mar = c(4, 0.5, 0, 0.5))


plot(
```

```
 mag[, 2],
 mag[, 1],
 type = "l",
 ylim = rev(c(max(mag[, 1]), min(mag[, 1]))),
 yaxs = "i",
 yaxt = "n",
 xlab = "Mag. suc.",
 ylab = ""
)

add_wavelet(
 wavelet = mag_wt,
 lowerPeriod = 0.15,
 upperPeriod = 80,
 lower_depth_time = NULL,
 upper_depth_time = NULL,
 n.levels = 100,
 plot.COI = TRUE,
 color_brewer = "grDevices",
 palette_name = "rainbow",
 plot_dir = FALSE,
 add_lines = NULL,
 add_points = NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 plot_horizontal = FALSE,
 period_ticks = 1,
 periodlab = "period (m)",
 main = NULL,
 yaxt = "n",
 xaxt = "s",
 depth_time_lab = ""
)

lines(log2(mag_track_solution[,2]),mag_track_solution[,1],lwd=4,lty=4)

mag_405 <- extract_signal(
 tracked_cycle_curve = mag_track_solution,
 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
)

plot(mag_405[,2],mag_405[,1],type="l",
    yaxt="n", yaxs = "i",
    xlab="405-kyr ecc")
```

```
mag_110 <- extract_signal(
 tracked_cycle_curve = mag_track_solution,
 wavelet = mag_wt,
 period_up = 1.25,
 period_down = 0.75,
 add_mean = TRUE,
 tracked_cycle_period = 405,
 extract_cycle = 110,
 tune = FALSE,
 plot_residual = FALSE
)

# amplitude of the ~110-kyr cycle from the wavelet: the tracked 405-kyr curve scaled to 110 kyr
track_110 <- cbind(mag_track_solution[, 1], mag_track_solution[, 2] * 110 / 405)
mag_110_amp <- extract_amplitude(mag_110, cycle = track_110, band = c(0.75, 1.33))
mag_110_mean <- mean(mag_110[, 2])

plot(mag_110[,2],mag_110[,1],type="l",
    yaxt="n", yaxs = "i",
    xlab="110-kyr ecc")

lines(mag_110_mean + mag_110_amp[, 2], mag_110_amp[, 1], col = "red")
lines(mag_110_mean - mag_110_amp[, 2], mag_110_amp[, 1], col = "red")
```

---

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>

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. <doi:10.1051/0004-6361:20041335>

`amtm_prewhiten` *Autoregressive prewhitening of a series for aMTM analysis*

Description

The `amtm_prewhiten` function prewhitens an evenly spaced series with an AR(p) filter whose coefficients are obtained from the autocovariance sequence of a low-variance aMTM spectrum (NW = 4, K = 7, zero padding to `2^(ceiling(log2(N)) + nfft_add)`) through the Yule-Walker equations. The one-step predictions minus the observations form the prewhitened series; the first `2*order` values are dropped and the mean is removed. The spectrum of the prewhitened series can be "recoloured" by dividing it by $|1 - \sum_j \phi_j e^{-i2\pi f j \Delta t}|^2$. Translated from the "Python PreWhiten Filter Example" (prewhiten(), correct_spec_for_pw()) distributed with amtm_signaldetect (MIT licence), itself based on an R prewhiten() routine (Wesley Burr / David Riegert).

Usage

```
amtm_prewhiten(data = NULL, order = 2, nw = 4, k = 7, nfft_add = 2)
```

Arguments

`data`	Input data set: 2-column matrix [depth/time, value], evenly spaced
`order`	Order of the AR filter `Default=2`
`nw`	Time-bandwidth product of the spectrum used for the ACVS `Default=4`
`k`	Number of tapers of that spectrum `Default=7`
`nfft_add`	Zero-padding exponent increment `Default=2`

Value

A list with `data` (2-column matrix of the prewhitened series, with the depth/time values of the retained samples), `ar_coef` (AR coefficients phi_1..phi_p) and `acvs` (autocovariance sequence used).

Author(s)

Michiel Arts (R translation); original Python code: Hector A. Salinas et al.

References

Percival, D. B., & Walden, A. T. (2020). Spectral Analysis for Univariate Time Series. Cambridge University Press. doi:10.1017/9781139235723

Examples

```
set.seed(3)
x <- as.numeric(stats::filter(rnorm(400), 0.9, method = "recursive"))
pw <- amtm_prewhiten(cbind(seq_along(x), x), order = 1)
pw$ar_coef
```

`amtm_spectrum` *Adaptive multitaper spectrum with background fit and periodic signal detection*

Description

The `amtm_spectrum` function computes the adaptive multitaper (aMTM) power spectral density (PSD) of an evenly spaced series, fits the PSD background by maximum likelihood with several noise models, selects the best model and detects periodic signals with the dual gamma-test / harmonic F-test criterion of Di Matteo et al. (2021) and Salinas et al. (2025).

Method: (1) Thomson's adaptively weighted multitaper PSD (Prieto's 'multitaper' implementation: unit-energy DPSS tapers, adaptive weights iterated until the relative change is below 9.5e-7, PSD rescaled so that its integral equals the variance, one-sided by doubling all bins except 0 and Nyquist). The half degrees of freedom are $\alpha_k = (\sum_j w_{jk}^2)^2 / \sum_j w_{jk}^4$. (2) Background models, fitted on the frequency range [2NW/T, f_Nyq - 2NW/T] (or a user range) by minimising the aMTM log-likelihood $M = 2\sum_k [\alpha_k S_k/B_k + \ln(\Gamma(\alpha_k)S_k) - \alpha_k \ln(\alpha_k S_k/B_k)]$: white noise WHT $B = c$ (analytical), power law PL $B = cf^{-\beta}$, bending power law BPL $B = cf^{-\beta}/(1 + (f/f_b)^{\gamma-\beta})$, and first-order autoregressive AR1 $B = c/(1 - 2\rho\cos(\pi f/f_{Nyq}) + \rho^2)$. (3) Model selection by the MERIT function (default in SPD_MTM) or AIC ($M + 2n_{free}$, with 0/1/1/3 free parameters besides c for WHT/PL/AR1/BPL). Adjacent aMTM frequencies are correlated over the 2W band, so these criteria tend to favour flexible models (BPL is sometimes chosen for pure white noise): inspect `model_stats` and use `bkg_model` to force a model when needed. The Python code minimises M with scipy SLSQP (or Powell) on the raw parameters; here `stats::nlminb` is used on (log c, beta[, gamma, log fb]) with the same bounds and starting values. Both reach the same optimum for PL (parameters within ~1e-5 relative); for BPL R reaches the same or a lower (better) M, because SLSQP can stop in a local minimum. (4) Gamma-test $\gamma_k = S_k/B_k$, whose threshold is taken from the mixture-of-gamma CDF built from the histogram (bin width 0.1) of α_k; harmonic F-test with threshold `qf(conf, 2, 2K-2)`. A frequency is a detected periodic signal when an F-test peak above its threshold falls inside a band where γ_k exceeds its threshold and that band is at least $NW/(2T)$ wide.

Translated from the Python package amtm_signaldetect v1.0.6 (Hector A. Salinas; functions get_amtm_specs, get_spectra_dofA, get_background_psdfit, get_gamtest_confs, get_ftest_confs, get_gftest_confpeaks) and from Prieto's Python 'multitaper' v1.2.0 (MTSpec, adaptspec, ftest), both MIT licensed. The AR1 model and the AIC/MERIT criteria follow the IDL code SPD_MTM (Di Matteo et al., 2021). This is not an Acycle function.

Cyclostratigraphic interpretation: for a depth series the frequency axis is in cycles per depth unit (e.g. cycles/m if depth is in m); the period (thickness) of a detected cycle is 1/frequency. The fitted background is the null hypothesis of "no cycle": use PL/BPL/AR1 for the red spectra typical of sedimentary proxies, and treat the gamma-test as the test for excess power above the red background and the F-test as the test for a phase-coherent sinusoid. Because sedimentation-rate changes smear orbital lines, a real Milankovitch signal in depth often passes the gamma-test over a band but gives only a moderate F-test; inspect both, use `freq_range` to exclude trends/very high frequencies, and compare the ratios of detected periods with expected orbital ratios (e.g. 405:100:~20 kyr) rather than relying on single detections. The frequency resolution is $2NW/T$ with T the series length in depth units.

Usage

```
amtm_spectrum(
  data = NULL,
  nw = 3,
  k = NULL,
  nfft = NULL,
  models = c("WHT", "PL", "AR1", "BPL"),
  selection = "MERIT",
  bkg_model = NULL,
  freq_range = NULL,
  conf_level = 0.99,
  prewhiten = FALSE,
  pw_order = 2,
  genplot = FALSE,
  verbose = FALSE
)
```

Arguments

data	Input data set: a matrix or data frame with 2 columns, the first being depth or time (evenly spaced, increasing), the second the proxy values.
nw	Time-bandwidth product NW Default=3
k	Number of tapers Default=NULL (uses round(2*nw-1), as Python)
nfft	Number of FFT points Default=NULL (uses the series length N, the Python default; values smaller than N are replaced by 2N+1, as in 'multitaper')
models	Background models to fit, any of "WHT","PL","AR1","BPL" Default=c("WHT","PL","AR1","BPL")
selection	Model-selection criterion, "MERIT" or "AIC" Default="MERIT"
bkg_model	Force this background model for the detection instead of the automatically selected one Default=NULL
freq_range	Frequency range c(f_low, f_high) (cycles per unit of the first column) for the background fit and the detection Default=NULL (uses [2NW/T, f_Nyq - 2NW/T])
conf_level	Confidence level used for the detection Default=0.99
prewhiten	Prewhiten the series with an AR model before the analysis (see amtm_prewhiten) Default=FALSE
pw_order	Order of the AR prewhitening filter Default=2
genplot	Plot the results with plot_amtm_spectrum Default=FALSE
verbose	Print progress and fit information Default=FALSE

Value

A list with elements

- freq, psd, alpha, ftest: aMTM frequency (cycles per unit), one-sided PSD, half degrees of freedom and harmonic F-test over [0, f_Nyq].

- `fit_index`: indices of `freq` used for the fit/detection; `fit_freq`: those frequencies.
- `background`: matrix (fit range x models) of fitted backgrounds.
- `params`: list of fitted parameters per model (WHT: c; PL: c, beta; AR1: c, rho; BPL: c, beta, gamma, fb (after the beta > gamma correction)).
- `model_stats`: data frame with log-likelihood M, AIC, MERIT and convergence code per model; `selected_model`; `background_selected`.
- `gamma`: gamma-statistic S/B of the selected model (fit range).
- `thresholds`: data frame with confidence levels (0.5, 0.9, 0.95, 0.99 and `conf_level`) and the gamma-test and F-test thresholds.
- `gamma_threshold`, `f_threshold`: thresholds at `conf_level`.
- `f_peaks`, `gamma_peaks`: indices (within the fit range) of F-test and gamma-test peaks above threshold; `gamma_bands`: data frame of gamma bands (f_start, f_end) wider than NW/(2T).
- `detected`: data frame of periodic signals passing both tests (frequency, period, ftest, gamma, psd, background).
- `prewhiten`: NULL or list(ar_coef, psd_recoloured).
- `settings`: list of the settings used (nw, k, nfft, dt, n, conf_level).

Author(s)

Michiel Arts (R translation); original Python code: Hector A. Salinas (amtm_signaldetect) and German A. Prieto (multitaper); original method and IDL code SPD_MTM: Simone Di Matteo

References

Di Matteo, S., Viall, N. M., & Kepko, L. (2021). Power spectral density background estimate and signal detection via the multitaper method. Journal of Geophysical Research: Space Physics, 126, e2020JA028748. doi:10.1029/2020JA028748

Salinas, H. A., et al. (2025). Journal of Geophysical Research: Space Physics. doi:10.1029/2025JA034006 (open-source Python aMTM signal detection)

Prieto, G. A. (2022). The multitaper spectrum analysis package in Python. Seismological Research Letters, 93(3), 1922-1929. doi:10.1785/0220210332

Thomson, D. J. (1982). Spectrum estimation and harmonic analysis. Proceedings of the IEEE, 70(9), 1055-1096. doi:10.1109/PROC.1982.12433

Examples

```
# synthetic depth series: AR(1) red noise + 2 cycles (periods 5 m and 1.25 m)
set.seed(1)
n <- 600
depth <- seq(0, by = 0.1, length.out = n)
red <- as.numeric(stats::filter(rnorm(n), 0.8, method = "recursive"))
x <- red + 1.2 * sin(2 * pi * depth / 5) + 0.8 * sin(2 * pi * depth / 1.25)
res <- amtm_spectrum(data = cbind(depth, x), nw = 3, genplot = FALSE)
res$selected_model
res$detected           # frequency in cycles/m, period in m
plot_amtm_spectrum(res)
```

`analyse_Awavelet` *Conduct the adaptive continuous wavelet transform on a time series or signal*

Description

Compute the continuous wavelet transform (CWT) using a complex Morlet wavelet with scale dependent wavelet width. The number of oscillatory cycles in the Morlet wavelet is allowed to vary smoothly with scale, enabling adaptive time frequency resolution similar in spirit to superlet based approaches.

Usage

```
analyse_Awavelet(
  data = NULL,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = FALSE,
  omega_min = 6,
  omega_max = 12,
  scaling = c("none", "log2", "linear", "sqrt", "quadratic", "power"),
  alpha = 1,
  n_simulations = 10,
  run_multicore = FALSE,
  wavelet = "morlet",
  wavelet_order = NULL
)

analyze_Awavelet(
  data = NULL,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = FALSE,
  omega_min = 6,
  omega_max = 12,
  scaling = c("none", "log2", "linear", "sqrt", "quadratic", "power"),
  alpha = 1,
  n_simulations = 10,
  run_multicore = FALSE,
  wavelet = "morlet",
  wavelet_order = NULL
)
```

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 the proxy record.
dj	Spacing between successive scales. Scales increase by powers of two as 2^(j * dj). Smaller values of dj increase frequency resolution at the expense of computational time. Default is 1/100.
lowerPeriod	Lowest period to be analysed. Defines the smallest scale of the transform.
upperPeriod	Highest period to be analysed. Defines the largest scale of the transform.
verbose	Logical. If TRUE, print interpolation diagnostics.
omega_min	Minimum number of oscillatory cycles of the Morlet wavelet.
omega_max	Maximum number of oscillatory cycles of the Morlet wavelet.
scaling	Character string defining how the number of wavelet cycles varies with scale. One of "log2", "linear", "sqrt", "quadratic", or "power".
alpha	Numeric. Exponent used when scaling = "power". Values greater than one emphasize high frequency sharpening, whereas values smaller than one emphasize low frequency sharpening.
n_simulations	Number of Monte Carlo simulations. Currently included for interface compatibility.
run_multicore	Logical. Currently included for interface compatibility.
wavelet	Mother wavelet, given as a name (PyWavelets style names with the order appended, such as "db4", "sym8", "cgau3", "gaus2" or "cmor1.5-1", also work) or as a function psi_hat(omega) that returns the Fourier transform of a user defined mother wavelet. Analytic (complex) wavelets, which give phase and amplitude and can be synchrosqueezed: "morlet" (default; aliases "gabor", "amor"), "morlet0" (exactly zero mean Morlet, admissible for a small omega_nr; alias "morlex"), "paul" (alias "cauchy"), "morse" (generalised Morse wavelet of Olhede and Walden, 2002; aliases "gmw", "airy"), "bump", "cmor" (complex Morlet with bandwidth and centre frequency as in Matlab/PyWavelets), "cgau" (analytic Gaussian derivative; "cmhat" is the complex Mexican hat), "shannon" (complex Shannon, box spectrum), "fbsp" (frequency B-spline), "meyer", "loggabor" (alias "lognormal"), "gammatone" (asymmetric in time: sharp onset, slow decay) and "hann" (raised cosine band). Real wavelets, which give a real transform (no phase or envelope): "dog" (derivative of Gaussian; "mexh", "ricker", "marr" and "sombrero" are the Mexican hat, order 2), "morl" (real Morlet), "hhat" (Hermitian hat), "poisson", and the orthogonal wavelets "haar", "db" (Daubechies), "sym" (Symlets) and "battle" (Battle-Lemarie spline wavelets), which are computed in the frequency domain from their filters. The period axis of the orthogonal wavelets is approximate because their spectra are broad and not symmetric. A narrow wavelet (Paul, low-order DOG, Shannon, Haar) resolves position well, a wide one (Morse with a large beta, Morlet with a high omega_nr, bump) resolves period well.
wavelet_order	Parameter(s) of the wavelet (see analyse_wavelet); when it is NULL the wavelet whose relative bandwidth equals that of a Morlet wavelet with omega_min to omega_max cycles is used at every scale. Parameters of the wavelet: paul: order

m (default 4); dog: derivative m (default 2); morse: c(gamma, beta) or beta alone with gamma = 3 (default c(3, 20)); bump: c(mu, sigma) (default c(5, 0.6)); cmor: c(bandwidth, centre frequency) (default c(1.5, 1)); cgau: derivative m (default 4); shannon: c(bandwidth, centre frequency) (default c(2, 1.5)); fbsp: c(m, bandwidth, centre frequency) (default c(2, 2, 1.5)); loggabor: c(centre angular frequency, log-width) (default c(6, 0.25)); meyer and hhat have no parameter; morlet0: omega_0 (default omega_nr); morl: omega_0 (default 5); poisson: order n (default 2); gammatone: c(n, centre angular frequency, bandwidth) (default c(4, 6, 1.5)); hann: c(centre angular frequency, half width) (default c(6, 3)); db: number of vanishing moments 1 to 20 (default 4, "haar" = "db1"); sym: 2 to 20 (default 8); battle: spline order 1, 3, 5 or 7 (default 3). For the Morlet wavelet use omega_nr.

Value

A list with class "analyze.Awavelet" containing:

- Wave: complex wavelet coefficients
- Phase: instantaneous phase
- Ampl: wavelet amplitude
- Power: wavelet power spectrum
- dt: sampling interval
- dj: scale spacing
- Power.avg: average power per period
- Period: physical periods
- Scale: wavelet scales
- coi.1: cone of influence x coordinates
- coi.2: cone of influence y coordinates
- nc: number of samples
- nr: number of scales
- axis.1: x axis values
- axis.2: log2 scaled periods
- omega_min: minimum wavelet cycles
- omega_max: maximum wavelet cycles
- omega: scale dependent wavelet cycles
- scaling: scaling method used
- alpha: power law exponent
- x: interpolated x values
- y: interpolated signal values

Author(s)

Adapted from the WaveletComp and biwavelet R packages, which are based on the MATLAB wavelet code by Torrence and Compo, with extensions for scale dependent wavelet width.

References

Torrence, C., and G. P. Compo (1998). A practical guide to wavelet analysis. Bulletin of the American Meteorological Society, 79, 61–78.

Morlet, J., G. Arens, E. Fourgeau, and D. Giard (1982). Wave propagation and sampling theory. Geophysics, 47, 203–236.

Examples

```
#Example 1. Using the Total Solar Irradiance data set of Steinhilber et al., (2012)
TSI_wt <-
 analyse_Awavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
omega_min = 6,
omega_max = 12,
scaling = "log2",
alpha = 1
 )


#Example 2. Using the magnetic susceptibility data set of Pas et al., (2018)
mag_wt <-
analyse_Awavelet(
data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_min = 6,
omega_max = 12,
scaling = "log2",
alpha = 1
)

#Example 3. Using the greyscale data set of Zeeden et al., (2013)
grey_wt <-
 analyse_Awavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
omega_min = 6,
omega_max = 12,
scaling = "log2",
alpha = 1,
 )
```

`analyse_chirp_eha` *Chirp-matched evolutive harmonic analysis (de-chirped windowed multitaper spectra)*

Description

Evolutive harmonic analysis (sliding-window multitaper spectra with the harmonic F-test, Thomson 1982; Meyers 2012) in which, in every window, the depth axis is first warped by a linear change of the sedimentation rate so that cycles whose period changes along the window become lines again. Within a window of length `window` the rate is modelled as r(z) = r0 (1 + g (z - z0)); the warped coordinate u = ln(1 + g (z - z0)) / g makes a cycle of fixed duration stationary in u, with the period it has at the window centre z0. For every window a grid of relative rate gradients `chirp` (g, per depth unit) is tried, and the gradient that concentrates the spectrum most (lowest Renyi entropy of the whitened power in the analysed period band, or the largest total F-test) is kept. The result is a period-depth map in which each cycle appears as a sharp line that moves with the sedimentation rate, plus the gradient itself (`chirp`), an independent estimate of where and how fast the rate changes.

Usage

```
analyse_chirp_eha(
  data,
  window = NULL,
  step = NULL,
  chirp = seq(-0.05, 0.05, by = 0.005),
  nw = 3,
  k = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  dj = 1/50,
  band = NULL,
  criterion = c("ftest", "template", "peak", "entropy"),
  astro_periods = NULL,
  sedrate = NULL,
  smooth_chirp = NULL,
  steer = NULL,
  pad = 8,
  detrend = TRUE,
  continuum = c("lowess", "median"),
  verbose = FALSE
)

analyze_chirp_eha(
  data,
  window = NULL,
  step = NULL,
```

```
  chirp = seq(-0.05, 0.05, by = 0.005),
  nw = 3,
  k = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  dj = 1/50,
  band = NULL,
  criterion = c("ftest", "template", "peak", "entropy"),
  astro_periods = NULL,
  sedrate = NULL,
  smooth_chirp = NULL,
  steer = NULL,
  pad = 8,
  detrend = TRUE,
  continuum = c("lowess", "median"),
  verbose = FALSE
)
```

Arguments

`data`	Record: first column depth, second column proxy (uniformly sampled; it is interpolated otherwise).
`window`	Window length (depth units), or several lengths, whose template scores are then averaged (short windows resolve abrupt changes, long ones resolve the lines). `NULL` with `criterion = "template"`: windows of about 12 and 25 cycles of the shortest cycle at the geometric mean of `sedrate`. A window should hold 15-25 cycles of the cycles that carry the evidence and be shorter than the scale over which the rate changes.
`step`	Step between window centres `Default=window / 20`.
`chirp`	Relative rate gradients tried (per depth unit): g = (dr/dz) / r at the window centre `Default=seq(-0.05, 0.05, by = 0.005)`; `"auto"` (with the template): 25 values up to a factor 3 change of the rate within the longest window.
`nw`	Multitaper time-bandwidth product `Default=3`; `k` tapers `Default=2 nw - 1`.
`k`	Number of Slepian tapers; by default `2 * nw - 1`.
`lowerPeriod, upperPeriod, dj`	Log2 period grid of the output.
`band`	Period range (depth units) within which the concentration criterion is evaluated `Default=c(lowerPeriod, min(upperPeriod, window / 3))`.
`criterion`	How the gradient of a window is chosen: `"ftest"` (default; sum of the harmonic F-test values above 2 in `band`, i.e. the strength of the lines), `"template"` (the match with the astronomical template, see `astro_periods`; pools all cycles and is the most robust in noisy records), `"peak"` (the highest whitened peak) or `"entropy"` (Renyi entropy of order 2 of the power in `band`; biased towards strong warps in noisy records).
`astro_periods, sedrate`	For `criterion = "template"`: the astronomical periods (vector, or data frame with `period` and `weight`) and the sedimentation-rate range searched; the de-

chirped spectrum of every window is then matched to the astronomical template (summed log line-to-continuum ratio at the periods of all cycles, over a grid of rates, as in eCOCO), the gradient with the best match is chosen, and the best rate per window is returned as `sedrate` with the score matrix `template_score` (windows x rates).

`smooth_chirp` The criterion is averaged over this length (depth units) along the record before the gradient is chosen, because the gradient changes slowly and the choice from one noisy window is unreliable `Default=window / 2`.

`steer` Optional tracked sedimentation rate (a `track_astro_hmm` or `track_astro_auto` result): every window is then de-chirped with the gradient of the track (nearest value of `chirp`) instead of the gradient chosen by `criterion`. Where the track is right, all cycles become sharp lines that follow the changes of the rate; where it is wrong, they do not, which makes the map a check of the track that is independent of the power the track was fitted to.

`pad` Zero-padding factor of the window spectra `Default=8`.

`detrend` Remove a linear trend from every window `Default=TRUE`.

`continuum` How the continuum against which a line is measured is estimated from the spectrum of a window: `"lowess"` (default; a smooth fit of log power against log period over the whole window spectrum, which is not pulled up by clusters of lines such as the precession triplet) or `"median"` (running median over 0.5 octave; it was pulled up inside line clusters, so that isolated aliases scored better than the true rate).

`verbose` Print the progress `Default=FALSE`.

Details

The ordinary EHA smears a cycle whose period changes by more than the spectral resolution within the window (with a 20 m window and a rate oscillating by a factor two over 54 m, the precession period changes by up to 60 percent inside a window), the wavelet transform smears it over the wavelet length, and the synchrosqueezed transforms only sharpen what the wavelet already resolves. The de-chirping here is the windowed-spectrum counterpart of the matched chirped wavelets of `analyse_chirplet_sst`; with the multitaper estimate it is far more robust to red noise than the squeezed maps.

Value

A list of class `analyze.wavelet` (usable in `plot_wavelet`, `track_astro_hmm`, `plot_astro_evidence`) with `Power` (periods x windows, de-chirped multitaper power), `Power_norm` (the power divided by the running median of its own window along the period axis, i.e. lines against their local continuum: the map to look at), `Ftest`, `Period`, `axis.1` (window centres), `axis.2` (log2 periods), `chirp` (the chosen gradient per window), `chirp_grid`, `criterion` (windows x gradients), the power without de-chirping `Power0` and the record.

References

Thomson, D.J. (1982). Spectrum estimation and harmonic analysis. Proc. IEEE 70, 1055-1096.

Meyers, S.R. (2012). Seeing red in cyclic stratigraphy: spectral noise estimation for astrochronology. Paleoceanography 27, PA3228.

Sinnesael, M. et al. (2019). The Cyclostratigraphy Intercomparison Project (CIP): consistency, merits and pitfalls. Earth-Science Reviews 199, 102965.

Examples

```
set.seed(1)
t <- seq(0, 3000, by = 1)
x_t <- sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 19) + 0.8 * sin(2 * pi * t / 41)
sr <- 2 + 0.7 * sin(2 * pi * t / 600)
zt <- c(0, cumsum(sr[-1] * diff(t)))
z <- seq(0, max(zt), by = 1)
x <- approx(zt, x_t, z)$y + rnorm(length(z), sd = 1.5)
ce <- analyse_chirp_eha(data.frame(z, x), window = 300, lowerPeriod = 10, upperPeriod = 300)
plot_wavelet(ce, whiten = TRUE)
```

---

analyse_chirp_wavelet *Chirp-matched wavelet scalogram with the astronomical template over rate and rate gradient*

Description

The wavelet counterpart of analyse_chirp_eha. At every scale the record is transformed with a family of chirped Morlet wavelets whose frequency sweep corresponds to a relative gradient of the sedimentation rate g = d ln(rate) / d depth (a cycle of fixed duration has d ln(period) / d depth = g, i.e. the chirp of the matched wavelet at scale s is c = -g s omega_0). The astronomical template is then scored, for every position, rate and gradient, on the scalogram of the wavelets matched to that gradient: the score is the weighted mean log ratio of the power at the depth periods of the cycles to the local continuum (running median along the period axis). track_chirp_hmm tracks the rate and the gradient jointly through this score. With `steer`, the scalogram of the gradient of a track is returned (every cell with the wavelet matched to the local gradient), as power and, with `squeeze = TRUE`, synchrosqueezed with the matched wavelets' instantaneous period.

Usage

```
analyse_chirp_wavelet(
  data,
  lowerPeriod,
  upperPeriod,
  dj = 1/50,
  omega_nr = 10,
  chirp = seq(-0.05, 0.05, by = 0.005),
  astro_periods,
  sedrate,
  step = NULL,
  smooth_depth = NULL,
```

```
  local_octaves = 0.75,
  chirp_max = 1.5,
  steer = NULL,
  squeeze = FALSE,
  continuum = c("lowess", "median"),
  verbose = FALSE
)

analyze_chirp_wavelet(
  data,
  lowerPeriod,
  upperPeriod,
  dj = 1/50,
  omega_nr = 10,
  chirp = seq(-0.05, 0.05, by = 0.005),
  astro_periods,
  sedrate,
  step = NULL,
  smooth_depth = NULL,
  local_octaves = 0.75,
  chirp_max = 1.5,
  steer = NULL,
  squeeze = FALSE,
  continuum = c("lowess", "median"),
  verbose = FALSE
)
```

Arguments

data	Record: first column depth, second column proxy.
lowerPeriod, upperPeriod, dj	Period grid.
omega_nr	Morlet omega_0 Default=10 (long wavelets: the de-chirping pays off most for them).
chirp	Relative rate gradients g (per depth unit) Default=seq(-0.05, 0.05, by = 0.005).
astro_periods	Astronomical periods (vector, or data frame with period and weight).
sedrate	Rate range searched c(lower, upper) (depth units per kyr).
step	Spacing of the positions at which the template is scored Default=5 dz.
smooth_depth	Smoothing of the power along the record before the template is scored (depth units) Default=2 x shortest period x geometric mean rate.
local_octaves	Half width (octaves) of the running median that defines the local continuum Default=0.75.
chirp_max	Largest dimensionless chirp of a wavelet (|g| x scale x omega_0) Default=1.5: a wavelet that would have to sweep by more than this within its own length is not matched (the plain wavelet is used), because the linear-chirp model does not hold over a wavelet longer than the scale of the rate changes (the 405-kyr wavelet spans a whole 1.2-Myr oscillation in CIP Case 3).

steer	Optional track (track_chirp_hmm, track_astro_hmm): the scalogram of the wavelets matched to its gradient is returned in Power.
squeeze	With steer: also synchrosqueeze the steered scalogram Default=FALSE.
continuum	Continuum estimate of a position's spectrum, see analyse_chirp_eha.
verbose	Print the progress.

Details

Compared with analyse_chirplet_sst, which picks the chirp of the strongest coefficient cell by cell (and therefore follows noise in a noisy record), the chirp here is a property of the sedimentation rate shared by all cycles at a position, so it is decided by all cycles together and by the continuity of the track.

Value

A list of class analyze.wavelet with Power (the steered scalogram if steer is given, otherwise the g = 0 Morlet scalogram), Power_sst (if squeezed), Period, axis.1, axis.2, and for track_chirp_hmm: template_score_all (rates x gradients x positions), sedrate_grid, chirp_grid, positions, dt (the position step).

Examples

```
set.seed(1)
t <- seq(0, 3000, by = 1)
x_t <- sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 19) + 0.8 * sin(2 * pi * t / 41)
sr <- 2 + 0.7 * sin(2 * pi * t / 600)
zt <- c(0, cumsum(sr[-1] * diff(t)))
z <- seq(0, max(zt), by = 1)
x <- approx(zt, x_t, z)$y + rnorm(length(z), sd = 1.5)
cw <- analyse_chirp_wavelet(data.frame(z, x), lowerPeriod = 10, upperPeriod = 300,
    chirp = seq(-0.004, 0.004,
                        by = 0.0005), astro_periods = c(41, 23.7, 19), sedrate = c(0.8, 5))
tr <- track_chirp_hmm(cw)
plot(tr$depth, tr$sedrate, type = "l"); lines(z, approx(zt, sr, z)$y, col = 2)
```

---

analyse_chirplet_sst *Synchrosqueezed wavelet-chirplet transform*

Description

Scalogram with chirped Morlet wavelets: at every scale and position the wavelet whose linear chirp (the change of its frequency along the record) matches the signal best is selected, and the coefficient is squeezed to the instantaneous period estimated with that matched wavelet. Where the period of a cycle changes quickly along the record (a changing sedimentation rate), the ordinary wavelet sees a frequency that sweeps across its bandwidth and the ridge blurs; the matched chirped wavelet follows the sweep, so the ridge stays sharp and its instantaneous period estimate is unbiased for a

locally linear chirp (synchrosqueezed chirplet transforms: Chen et al. 2023; Li et al. 2026). The selected chirp gives, as a by-product, a map of the local rate of change of the period, i.e. of the sedimentation rate.

Usage

```
analyse_chirplet_sst(
  data = NULL,
  lowerPeriod = 2,
  upperPeriod = 1024,
  dj = 1/100,
  omega_nr = c(8, 12, 16),
  chirp = seq(-2, 2, by = 0.25),
  n_iter = 4,
  threshold = "riskshrink",
  smooth = 0.01,
  combine = c("geometric", "min", "mean"),
  verbose = FALSE
)

analyze_chirplet_sst(
  data = NULL,
  lowerPeriod = 2,
  upperPeriod = 1024,
  dj = 1/100,
  omega_nr = c(8, 12, 16),
  chirp = seq(-2, 2, by = 0.25),
  n_iter = 4,
  threshold = "riskshrink",
  smooth = 0.01,
  combine = c("geometric", "min", "mean"),
  verbose = FALSE
)
```

Arguments

`data`	Record: first column depth, second column proxy.
`lowerPeriod`, `upperPeriod`, `dj`	Period axis.
`omega_nr`	Morlet omega_0 of the orders `Default=c(8, 12, 16)`.
`chirp`	Dimensionless chirp values that are tried `Default=seq(-2, 2, by = 0.25)`.
`n_iter`	Iterations of the frequency reassignment (multisynchrosqueezing) `Default=4`.
`threshold`	Noise threshold as in `analyse_wavelet_sst` `Default="riskshrink"`.
`smooth`	Gaussian smoothing along the period axis (octaves) `Default=0.01`.
`combine`	How the orders are combined `Default="geometric"`.
`verbose`	Print the progress `Default=FALSE`.

Details

Several omega_nr values are combined as in analyse_superlet_sst (geometric mean of the squeezed amplitudes). The chirp parameter is dimensionless: chirp = c means that the frequency of the wavelet changes by c / omega_nr (relative) over one scale length (the period times omega_nr / 2 pi).

Value

A list with the structure of analyse_wavelet (so that plot_wavelet, track_astro_hmm etc. work on it) with in addition Chirp: the relative rate of change of the instantaneous period along the record (per depth unit; negative where the period shortens, i.e. where the sedimentation rate falls) of the matched wavelet at every period and position, averaged over the orders.

References

Chen, Z., Jiang, Q., Lin, J. and Shu, C. (2023). Synchrosqueezed chirplet transforms. Also: Li, S. et al. (2026). High-order synchrosqueezed wavelet-chirplet transform for instantaneous frequency and chirprate estimation. J. Comput. Appl. Math.

Examples

```
z <- seq(0, 600, by = 0.5)
sr <- 1 + z / 600                       # sedimentation rate doubles along the record
t <- cumsum(c(0, diff(z) / sr[-1]))
x <- sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 41) + rnorm(length(z), sd = 0.5)
cs <- analyse_chirplet_sst(data.frame(z, x), lowerPeriod = 10, upperPeriod = 200)
plot_wavelet(cs)
```

---

analyse_conceft *Multitaper synchrosqueezing (ConceFT) of a record*

Description

Computes the "Concentration of Frequency and Time" (ConceFT) representation of Daubechies, Wang and Wu (2016), a multitaper version of the wavelet synchrosqueezing transform that is robust against noise. The synchrosqueezed transform of a noisy record contains spurious short ridges because noise also has an instantaneous frequency. ConceFT uses a set of K orthonormal generalised Morse wavelets (Olhede and Walden 2002) and computes the synchrosqueezed transform for many random linear combinations of them (random unit vectors). The noise in these transforms is (nearly) independent while the ridges of the cycles are the same, so the average keeps the cycles and averages the noise away, with much less broadening than averaging the scalograms of the separate wavelets (method = "multitaper", which is also available).

Usage

```
analyse_conceft(
  data = NULL,
  lowerPeriod = 2,
  upperPeriod = 1024,
  dj = 1/100,
  gamma = 3,
  beta = 20,
  K = 2,
  n_comb = 20,
  method = c("conceft", "multitaper"),
  sst_order = 1,
  n_iter = 1,
  threshold = "riskshrink",
  seed = 1,
  verbose = FALSE
)

analyze_conceft(
  data = NULL,
  lowerPeriod = 2,
  upperPeriod = 1024,
  dj = 1/100,
  gamma = 3,
  beta = 20,
  K = 2,
  n_comb = 20,
  method = c("conceft", "multitaper"),
  sst_order = 1,
  n_iter = 1,
  threshold = "riskshrink",
  seed = 1,
  verbose = FALSE
)
```

Arguments

`data`	Input is a time series with the first column being depth or time and the second column being a proxy.
`lowerPeriod, upperPeriod`	Shortest and longest period.
`dj`	Spacing of the period axis in octaves `Default=1/100`.
`gamma, beta`	Parameters of the generalised Morse wavelets `Default=3` and `Default=20` (Lilly's default family; Daubechies et al. 2016 used gamma = 9, beta = 30). A larger beta gives longer wavelets (better frequency resolution, worse depth/time resolution).
`K`	Number of orthonormal Morse wavelets `Default=2`. More tapers suppress noise better but broaden ridges.

n_comb	Number of random combinations Default=20.
method	"conceft" (default) or "multitaper" (average of the synchrosqueezed transforms of the K orthonormal wavelets themselves).
sst_order, n_iter, threshold	Settings of the synchrosqueezing of every combination, see analyse_wavelet_sst.
seed	Seed of the random combinations (for reproducible results) Default=1.
verbose	Print messages.

Value

A list of class analyze.wavelet (see analyse_wavelet_sst) with the averaged power in Power; Wave has this power with the phase of the synchrosqueezed transform of the first (zero order) Morse wavelet, so the extraction functions can use it.

References

Daubechies, I., Wang, Y., Wu, H.-T., 2016. ConceFT: concentration of frequency and time via a multitapered synchrosqueezed transform. Philosophical Transactions of the Royal Society A 374, 20150193.

Olhede, S.C., Walden, A.T., 2002. Generalized Morse wavelets. IEEE Transactions on Signal Processing 50, 2661-2670.

Lilly, J.M., Olhede, S.C., 2012. Generalized Morse wavelets as a superfamily of analytic wavelets. IEEE Transactions on Signal Processing 60, 6036-6041.

Examples

```
z <- 0:999
x <- sin(2 * pi * z / 40) + sin(2 * pi * z / 50) + rnorm(length(z), sd = 1.5)
cf <- analyse_conceft(data.frame(z, x), lowerPeriod = 20, upperPeriod = 100, dj = 1 / 50,
    n_comb = 10)
plot_wavelet(cf)
```

---

analyse_dechirped *De-chirped scalogram of any transform: the record warped to time with a track, transformed, mapped back to depth*

Description

The universal form of the de-chirping of analyse_chirp_eha: the record is converted to the time of a tracked sedimentation rate (any wr_track: track_chirp_hmm, track_astro_hmm, track_astro_auto), so that all cycles become stationary where the track is right, the chosen transform (Morlet wavelet, superlet-SST, multisynchrosqueezing, chirplet-SST or multitaper EHA) is computed in that time domain (the "tuned" scalogram, with the periods in kyr), and the result is mapped back onto the depth axis with the period axis in depth units at every position (period in depth = period in kyr x local rate), so that the cycles are shown as sharp lines that follow the changes of the rate. Where the track is wrong the lines do not straighten in time and do not sharpen in depth, which makes the map a check of the track.

Usage

```
analyse_dechirped(
  data,
  track,
  transform = c("wavelet", "superlet_sst", "wavelet_sst", "chirplet_sst", "eha",
    "combined"),
  lowerPeriod = 10,
  upperPeriod = 600,
  dj = 1/50,
  omega_nr = 10,
  window = NULL,
  n_sim = 0,
  seed = 1,
  ...
)
```

Arguments

data	Record: first column depth, second column proxy.
track	A wr_track (or track_astro_auto result) with depth, sedrate and time.
transform	"wavelet", "superlet_sst", "wavelet_sst" (multisynchrosqueezing), "chirplet_sst", "eha" or "combined": the geometric mean of the rank-normalised Morlet, multitaper EHA and superlet-SST maps (the Morlet is best at the long periods, the EHA at the short ones and the superlet-SST localises the periods best; the product keeps what all three agree on).
lowerPeriod, upperPeriod	Period range in kyr Default=10 and 600.
dj	Period spacing Default=1/50.
omega_nr	Morlet omega_0 for "wavelet" (and the orders of the superlets / chirplets when a vector) Default=10.
window	Window length in kyr for "eha" Default=20 x lowerPeriod.
n_sim	Number of AR1 surrogates for the significance Default=0.
seed	Seed of the surrogates.
...	Further arguments of the transform.

Details

With n_sim > 0 the same warp and transform are applied to AR1 surrogates of the record and the power is compared with the 95th percentile of the surrogates cell by cell (the warp changes the noise as well, so wavy structure in the de-chirped map is only evidence where it exceeds the warped noise).

Value

A list of class wr_dechirped with tuned (the transform in time: an analyze.wavelet object with Period in kyr, usable in plot_wavelet), depth (an analyze.wavelet object on the depth axis with

`Period` in depth units: the de-chirped scalogram), `Power_norm` (the depth map with every column divided by its smooth continuum), and with `n_sim > 0` `Power_sig` (ratio to the 95th percentile of the warped surrogates) and `Power_pval`.

Examples

```
set.seed(1)
t <- seq(0, 2400, by = 1)
x_t <- sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 19) + 0.8 * sin(2 * pi * t / 41)
sr <- 2 + 0.7 * sin(2 * pi * t / 600)
zt <- c(0, cumsum(sr[-1] * diff(t)))
z <- seq(0, max(zt), by = 1)
x <- approx(zt, x_t, z)$y + rnorm(length(z), sd = 1)
res <- track_astro_auto(data.frame(z, x), sedrate = c(0.8, 5), astro_periods = c(41, 23.7, 19))
dc <- analyse_dechirped(data.frame(z, x), res$track, "superlet_sst", lowerPeriod = 10,
    upperPeriod = 120)
plot_dechirped(dc)
```

---

analyse_superlet *Conduct the superlet transform on a time series or signal*

Description

Compute the superlet transform using a complex Morlet wavelet following the approach of Moca et al. (2021). The superlet transform increases time frequency resolution by combining multiple wavelet orders at each frequency. Both the frequency sampling and the scaling of superlet order with frequency can be explicitly controlled.

Usage

```
analyse_superlet(
  data,
  upperPeriod,
  lowerPeriod,
  Nf,
  c1,
  o = NULL,
  mult = TRUE,
  order_scaling = c("none", "log2", "linear", "sqrt", "quadratic", "power"),
  order_alpha = 1,
  verbose = FALSE,
  detrend = c("linear", "mean", "none"),
  standardize = TRUE,
  run_multicore = FALSE,
  n_cores = NULL,
  standardise = NULL,
  wavelet = "morlet",
```

```
  wavelet_order = NULL
)

analyze_superlet(
  data,
  upperPeriod,
  lowerPeriod,
  Nf,
  c1,
  o = NULL,
  mult = TRUE,
  order_scaling = c("none", "log2", "linear", "sqrt", "quadratic", "power"),
  order_alpha = 1,
  verbose = FALSE,
  detrend = c("linear", "mean", "none"),
  standardize = TRUE,
  run_multicore = FALSE,
  n_cores = NULL,
  standardise = NULL,
  wavelet = "morlet",
  wavelet_order = NULL
)
```

Arguments

`data`	Input data as a matrix or data frame. The first column must represent depth or time, and the second column the signal or proxy record.
`upperPeriod`	Maximum period to be analysed. Controls the lowest analysed frequency.
`lowerPeriod`	Minimum period to be analysed. Controls the highest analysed frequency.
`Nf`	Number of frequencies used to construct the superlet spectrum.
`c1`	Base number of cycles of the Morlet wavelet. Acts as the fundamental wavelet width.
`o`	numeric vector of length two defining the minimum and maximum superlet order.
`mult`	Logical. If TRUE, the number of cycles increases multiplicatively with superlet order ($c_i = ic_1$). If FALSE, cycles increase additively ($c_i = c_1 + i - 1$, Moca et al., 2021).
`order_scaling`	Character string defining how the superlet order increases from `o[1]` (lowest frequency) to `o[2]` (highest frequency). One of `"none"` (default, same as `"log2"`: linear in log frequency), `"log2"`, `"linear"` (linear in frequency, as in Moca et al., 2021), `"sqrt"`, `"quadratic"` or `"power"`.
`order_alpha`	Numeric. Exponent used when `scaling = "power"`. Values greater than one emphasize high frequency sharpening, whereas values smaller than one emphasize low frequency sharpening.
`verbose`	Logical. If TRUE, print progress and interpolation information.

detrend — Removal of a trend before the transform: "linear" (default), "mean" or "none". A mean or trend that is not removed leaks into all periods and gives large edge effects, which is a problem for stratigraphic records (for example grey values or counts) that do not have a zero mean.

standardize — Logical. If TRUE (default) the record is divided by its standard deviation, as in analyse_wavelet, so that the power does not depend on the units of the proxy.

run_multicore — Logical. Compute the frequencies on several cores Default=FALSE.

n_cores — Number of cores used when run_multicore = TRUE (NULL: a safe automatic number; see analyse_wavelet). Results do not depend on the number of cores.

standardise — British spelling of standardize; if given it overrides standardize.

wavelet — Mother wavelet, given as a name (PyWavelets style names with the order appended, such as "db4", "sym8", "cgau3", "gaus2" or "cmor1.5-1", also work) or as a function psi_hat(omega) that returns the Fourier transform of a user defined mother wavelet. Analytic (complex) wavelets, which give phase and amplitude and can be synchrosqueezed: "morlet" (default; aliases "gabor", "amor"), "morlet0" (exactly zero mean Morlet, admissible for a small omega_nr; alias "morlex"), "paul" (alias "cauchy"), "morse" (generalised Morse wavelet of Olhede and Walden, 2002; aliases "gmw", "airy"), "bump", "cmor" (complex Morlet with bandwidth and centre frequency as in Matlab/PyWavelets), "cgau" (analytic Gaussian derivative; "cmhat" is the complex Mexican hat), "shannon" (complex Shannon, box spectrum), "fbsp" (frequency B-spline), "meyer", "loggabor" (alias "lognormal"), "gammatone" (asymmetric in time: sharp onset, slow decay) and "hann" (raised cosine band). Real wavelets, which give a real transform (no phase or envelope): "dog" (derivative of Gaussian; "mexh", "ricker", "marr" and "sombrero" are the Mexican hat, order 2), "morl" (real Morlet), "hhat" (Hermitian hat), "poisson", and the orthogonal wavelets "haar", "db" (Daubechies), "sym" (Symlets) and "battle" (Battle-Lemarie spline wavelets), which are computed in the frequency domain from their filters. The period axis of the orthogonal wavelets is approximate because their spectra are broad and not symmetric. A narrow wavelet (Paul, low-order DOG, Shannon, Haar) resolves position well, a wide one (Morse with a large beta, Morlet with a high omega_nr, bump) resolves period well.

wavelet_order — Parameter(s) of the wavelet (see analyse_wavelet); when it is NULL the wavelet with the same relative bandwidth as a Morlet wavelet with the superlet cycles is used at every frequency (cycles are converted as omega_0 = 2 pi cycles / 5). Parameters: paul: order m (default 4); dog: derivative m (default 2); morse: c(gamma, beta) or beta alone with gamma = 3 (default c(3, 20)); bump: c(mu, sigma) (default c(5, 0.6)); cmor: c(bandwidth, centre frequency) (default c(1.5, 1)); cgau: derivative m (default 4); shannon: c(bandwidth, centre frequency) (default c(2, 1.5)); fbsp: c(m, bandwidth, centre frequency) (default c(2, 2, 1.5)); loggabor: c(centre angular frequency, log-width) (default c(6, 0.25)); meyer and hhat have no parameter; morlet0: omega_0 (default omega_nr); morl: omega_0 (default 5); poisson: order n (default 2); gammatone: c(n, centre angular frequency, bandwidth) (default c(4, 6, 1.5)); hann: c(centre angular frequency, half width) (default c(6, 3)); db: number of vanishing moments 1 to 20 (default 4, "haar" = "db1"); sym: 2

to 20 (default 8); `battle`: spline order 1, 3, 5 or 7 (default 3). For the Morlet wavelet use `omega_nr`.

Value

A list with class `"analyze.superlet"` containing Power: time frequency power spectrum dt: sampling interval after interpolation dj: number of frequencies Power.avg: average spectral power Period: physical periods corresponding to frequencies nc: number of columns in the input data nr: number of frequency levels axis.1: x axis values (time or depth) axis.2: y axis values (log2 scaled periods) c1: base number of wavelet cycles o: numeric vector of length two defining the minimum and maximum superlet order. If NULL, all frequencies are analysed using order one. freq.scale: scaling of the frequencies order.mode: order.mode ("none", "log2", "linear", "sqrt", "power") order: superlet order of every period coi: distance from the start and end of the record within which edge effects are important (cone of influence), for every period coi.1, coi.2: the cone of influence as polygons (x and log2 period), drawn by the plot functions x: interpolated x values y: interpolated signal values

Author(s)

Adapted from MATLAB implementations by V. V. Moca et al. (2021)

References

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., & Mureșan, R. C. (2021). Time-frequency super-resolution with superlets. Nature Communications, 12(1), 337. doi:10.1038/s41467020205399

Examples

```
## Example 1. Using the Total Solar Irradiance data set of Steinhilber et al. (2012)
TSI_sl <-
  analyse_superlet(
    data = TSI,
    upperPeriod = 8192,
    lowerPeriod = 16,
    Nf = 20,
    c1 = 2,
    o = c(1, 5),
    mult = TRUE,
    order_scaling = "log2",
    order_alpha = 1,
    verbose = FALSE
  )


## Example 2. Using the magnetic susceptibility data set of Pas et al. (2018)
mag_sl <-
  analyse_superlet(
    data = mag,
    upperPeriod = 254,
    lowerPeriod = 0.1,
    Nf = 20,
```

```
    c1 = 2,
    o = c(1, 5),
    mult = TRUE,
    order_scaling = "log2",
    order_alpha = 1,
    verbose = FALSE
  )


## Example 3. Using the greyscale data set of Zeeden et al. (2013)
grey_sl <-
  analyse_superlet(
    data = grey,
    upperPeriod = 256,
    lowerPeriod = 0.02,
    Nf = 20,
    c1 = 2,
    o = c(2,5),
    mult = TRUE,
    order_scaling = "log2",
    order_alpha = 1,
    verbose = FALSE)
```

`analyse_superlet_sst` *Superlet synchrosqueezing: consensus of synchrosqueezed transforms with short and long wavelets*

Description

Combines the idea of superlets (Moca et al. 2021) with synchrosqueezing. A short wavelet (few cycles) follows fast changes of the period, for instance where the sedimentation rate changes, but smears the energy over a broad range of periods; a long wavelet (many cycles) separates close periods but blurs changes along the record. Superlets combine a set of wavelets with an increasing number of cycles by the geometric mean of their amplitudes, which keeps the energy only where all of them agree. Here the wavelet transforms are first synchrosqueezed (first or second order, see `analyse_wavelet_sst`), slightly smoothed along the period axis, and then combined by the geometric mean. A cycle that is present in the record is found by every order at (nearly) the same period and survives; noise is squeezed to different, random periods by the different orders and is strongly suppressed; ridges of the long wavelets that lag behind a change of the period are removed by the short wavelets and vice versa. This gives sharp ridges that follow changes of the sedimentation rate and separate close cycles in noisy records.

The defaults (orders with omega_0 = 8, 12 and 16, second-order synchrosqueezing with 4 iterations of the reassignment, smoothing of 0.01 octave) gave the most concentrated and best separated ridges in synthetic tests with eccentricity-modulated precession, obliquity, changing sedimentation rates and red noise (see NEWS).

Usage

```
analyse_superlet_sst(
  data = NULL,
  lowerPeriod = 2,
  upperPeriod = 1024,
  dj = 1/100,
  c1 = 4,
  o = 5,
  mult = TRUE,
  omega_nr = c(8, 12, 16),
  wavelet = "morlet0",
  wavelet_order = NULL,
  sst_order = 2,
  n_iter = 4,
  threshold = "riskshrink",
  smooth = 0.01,
  combine = c("geometric", "min", "mean"),
  verbose = FALSE
)

analyze_superlet_sst(
  data = NULL,
  lowerPeriod = 2,
  upperPeriod = 1024,
  dj = 1/100,
  c1 = 4,
  o = 5,
  mult = TRUE,
  omega_nr = c(8, 12, 16),
  wavelet = "morlet0",
  wavelet_order = NULL,
  sst_order = 2,
  n_iter = 4,
  threshold = "riskshrink",
  smooth = 0.01,
  combine = c("geometric", "min", "mean"),
  verbose = FALSE
)
```

Arguments

`data`	Input is a time series with the first column being depth or time and the second column being a proxy.
`lowerPeriod, upperPeriod`	Shortest and longest period.
`dj`	Spacing of the period axis in octaves `Default=1/100`.
`c1, o, mult`	Superlet style definition of the orders, used when `omega_nr = NULL`: `o` orders

with `i * c1` cycles (`mult = TRUE`) or `c1 + i - 1` cycles (`mult = FALSE`); cycles are converted to omega_0 = 2 pi cycles / 5, as in `analyse_superlet`.

`omega_nr` omega_0 (Morlet equivalent central angular frequencies) of the orders `Default=c(8, 12, 16)`. Set to `NULL` to use `c1`, `o` and `mult`.

`wavelet`, `wavelet_order` Mother wavelet, see `analyse_wavelet` `Default="morlet0"` (the exactly zero-mean Morlet wavelet, which stays admissible for few cycles). The orders of other wavelets get the same relative bandwidth as the Morlet wavelet with the same number of cycles.

`sst_order` Order of the synchrosqueezing (1 or 2) `Default=2`.

`n_iter` Iterations of the frequency reassignment (multisynchrosqueezing) of every order, see `analyse_wavelet_sst` `Default=4`.

`threshold` Noise threshold of the synchrosqueezing, see `analyse_wavelet_sst`.

`smooth` Standard deviation (in octaves) of the Gaussian smoothing along the period axis before the orders are combined. It absorbs the small differences in the period estimated by the different orders `Default=0.01`.

`combine` `"geometric"` (default, as superlets), `"min"` (strictest consensus) or `"mean"`.

`verbose` Print messages.

Value

A list of class `analyze.wavelet` (see `analyse_wavelet_sst`) that can be plotted with `plot_wavelet` and used by the tracking functions, with the extra elements `superlet = TRUE`, `omega_nr_all` (omega_0 of every order) and `Power_orders` (the squeezed power of every order, a list).

References

Moca, V.V., Barzan, H., Nagy-Dabacan, A., Muresan, R.C., 2021. Time-frequency super-resolution with superlets. Nature Communications 12, 337.

Daubechies, I., Lu, J., Wu, H.-T., 2011. Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool. Applied and Computational Harmonic Analysis 30, 243-261.

Oberlin, T., Meignen, S., Perrier, V., 2015. Second-order synchrosqueezing transform or invertible reassignment? Towards ideal time-frequency representations. IEEE Transactions on Signal Processing 63, 1335-1344.

Examples

```
z <- 0:1499
x <- ifelse(z < 750, sin(2 * pi * z / 40), sin(2 * pi * z / 44)) + rnorm(length(z), sd = 1)
s <- analyse_superlet_sst(data.frame(z, x), lowerPeriod = 20, upperPeriod = 100, dj = 1 / 50)
plot_wavelet(s)
```

`analyse_wavelet` *Conduct the continuous wavelet transform on a time series/signal*

Description

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

Usage

```
analyse_wavelet(
  data = NULL,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = FALSE,
  omega_nr = 8,
  wavelet = "morlet",
  wavelet_order = NULL,
  pval = FALSE,
  n_simulations = 10,
  run_multicore = FALSE,
  n_cores = NULL
)

analyze_wavelet(
  data = NULL,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = FALSE,
  omega_nr = 8,
  wavelet = "morlet",
  wavelet_order = NULL,
  pval = FALSE,
  n_simulations = 10,
  run_multicore = FALSE,
  n_cores = NULL
)
```

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. The CWT analyses analyses the signal using successive periods which increase by the power of 2 (e.g.2^0=1,2^1=2,2^2=4,2^3=8,2^4=16). To have more resolution in-between these steps the dj parameter exists, the dj parameter specifies how many extra steps/spacing in-between the power of 2

scaled CWT is added. The amount of steps is 1/x with a higher x indicating a smaller spacing. Increasing the increases the computational time of the CWT Default=1/200.

lowerPeriod — Lowest period to be analysed Default=2. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyse the signal for a specific range of cycles. 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 analysed Default=1024. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyse the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.

verbose — Print text Default=FALSE.

omega_nr — The omega number: central angular frequency of the Morlet wavelet. A larger value gives a longer wavelet with more oscillations, a better frequency resolution and a worse resolution in depth/time

wavelet — Mother wavelet, given as a name (PyWavelets style names with the order appended, such as "db4", "sym8", "cgau3", "gaus2" or "cmor1.5-1", also work) or as a function psi_hat(omega) that returns the Fourier transform of a user defined mother wavelet. Analytic (complex) wavelets, which give phase and amplitude and can be synchrosqueezed: "morlet" (default; aliases "gabor", "amor"), "morlet0" (exactly zero mean Morlet, admissible for a small omega_nr; alias "morlex"), "paul" (alias "cauchy"), "morse" (generalised Morse wavelet of Olhede and Walden, 2002; aliases "gmw", "airy"), "bump", "cmor" (complex Morlet with bandwidth and centre frequency as in Matlab/PyWavelets), "cgau" (analytic Gaussian derivative; "cmhat" is the complex Mexican hat), "shannon" (complex Shannon, box spectrum), "fbsp" (frequency B-spline), "meyer", "loggabor" (alias "lognormal"), "gammatone" (asymmetric in time: sharp onset, slow decay) and "hann" (raised cosine band). Real wavelets, which give a real transform (no phase or envelope): "dog" (derivative of Gaussian; "mexh", "ricker", "marr" and "sombrero" are the Mexican hat, order 2), "morl" (real Morlet), "hhat" (Hermitian hat), "poisson", and the orthogonal wavelets "haar", "db" (Daubechies), "sym" (Symlets) and "battle" (Battle-Lemarie spline wavelets), which are computed in the frequency domain from their filters. The period axis of the orthogonal wavelets is approximate because their spectra are broad and not symmetric. A narrow wavelet (Paul, low-order DOG, Shannon, Haar) resolves position well, a wide one (Morse with a large beta, Morlet with a high omega_nr, bump) resolves period well.

wavelet_order — Parameter(s) of the wavelet: paul: order m (default 4); dog: derivative m (default 2); morse: c(gamma, beta) or beta alone with gamma = 3 (default c(3, 20)); bump: c(mu, sigma) (default c(5, 0.6)); cmor: c(bandwidth, centre frequency) (default c(1.5, 1)); cgau: derivative m (default 4); shannon: c(bandwidth, centre frequency) (default c(2, 1.5)); fbsp: c(m, bandwidth, centre frequency) (default c(2, 2, 1.5)); loggabor: c(centre angular frequency, log-width) (default c(6, 0.25)); meyer and hhat have no parameter; morlet0: omega_0 (default omega_nr); morl: omega_0 (default 5); poisson: order n (default 2); gammatone: c(n, centre angular frequency, bandwidth) (default c(4, 6, 1.5)); hann: c(centre angular frequency, half width) (de-

fault c(6, 3)); db: number of vanishing moments 1 to 20 (default 4, "haar" = "db1"); sym: 2 to 20 (default 8); battle: spline order 1, 3, 5 or 7 (default 3). For the Morlet wavelet use omega_nr. The period axis is always the Fourier period, i.e. the period of the sinusoid at which the wavelet has its maximum response, so the results of different wavelets are comparable.

pval — calculate the P-value Default=FALSE. The p-value is based on Monte Carlo modelling runs on surrogate data generated based on autocorrelated noise (red noise) the calculated using a windowed (the window is half the size of the data set) temporal autocorrelation and on shuffling the data set resulting in a random data sets which has similar spectral characteristics to the original data set.The shuffling of the data set creates white noise which ensures that high amplitude high frequency/short period cycles do not result in statistical significant peaks. The part of the data generated using the autocorrelated noise (red noise) based on the windowed (the window is half the size of the data set) temporal autocorrelation represent a spectral signature similar to to that of the original data. The original data might include spectral peaks which are the result of astronomical forcing. The result is that the spectral power profile is biased towards rejecting the 0-hypothesis (e.g. no astronomical forcing). By combining the shuffling of the data set with autocorrelated noise a surrogate data set is created which rejects high amplitude high frequency/short period cycles and a reduced biased towards towards rejecting the 0-hypothesis if the data was solely the result of autocorrelated noise

n_simulations — Number of simulation to be ran to generate the p-value

run_multicore — Run p-value calculation with one core or multiple cores

n_cores — Number of cores used when ‘run_multicore = TRUE‘ (‘Default = NULL‘: an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set ‘options(waverider.n_cores = ...)‘ to change the default. A value > 1 also switches on parallel computing. Results do not depend on the number of cores.

Value

The output is a list (wavelet object) which contain 20 objects which are the result of the continuous wavelet transform (CWT). Object 1: Wave - Wave values of the wavelet Object 2: Phase - Phase of the wavelet Object 3: Ampl - Amplitude values of the wavelet Object 4: Power - Power values of the wavelet Object 5: dt - Step size Object 6: dj - Scale size Object 7: Power.avg - Average power values Object 8: Period - Period values Object 9: Scale - Scale value Object 10: coi.1 - Cone of influence values 1 Object 11: coi.2 - Cone of influence values 2 Object 12: nc - Number of columns Object 13: nr - Number of rows Object 14: axis.1 - axis values 1 Object 15: axis.2 - axis values 2 Object 16: omega_nr - Number of cycles in the wavelet Object 17: x - x values of the data set Object 18: y - y values of the data set Object 19: average p value of the spectral power Object 20: p value of spectral power

Author(s)

Code based on on the "WaveletComp" function of the ’WaveletComp’ R package and "wt" function of the ’biwavelet’ R 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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#Example 1. Using the Total Solar Irradiance data set of Steinhilver et al., (2012)
TSI_wt <-
 analyse_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6,
   pval=FALSE,
   n_simulations=10,
   run_multicore = FALSE
 )


#Example 2. Using the magnetic susceptibility data set of Pas et al., (2018)
mag_wt <-
analyse_wavelet(
data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10,
pval=FALSE,
n_simulations=10,
run_multicore = FALSE
)

#Example 3. Using the greyscale data set of Zeeden et al., (2013)
grey_wt <-
 analyse_wavelet(
   data = grey,
   dj = 1/200,
```

```
  lowerPeriod = 0.02,
  upperPeriod = 256,
  verbose = FALSE,
  omega_nr = 8,
  pval=FALSE,
  n_simulations=10,
  run_multicore = FALSE
)
```

---

analyse_wavelet_coherence

Cross-wavelet coherence analysis

Description

Computes the cross-wavelet transform and wavelet coherence between two time/depth series using a superlet-based wavelet approach. The function internally interpolates both input series onto a common grid, computes individual wavelet transforms, and derives cross-wavelet power, phase, and coherence with configurable smoothing.

Usage

```
analyse_wavelet_coherence(
  data_1,
  data_2,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = FALSE,
  omega_nr = 8,
  wavelet = "morlet",
  wavelet_order = NULL,
  n_simulations = 10,
  window.type.t = 4,
  window.type.s = 4,
  abs_window_t = 500,
  abs_window_s = 0.5
)

analyze_wavelet_coherence(
  data_1,
  data_2,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = FALSE,
```

```
  omega_nr = 8,
  wavelet = "morlet",
  wavelet_order = NULL,
  n_simulations = 10,
  window.type.t = 4,
  window.type.s = 4,
  abs_window_t = 500,
  abs_window_s = 0.5
)
```

Arguments

data_1	First input dataset as a two-column matrix or data.frame. First column must contain time/depth, second column the signal.
data_2	Second input dataset as a two-column matrix or data.frame. Must have the same structure as data_1.
dj	Spacing between discrete scales. Smaller values increase resolution. Default is 1/100.
lowerPeriod	Lower bound of the period range to analyze.
upperPeriod	Upper bound of the period range to analyze.
verbose	Logical; if TRUE, prints progress messages.
omega_nr	Number of oscillations in the Morlet/superlet wavelet.
wavelet	Mother wavelet, given as a name (PyWavelets style names with the order appended, such as "db4", "sym8", "cgau3", "gaus2" or "cmor1.5-1", also work) or as a function psi_hat(omega) that returns the Fourier transform of a user defined mother wavelet. Analytic (complex) wavelets, which give phase and amplitude and can be synchrosqueezed: "morlet" (default; aliases "gabor", "amor"), "morlet0" (exactly zero mean Morlet, admissible for a small omega_nr; alias "morlex"), "paul" (alias "cauchy"), "morse" (generalised Morse wavelet of Olhede and Walden, 2002; aliases "gmw", "airy"), "bump", "cmor" (complex Morlet with bandwidth and centre frequency as in Matlab/PyWavelets), "cgau" (analytic Gaussian derivative; "cmhat" is the complex Mexican hat), "shannon" (complex Shannon, box spectrum), "fbsp" (frequency B-spline), "meyer", "loggabor" (alias "lognormal"), "gammatone" (asymmetric in time: sharp onset, slow decay) and "hann" (raised cosine band). Real wavelets, which give a real transform (no phase or envelope): "dog" (derivative of Gaussian; "mexh", "ricker", "marr" and "sombrero" are the Mexican hat, order 2), "morl" (real Morlet), "hhat" (Hermitian hat), "poisson", and the orthogonal wavelets "haar", "db" (Daubechies), "sym" (Symlets) and "battle" (Battle-Lemarie spline wavelets), which are computed in the frequency domain from their filters. The period axis of the orthogonal wavelets is approximate because their spectra are broad and not symmetric. A narrow wavelet (Paul, low-order DOG, Shannon, Haar) resolves position well, a wide one (Morse with a large beta, Morlet with a high omega_nr, bump) resolves period well.
wavelet_order	Parameter(s) of the wavelet: paul: order m (default 4); dog: derivative m (default 2); morse: c(gamma, beta) or beta alone with gamma = 3 (default c(3,

20)); bump: c(mu, sigma) (default c(5, 0.6)); cmor: c(bandwidth, centre frequency) (default c(1.5, 1)); cgau: derivative m (default 4); shannon: c(bandwidth, centre frequency) (default c(2, 1.5)); fbsp: c(m, bandwidth, centre frequency) (default c(2, 2, 1.5)); loggabor: c(centre angular frequency, log-width) (default c(6, 0.25)); meyer and hhat have no parameter; morlet0: omega_0 (default omega_nr); morl: omega_0 (default 5); poisson: order n (default 2); gammatone: c(n, centre angular frequency, bandwidth) (default c(4, 6, 1.5)); hann: c(centre angular frequency, half width) (default c(6, 3)); db: number of vanishing moments 1 to 20 (default 4, "haar" = "db1"); sym: 2 to 20 (default 8); battle: spline order 1, 3, 5 or 7 (default 3). For the Morlet wavelet use omega_nr. Controls time-frequency resolution trade-off.

n_simulations Number of simulations (reserved for future significance testing). Currently not used internally.

window.type.t Type of smoothingw indow applied in the time direction. Options: "none", "bar" (Bartlett), "tri" (triangular), "box" (boxcar), "han" (Hanning, default), "ham" (Hamming) or "bla" (Blackman)

window.type.s Type of smoothing window applied in the scale direction. Same options as window.type.t.

abs_window_t Absolute smoothing window size in the time direction (same units as input data).

abs_window_s Absolute smoothing window size in the scale direction (in units of dj).

Value

A list of class containing: Wave Smoothed cross-wavelet transform (complex) Coherence Wavelet coherence matrix Phase Phase difference between the two signals Coh.avg Average coherence over time Period (m)vector corresponding to scales Scale Wavelet scales dt Time step of interpolated grid dj Scale resolution axis.1 Time/depth axis axis.2 Period axis nc, nr Dimensions of the wavelet matrices x1, y1 Interpolated first dataset x2, y2 Interpolated second dataset

Author(s)

plotting code based on the "analyze.coherency" function of the 'WaveletComp' R package

References

Torrence, C., and G. P. Compo (1998). A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society*, 79(1), 61–78.

Examples

```
# Generate two synthetic signals
t <- seq(0, 1000, by = 1)
x1 <- sin(2 * pi * t / 100) + rnorm(length(t), 0, 0.5)
x2 <- sin(2 * pi * t / 100 + pi/4) + rnorm(length(t), 0, 0.5)

data_1 <- cbind(t, x1)
data_2 <- cbind(t, x2)
```

```
coh <- analyse_wavelet_coherence(
  data_1 = data_1,
  data_2 = data_2,
  lowerPeriod = 2,
  upperPeriod = 256
)
```

`analyse_wavelet_sst` *Wavelet synchrosqueezing transform*

Description

Computes the continuous wavelet transform of a record and sharpens it with synchrosqueezing (Daubechies, Lu and Wu 2011; Thakur et al. 2013). A wavelet scalogram smears the energy of a cycle over neighbouring periods because the wavelet has a finite bandwidth. Synchrosqueezing estimates the instantaneous frequency of every wavelet coefficient from the phase of the transform (the time derivative of the coefficient divided by the coefficient) and moves the coefficient to that frequency. The result is a time-frequency representation in which a cycle appears as a thin ridge at its instantaneous period, so that closely spaced or slowly changing cycles are better separated than in the scalogram, and the transform stays invertible: summing the squeezed coefficients gives back the record, so a cycle can be reconstructed from a band of the squeezed transform.

The result has the same structure as the output of `analyse_wavelet` (class `analyze.wavelet`), so it can be plotted with `plot_wavelet` and used by the tracking and extraction functions (e.g. `track_period_wavelet`, `extract_signal`) in place of the scalogram.

Usage

```
analyse_wavelet_sst(
  data = NULL,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  omega_nr = 12,
  wavelet = "morlet",
  wavelet_order = NULL,
  threshold = "riskshrink",
  sst_order = 1,
  type = c("sst", "set", "reassign"),
  n_iter = 1,
  adaptive_window = NULL,
  adaptive_on = c("sst", "cwt"),
  verbose = FALSE
)
```

```
analyze_wavelet_sst(
  data = NULL,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  omega_nr = 12,
  wavelet = "morlet",
  wavelet_order = NULL,
  threshold = "riskshrink",
  sst_order = 1,
  type = c("sst", "set", "reassign"),
  n_iter = 1,
  adaptive_window = NULL,
  adaptive_on = c("sst", "cwt"),
  verbose = FALSE
)
```

Arguments

`data`	Input is a time series with the first column being depth or time and the second column being a proxy.
`dj`	Spacing of the scales in octaves (as in `analyse_wavelet`); also the spacing of the squeezed period axis.
`lowerPeriod, upperPeriod`	Shortest and longest period of the transform.
`omega_nr`	Central angular frequency of the Morlet wavelet (see `analyse_wavelet`); several values (e.g. `c(6, 10, 16, 24)`) select adaptive synchrosqueezing, see `adaptive_window`. Synchrosqueezing separates cycles that are further apart than the bandwidth of the wavelet, so a larger value (10 to 12) separates close cycles better; the interference of two close cycles otherwise makes the ridges wave.
`wavelet, wavelet_order`	Mother wavelet and its order, see `analyse_wavelet`. Synchrosqueezing needs an analytic wavelet of `analyse_wavelet` (not the real wavelets dog, morl, hhat, poisson, haar, db, sym and battle).
`threshold`	Coefficients with a modulus below this value are not squeezed, because the phase of a coefficient that is dominated by noise gives a meaningless frequency, and squeezing them scatters noise over the whole spectrum. `"riskshrink"` (default) estimates the noise level from the finest scales as `sqrt(2 log(N)) * 1.4826 * MAD(|W|)` (Donoho and Johnstone 1994; as in the Synchrosqueezing Toolbox of Brevdo and Thakur and in `ssqueezepy`); a number is used directly, in standard deviations of the (standardised) record, since the transform is normalised so that white noise gives the same modulus at every scale; `"none"` uses a negligible value (sqrt of the machine epsilon), which squeezes everything.
`sst_order`	1 (default) for the classic synchrosqueezing, 2 for the second-order synchrosqueezing transform (Oberlin, Meignen and Perrier 2015; Oberlin and Meignen 2017). The first-order estimate of the instantaneous frequency is the frequency at the centre of gravity (in time) of the part of the signal that a wavelet sees, which lags

behind when the period changes quickly (a changing sedimentation rate makes the cycle a chirp). The second-order estimate measures the rate of change of the frequency (the chirp rate) from additional transforms and corrects for it, so that ridges follow fast changes of the sedimentation rate without giving up frequency resolution.

`type` `"sst"` (default) squeezes every coefficient to its instantaneous period; `"set"` is the synchroextracting transform (Yu et al. 2017), which keeps only the coefficients whose instantaneous period equals the period of their own scale (the ridge itself) and gives the thinnest ridges, at the cost of discarding the energy away from the ridge (it cannot be used to reconstruct the full signal).

`n_iter` Number of synchrosqueezing iterations (multisynchrosqueezing, Yu et al. 2019) `Default=1`.

`adaptive_window`

Only used when `omega_nr` has several values (adaptive synchrosqueezing): the transform is computed for every `omega_nr` and at every position the one with the most concentrated energy (lowest Renyi entropy of order 3 within a moving window of this length, in the units of the depth/time axis) is used. A short wavelet is then chosen where the period changes quickly and a long wavelet where close cycles need to be separated. `Default=NULL`: three times the geometric mean of `lowerPeriod` and `upperPeriod`.

`adaptive_on` Transform on which the entropy of the adaptive selection is measured: `"sst"` (default), the squeezed transform, which picks short wavelets where the period changes quickly, but in very noisy records also prefers short wavelets (noise is squeezed into sparse spikes); or `"cwt"`, the wavelet transform, which is robust against noise but nearly always prefers the longest wavelet.

`verbose` Print messages.

Details

Every coefficient W(s, t) is moved to the period that its instantaneous frequency gives. With the Torrence and Compo (1998) normalisation the reconstruction sum W / sqrt(s) is conserved: the squeezed coefficients are stored such that `Re(Wave) / sqrt(Scale)` summed over the periods equals the same sum of the scalogram, which is what `extract_signal` and related functions use. Coefficients whose instantaneous period falls outside `lowerPeriod` - `upperPeriod` are discarded, so very strong cycles at the edge of the range lose part of their energy there.

The ridge of a stationary cycle is sharp; for a cycle with a changing period the ridge follows the period (the instantaneous frequency). The power of the squeezed transform is not proportional to the squared amplitude of the cycle in the same way as in a scalogram, because energy is concentrated.

Value

A list of class `analyze.wavelet` like `analyse_wavelet` (`Wave`, `Power`, `Ampl`, `Phase`, `Period`, `Scale`, `coi.1`, `coi.2`, `axis.1`, `axis.2`, ...) for the squeezed transform, with the extra elements `synchrosqueezed = TRUE`, `Wave_cwt` and `Power_cwt` (the unsqueezed transform) and `inst_period` (matrix with the instantaneous period of every coefficient).

References

Daubechies, I., Lu, J., Wu, H.-T., 2011. Synchrosqueezed wavelet transforms: an empirical mode decomposition-like tool. Applied and Computational Harmonic Analysis 30, 243-261.

Thakur, G., Brevdo, E., Fuckar, N.S., Wu, H.-T., 2013. The synchrosqueezing algorithm for time-varying spectral analysis: robustness properties and new paleoclimate applications. Signal Processing 93, 1079-1094.

Torrence, C., Compo, G.P., 1998. A practical guide to wavelet analysis. Bulletin of the American Meteorological Society 79, 61-78.

Donoho, D.L., Johnstone, I.M., 1994. Ideal spatial adaptation by wavelet shrinkage. Biometrika 81, 425-455.

Oberlin, T., Meignen, S., Perrier, V., 2015. Second-order synchrosqueezing transform or invertible reassignment? Towards ideal time-frequency representations. IEEE Transactions on Signal Processing 63, 1335-1344.

Oberlin, T., Meignen, S., 2017. The second-order wavelet synchrosqueezing transform. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 3994-3998.

Yu, G., Wang, Z., Zhao, P., 2019. Multisynchrosqueezing transform. IEEE Transactions on Industrial Electronics 66, 5441-5455.

Auger, F., Flandrin, P., 1995. Improving the readability of time-frequency and time-scale representations by the reassignment method. IEEE Transactions on Signal Processing 43, 1068-1089.

Yu, G., Yu, M., Xu, C., 2017. Synchroextracting transform. IEEE Transactions on Industrial Electronics 64, 8042-8054.

Sheu, Y.-L., Hsu, L.-Y., Chou, P.-T., Wu, H.-T., 2017. Entropy-based time-varying window width selection for nonlinear-type time-frequency analysis. International Journal of Data Science and Analytics 3, 231-245.

Examples

```
t <- seq(0, 1000, by = 2)
x <- sin(2 * pi * t / 45) + sin(2 * pi * t / 60) + rnorm(length(t), sd = 0.2)
cwt <- analyse_wavelet(data.frame(t, x), lowerPeriod = 20, upperPeriod = 150, dj = 1 / 40)
sst <- analyse_wavelet_sst(data.frame(t, x), lowerPeriod = 20, upperPeriod = 150, dj = 1 / 40)
op <- par(mfrow = c(1, 2))
plot_wavelet(cwt)
plot_wavelet(sst)
par(op)
```

`analyse_Xsuperlet` *Conduct the cross superlet transform on a time series or signal*

Description

Compute the cross superlet transform using a complex Morlet wavelet based on the approach of Moca et al. (2021) and the "analyze.coherency" function of the WaveletComp package. The superlet transform increases time frequency resolution by combining multiple wavelet orders at each frequency.

Usage

```
analyse_Xsuperlet(
  data_1 = NULL,
  data_2 = NULL,
  upperPeriod = 1024,
  lowerPeriod = 2,
  Nf = 128,
  c1 = 3,
  o = c(1, 10),
  mult = TRUE,
  order_scaling = "log2",
  order_alpha = 1,
  verbose = FALSE,
  detrend = "linear",
  standardize = TRUE,
  standardise = NULL,
  wavelet = "morlet",
  wavelet_order = NULL
)

analyze_Xsuperlet(
  data_1 = NULL,
  data_2 = NULL,
  upperPeriod = 1024,
  lowerPeriod = 2,
  Nf = 128,
  c1 = 3,
  o = c(1, 10),
  mult = TRUE,
  order_scaling = "log2",
  order_alpha = 1,
  verbose = FALSE,
  detrend = "linear",
  standardize = TRUE,
  standardise = NULL,
  wavelet = "morlet",
  wavelet_order = NULL
)
```

Arguments

data_1	First input data set as a matrix or data frame. The first column must represent depth or time, and the second column the signal or proxy record.
data_2	Second input data as a matrix or data frame. The first column must represent depth or time, and the second column the signal or proxy record.
upperPeriod	Maximum period to be analysed. Controls the lowest analysed frequency.
lowerPeriod	Minimum period to be analysed. Controls the highest analysed frequency.

`Nf` Number of frequencies used to construct the superlet spectrum. Frequencies are logarithmically spaced between the limits defined by `upperPeriod` and `lowerPeriod`.

`c1` Base number of cycles of the Morlet wavelet. Acts as the fundamental wavelet width.

`o` numeric vector of length two defining the minimum and maximum superlet order. If NULL, all frequencies are analysed using order one.

`mult` Logical. If TRUE, the number of cycles increases multiplicatively with superlet order ($c_i = ic_1$). If FALSE, cycles increase additively ($c_i = c_1 + i - 1$, Moca et al., 2021).

`order_scaling` Character string defining how the number of wavelet cycles varies with scale. One of `"log2"`, `"linear"`, `"sqrt"`, `"quadratic"`, or `"power"`.

`order_alpha` Numeric. Exponent used when `scaling = "power"`. Values greater than one emphasize high frequency sharpening, whereas values smaller than one emphasize low frequency sharpening.

`verbose` Logical. If TRUE, print progress and interpolation information.

`detrend, standardize`
Passed to `analyse_superlet`: removal of a linear trend (or mean) and standardisation of both records before the transform `Default="linear"` and `TRUE`.

`standardise` British spelling of `standardize`; if given it overrides `standardize`.

`wavelet` Mother wavelet, given as a name (PyWavelets style names with the order appended, such as `"db4"`, `"sym8"`, `"cgau3"`, `"gaus2"` or `"cmor1.5-1"`, also work) or as a function `psi_hat(omega)` that returns the Fourier transform of a user defined mother wavelet. Analytic (complex) wavelets, which give phase and amplitude and can be synchrosqueezed: `"morlet"` (default; aliases `"gabor"`, `"amor"`), `"morlet0"` (exactly zero mean Morlet, admissible for a small omega_nr; alias `"morlex"`), `"paul"` (alias `"cauchy"`), `"morse"` (generalised Morse wavelet of Olhede and Walden, 2002; aliases `"gmw"`, `"airy"`), `"bump"`, `"cmor"` (complex Morlet with bandwidth and centre frequency as in Matlab/PyWavelets), `"cgau"` (analytic Gaussian derivative; `"cmhat"` is the complex Mexican hat), `"shannon"` (complex Shannon, box spectrum), `"fbsp"` (frequency B-spline), `"meyer"`, `"loggabor"` (alias `"lognormal"`), `"gammatone"` (asymmetric in time: sharp onset, slow decay) and `"hann"` (raised cosine band). Real wavelets, which give a real transform (no phase or envelope): `"dog"` (derivative of Gaussian; `"mexh"`, `"ricker"`, `"marr"` and `"sombrero"` are the Mexican hat, order 2), `"morl"` (real Morlet), `"hhat"` (Hermitian hat), `"poisson"`, and the orthogonal wavelets `"haar"`, `"db"` (Daubechies), `"sym"` (Symlets) and `"battle"` (Battle-Lemarie spline wavelets), which are computed in the frequency domain from their filters. The period axis of the orthogonal wavelets is approximate because their spectra are broad and not symmetric. A narrow wavelet (Paul, low-order DOG, Shannon, Haar) resolves position well, a wide one (Morse with a large beta, Morlet with a high omega_nr, bump) resolves period well.

`wavelet_order` Parameter(s) of the wavelet (see `analyse_superlet`): `paul`: order m (default 4); `dog`: derivative m (default 2); `morse`: `c(gamma, beta)` or beta alone with gamma = 3 (default `c(3, 20)`); `bump`: `c(mu, sigma)` (default `c(5, 0.6)`); `cmor`: `c(bandwidth, centre frequency)` (default `c(1.5, 1)`); `cgau`: derivative m (default 4); `shannon`: `c(bandwidth, centre frequency)` (default `c(2, 1.5)`);

fbsp: c(m, bandwidth, centre frequency) (default c(2, 2, 1.5)); loggabor: c(centre angular frequency, log-width) (default c(6, 0.25)); meyer and hhat have no parameter; morlet0: omega_0 (default omega_nr); morl: omega_0 (default 5); poisson: order n (default 2); gammatone: c(n, centre angular frequency, bandwidth) (default c(4, 6, 1.5)); hann: c(centre angular frequency, half width) (default c(6, 3)); db: number of vanishing moments 1 to 20 (default 4, "haar" = "db1"); sym: 2 to 20 (default 8); battle: spline order 1, 3, 5 or 7 (default 3). For the Morlet wavelet use omega_nr.

Value

A list with class "analyze.Xsuperlet" containing Wave: complex wavelet coefficients Power: time frequency power spectrum dt: sampling interval after interpolation Phase: instantaneous phase of the signal dj: number of frequencies Power.avg: average spectral power Period: physical periods corresponding to frequencies nc: number of columns in the input data nr: number of frequency levels axis.1: x axis values (time or depth) axis.2: y axis values (log2 scaled periods) c1: base number of wavelet cycles o: numeric vector of length two defining the minimum and maximum superlet order. If NULL, all frequencies are analysed using order one. x1: interpolated x values first data set y1: interpolated signal values first data set x2: interpolated x values second data set y2: interpolated signal values second data set

Author(s)

The "analyse_Xsuperlet" that generates the input for the plotting function is based on the matlab code in Moca et al. (2021) and the the "analyze.coherency" function of the 'WaveletComp' R package

References

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., & Mureșan, R. C. (2021). Time-frequency super-resolution with superlets. Nature Communications, 12(1), 337. doi:10.1038/s41467020205399

Examples

```
#Example 1. A cross superlet of two etp solutions with noise overprint
etp_1 <- astrochron::etp(
 tmin = 0,
 tmax = 300,
 dt = 1,
 eWt = 1.5,
 oWt = 0.75,
 pWt = 1,
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
 verbose = FALSE
)

etp_2 <- astrochron::etp(
 tmin = 0,
 tmax = 300,
```

```
 dt = 1,
 eWt = 1,
 oWt = 0.5,
 pWt = 1.5,
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
 verbose = FALSE
)

etp_1[, 2] <- etp_1[, 2] + wr_coloured_noise(
 nrow(etp_1),
 sd = sd(etp_1[, 2]) / 1.5,
 mean = mean(etp_1[, 2]),
 phi = 0.9
)
etp_2[, 2] <- etp_2[, 2] + wr_coloured_noise(
 nrow(etp_2),
 sd = sd(etp_2[, 2]) / 1.5,
 mean = mean(etp_2[, 2]),
 phi = 0.9
)

Xetp <- analyse_Xsuperlet(
 data_1 = etp_1,
 data_2  = etp_2,
 upperPeriod = 256,
 lowerPeriod = 2,
 Nf = 32,
 c1 = 3,
 o = c(1, 10),
 mult = TRUE,
 order_scaling = "log2",
 order_alpha = 1,
 verbose = FALSE
)
```

`analyse_Xwavelet` *Conduct the cross wavelet transform on a time series or signal*

Description

Compute the cross wavelet transform using a complex Morlet wavelet based on the "analyze.coherency" function of the WaveletComp package.

Usage

```
analyse_Xwavelet(
  data_1,
```

```
  data_2,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = FALSE,
  omega_nr = 4,
  wavelet = "morlet",
  wavelet_order = NULL
)

analyze_Xwavelet(
  data_1,
  data_2,
  dj = 1/100,
  lowerPeriod = 2,
  upperPeriod = 1024,
  verbose = FALSE,
  omega_nr = 4,
  wavelet = "morlet",
  wavelet_order = NULL
)
```

Arguments

data_1	First input data set as a matrix or data frame. The first column must represent depth or time, and the second column the signal or proxy record.
data_2	Second input data as a matrix or data frame. The first column must represent depth or time, and the second column the signal or proxy record.
dj	Spacing between successive scales. The CWT analyses analyses the signal using successive periods which increase by the power of 2 (e.g.2^0=1,2^1=2,2^2=4,2^3=8,2^4=16). To have more resolution in-between these steps the dj parameter exists, the dj parameter specifies how many extra steps/spacing in-between the power of 2 scaled CWT is added. The amount of steps is 1/x with a higher x indicating a smaller spacing. Increasing the increases the computational time of the CWT
lowerPeriod	Minimum period to be analysed. Controls the highest analysed frequency.
upperPeriod	Maximum period to be analysed. Controls the lowest analysed frequency.
verbose	Logical. If TRUE, print progress and interpolation information.
omega_nr	. omega number: central angular frequency of the Morlet wavelet
wavelet	Mother wavelet, given as a name (PyWavelets style names with the order appended, such as "db4", "sym8", "cgau3", "gaus2" or "cmor1.5-1", also work) or as a function psi_hat(omega) that returns the Fourier transform of a user defined mother wavelet. Analytic (complex) wavelets, which give phase and amplitude and can be synchrosqueezed: "morlet" (default; aliases "gabor", "amor"), "morlet0" (exactly zero mean Morlet, admissible for a small omega_nr; alias "morlex"), "paul" (alias "cauchy"), "morse" (generalised Morse wavelet of Olhede and Walden, 2002; aliases "gmw", "airy"), "bump", "cmor" (complex Morlet with bandwidth and centre frequency as in Matlab/PyWavelets),

"cgau" (analytic Gaussian derivative; "cmhat" is the complex Mexican hat), "shannon" (complex Shannon, box spectrum), "fbsp" (frequency B-spline), "meyer", "loggabor" (alias "lognormal"), "gammatone" (asymmetric in time: sharp onset, slow decay) and "hann" (raised cosine band). Real wavelets, which give a real transform (no phase or envelope): "dog" (derivative of Gaussian; "mexh", "ricker", "marr" and "sombrero" are the Mexican hat, order 2), "morl" (real Morlet), "hhat" (Hermitian hat), "poisson", and the orthogonal wavelets "haar", "db" (Daubechies), "sym" (Symlets) and "battle" (Battle-Lemarie spline wavelets), which are computed in the frequency domain from their filters. The period axis of the orthogonal wavelets is approximate because their spectra are broad and not symmetric. A narrow wavelet (Paul, low-order DOG, Shannon, Haar) resolves position well, a wide one (Morse with a large beta, Morlet with a high omega_nr, bump) resolves period well.

`wavelet_order` Parameter(s) of the wavelet: paul: order m (default 4); dog: derivative m (default 2); morse: c(gamma, beta) or beta alone with gamma = 3 (default c(3, 20)); bump: c(mu, sigma) (default c(5, 0.6)); cmor: c(bandwidth, centre frequency) (default c(1.5, 1)); cgau: derivative m (default 4); shannon: c(bandwidth, centre frequency) (default c(2, 1.5)); fbsp: c(m, bandwidth, centre frequency) (default c(2, 2, 1.5)); loggabor: c(centre angular frequency, log-width) (default c(6, 0.25)); meyer and hhat have no parameter; morlet0: omega_0 (default omega_nr); morl: omega_0 (default 5); poisson: order n (default 2); gammatone: c(n, centre angular frequency, bandwidth) (default c(4, 6, 1.5)); hann: c(centre angular frequency, half width) (default c(6, 3)); db: number of vanishing moments 1 to 20 (default 4, "haar" = "db1"); sym: 2 to 20 (default 8); battle: spline order 1, 3, 5 or 7 (default 3). For the Morlet wavelet use omega_nr.

Value

A list with class "analyze.Xwavelet" containing Wave: complex wavelet coefficients Power: time frequency power spectrum dt: sampling interval after interpolation Phase: instantaneous phase of the signal dj: number of frequencies Power.avg: average spectral power Period: physical periods corresponding to frequencies nc: number of columns in the input data nr: number of frequency levels axis.1: x axis values (time or depth) axis.2: y axis values (log2 scaled periods) c1: base number of wavelet cycles o: numeric vector of length two defining the minimum and maximum superlet order. If NULL, all frequencies are analysed using order one. x1: interpolated x values first data set y1: interpolated signal values first data set x2: interpolated x values second data set y2: interpolated signal values second data set

Author(s)

The "analyse_Xsuperlet" that generates the input for the plotting function is based on the matlab code in Moca et al. (2021) and the the "analyze.coherency" function of the 'WaveletComp' R package

References

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., & Mureșan, R. C. (2021). Time-frequency super-resolution with superlets. Nature Communications, 12(1), 337. doi:10.1038/s41467020205399

Examples

```
#Example 1. A cross superlet of two etp solutions with noise overprint
etp_1 <- astrochron::etp(
 tmin = 0,
 tmax = 500,
 dt = 2,
 eWt = 1.5,
 oWt = 0.75,
 pWt = 1,
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
 verbose = FALSE
)

etp_2 <- astrochron::etp(
 tmin = 0,
 tmax = 500,
 dt = 2,
 eWt = 1,
 oWt = 0.5,
 pWt = 1.5,
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
 verbose = FALSE
)

etp_1[, 2] <- etp_1[, 2] + wr_coloured_noise(
 nrow(etp_1),
 sd = sd(etp_1[, 2]) / 1.5,
 mean = mean(etp_1[, 2]),
 phi = 0.9
)
etp_2[, 2] <- etp_2[, 2] + wr_coloured_noise(
 nrow(etp_2),
 sd = sd(etp_2[, 2]) / 1.5,
 mean = mean(etp_2[, 2]),
 phi = 0.9
)

Xetp <- analyse_Xwavelet(
 data_1 = etp_1,
 data_2  = etp_2,
 upperPeriod = 512,
     dj = 1/20,
 lowerPeriod = 4,
 verbose = FALSE,
 omega_nr = 8
)
```

anchor_points_Bisciaro_al *XRF records of the Bisciaro Fm*

Description

data set consist of the tie points between the Bisciaro_al record of Arts (2014) and the la2011 solution of laskar et al., (20111)

Details

The data set is a matrix with the 4 columns. The first column is the depth/time of the al proxy record tie-points. The second column is the time value of the la2011 astronomical solution tie-points. The third column is the Al value of the a; tie-point. The fourth column is the eccentricity value of the la2011 astronomical solution tie-point.

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

Laskar, J., M. Gastineau, J. B. Delisle, A. Farrés, and A. Fienga (2011b), Strong chaos induced by close encounters with Ceres and Vesta, Astron. Astrophys., 532, L4,<doi:10.1051/0004-6361/201117504>

anchor_points_grey *Example anchor points for the grey scale data set of Zeeden et al., (2013)*

Description

An example of anchor points generated using `astro_anchor` function
The anchor points were generated for the `grey` grey data set of
Zeeden et al. (2013) and anchored to the `astrosignal_example`
astronomical solution which is a pre-generated ETP (eccentricity-tilt-precession)
solution(p-0.5t based on the la2004 solution) based on Laskar et al., (20004)
astronomical solution.

Details

Column 1: depth proxy record
Column 2: time astronomical solution
Column 3: y-scale value proxy record
Column 4: y-scale value astronomical solution

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>

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. <doi:10.1051/0004-6361:20041335>

`anchor2time` *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

```
anchor2time(
  anchor_points = NULL,
  data = NULL,
  genplot = FALSE,
  keep_editable = FALSE
)
```

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=FALSE` then 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 set to TRUE to allow for anchoring using the GUI
`keep_editable`	Keep option to add extra features after plotting `Default=FALSE`

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 data set. Plot 2: the depth time plot. Plot 3: the data set in the time domain.

Examples

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

grey_time <- anchor2time(anchor_points=age_model_zeeden,
data=grey,
genplot=FALSE,
keep_editable=FALSE)
```

`as_wr_track` *Build a track (depth, sedimentation rate, time) from a tracked period curve, a rate curve or an age model*

Description

The tracking functions return `wr_track` objects; this builds one from the curves older workflows produce, so that they can be given to `analyse_dechirped` and `check_agemodel`: a tracked period curve (`track_period_wavelet`: depth and the period of one astronomical cycle in depth units), a sedimentation-rate curve (e.g. from an EHA/sedrate workflow) or an age-depth model.

Usage

```
as_wr_track(
  x,
  type = c("period", "sedrate", "age"),
  cycle = 405,
  rate_factor = 1,
  step = NULL
)
```

Arguments

`x`	Data frame or matrix: depth in the first column and, in the second, the tracked period (depth units), the sedimentation rate (depth units per kyr, times `rate_factor`) or the age (kyr, times `rate_factor`).
`type`	`"period"` (default), `"sedrate"` or `"age"`.
`cycle`	Period (kyr) of the tracked cycle for `type = "period"` `Default=405`.
`rate_factor`	Multiplier that brings the second column to depth units per kyr (rate) or to kyr (age): `0.01` for a rate in cm/kyr with depth in m, `1000` for an age in Ma `Default=1`.
`step`	Depth step of the output; `NULL` keeps the input positions `Default=NULL`.

Value

A `wr_track` data frame with `depth`, `sedrate` and `time` (kyr from the top of the curve, increasing with depth).

Examples

```
curve <- data.frame(depth = 0:100, period = 12 + 3 * sin(0:100 / 10))
tr <- as_wr_track(curve, "period", cycle = 405)
head(tr)
```

`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
- La2004 Obliquity solution available via the getLaskar function
- La2004 Precession solution available via the getLaskar function
- La2010a Eccentricity solution available via the getLaskarfunction
- La2010a Obliquity solution
- La2010a Precession solution
- La2010b Eccentricity solution available via the getLaskar function
- La2010b Obliquity solution
- La2010b Precession solution
- La2010c Eccentricity solution available via the getLaskar function
- La2010c Obliquity solution
- La2010c Precession solution
- La2010d Eccentricity solution available via the getLaskar function
- La2010d Obliquity solution
- La2010d Precession solution
- La2011 Eccentricity solution available via the getLaskar function
- 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' R 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,
  pts = 3,
  verbose = FALSE,
  time_dir = TRUE,
  genplot = FALSE
)
```

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	Specifies the upper period of the astronomical cycle which is extracted from an astronomical solution. The astro_period_up is a factor with which the astronomical component is multiplied by. Default=1.2
astro_period_down	Specified the lower period of the astronomical cycle which is extracted from an astronomical solution. The astro_period_down 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	Specifies the upper period of the astronomical cycle to be extracted from the proxy record. The upper bound value is a factor with which the (proxy_period_cycle) value is multiplied by. Default=1.2.
proxy_period_down	Specifies the upper period of the astronomical cycle to be extracted from the proxy record. The lower bound value is a factor with which the astronomical component (proxy_period_cycle) value is multiplied by. Default=0.8.
proxy_period_cycle	Period in kyr of the astronomical cycle/component which is to be extracted from the proxy record.

pts	The pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm `Default=3`
verbose	print text `Default=FALSE` set verbose to TRUE to allow for anchoring using text feedback commands
time_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then time_dir should be set to TRUE if time decreases with depth/time values (eg stratigraphic logs where 0m is the bottom of the section) then time_dir should be set to FALSE `time_dir=TRUE`
genplot	Generate plot `Default="FALSE"`

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. If genplot is set to true then at plot of the of the achored points will be plotted

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. <doi:10.1051/0004-6361:20041335>

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 <doi:10.1051/0004-6361/201116836>

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. <doi:10.1051/0004-6361/201117504>

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>

Zeebe, R. E. and Lourens, L. J. Geologically constrained astronomical solutions for the Cenozoic era, Earth and Planetary Science Letters, 2022 <doi:10.1016/j.epsl.2022.117595>

Richard E. Zeebe Lucas J. Lourens ,Solar System chaos and the Paleocene–Eocene boundary age constrained by geology and astronomy.Science365,926-929(2019) <doi:10.1126/science.aax0612>

Zeebe, Richard E. "Numerical solutions for the orbital motion of the Solar System over the past 100 Myr: limits and new results." The Astronomical Journal 154, no. 5 (2017): 193. <doi:10.3847/1538-3881/aa8cce>

Stephen R. Meyers,Cyclostratigraphy and the problem of astrochronologic testing, Earth-Science Reviews,Volume 190,2019,Pages 190-223,ISSN 0012-8252 <doi:10.1016/j.earscirev.2018.11.015>

Examples

```
if (interactive()) { # needs mouse input (identify())
# Use the grey_track example tracking points to anchor the grey scale data set
# of Zeeden et al., (2013) to the p-0.5t la2004 solution

grey_wt <-
 analyse_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
   omega_nr = 8
 )
#Use the pre-tracked grey_track curve which traced the precession cycle
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,pts=3)

#use the astrosignal_example to tune to which is an \cr
# ETP solution (p-0.5t la2004 solution)
astrosignal_example <- na.omit(astrosignal_example)
astrosignal_example[,2] <- -1*astrosignal_example[,2]
astrosignal <- as.data.frame(astrosignal_example)

#anchor the synthetic insolation curve extracted from the grey scale record to the insolation curve.

anchor_pts <- 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,
pts=3,
verbose=FALSE, #set verbose to TRUE to allow for anchoring using text feedback commands
genplot=FALSE
)}
```

`astro_anchor_app` *Interactive Shiny app to tie a proxy record to an astronomical solution (tune live while tying)*

Description

A browser-based version of `astro_anchor_interactive`. One window shows three linked rows: the proxy record, the astronomical solution and, underneath, the record tuned to the current tie points together with a cycle extracted from it (and optionally the astronomical solution and the sedimentation rate). The tuned row is updated while tie points are made, moved or removed, so the effect of every change on the fit can be judged on the go. Tie points are numbered and colour coded in all rows and joined by lines between the two records.

Usage

```
astro_anchor_app(
  astro_solution = NULL,
  proxy_signal = NULL,
  proxy_min_or_max = "max",
  astrosolution_min_or_max = "max",
  pts = 3,
  ties = NULL,
  time_dir = TRUE,
  peaks_in_view = 12,
  tie_on = c("raw", "cycle"),
  proxy_flow = NULL,
  proxy_fhigh = NULL,
  proxy_roll = 3,
  inspect_flow = NULL,
  inspect_fhigh = NULL,
  inspect_roll = 3,
  sedrate_factor = 1,
  launch_browser = TRUE
)
```

Arguments

`astro_solution`	Astronomical solution (matrix or data frame, first column time/age, second column value), or a named list of such solutions to choose from in the app. In the app the target can also be switched to a metronome (a cosine with a period, phase and time interval you set), or to a Laskar solution (eccentricity, obliquity, climatic precession or ETP of La2004, La2010a-d, La2011 eccentricity, or 65 N insolation) that is downloaded on request with `getLaskar` when you press the download button.
`proxy_signal`	Proxy record (first column depth/time, second column value).

`proxy_min_or_max`
: Peaks to offer when snapping on the proxy record: maxima (`"max"`) or minima (`"min"`) `Default="max"`.

`astrosolution_min_or_max`
: Peaks to offer on the astronomical solution: maxima or minima `Default="max"`.

`pts`
: Number of points to the left and right that a point has to exceed to count as a peak (see `max_detect`) `Default=3`.

`ties`
: Optional existing tie points to start from: a data frame with the columns `data_x` and `insolation_x` (or the output of `astro_anchor`). The astronomical positions are moved to the nearest maximum/minimum, the proxy positions are used as given.

`time_dir`
: `TRUE` if time increases with increasing depth/time values of the proxy record, `FALSE` if it decreases `Default=TRUE`.

`peaks_in_view`
: Approximate number of peaks shown in a window at the start `Default=12`.

`tie_on`
: Start with the tie points on the raw proxy record (`"raw"`) or on a cycle extracted from it (`"cycle"`) `Default="raw"`. Can be changed in the app.

`proxy_flow`, `proxy_fhigh`
: Low and high cut-off frequencies (cycles per unit of the proxy record) of the Taner filter that extracts the proxy cycle `Default=NULL`: the strongest period of the wavelet spectrum -/+ 25 percent. They can be changed in the app.

`proxy_roll`
: Roll-off of that filter as the exponent `x` of `roll = 10^x` (see `taner`) `Default=3`.

`inspect_flow`, `inspect_fhigh`
: Low and high cut-off frequencies (cycles per time unit of the astronomical solution) of the Taner filter that extracts the cycle from the tuned record `Default=NULL`: the median spacing of the peaks of the astronomical solution -/+ 25 percent.

`inspect_roll`
: Roll-off exponent of that filter `Default=3`.

`sedrate_factor`
: Factor applied to the sedimentation rate (depth units per time unit) that is shown, for example 100 to go from metres/kyr to cm/kyr `Default=1`.

`launch_browser`
: Open the app in the default web browser `Default=TRUE`; `FALSE` prints the local address (or uses the RStudio viewer when R is run there).

Details

What is tied. Tie points on the astronomical solution always sit on a maximum or minimum (see `astrosolution_min_or_max`). On the proxy side they can be placed on the *raw* record or on a cycle that is extracted from it with a Taner filter (`taner`; choose *Cycle extracted from the proxy* and give `flow`, `fhigh` and the exponent of the roll-off, `roll = 10^x`). The tuning is always applied to the raw record. Proxy tie points do not have to lie on a maximum or minimum: with *Snap to peaks* switched off they are placed exactly where you click, and every proxy tie point can be moved left and right with the buttons (fine and coarse steps), by typing its position, by typing its offset from the nearest maximum/minimum, or by dragging. The cycle shown in the tuned record is extracted with the same kind of Taner filter.

Tie a point: click one record (red) and then the other (in any order). **Select** a tie point by clicking it. **Move** it by clicking a free position of one of the records afterwards, by dragging over the record from the tie point to the new position, by the arrow buttons, or by editing the numbers. **Untie** it by

double clicking it or with the *Untie selected* button. Tie points in the wrong order (the tuned time would run backwards) are marked with a black cross; the tuned row waits until they are fixed.

Target and windows. The card *Astronomical solution* selects the target (a provided solution, a metronome, or a Laskar solution), the time interval it spans and whether maxima or minima are tied. Two sliders above the plot show the visible window (from, to) of the proxy record and of the astronomical solution; type new numbers to set them. With the mouse, the *wheel* zooms the record under the pointer around the pointer and *Shift + drag* (or dragging with the right or middle mouse button) moves it from left to right. *Sync* brings the astronomical window to the partners of the tie points in the proxy window, *Show both records in full* resets both.

The sidebar also has buttons to undo changes and to finish (*Done*). Closing the browser tab also finishes and returns the tie points.

The app runs on the local computer only (a Shiny app started with `runApp`); nothing is sent to other servers. The package **shiny** has to be installed.

The result has the same four columns as the output of `astro_anchor` and can be used by `plot_astro_anchor` and `anchor2time`. `data_y` and `insolation_y` are the values of the records at the tie positions.

Value

A data frame with the tie points sorted by depth/time: `data_x`, `data_y` (proxy), `insolation_x`, `insolation_y` (astronomical solution). Without an interactive session and with `ties` the starting tie points are returned without starting the app (those on the astronomical solution moved to the nearest peak).

See Also

`astro_anchor_interactive`, `astro_anchor`, `plot_astro_anchor`, `anchor2time`

Examples

```
# Without an interactive session (as here) the starting tie points are returned as they are.
astrosignal <- na.omit(astrosignal_example)
tie_start <- data.frame(data_x = c(155, 160, 165), insolation_x = c(5040, 5400, 5560))
proxy <- grey[grey[, 1] < 175, ]
astro_anchor_app(astro_solution = astrosignal, proxy_signal = proxy, ties = tie_start)
# in an interactive session, leave out 'ties' to start the app and tie the records with the mouse
```

`astro_anchor_interactive` *Interactively tie a proxy record to an astronomical solution (random order, tie and untie, inspect)*

Description

A more flexible alternative to astro_anchor. The proxy record (top) and the astronomical solution (bottom) are shown in one window with their peaks (or troughs). Tie points can be set, removed and changed in any order: click a peak of the proxy record and then a peak of the astronomical solution to tie them (or first the astronomical peak and then the proxy peak); click a tied peak again to untie it. The windows can be zoomed and moved with the buttons above the plot, so that long records can be tied piece by piece. Press *Tune and inspect* at any moment: a second window shows the record tuned to the current tie points together with a cycle extracted from it, the astronomical solution and the sedimentation rate between the tie points, so that the result can be judged while tie points are still being added or removed.

Usage

```
astro_anchor_interactive(
  astro_solution = NULL,
  proxy_signal = NULL,
  proxy_min_or_max = "max",
  astrosolution_min_or_max = "max",
  pts = 3,
  ties = NULL,
  time_dir = TRUE,
  peaks_in_view = 12,
  inspect_period = NULL,
  inspect_band = 0.25,
  sedrate_factor = 1,
  inspect_wavelet = FALSE,
  locator_fun = NULL
)
```

Arguments

astro_solution	Astronomical solution (matrix or data frame, first column time/age, second column value).
proxy_signal	Proxy record (first column depth/time, second column value). Use extract_signal first if a single cycle of the record should be tied.
proxy_min_or_max	Tie maxima ("max") or minima ("min") of the proxy record Default="max".
astrosolution_min_or_max	Tie maxima or minima of the astronomical solution Default="max".
pts	Number of points to the left and right that a point has to exceed to count as a peak (see max_detect) Default=3.
ties	Optional existing tie points to start from: a data frame with the columns data_x and insolation_x (or the output of astro_anchor). They are snapped to the nearest peaks.
time_dir	TRUE if time increases with increasing depth/time values of the proxy record, FALSE if it decreases (the proxy axis is then reversed) Default=TRUE.
peaks_in_view	Approximate number of peaks shown in a window at the start Default=12.

`inspect_period` Period (in the time units of the astronomical solution) of the cycle that is extracted from the tuned record in the inspection window `Default=NULL`: the median spacing of the peaks of the astronomical solution.

`inspect_band` Relative half width of the extracted band, `0.25` gives 0.75 to 1.25 times the period `Default=0.25`.

`sedrate_factor` Factor applied to the sedimentation rate (depth units per time unit) in the inspection window, for example 100 to go from metres/kyr to cm/kyr `Default=1`.

`inspect_wavelet`
Also show the wavelet scalogram of the tuned record in the inspection window `Default=FALSE`.

`locator_fun` Function that returns the next mouse click in normalised device coordinates, like `locator(1)`. `Default=NULL`: `locator`. Another function (for example one that replays a list of clicks) allows scripted use and testing. When no interactive device is available and `ties` is given, the snapped tie points are returned without opening a window.

Details

Buttons (click them with the left mouse button): *Tune and inspect* (open or update the inspection window), *Undo* (remove the last change), *Clear* (remove all tie points), *Zoom +*/*Zoom -* (both windows), *P <*, *P >* (move the proxy window), *A <*, *A >* (move the astronomical window), *Fit* (show the whole records), *Sync* (move the astronomical window to the partners of the tie points in the proxy window, or the other way round if there are none), *Wavelet* (switch the wavelet scalogram of the tuned record in the inspection window on or off) and *Done*. A right click also finishes. Tie points that are in the wrong order (the tuned time would run backwards) are drawn in orange, and the tuning refuses to run until they are fixed.

The result has the same four columns as the output of `astro_anchor` and can be used directly by `plot_astro_anchor` and `anchor2time`.

Value

A data frame with the tie points sorted by depth/time: `data_x`, `data_y` (proxy), `insolation_x`, `insolation_y` (astronomical solution).

See Also

`astro_anchor`, `plot_astro_anchor`, `anchor2time`

Examples

```
# Tie points can be given as a starting point and are snapped to the nearest peaks;
# without an interactive graphics device (as here) they are returned as they are.
astrosignal <- na.omit(astrosignal_example)
tie_start <- data.frame(data_x = c(155, 160, 165), insolation_x = c(5040, 5400, 5560))
proxy <- grey[grey[, 1] < 175, ]
ties <- astro_anchor_interactive(astro_solution = astrosignal,
                                 proxy_signal = proxy,
                                 ties = tie_start)
ties
```

```
# in an interactive session, leave out 'ties' to tie the peaks with the mouse
```

`astro_ctest` *Consistency-test inversion (the estimator used in the paper)*

Description

After Zeeden et al. (2023): draw a precession constant and a set of secular frequencies, predict every term, and keep the draw only if all measured windows are satisfied at once. Two details matter. First, the secular frequencies are sampled as periods with the Laskar et al. (2011) uncertainties scaled by sqrt(age / 100 Myr). Second, because a record tuned to a nominal 405 kyr inherits the error in the true g2-g5 period, every measured period is rescaled by (sampled 405-kyr period) / 405 before it is compared with the prediction; this is what allows the tuned time scale itself to stretch slightly.

Usage

```
astro_ctest(
  windows,
  age_ma,
  nsim = 3000,
  k_range = c(10, 34),
  anchor_kyr = 405,
  seed = 1,
  block = 2e+05,
  max_attempts = 5e+07,
  verbose = TRUE
)
```

Arguments

`windows`	named list, term -> c(low, high) in kyr, as in astro_invert()
`age_ma`	age of the record, or c(min, max)
`nsim`	number of accepted draws to collect
`k_range`	prior range for the precession constant, kyr
`anchor_kyr`	the period assumed when the record was tuned
`seed`	Random seed for reproducibility.
`block`	Number of random draws per block.
`max_attempts`	Maximum number of draws before the search is aborted.
`verbose`	Print progress messages.

`astro_from_peaks` *Astronomical parameters from spectral peaks*

Description

Wraps ctest() and ctest_earthmoon(): give the wavelength ranges (m) of the cycles you identified in a depth-domain spectrum and an approximate age, and get back the secular frequencies g2-g5 and s3, the precession constant k, the Earth-Moon distance and the length of day, with uncertainties.

Usage

```
astro_from_peaks(
  age,
  e405,
  e125,
  e95,
  obl = NULL,
  prec_long = NULL,
  prec_mid = NULL,
  prec_short = NULL,
  age_tol = 5,
  nsim = 2000,
  ptest = c(10000, 28000),
  mode = c("auto", "PT", "T", "P"),
  sampleorb = "U",
  method = c("angular_momentum", "astrogeo22"),
  s3_uncertainty = c("published", "scaled"),
  seed = NULL,
  verbose = TRUE,
  data_dir = NULL
)
```

Arguments

`age`	approximate age in Ma: c(min, max), or a single value (then 'age_tol' Ma is used on either side)
`e405`, `e125`, `e95`	c(min, max) wavelength (m) of the 405-kyr, ~125-kyr and ~95-kyr eccentricity cycles (all three are required)
`obl`	c(min, max) wavelength (m) of the obliquity cycle, if identified
`prec_long`, `prec_mid`, `prec_short`	c(min, max) wavelength (m) of the precession cycles, longest to shortest (k+g5; k+g2; k+g4/k+g3). Give what you have; all three are needed for the precession test.
`age_tol`	half-width (Ma) used when 'age' is a single value
`nsim`, `ptest`, `sampleorb`, `s3_uncertainty`, `seed`, `verbose`	passed to ctest()

mode	"auto" picks PT / T / P from what you supplied
method	how k is converted to an Earth-Moon state: "angular_momentum" or "astrogeo22"
data_dir	folder with the AstroGeo22 table

Value

object of class 'astro_from_peaks': '$frequencies' (g, s, k in both arcsec/yr and yr), '$earthmoon' (distance, day length, months, days per year), '$periods' (the astronomical periods implied), '$reference' (the AstroGeo22 model at the same age) and the underlying '$ctest'

---

astro_from_periods *Astronomical parameters from a tuned record (cycles already in kyr)*

Description

Same output as astro_from_peaks(), but the input is cycle periods in kyr and the time scale is set by the tuning target rather than by a sampled sedimentation rate. Give only the terms you actually resolved.

Usage

```
astro_from_periods(
  cycles,
  age,
  anchor = 405,
  anchor_term = "e405",
  nsim = 2000,
  ptest = c(10000, 28000),
  method = c("angular_momentum", "astrogeo22"),
  s3_uncertainty = c("published", "scaled"),
  seed = NULL,
  verbose = TRUE,
  data_dir = NULL,
  ...
)
```

Arguments

cycles	named list of c(min, max) periods in kyr (see astro_mc())
age	approximate age in Ma
anchor	period assumed for the tuning target, kyr (405 by default)
anchor_term	the term you tuned to
nsim, ptest, s3_uncertainty, seed, verbose	passed to astro_mc()

method	"angular_momentum" or "astrogeo22"
data_dir	folder with the AstroGeo22 table
...	further arguments to astro_mc()

Value

object of class ‘astro_from_peaks‘ (see astro_from_peaks())

`astro_invert` *Joint inversion of several measured terms, with Monte Carlo uncertainty*

Description

Joint inversion of several measured terms, with Monte Carlo uncertainty

Usage

```
astro_invert(
  windows,
  nsim = 5000,
  obl_deg = AC$obl0_deg,
  k_range = c(10, 34),
  seed = 1,
  gs = LA2004
)
```

Arguments

windows	named list: term -> c(low, high) measured period range in kyr. Names must be terms of K_TERMS(), e.g. list("k+s3" = c(29.0, 30.4)).
nsim	number of Monte Carlo draws
obl_deg	obliquity used in the precession relation
k_range	prior range for k, kyr (draws outside it are rejected)
seed	Random seed for reproducibility.
gs	Vector of fundamental g frequencies (‘NULL‘ = LA2004 values).

Value

list(k = summary of k, state = summary of the Earth-Moon state, draws = the raw draws)

`astro_mc` *Monte Carlo consistency test for tuned (time-domain) or depth-domain cycles*

Description

Monte Carlo consistency test for tuned (time-domain) or depth-domain cycles

Usage

```
astro_mc(
  constraints,
  age,
  domain = c("time", "depth"),
  anchor = 405,
  anchor_term = "e405",
  scale_range = NULL,
  g4g3_range_kyr = NULL,
  s4s3_range_kyr = NULL,
  age_tol = 5,
  nsim = 2000,
  ptest = c(10000, 28000),
  sampleorb = c("U", "N"),
  s3_uncertainty = c("published", "scaled"),
  seed = NULL,
  block = 1e+05,
  max_attempts = 5e+07,
  verbose = TRUE
)
```

Arguments

`constraints`	named list of c(min, max) ranges. Names are astronomical terms: "e405", "e125" (= e130 and e124), "e95" (= e99 and e95), "k+s3" (or "obl"), "k+g5", "k+g2", "k+g4", "k+g3", "prec_short" (= k+g4 and k+g3). Several terms in one band: "k+g4|k+g3" (both must fit). A peak you cannot assign: "any:k+g5|k+g2" (at least one must fit). Units: kyr when ‘domain = "time"‘, metres when ‘domain = "depth"‘.
`age`	approximate age in Ma, c(min, max) or a single value with ‘age_tol‘
`domain`	"time" for a tuned record, "depth" for a depth-domain spectrum
`anchor`	the period you assumed for the tuning target, in kyr (default 405 for a 405-kyr tuned record); NULL means the time scale is sampled freely within ‘scale_range‘
`anchor_term`	the term you tuned to (default "e405" = g2-g5)
`scale_range`	sedimentation rate range (m/kyr) when ‘domain = "depth"‘, or the allowed stretch of the time scale when ‘anchor‘ is NULL

g4g3_range_kyr, s4s3_range_kyr
: accepted for a common interface

age_tol, nsim, ptest, sampleorb, s3_uncertainty, seed, block, max_attempts, verbose
: as in ctest()

Value

object of class c("astro_mc", "ctest"), so print(), plot() and ctest_earthmoon() work on it

---

astro_obliquity *Obliquity at a given age, from AstroGeo22 if available, else today's value*

Description

Obliquity at a given age, from AstroGeo22 if available, else today's value

Usage

```
astro_obliquity(age_ma)
```

Arguments

age_ma
: Age(s) in Ma.

---

astro_periods_tracking *Astronomical periods and weights for tracking, for any age*

Description

Returns the main eccentricity, obliquity and precession periods with weights proportional to their typical amplitude, for use in track_astro_hmm. The obliquity and precession periods follow from the precession constant k at the given age (AstroGeo22, via astro_terms) or from a k given directly (e.g. estimated with test_tuned), so that the periods fit deep-time records such as the Silurian and Devonian. The eccentricity periods are the La2004 values; the 405 kyr period is stable through the Phanerozoic, the short eccentricity periods are less certain before about 60 Ma.

Usage

```
astro_periods_tracking(
  age_ma = 0,
  k_yr = NULL,
  include = c("eccentricity", "obliquity", "precession")
)
```

Arguments

age_ma	Age in Ma Default=0.
k_yr	Optional precession constant (arcsec/yr) instead of the AstroGeo22 value for age_ma.
include	Which families to include Default=c("eccentricity", "obliquity", "precession"). When only "eccentricity" is included, the 405-kyr cycle gets weight 1 (instead of 0.5): in records in which only the eccentricity is resolved, the stable 405-kyr cycle carries the age model.

Value

A data frame with term, family, period (kyr) and weight.

Examples

```
astro_periods_tracking(0)
astro_periods_tracking(420)        # Silurian
```

---

astro_term_search *Which term could a measured period be?*

Description

Enumerates single terms and low-order combinations and returns those whose period matches, so an unidentified peak can be checked against the theory.

Usage

```
astro_term_search(
  period_kyr,
  age_ma = 0,
  tol = 0.03,
  max_order = 2,
  k_yr = NULL,
  data_dir = NULL
)
```

Arguments

period_kyr	the measured period
age_ma	age, for the precession constant used in the k-bearing terms
tol	relative tolerance (0.03 = 3%)
max_order	1 = single terms only, 2 = also sums/differences of two
k_yr	Precession constant in arcsec/yr (‘NULL‘ = taken from ‘astrogeo22(age_ma)$p_arcsec‘).
data_dir	Directory with the data files (‘NULL‘ = the data installed with the package).

Value

data.frame of candidate terms sorted by |offset from the measurement|

`astro_terms` *The menu of named astronomical terms at a given age*

Description

The menu of named astronomical terms at a given age

Usage

```
astro_terms(age_ma = 0, k_yr = NULL, data_dir = NULL)
```

Arguments

`age_ma`	age in Ma; the precession constant comes from AstroGeo22
`k_yr`	optional: your own precession constant (yr) instead of the model
`data_dir`	Directory with the data files ('NULL' = the data installed with the package).

Value

data.frame from the longest to the shortest period

`astrogeo22` *Earth-Moon state from the AstroGeo22 model at a given age*

Description

Earth-Moon state from the AstroGeo22 model at a given age

Usage

```
astrogeo22(age_ma, interpolate = TRUE, data_dir = NULL)
```

Arguments

`age_ma`	age in Ma (0 to 4429); vectorised
`interpolate`	TRUE (linear between the 1-Myr rows) or FALSE (nearest, which is what calculate_orbit9.m does)
`data_dir`	Directory with the data files ('NULL' = the data installed with the package).

Value

data.frame: age_ma, a_RE, a_km, lod_h, obliquity_deg, p_arcsec, p_kyr and the model's min/max envelope for each

`astroMCBchron` *Joint Bayesian age-depth model of radioisotopic dates and a floating astrochronology (astroMCBchron)*

Description

`astroMCBchron()` combines radioisotopic dates (e.g. CA-ID-TIMS U-Pb) with a floating astrochronology (the tracked period of an astronomical cycle, typically the 405-kyr eccentricity cycle, as obtained with `track_period_wavelet`, `completed_series` or `retrack_wt_MC`) in one joint posterior distribution. The floating astrochronology is not treated as exact: between each pair of dates its duration is multiplied by a stretch factor whose prior width is set by the tracking uncertainty (the spread between independently tracked proxies), and the duration of every hiatus is estimated together with the age model. The dates enter the likelihood only, so the astrochronology and the dates remain independent sources of information.

The model (Arts et al., in prep.):

- the record is divided into intervals bounded by the dated horizons and the top and base of the record. In interval *j* the floating duration (number of tracked cycles x `target_period`) is multiplied by a stretch factor *s_j*, with log *s_j* ~ N(0, tau_j^2 + tau0^2). tau_j is the standard error of the floating duration of the interval (between-proxy spread / sqrt(n proxies)) and tau0 a common additional slack with a half-normal prior (sd = `tau0_sd`). The two outer intervals, which are not bracketed by dates, are tied to their neighbours (sd = `outer_sd` in log units);
- each hiatus has a duration H with a normal prior (`hiatuses$mean`, `hiatuses$sd`), truncated at zero;
- the stratigraphic position of each date may vary uniformly within +/- `thickness`;
- the floating astrochronology is drawn from the realisations of the tracked period (one realisation per segment and MCMC state);
- the age of every position follows from the age of an anchor date plus the cumulative stretched floating durations and the hiatuses;
- the dates are screened in two stages: a robust Student-t stage flags dates that move more than `outlier_z` standard deviations; in the second stage the retained dates have a Gaussian likelihood and flagged dates only enter as a weak guard (`outlier_k` times their uncertainty);
- the within-interval uncertainty is added as a monotone gamma (Dirichlet) bridge between the dates (free gamma walks beyond the outermost dates), with a dispersion per cycle equal to the relative between-proxy spread of the cycle durations; the reported 2 sd is computed analytically (Rao-Blackwellised) from the posterior draws;
- optionally, every date is predicted from all other dates and the astrochronology (leave-one-out), which tests the consistency between dates and cycles.

With `bchron = TRUE`, a U-Pb-only Bayesian age-depth model with the same dates (`modBchron`, a pure-R implementation of modifiedBChron; Trayler et al., 2020; Haslett and Parnell, 2008) is run alongside, on the same prediction grid, so that both models and the reduction in uncertainty can be compared directly.

Usage

```
astroMCBchron(
  dates,
  tracks,
  hiatuses = NULL,
  position_type = c("depth", "height"),
  age_unit = c("Ma", "ka"),
  target_period = 405,
  proxy_tracks = NULL,
  tau = NULL,
  tau_method = c("se", "sd"),
  eps = NULL,
  tau0_sd = 0.03,
  outer_sd = 0.2,
  anchor = NULL,
  two_stage = TRUE,
  nu = 4,
  outlier_z = 3,
  outlier_k = 4,
  n_iter = 3e+05,
  thin = 50,
  burn_frac = 0.2,
  n_tracks = 1000,
  loo = TRUE,
  loo_iter = NULL,
  predict_positions = NULL,
  max_curves = 5000,
  n_bridge = NULL,
  date_loglik = NULL,
  bchron = FALSE,
  bchron_split = FALSE,
  bchron_args = list(),
  seed = NULL,
  verbose = TRUE,
  genplot = FALSE
)
```

Arguments

dates	data.frame with one row per dated horizon and columns `id`, `age`, `sd` (1 sigma, same unit as `age`), `position` and, optionally, `thickness` (half thickness of the stratigraphic position uncertainty; default 0). Ages are given in `age_unit`. Use the internal (analytical) uncertainty when all dates come from the same laboratory and tracer.
tracks	list with one element per continuous segment (number of hiatuses + 1), ordered from the youngest (top) to the oldest (base) segment. Each element is either (i) a list with `position` (vector) and `period` (matrix, positions x realisations, or a vector), (ii) the output of `retrack_wt_MC` style objects, i.e. a list whose

first element is a matrix with the position in its first column and whose second element is the matrix of period realisations, or (iii) a matrix/data.frame with position in the first column and one or more period realisations in the other columns. Periods are in position units (e.g. m) per `target_period`.

`hiatuses` optional data.frame with columns `position`, `mean` and `sd` (1 sigma; duration prior in kyr). Default `NULL` (no hiatus).

`position_type` "depth" (positions increase downwards, default) or "height" (positions increase upwards).

`age_unit` unit of `dates$age`, `dates$sd` and of the output: "Ma" (default) or "ka" (= kyr).

`target_period` period of the tracked cycle (kyr). Default 405.

`proxy_tracks` optional list (one element per segment) of named lists of single per-proxy tracks (data.frame or matrix with position and period). Used to compute tau (stretch prior) and eps (within-interval spread) from the between-proxy spread.

`tau` optional numeric vector (one value per interval between consecutive dates, log units, or a single value used for all intervals) that overrides the computed stretch-prior widths.

`tau_method` "se" (between-proxy sd / sqrt(n), default) or "sd" (between-proxy sd). Only used with `proxy_tracks`. Without `proxy_tracks`, tau is the sd of the log floating durations over the track realisations.

`eps` optional within-interval relative spread per cycle: a single number, or a data.frame with `position` and `eps`. Default: computed from `proxy_tracks` (between-proxy standard error per cycle) or, without them, from the spread of the track realisations.

`tau0_sd` sd of the half-normal prior on the common slack tau0 (log units). Default 0.03. Set to 0 to fix tau0 at 0.

`outer_sd` sd (log units) tying the stretch factors of the outer intervals to their neighbours. Default 0.2.

`anchor` id of the date used as anchor. Default: the most precise date.

`two_stage` logical; robust two-stage outlier screening (default TRUE).

`nu` degrees of freedom of the Student-t likelihood in stage 1. Default 4.

`outlier_z` flag a date when its stage-1 shift exceeds `outlier_z` x its sd.

`outlier_k` flagged dates enter stage 2 with `outlier_k` x their sd.

`n_iter` number of MCMC iterations (stage 2; stage 1 uses half). Default 3e5.

`thin` thinning interval. Default 50.

`burn_frac` fraction of iterations discarded as burn-in (and used for adapting the proposal widths). Default 0.2.

`n_tracks` number of track realisations drawn per segment. Default 1000.

`loo` logical; leave-one-out prediction of every date. Default TRUE.

`loo_iter` iterations per leave-one-out run. Default `n_iter / 3`.

`predict_positions`
positions at which the model is evaluated. Default: 501 equidistant positions spanning the dates and the tracks.

max_curves	maximum number of age-depth curves stored. Default 5000.
n_bridge	bridge realisations per posterior draw used for the percentiles (default: enough for >= 10,000 curves, at most 20).
date_loglik	optional list (one function per date, same order as dates) returning the log-likelihood of an age (in age_unit); replaces the Gaussian date likelihood (e.g. for Bayesian eruption-age estimates). Disables the two-stage screening.
bchron	logical; also run the U-Pb-only modBchron model with the same dates. Default FALSE.
bchron_split	logical; with hiatuses, run one modBchron model per segment (dates within the segment) and estimate the hiatus duration from the difference between the models at the hiatus (as in Cramer et al., 2026). Default FALSE (one model through all dates).
bchron_args	list of further arguments passed to modBchron (e.g. MC, burn, truncateUp, truncateDown).
seed	optional random seed.
verbose	logical; print progress. Default TRUE.
genplot	logical; plot the result. Default FALSE.

Value

An object of class astroMCBchron: a list with

model matrix of age-depth curves (rows = predictPositions, columns = realisations), in age_unit; the same layout as the model element of modBchron output.

predictPositions positions of the rows of model.

summary data.frame: position, mean, median, 2.5 and 97.5 percentiles and the analytical 2 sd (age_unit, 2 sd in kyr).

dates data.frame: date, posterior age and 2 sd, shift (kyr) and shift in sd of the date, outlier flag, leave-one-out prediction and z-score.

stretch matrix of posterior stretch factors (draws x intervals).

hiatus matrix of posterior hiatus durations (kyr).

tau, tau0, eps stretch-prior widths, posterior of the common slack and the within-interval spread profile.

stage1, loo stage-1 screening and leave-one-out tables.

bchron, comparison the modBchron model(s) and a comparison table of the 2 sd of both models (if bchron = TRUE).

posterior, acceptance, call, settings raw posterior states, acceptance rates and settings.

References

Haslett, J., Parnell, A., 2008. A simple monotone process with application to radiocarbon-dated depth chronologies. Journal of the Royal Statistical Society C 57, 399-418.

Trayler, R.B., Schmitz, M.D., Cuitino, J.I., Kohn, M.J., Bargo, M.S., Kay, R.F., Stromberg, C.A.E., Vizcaino, S.F., 2020. An improved approach to age-modeling in deep time: Implications for the Santa Cruz Formation, Argentina. GSA Bulletin 132, 233-244.

Cramer, B.D., Schmitz, M.D., Trayler, R.B., Stolfus, B.M., Peate, D.W., McAdams, N.E.B., Arts, M., Da Silva, A.-C., 2026. Quantifying uncertainty in the structure and duration of Silurian biogeochemical events. Scientific Reports.

Examples

```
## synthetic example: accumulation rate varying between 1 and 3 cm/kyr (so that interpolating
## between the dates is wrong), a 300-kyr hiatus at 60 m, five dates with 1 sigma = 80 kyr and a
## floating 405-kyr astrochronology (the tracked period in m, 40 realisations with 2 % noise)
set.seed(1)
rate <- function(z) 0.02 * (1 + 0.5 * sin(2 * pi * z / 40))                  # m/kyr
zz <- seq(0, 100, 0.05)
age_z <- 400000 + c(0, cumsum(diff(zz) / rate(zz[-1]))) + ifelse(zz > 60, 300, 0)  # kyr
true_age <- function(z) approx(zz, age_z, z)$y
z_dates <- c(10, 35, 55, 70, 95)
dates <- data.frame(id = paste0("ash", 1:5), position = z_dates,
                    age = true_age(z_dates) / 1000 + rnorm(5, 0, 0.08), sd = 0.08,
                    thickness = 0.5)
trk <- function(z) list(position = z, period = sapply(1:40, function(i)
  405 * rate(z) * exp(rnorm(1, 0, 0.02))))
fit <- astroMCBchron(dates, tracks = list(trk(seq(0, 60, 0.25)), trk(seq(60, 100, 0.25))),
                     hiatuses = data.frame(position = 60, mean = 280, sd = 60),
                     n_iter = 5000, loo = FALSE, bchron = TRUE,
                     bchron_args = list(MC = 1000, burn = 300))
fit
plot(fit)
# error against the truth: astrochronology + dates against the dates alone
s <- fit$summary
mean(abs(s$median - true_age(s$position) / 1000)) * 1000                     # kyr
```

`astrosignal_example` *An ETP astronomical solution*

Description

The `astrosignal_example` is a pre-generated ETP
(eccentricity-tilt-precession) (p-0.5t based on the la2004 solution)
the `astrosignal_example` can be used to anchor the `grey`
data set to an astronomical solution eg. `astrosignal_example`
using the `astro_anchor` function. the data set was generated using the
`etp` function of the 'astrochron' R package. The pre-generated ETP spans 5000 to 6000kyr.

Details

Column 1: time (kyr)
Column 2: ETP

Author(s)

Generated using the etp function of the astrochron-package.

References

Stephen R. Meyers,Cyclostratigraphy and the problem of astrochronologic testing, Earth-Science Reviews,Volume 190,2019,Pages 190-223,ISSN 0012-8252 <doi:10.1016/j.earscirev.2018.11.015>

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. <doi:10.1051/0004-6361:20041335>

`bayes_changepoint` *Bayesian change point analysis (Ruggieri, 2013)*

Description

The `bayes_changepoint` function applies the Bayesian change point algorithm of Ruggieri (2013) to a proxy record. The record is modelled as a sequence of segments, each described by its own linear regression (intercept + trend) with conjugate normal / scaled-inverse-chi-square priors. The algorithm

1. calculates the marginal likelihood (log form) of every possible sub-interval of the data (Eq. 2 of Ruggieri, 2013),
2. builds the partition function by forward recursion (dynamic programming, Eqs. 3-4),
3. calculates the exact posterior distribution of the number of change points (Eqs. 5-7) and
4. draws `n_sim` solutions (number of change points, their locations and the segment regressions) by stochastic backtrace.

The posterior probability of a change point at each data point, the posterior-mean regression model and its R^2 are returned as in Acycle. In addition the per-sample models are kept, from which pointwise credible intervals of the model are derived (these reflect the uncertainty in the number and location of the change points; each sampled segment uses its posterior-mean regression coefficients, as in Acycle).

Translated from Acycle v3.1 bayes_cpt.m, partition_fn.m, pick_k1.m and nCk.m (Mingsong Li, modified from the code of Eric Ruggieri); default hyperparameters as in the Acycle "Changepoint" menu (AC.m).

Usage

```
bayes_changepoint(
  data = NULL,
  k_max = 10,
  d_min = 1,
  k_0 = 0.01,
  v_0 = NULL,
  sig_0 = NULL,
```

```
  n_sim = 500,
  seed = NULL,
  ci = c(0.025, 0.975),
  stable = FALSE,
  genplot = FALSE,
  verbose = FALSE
)
```

Arguments

data	Input data set, a matrix with 2 columns: first column depth or time, second column the proxy. Data points are treated as equally spaced (the regression uses the sample index as predictor, as in Acycle).
k_max	Maximum number of change points Default=10
d_min	Minimum distance (in data points) between adjacent change points, i.e. minimum segment length Default=1
k_0	Hyperparameter (variance scaling) of the prior on the regression coefficients Default=0.01
v_0	Hyperparameter (pseudo data points) of the scaled inverse chi-square prior on the variance Default=NULL which equals ceiling(abs(x[n]-x[1])/44) as in Acycle.
sig_0	Hyperparameter (pseudo data variance) of the scaled inverse chi-square prior on the variance Default=NULL which equals the variance of the proxy.
n_sim	Number of sampled solutions Default=500
seed	Seed for the random number generator, Default=NULL (no seed is set).
ci	Probabilities of the pointwise credible interval of the model Default=c(0.025,0.975)
stable	Use numerically safe log-space arithmetic (shifted log-sum-exp, lgamma, log binomial coefficients). With the default FALSE the arithmetic of Acycle is reproduced exactly (log(sum(exp(.))) in linear space), which under- or overflows for long records or records with a very small/large variance; a warning is given when that happens. Default=FALSE
genplot	Generate a plot Default=FALSE
verbose	Print a progress bar Default=FALSE

Value

A list with elements

- model: matrix [x, posterior-mean model] (Acycle "BayesRegModel").
- cpt: matrix [x, posterior probability of a change point] (Acycle "BayesChangepoint"). A change point at index i means that a new segment starts at index i+1.
- r2: $R^2 = 1 - \sum(y - model)^2 / \sum y^2$ as in Acycle.
- k_posterior: matrix [number of change points (0..k_max), posterior probability].
- model_ci: matrix [x, lower, median, upper] pointwise credible interval of the sampled models.

- `beta`: matrix [intercept, slope] of the posterior-mean regression coefficients at each data point (predictor = sample index).
- `samples`: `n_sim` x `k_max` matrix with the index of the sampled change points (0 = none), as Acycle's samp_holder.
- `samples_x`: same as `samples` expressed in the unit of the first column (NA = none).
- `n_cpt`: sampled number of change points for every solution.
- `sample_models`: N x `n_sim` matrix of the sampled models.
- `log_partition`: the k_max x N partition function (log form).
- `param`: list with the used hyperparameters.

Author(s)

Michiel Arts (R translation); original MATLAB code: Eric Ruggieri, modified for Acycle by Mingsong Li.

References

Ruggieri, E. (2013). A Bayesian approach to detecting change points in climatic records. International Journal of Climatology, 33, 520-528. <doi:10.1002/joc.3447>

Li, M., Hinnov, L., Kump, L. (2019). Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences, 127, 12-22. <doi:10.1016/j.cageo.2019.02.011>

Examples

```
# a record with two changes in mean and one change in trend
set.seed(2)
x <- 1:150
y <- c(rep(0, 50), rep(2, 50), 2 + 0.05 * (1:50)) + stats::rnorm(150, sd = 0.4)
bcp <- bayes_changepoint(data = cbind(x, y), k_max = 6, d_min = 5,
                         k_0 = 0.01, v_0 = 2, n_sim = 500, seed = 1)
bcp$k_posterior
plot_bayes_changepoint(bcp)
```

---

bayes_changepoint_r2 *R-squared of the Bayesian change point model as a function of the maximum number of change points*

Description

The `bayes_changepoint_r2` function runs the Bayesian change point algorithm of Ruggieri (2013) (see `bayes_changepoint`) for a maximum number of change points k = 1, 2, ..., `k_max` and returns the R^2 of the posterior-mean model for each k. This helps to choose `k_max`. Translated from Acycle v3.1 bayes_cpt_r2.m (Mingsong Li, modified from the code of Eric Ruggieri). Note: the Acycle code loops over k but always uses the full `k_max` inside the loop (the loop index is unused), so it

returns k_max replicate R2 values of the same model; here the loop index is used as the maximum number of change points, as intended. The sub-interval likelihoods and the partition function are calculated once (the rows of the partition function do not depend on k_max).

Usage

```
bayes_changepoint_r2(
  data = NULL,
  k_max = 10,
  d_min = 1,
  k_0 = 0.01,
  v_0 = NULL,
  sig_0 = NULL,
  n_sim = 500,
  seed = NULL,
  stable = FALSE,
  genplot = FALSE,
  verbose = FALSE
)
```

Arguments

data	Input data set, a matrix with 2 columns: first column depth or time, second column the proxy. Data points are treated as equally spaced (the regression uses the sample index as predictor, as in Acycle).
k_max	Maximum number of change points Default=10
d_min	Minimum distance (in data points) between adjacent change points, i.e. minimum segment length Default=1
k_0	Hyperparameter (variance scaling) of the prior on the regression coefficients Default=0.01
v_0	Hyperparameter (pseudo data points) of the scaled inverse chi-square prior on the variance Default=NULL which equals ceiling(abs(x[n]-x[1])/44) as in Acycle.
sig_0	Hyperparameter (pseudo data variance) of the scaled inverse chi-square prior on the variance Default=NULL which equals the variance of the proxy.
n_sim	Number of sampled solutions Default=500
seed	Seed for the random number generator, Default=NULL (no seed is set).
stable	Use numerically safe log-space arithmetic (shifted log-sum-exp, lgamma, log binomial coefficients). With the default FALSE the arithmetic of Acycle is reproduced exactly (log(sum(exp(.))) in linear space), which under- or overflows for long records or records with a very small/large variance; a warning is given when that happens. Default=FALSE
genplot	Generate a plot Default=FALSE
verbose	Print a progress bar Default=FALSE

Value

A matrix with columns `k_max` and `r2`.

Author(s)

Michiel Arts (R translation); original MATLAB code: Eric Ruggieri, modified for Acycle by Mingsong Li.

References

Ruggieri, E. (2013). A Bayesian approach to detecting change points in climatic records. International Journal of Climatology, 33, 520-528. <doi:10.1002/joc.3447>

Examples

```
set.seed(2)
x <- 1:150
y <- c(rep(0, 50), rep(2, 50), 2 + 0.05 * (1:50)) + stats::rnorm(150, sd = 0.4)
r2 <- bayes_changepoint_r2(data = cbind(x, y), k_max = 6, d_min = 5,
                           v_0 = 2, n_sim = 200, seed = 1, genplot = TRUE)
```

---

bayes_spectrum *Bayesian probability spectrum*

Description

Calculates the Bayesian probability spectrum of Gregory (2005) as used by Weedon et al. (2019) and Weedon (2020): the data are pre-whitened with the lag-1 autocorrelation coefficient, linearly detrended, tapered with a 5% split cosine taper and standardised; the variance-normalised Lomb-Scargle periodogram (routine PERIOD of Press et al., 1992) is then converted into the posterior probability of a single sinusoid at each frequency, $p(f) \propto (1 - 2P(f)/(N\overline{d^2}))^{(2-N)/2}$, normalised so that the probabilities sum to one. Translated from Acycle v3.1 specbayes.m (Mingsong Li, after the FORTRAN program of Graham P. Weedon).

Usage

```
bayes_spectrum(data = NULL, genplot = FALSE, verbose = FALSE)
```

Arguments

`data`	Matrix or data frame with two columns: depth/time and proxy value. Rows are sorted on the first column and non-finite rows removed.
`genplot`	Plot the periodogram and the Bayesian probability. `Default=FALSE`
`verbose`	Print the frequency with maximum probability. `Default=FALSE`

Value

A list with elements `freq` (frequencies), `period` (1/freq), `power` (variance-normalised Lomb-Scargle periodogram of the pre-whitened, detrended, tapered and standardised data), `bayes_prob` (normalised Bayesian probability), `max_prob`, `freq_max` (frequency of maximum probability) and `rho1` (lag-1 autocorrelation used for pre-whitening).

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li, translated from the FORTRAN program of Graham P. Weedon.

References

Gregory, P.C., 2005. Bayesian Logical Data Analysis for the Physical Sciences. Cambridge University Press.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in FORTRAN, 2nd ed. Cambridge University Press.

Weedon, G.P., Page, K.N., Jenkyns, H.C., 2019. Cyclostratigraphy, stratigraphic gaps and the duration of the Hettangian (Jurassic): insights from the Blue Lias Formation of Southern Britain. Geological Magazine 156, 1469-1509.

Weedon, G.P., 2020. Confirmed detection of Palaeogene and Jurassic orbitally-forced sedimentary cycles in the depth domain using False Discovery Rates and Bayesian probability spectra. Boletin Geologico y Minero 131, 207-230.

Examples

```
set.seed(2)
n <- 400
t <- 1:n
noise <- as.numeric(stats::filter(stats::rnorm(n), 0.5,
  method = "recursive"))
x <- noise + 1.5 * sin(2 * pi * t / 23)
res <- bayes_spectrum(data = cbind(t, x), genplot = TRUE)
1 / res$freq_max
```

Bisciaro_al_wt_track *Period of the short kyr ecc cycle in the Al record of the Bisciaro Fm*

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked in the wavelet scalogram of the aluminium (XRF) record of the Bisciaro Formation
The period was tracked using the `track_period_wavelet` function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the Aluminium (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

`Bisciaro_ca_wt_track` *Period of the short kyr ecc cycle in the Ca record of the Bisciaro Fm*

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked in the wavelet scalogram of the calcium (XRF) record of the Bisciaro Formation
The period was tracked using the `track_period_wavelet` function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the calcium (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

`Bisciaro_Mg_wt_track` *Period of the short kyr ecc cycle in the Mg record of the Bisciaro Fm*

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked in the wavelet scalogram of the magnesium (XRF) record of the Bisciaro Formation
The period was tracked using the `track_period_wavelet` function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the Magnesium (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

`Bisciaro_Mn_wt_track` *Period of the short kyr ecc cycle in the Mn record of the Bisciaro Fm*

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked
in the wavelet scalogram of the manganese (XRF) record of the Bisciaro Formation
The period was tracked using the `track_period_wavelet` function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the manganese (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

`Bisciaro_sial_wt_track`
Period of the short kyr ecc cycle in the si/Al record of the Bisciaro Fm

---

Description

Data points which give the period (in meters) of the short kyr eccentricity cycle tracked
in the wavelet scalogram of the silicon/aluminium (XRF) record of the Bisciaro Formation
The period was tracked using the `track_period_wavelet` function
The tracking is based on a reinterpretation of Arts (2014)

Details

Column 1: depth proxy record
Column 2: period tracked in the wavelet scalogram of the silicon/aluminium (XRF) record

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

`Bisciaro_XRF` *XRF records of the Bisciaro Fm*

Description

XRF proxy records from the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy)

Details

Column 1: depth proxy record
Column 2-71: XRF proxy records

References

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

---

`bispectral_analysis` *Bispectral analysis: bispectrum, biphase and squared bicoherence*

Description

The `bispectral_analysis` function estimates the complex bispectrum $B(f_1, f_2) = mean[X(f_1)X(f_2)conj(X(f_1+f_2))]$, the biphase $argB$ and the Kim and Powers (1979) magnitude-squared bicoherence (bounded in [0,1]) of an evenly or unevenly sampled series. Two estimators are available: a direct FFT bispectrum averaged over tapered (WOSA/Welch) segments, and a two-dimensional hexagonal frequency-smoothed direct estimate on the native DFT grid. The numerical domain is the non-redundant positive principal domain ($k_2 \leq k_1$, $k_1 + k_2$ below Nyquist). Before estimation the series is cleaned (non-finite rows removed, sorted, duplicate coordinates averaged), regularised by interpolation when the sampling is irregular, detrended and optionally standardised. Optional inference: an analytical Beta(1,R-1) pointwise reference, pointwise Monte Carlo surrogate thresholds, or a global maximum-statistic surrogate test (family-wise error control over the full computed principal domain) using Fourier-transform (FT) phase-randomised or IAAFT surrogates (Schreiber

and Schmitz, 1996) with plus-one Monte Carlo p-values. High bicoherence indicates repeatable quadratic phase coupling at $(f_1, f_2, f_1 + f_2)$; it is not by itself evidence of causality or energy transfer. Translated from Acycle v3.1 bispectralAnalyze.m, bispectralDefaults.m, bispectralPreprocess.m, bispectralEstimate.m and bispectralSignificance.m (Mingsong Li).

Usage

```
bispectral_analysis(
  data = NULL,
  estimator = NULL,
  n_segments = NULL,
  overlap = 50,
  window = "hann",
  segment_detrend = "mean",
  smoothing_span = 3,
  smoothing_kernel = "daniell",
  detrend = "linear",
  poly_order = 2,
  standardize = TRUE,
  interpolate = "auto",
  interp_method = "linear",
  sample_interval = NULL,
  irregular_tol = 0.01,
  gap_warning_factor = 5,
  input_policy = "prepare",
  nfft = NULL,
  zero_pad = 1,
  max_freq_bins = 512,
  significance = "none",
  conf_level = 0.95,
  n_sim = 999,
  surrogate_type = "iaaft",
  iaaft_iter = 200,
  iaaft_tol = 0.02,
  max_attempts = NULL,
  seed = NULL,
  genplot = FALSE,
  verbose = FALSE,
  standardise = NULL
)
```

Arguments

`data`	Input data set: a matrix or data frame with 2 columns, the first being depth/time and the second the proxy value.
`estimator`	Estimator: "wosa" (segment averaged) or "frequency-smoothed". `NULL` chooses as Acycle does: "wosa", unless three 32-point segments do not fit, then "frequency-smoothed". `Default=NULL`

n_segments	Number of WOSA segments (>= 3). NULL uses min(8, maximum number of 32-point segments) as in Acycle. Default=NULL
overlap	Segment overlap in percent, in [0,90). Default=50
window	Taper: "hann", "hamming", "blackman" or "rectangular". Default="hann"
segment_detrend	Detrending applied to each segment (or the whole record for the frequency-smoothed estimator): "none", "mean" or "linear". Default="mean"
smoothing_span	Odd span (>= 3) of the hexagonal frequency-smoothing kernel (3 gives 7 triads). Default=3
smoothing_kernel	"daniell" (uniform) or "cosine" weights. Default="daniell"
detrend	Whole-record detrending: "none", "mean", "linear" or "polynomial". Default="linear"
poly_order	Polynomial order (1-10) for detrend="polynomial". Default=2
standardize	Standardise the series to zero mean and unit standard deviation after detrending. Default=TRUE
interpolate	"auto" (interpolate only irregular series), "always" or "never". Default="auto"
interp_method	Interpolation method: "linear", "pchip" or "makima". Default="linear"
sample_interval	Sample interval of the regular grid; NULL uses the median original spacing. May not be coarser than the median spacing. Default=NULL
irregular_tol	Maximum relative spacing departure treated as regular (input_policy "prepare"). Default=0.01
gap_warning_factor	Largest gap (in sample intervals) above which a warning is stored. Default=5
input_policy	"prepare" (clean the input) or "strict" (refuse non-finite, unsorted or duplicated rows). Default="prepare"
nfft	FFT length for the WOSA estimator; NULL uses round(segment length x zero_pad). Ignored by the frequency-smoothed estimator. Default=NULL
zero_pad	Zero-padding factor for the WOSA estimator. Default=1
max_freq_bins	Maximum number of computed frequency-axis bins (>= 16); larger grids are decimated with a constant stride. Default=512
significance	Inference: "none", "analytical", "pointwise" (pointwise surrogate thresholds) or "global" (maximum-statistic surrogate test). Default="none"
conf_level	Confidence level in (0.5,1). Default=0.95
n_sim	Number of accepted surrogates (19-99999). Default=999
surrogate_type	"iaaft" or "ft" (phase randomisation). Default="iaaft"
iaaft_iter	Maximum number of IAAFT iterations. Default=200
iaaft_tol	Maximum accepted relative Fourier-amplitude error of an IAAFT surrogate (non-DC, non-Nyquist bins); worse surrogates are rejected and replaced. Default=0.02
max_attempts	Maximum number of surrogate attempts; NULL allows n_sim + max(10, ceiling(0.1 n_sim)). Default=NULL

seed	Seed for the random number generator (NULL leaves the generator untouched). Default=NULL
genplot	Plot the result with plot_bispectral. Default=FALSE
verbose	Print progress (progress bar for the surrogates). Default=FALSE
standardise	British spelling of standardize; if given it overrides standardize.

Value

A list with elements frequency (computed frequency axis, cycles per unit), bispectrum (complex matrix; rows = f2, columns = f1; NA outside the principal domain), bispectrum_magnitude, biphase (radians, NA where undefined), bicoherence_squared, denominator, valid_mask, principal_domain_mask, power (segment- averaged power on the frequency axis), pair_index, pair_bicoherence_squared, meta (estimator settings and diagnostics), processed_data (2-column processed series), preprocessing (list), significance (list with method, threshold, threshold_matrix, p_value, significant_mask, surrogate_maxima, surrogate_spectral_errors, attempt/rejection counts and warnings), significant_mask and warnings.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li (Acycle bispectral module).

References

Kim, Y.C., Powers, E.J., 1979. Digital bispectral analysis and its applications to nonlinear wave interactions. IEEE Transactions on Plasma Science 7, 120-131.

Choudhury, M.A.A.S., Shah, S.L., Thornhill, N.F., 2008. Diagnosis of Process Nonlinearities and Valve Stiction. Springer.

Schreiber, T., Schmitz, A., 1996. Improved surrogate data for nonlinearity tests. Physical Review Letters 77, 635-638.

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22.

Examples

```
# quadratic phase coupling: f3 = f1 + f2 with phase3 = phase1 + phase2 + 0.4
set.seed(1)
seg_len <- 256
n_seg <- 16
t <- 0:(seg_len * n_seg - 1)
y <- numeric(length(t))
f1 <- 30 / seg_len
f2 <- 17 / seg_len
for (s in seq_len(n_seg)){
  idx <- (s - 1) * seg_len + seq_len(seg_len)
  p1 <- 2 * pi * runif(1)
  p2 <- 2 * pi * runif(1)
  y[idx] <- cos(2 * pi * f1 * t[idx] + p1) +
    0.85 * cos(2 * pi * f2 * t[idx] + p2) +
    0.65 * cos(2 * pi * (f1 + f2) * t[idx] + p1 + p2 + 0.4) +
```

```
    0.18 * rnorm(seg_len)
}
res <- bispectral_analysis(data = cbind(t, y), n_segments = 16,
  overlap = 0, detrend = "none", max_freq_bins = 256,
  significance = "global", n_sim = 19, seed = 1, genplot = TRUE)
```

`bispectral_surrogate` *FT and IAAFT surrogates for bispectral significance testing*

Description

The `bispectral_surrogate` function generates null-hypothesis surrogate series. Fourier-transform (FT) phase-randomised surrogates preserve the periodogram exactly (all Fourier amplitudes, including DC and the even-length Nyquist bin) while randomising the phases. Iterative amplitude-adjusted Fourier-transform (IAAFT) surrogates (Schreiber and Schmitz, 1996) preserve the observed marginal distribution exactly and iteratively approach the observed Fourier amplitudes; the iteration stops early when the relative spectral error changes by less than 1e-10. The reported relative spectral error is evaluated only on the independent positive frequencies (DC and the even-length Nyquist bin excluded). Translated from Acycle v3.1 bispectralSurrogate.m (Mingsong Li).

Usage

```
bispectral_surrogate(
  data = NULL,
  type = "iaaft",
  iaaft_iter = 200,
  n_sim = 1,
  seed = NULL,
  verbose = FALSE
)
```

Arguments

`data`	Input: a numeric vector, or a 2-column matrix [depth/time, value] of which the second column is used.
`type`	"iaaft" or "ft" (phase randomisation). `Default="iaaft"`
`iaaft_iter`	Maximum number of IAAFT iterations. `Default=200`
`n_sim`	Number of surrogates to generate. `Default=1`
`seed`	Seed for the random number generator (`NULL` leaves the generator untouched). `Default=NULL`
`verbose`	Show a progress bar. `Default=FALSE`

Value

A list with `surrogates` (matrix, one surrogate per column), `type`, `iterations` (completed IAAFT iterations per surrogate, 1 for FT) and `relative_spectral_error` (per surrogate).

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li (Acycle bispectral module).

References

Theiler, J., Eubank, S., Longtin, A., Galdrikian, B., Farmer, J.D., 1992. Testing for nonlinearity in time series: the method of surrogate data. Physica D 58, 77-94.

Schreiber, T., Schmitz, A., 1996. Improved surrogate data for nonlinearity tests. Physical Review Letters 77, 635-638.

Examples

```
set.seed(2)
x <- exp(0.35 * as.numeric(arima.sim(list(ar = 0.75), n = 512)))
sur <- bispectral_surrogate(data = x, type = "iaaft", n_sim = 5, seed = 3)
# marginal distribution preserved exactly
max(abs(sort(sur$surrogates[, 1]) - sort(x)))
sur$relative_spectral_error
```

---

bundling_ratios *Precession cycles per eccentricity bundle, predicted*

Description

Precession cycles per eccentricity bundle, predicted

Usage

```
bundling_ratios(
  age_ma = 0,
  k_yr = NULL,
  ecc_terms = c("g2-g5", "g3-g2", "g4-g2", "g3-g5", "g4-g5"),
  prec_terms = c("k+g5", "k+g2", "k+g3", "k+g4", "k+s3"),
  data_dir = NULL
)
```

Arguments

age_ma	age in Ma (the precession constant comes from AstroGeo22)
k_yr	optional: your own precession constant (yr) instead of the model
ecc_terms	eccentricity terms (rows)
prec_terms	precession and obliquity terms (columns)
data_dir	folder with the data tables

Value

matrix of bundling ratios, eccentricity terms in rows, precession (and obliquity) terms in columns

calculate_orbit9 *Nine astronomical periods at a given age*

Description

Nine principal orbital periods (calculate_orbit9.m in Acycle): La2004 secular frequencies g_i, s_i and the AstroGeo22 precession constant k(t) at the age rounded to the nearest Ma.

Usage

```
calculate_orbit9(age_ma, data_dir = NULL)
```

Arguments

age_ma	age in Ma (single finite number).
data_dir	folder holding AstroGeo22.txt and La04gisi.csv `Default=NULL` (searched with coco_data_dir()).

Value

data.frame with columns term, group, freq_arcsec_yr, period_yr, period_kyr.

chaos_scenario *Shift the fundamental frequencies and see what every cycle does*

Description

Shift the fundamental frequencies and see what every cycle does

Usage

```
chaos_scenario(
  g4_g3_kyr = NULL,
  s4_s3_kyr = NULL,
  hold = c("mean", "g3", "g4"),
  age_ma = 0,
  k_yr = NULL,
  data_dir = NULL
)
```

Arguments

g4_g3_kyr	the period of the "2.4 Myr" term under the scenario (NULL keeps La2004); 'hold' says how g3 and g4 are moved to achieve it
s4_s3_kyr	the period of the "1.2 Myr" term under the scenario
hold	"mean" moves g3 and g4 symmetrically about their mean (default), "g3" holds g3 and moves g4 only, "g4" holds g4 and moves g3 only
age_ma	age, for the precession constant used in the k-bearing terms
k_yr	Precession constant in arcsec/yr ('NULL' = taken from 'astrogeo22(age_ma)$p_arcsec').
data_dir	Directory with the data files ('NULL' = the data installed with the package).

Value

the astro_terms() table with the scenario period, the La2004 period, and the change, so it is obvious which cycles move and which do not

---

check_agemodel *Check an age model against the de-chirped record: where do the astronomical cycles bend away?*

Description

An existing track (a tracked period curve, a sedimentation-rate curve or an age model, see as_wr_track) is used to de-chirp the record with analyse_dechirped: in the time of the track every astronomical cycle should be a horizontal line at its period. Wherever the track is wrong the lines bend away from the expected periods, and the local offset of the ridge gives the correction of the sedimentation rate that the record itself asks for: an observed period P_obs where P was expected means the true rate is P_obs / P times the tracked rate (the track ran through that interval too fast or too slowly by that factor). The offset is measured per cycle and per transform (the smooth Morlet wavelet, the multitaper EHA, the superlet-SST, the chirplet-SST and the combined map each see the ridges differently). The offset is found by a comb search: a wrong rate shifts every cycle by the same factor, so the log-power at all expected periods is summed for every common shift of the comb (as the harmonic product spectrum of pitch detection does) and the best shift is the correction; the cycles are then checked one by one at that shift. The corrections are combined by a weighted median and an interval is flagged when the combined correction exceeds tolerance with at least min_cycles cycles and min_transforms transforms agreeing on its direction. Optionally a blind chirp-matched re-track (track_astro_auto) is run for comparison.

Usage

```
check_agemodel(
  data,
  track,
  age_ma = 0,
  astro_periods = NULL,
  transforms = c("wavelet", "eha", "superlet_sst", "chirplet_sst"),
```

```
  band = 0.3,
  max_shift = 1.2,
  tolerance = 0.1,
  min_cycles = 2,
  min_transforms = 2,
  min_strength = 1.5,
  min_contrast = 1.5,
  eha_window = NULL,
  n_max = 6000,
  clip = TRUE,
  min_length = NULL,
  harmonics = TRUE,
  retrack = TRUE,
  sedrate = NULL,
  dj = 1/50,
  verbose = TRUE,
  ...
)
```

Arguments

data	Data frame or matrix with depth and the proxy.
track	The existing track (wr_track or track_astro_auto result).
age_ma	Age (Ma) used to pick the astronomical periods with astro_periods_tracking when astro_periods is not given Default=0.
astro_periods	Periods (kyr) of the cycles to measure; by default the 405, 124 and 95 kyr eccentricity cycles, the main obliquity term and the two strongest precession terms for age_ma.
transforms	Which transforms of analyse_dechirped to use Default=c("wavelet", "eha", "superlet_sst", "chirplet_sst").
band	Half-width (octaves) of the band around each (shifted) expected period in which a cycle's own ridge is looked for; it is narrowed where two cycles are closer than that Default=0.3.
max_shift	Largest common shift (octaves) of the comb that is searched, i.e. a rate error of up to a factor 2^max_shift; a best shift at the edge of the grid is discarded Default=1.2.
tolerance	Relative rate correction above which an interval is flagged Default=0.1.
min_cycles, min_transforms	Minimum number of cycles and transforms supporting a flag Default=2.
min_strength	Minimum ridge prominence (ratio of the band maximum to the median of the column) for an offset to count; squeezed transforms use a tenth of it on their sparse maps Default=1.5.
min_contrast	Minimum contrast of the best shift (its score above the median over all shifts, in MADs) for a column to count Default=1.5.
eha_window	Window (kyr) of the multitaper EHA in the time of the track; by default 2.5 times the longest cycle, so that the 405-kyr cycle is resolved Default=NULL.

n_max	Records longer than this are resampled in depth before the transforms Default=6000.
clip	Restrict the record to the depth range of the track (beyond it the track is extrapolated with its end rates, which can dominate the time axis) Default=TRUE.
min_length	Shortest flagged interval kept and shortest gap merged, in kyr of the track; by default half the longest cycle Default=NULL.
harmonics	Add the 203 and 135 kyr harmonics of the 405-kyr cycle to the comb (weight 0.4), so that a non-sinusoidal long eccentricity cycle is not read as shifted 124 and 95 kyr cycles Default=TRUE.
retrack	Run a blind chirp-matched re-track for comparison Default=TRUE.
sedrate	Rate range for the blind re-track; by default half to twice the range of the existing track.
dj	Period resolution (octaves) Default=1/50.
verbose	Print progress Default=TRUE.
...	Passed to analyse_dechirped.

Details

The measured correction is only as good as the ridges: a cycle that is not expressed in the record gives no offset (its strength falls below min_strength and it is dropped), and an interval flagged by one weak cycle alone is not flagged. The flagged intervals are a list of places to look at in the maps of plot_agemodel_check, not a new age model; use track_astro_auto or track_chirp_hmm to re-track.

Value

A list of class wr_agemodel_check: profile (data frame with depth, time, sedrate of the track, one column of log2 rate correction per transform, correction (the combined factor), n_cycles, n_transforms, flag), intervals (data frame of the flagged intervals: from, to, correction, cycles, transforms), offsets (per transform: shift, the comb shift per time column; offset, the log2 offset of every cycle at that shift; offset_single, the offset of every cycle searched on its own around its expected period), dechirped (the wr_dechirped objects), retrack (the blind track, if run), astro_periods, track, data.

See Also

plot_agemodel_check, analyse_dechirped, as_wr_track

Examples

```
set.seed(1)
t <- seq(0, 3000, 1)
x <- cos(2 * pi * t / 405) + 0.8 * cos(2 * pi * t / 100) + 0.6 * cos(2 * pi * t / 41) +
  rnorm(length(t), 0, 0.8)
rate <- ifelse(t < 1500, 0.05, 0.08)      # a step in the rate the age model does not know
z <- cumsum(rate); data <- data.frame(z, x)
track <- as_wr_track(data.frame(z, 0.065), "sedrate")   # a constant-rate age model
chk <- check_agemodel(data, track, age_ma = 0, transforms = c("wavelet", "superlet_sst"),
    retrack = FALSE)
```

```
chk$intervals
plot_agemodel_check(chk)
```

`chirp_eha_significance` *Significance of the lines of a (de-chirped) EHA map against warped noise*

Description

The de-chirped map of `analyse_chirp_eha` warps the noise inside every window as well as the cycles, so wavy structure in it is not by itself evidence. This function runs the same analysis, with the same windows, gradients (or `steer`) and settings, on AR1 surrogates of the record and compares the observed line-to-continuum ratios with those of the surrogates cell by cell. Lines that are only artefacts of the warping appear in the surrogates as well and are not significant.

Usage

```
chirp_eha_significance(ce, data, n_sim = 20, rho = NULL, seed = 1)
```

Arguments

`ce`	Result of `analyse_chirp_eha`.
`data`	The record it was computed from.
`n_sim`	Number of surrogates `Default=20`.
`rho`	Lag-1 autocorrelation of the surrogates `Default=NULL`: estimated from the record.
`seed`	Seed.

Value

`ce` with `Power_sig` (observed `Power_norm` divided by the 95th percentile of the surrogates: above 1 is significant at that level), `Power_pval` (fraction of surrogates at or above the observed value) and `surrogate_q95`.

coco_adaptive *Adaptive COCO*

Description

Modern Acycle adaptive COCO: nine-term orbital target synthesised on the finite record at every tested rate, amplitudes fitted to the spectrum, AR(1) Monte Carlo null with global (max-over-rate) and local p-values. Translated from Acycle corrcoefslices_rankNew.m (Mingsong Li).

Usage

```
coco_adaptive(
  data,
  orbit9 = orbit9_default(),
  sr = seq(0.5, 20, by = 0.1),
  pad = 5000,
  red = 1,
  nsim = 1000,
  slices = 1,
  method = c("Pearson", "Spearman"),
  max_frequency = NULL,
  amplitude_mode = c("four-group-area", "per-orbit", "fixed"),
  target_model = c("coherent-nine", "phase-averaged"),
  seed = 1,
  batch = NULL,
  verbose = TRUE
)
```

Arguments

data	two columns depth (m), value. Irregular data are linearly interpolated at the median spacing (as in Acycle).
orbit9	nine orbital periods (kyr)
sr	sedimentation-rate grid (cm/kyr)
pad	zero padding Default=5000
red	0 none, 1 classical AR(1) subtraction, 2 robust AR(1), 3 smoothed-window-average (SWA) background
nsim	number of AR(1) Monte Carlo simulations
slices	number of slices whose spectra are averaged Default=1
method	"Pearson" or "Spearman"
max_frequency	highest target frequency Default=NULL
amplitude_mode	"four-group-area", "per-orbit" or "fixed"
target_model	"coherent-nine" or "phase-averaged"
seed	random seed Default=1

batch	Monte Carlo batch size Default=NULL
verbose	progress messages Default=TRUE

Value

list with ‘result‘ data.frame: sr, rho, p_param, p_global, p_local, norbit, pcoco

---

`coco_adaptive_evaluate` *Evaluate Adaptive COCO for a batch of spectra (cocoAdaptiveEvaluate.m)*

Description

Evaluate Adaptive COCO for a batch of spectra (cocoAdaptiveEvaluate.m)

Usage

```
coco_adaptive_evaluate(
  data_power,
  data_frequency,
  n_target,
  spacing,
  pad,
  orbit9,
  sr_grid,
  method = "Pearson",
  max_frequency = NULL,
  target_model = c("coherent-nine", "phase-averaged"),
  amplitude_mode = c("four-group-area", "per-orbit", "fixed"),
  observed_p = FALSE
)
```

Arguments

data_power	nf-by-k matrix: one-sided pad-point periodograms (cycles/m grid)
data_frequency	Frequencies of the data spectrum.
n_target	number of samples of the analysed record / window
spacing	depth sampling interval of that record (m)
pad	Zero-padding factor used for the spectrum.
orbit9	Orbital periods (kyr) used as target frequencies.
sr_grid	Grid of sedimentation rates to evaluate.
method	Correlation method, ‘"Pearson"‘ or ‘"Spearman"‘.
max_frequency	Maximum frequency evaluated (‘NULL‘ = 1.2 times the maximum of ‘1/orbit9‘).

target_model	"coherent-nine" (default) or "phase-averaged"
amplitude_mode	"four-group-area" (Acycle default for Adaptive COCO/eCOCO), "per-orbit" (adaptive9a), "fixed" (1.0/0.8/0.6 weights)
observed_p	Observed p-value(s) to compare against.

`coco_classic` *COCO analysis (Li et al. 2018)*

Description

Correlation coefficient between the (AR(1)-corrected) data periodogram and an astronomical target over a grid of sedimentation rates, with the H0 (no orbital forcing) Monte Carlo test of Li et al. (2018). Translated from Acycle corrcoefslices.m / cyclecorr.m (Mingsong Li).

Usage

```
coco_classic(
  data,
  target,
  orbit = NULL,
  sr = seq(0.2, 20, by = 0.2),
  dt = NULL,
  pad = 5000,
  red = 1,
  nsim = 0,
  slices = 1,
  adjust = 0,
  method = c("Pearson", "Spearman"),
  seed = NULL,
  keep_null = FALSE,
  verbose = TRUE
)
```

Arguments

data	two-column matrix/data.frame: depth (m), proxy value; evenly spaced
target	two-column target periodogram (cycles/kyr, power), from coco_target_from_periods() or coco_target_from_series()
orbit	orbital periods (kyr); used to count the number of contributing astronomical parameters. Default: the periods stored with the target (coco_target_from_periods), else orbit7_default()
sr	tested sedimentation rates (cm/kyr)
dt	sampling interval `Default=NULL` (median spacing)
pad	zero padding of the data periodogram

red	0 none, 1 subtract AR(1) (default, as in paper), 2 ratio AR(1), 3 robust AR(1), 50-99 remove red-noise confidence level
nsim	number of Monte Carlo simulations (paper: 2000)
slices	number of equal slices whose periodograms are averaged
adjust	1 = adjust target band power to data (7-period target only)
method	"Pearson" (paper) or "Spearman"
seed	random seed Default=NULL
keep_null	return the Monte Carlo null matrix Default=FALSE
verbose	progress messages Default=TRUE

Value

list with data.frame ‘result‘ (sr, rho, p_param, p_h0, norbit, pcoco)

`coco_fixed_target_weights` *Fixed sinusoid-amplitude weights for a COCO target (cocoFixedTargetWeights.m)*

Description

Fixed sinusoid-amplitude weights for a COCO target (cocoFixedTargetWeights.m)

Usage

```
coco_fixed_target_weights(orbit)
```

Arguments

orbit	vector of target periods (kyr)

`coco_prepare_regular_data` *Prepare a depth series for COCO (cocoPrepareRegularData.m)*

Description

Removes non-finite rows, sorts by depth, averages duplicate depths and linearly interpolates to the median step when the spacing is uneven.

Usage

```
coco_prepare_regular_data(raw, verbose = TRUE, min_points = 4)
```

Arguments

raw	two-column matrix [depth (m), value]
verbose	print what was changed
min_points	minimum number of points required

---

coco_target_fixed *Fixed-weight astronomical target spectrum (Fixed-target COCO)*

Description

Builds the fixed-relative-weight astronomical target used by Acycle's Fixed-target COCO: nine (or other) orbital sine terms with the historical amplitudes 1.0 (eccentricity), 0.8 (obliquity) and 0.6 (precession) (cocoFixedTargetWeights.m), sampled on the data grid mapped to time at sedimentation rate `sr`, linearly detrended, and turned into a rectangular-window periodogram of length max(pad, n) that is linearly interpolated at the requested temporal frequencies (0 outside the target band). Unlike the adaptive target, the amplitudes never depend on the data spectrum. Translated from Acycle v3.1 freq2targetFixed.m and cocoFixedTargetWeights.m (Mingsong Li).

Usage

```
coco_target_fixed(
  data = NULL,
  pad = 5000,
 orbit9 = c(405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897, 23.682, 22.3758,
    18.9519),
  freq = NULL,
  sr = NULL
)
```

Arguments

data	Two-column matrix (depth, value) defining the sampling grid; only the depth column is used. NULL uses 5 x max(orbit9) unit-spaced samples. `Default=NULL`
pad	Periodogram length. `Default=5000`
orbit9	Orbital periods (kyr). `Default=c(405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897, 23.6820, 22.3758, 18.9519)`
freq	Temporal frequencies (cycles/kyr) at which the target power is returned. `Default=NULL`
sr	Sedimentation rate (cm/kyr). `Default=NULL`

Value

Numeric vector of target power at `freq`.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li

References

Li, M., Kump, L.R., Hinnov, L.A., Mann, M.E., 2018. Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing. Earth and Planetary Science Letters 501, 165-179.

Examples

```
depth <- seq(0, 20, by = 0.05)
f <- seq(0, 0.06, by = 0.0005)
la <- coco_target_fixed(cbind(depth, 0), pad = 5000, freq = f, sr = 5)
plot(f, la, type = "l", xlab = "Frequency (cycles/kyr)", ylab = "Power")
```

---

coco_target_from_periods

Target periodogram from a list of orbital periods (period2spectrum.m)

Description

Sum of sinusoids at the orbital periods, sampled from t1 to t2 (kyr), and its zero-padded periodogram between f1 and f2 (cycles/kyr).

Usage

```
coco_target_from_periods(
  orbit = NULL,
  age_ma = NULL,
  t1 = 3e+05,
  t2 = 308000,
  step = 1,
  f1 = 0,
  f2 = NULL,
  pad = 10000,
  weights = NULL
)
```

Arguments

orbit	periods in kyr, longest first (e.g. c(405,125,95,41,23,22,19)) Default=NULL
age_ma	alternative to 'orbit': use the nine periods at this age (calculate_orbit9(); La2004 g/s + AstroGeo22 precession constant) Default=NULL
t1, t2, step	time axis of the synthetic series (kyr)
f1, f2	frequency interval kept (cycles/kyr); f2 defaults to 1.2 x the highest orbital frequency (Acycle GUI)
pad	zero padding of the periodogram Default=10000
weights	sinusoid amplitudes; default 1.0 eccentricity, 0.8 obliquity, 0.6 precession (as in period2spectrum.m)

Details

The returned target carries attribute "orbit" so coco_classic() uses the same periods to count contributing parameters.

Value

two-column matrix: frequency (cycles/kyr), power

`coco_target_from_series` *Target periodogram from an astronomical time series (e.g. La2004 ETP)*

Description

Periodogram of an astronomical series, or of an ETP series built from eccentricity, obliquity and precession as in gentarget.m.

Usage

```
coco_target_from_series(
  time_kyr,
  value = NULL,
  ecc = NULL,
  obl = NULL,
  prec = NULL,
  ne = 1,
  nt = 0.6,
  np = 0.5,
  f1 = 0,
  f2 = 0.06,
  pad = 10000
)
```

Arguments

`time_kyr`	evenly spaced time (kyr)
`value`	astronomical series `Default=NULL`; OR give ecc, obl, prec to build ETP as in gentarget.m
`ecc`	Eccentricity series used instead of ‘value‘.
`obl`	Obliquity series used instead of ‘value‘.
`prec`	Precession series used instead of ‘value‘.
`ne`, `nt`, `np`	ETP weights of eccentricity, obliquity and precession
`f1`, `f2`	frequency interval kept (cycles/kyr)
`pad`	zero padding `Default=10000`

Value

two-column matrix: frequency (cycles/kyr), power

`compare_published` *Compare a measured precession constant with published models*

Description

Compare a measured precession constant with published models

Usage

```
compare_published(age_ma, k_yr = NULL, obliquity_deg = NULL, data_dir = NULL)
```

Arguments

`age_ma`	age of the record (Ma)
`k_yr`	measured precession constant in years (optional)
`obliquity_deg`	obliquity for the angular-momentum inversion; default is the AstroGeo22 value at 'age_ma'
`data_dir`	Directory with the data files ('NULL' = the data installed with the package).

Value

data.frame with one row per source: precession period and frequency, Earth-Moon distance, LOD, obliquity, plus the k+g and k+s periods

---

`completed_series` *Complete the tracking of the period a cycle in a scalogram*

Description

Use the traced series and the existing wavelet or superlet scalogram to complete the tracking of a cycle in scalogram. The selected points using the `track_period_wavelet` or `track_period` 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` or `track_period`. 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(
  scalogram = NULL,
  tracked_curve = NULL,
  period_up = 1.2,
  period_down = 0.8,
  extrapolate = TRUE,
  genplot = FALSE,
  keep_editable = FALSE,
  ...
)
```

Arguments

scalogram	Wavelet or superlet object created using the analyse_wavelet or or track_period function.
tracked_curve	Tracked period result from the track_period_wavelet or or track_period function.
period_up	The period_up parameter is the factor with which the linear interpolated tracked_curve curve is multiplied by. This linear interpolated tracked_curve multiplied by the period_up factor is the upper boundary which is used for detecting the spectral peak nearest to the linear interpolated tracked_curve curve. If no spectral peak is detected within the specified boundary the interpolated value is used instead. between spectral peaks Default=1.2,
period_down	The period_down parameter is the factor with which the linear interpolated tracked_curve curve is multiplied by. This linear interpolated tracked_curve multiplied by the period_down factor is the lower boundary which is used for detecting the spectral peak nearest to the linear interpolated tracked_curve curve. If no spectral peak is detected within the specified boundary the interpolated value is used instead. between spectral peaks Default=0.8,
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.
keep_editable	Keep option to add extra features after plotting Default=FALSE
...	older inputs from previous version

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 grey_track example points to complete the tracking of the
# precession cycle in the wavelet spectra of the grey scale data set
# of Zeeden et al., (2013).
```

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

#The ~22kyr precession cycle is between 0.25 and 1m The grey_track data
#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 (meters)",
#x_lab = "depth (meters)"
#)



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

`ctest` *CTest: astronomical parameters from interpreted spectral peaks*

Description

CTest: astronomical parameters from interpreted spectral peaks

Usage

```
ctest(
  data,
  nsim = 1000,
  mode = c("PT", "P", "T"),
```

```
  ptest = c(10000, 28000),
  sampleorb = c("U", "N"),
  seed = NULL,
  max_attempts = 5e+07,
  verbose = TRUE,
  engine = c("vectorized", "loop"),
  block = 1e+05,
  s3_uncertainty = c("published", "scaled")
)
```

Arguments

data	8x2 table (matrix, data.frame or tibble) as read from the published spreadsheet, or built with ctest_input()
nsim	number of successful samples to collect
mode	"PT", "P" or "T" (see file header)
ptest	range of precession constants to sample, in years
sampleorb	"U" uniform within 2 sigma (used in the paper) or "N" normal
seed	optional RNG seed
max_attempts	abort if no nsim successes are found within this many draws
verbose	print progress every 10%
engine	"vectorized" (default, draws in blocks; fast) or "loop", which follows the published draw order exactly and reproduces mmc3/mmc4/mmc5 sample for sample under the same seed
block	number of draws per block for the vectorized engine
s3_uncertainty	"published" keeps the published code's draw for s3, rnorm(1, s3, ds3*sqrt(t/1e8)), in which the relative uncertainty ds3 is applied as an absolute one in years, so s3 is effectively fixed. "scaled" applies it the same way as for g2-g5 (s3 * ds3 * sqrt(t/1e8)), which gives s3 a realistic spread. Use "published" to reproduce the paper.

Value

list with ‘summary‘ (table as in the paper), ‘samples‘ (all accepted samples), ‘sampled_range‘ of g/s, and ‘acceptance‘

---

ctest_combine *Combine several CTest results by multiplying their densities*

Description

Reproduces the combination step of mmc8 (reconstruct_parameters_Devonian.R), using a base-R weighted quantile instead of Hmisc::wtd.quantile.

Usage

```
ctest_combine(results, what = "k", n = 1000)
```

Arguments

results	list of ctest objects (or numeric vectors of samples)
what	column of 'samples' to combine (default the precession constant)
n	grid size

Value

list: grid 'x', combined density, mean (median of the combined density) and sd (half the interquartile span, as in the paper)

---

ctest_earthmoon *Earth-Moon distance, length of day and obliquity for CTest samples*

Description

Each sampled precession constant is converted with precession_frequency() (Farhat et al. 2022, eq. A.7), either by inverting the AstroGeo22 curve or by assuming Earth-Moon angular-momentum conservation.

Usage

```
ctest_earthmoon(
  x,
  age_ma = NULL,
  method = c("angular_momentum", "astrogeo22"),
  obliquity_deg = NULL,
  probs = c(0.025, 0.5, 0.975),
  data_dir = NULL
)
```

Arguments

x	a ctest object, or a numeric vector of precession constants (yr)
age_ma	age of the record; used for the obliquity of the angular-momentum inversion and for the comparison row
method	"angular_momentum" (default) or "astrogeo22"
obliquity_deg	obliquity used in the angular-momentum inversion; NULL takes it from AstroGeo22 at 'age_ma'
probs	quantiles to report
data_dir	folder with the AstroGeo22 table

Value

list with ‘summary‘, ‘samples‘, ‘method‘, ‘age_ma‘ and, when an age is known, ‘model_at_age‘

`ctest_input` *Build the CTest input table from interpreted cycle wavelengths*

Description

Build the CTest input table from interpreted cycle wavelengths

Usage

```
ctest_input(
  age,
  e405,
  e125,
  e95,
  obl = c(NA, NA),
  p1 = c(NA, NA),
  p2 = c(NA, NA),
  p3 = c(NA, NA)
)
```

Arguments

`age`	c(min, max) age of the record in Ma
`e405`, `e125`, `e95`	c(min, max) wavelength (m) of the 405-kyr, ~125-kyr and ~95-kyr eccentricity cycles in the depth-domain spectrum
`obl`	c(min, max) wavelength (m) of the obliquity cycle (k+s3)
`p1`, `p2`, `p3`	c(min, max) wavelength (m) of the precession cycles, in the row order of the published spreadsheet, i.e. by DECREASING period: p1 = longest (k+g5, ~23.7 kyr today), p2 = k+g2 (~22.4 kyr), p3 = shortest (k+g4 and k+g3 together, ~19 kyr). Note: the comments inside the published functions label these rows the other way round, but the code tests them in the order used here (row 6 against k+g5, row 7 against k+g2, row 8 against k+g4 and k+g3).

Value

the 8x2 matrix used by ctest(), in the published layout (column 1 = lower frequency = longer wavelength)

curve2sedrate *Convert a tracked tracked to a sedimentation rate curve*

Description

Converts the period of a tracked cycle to a sedimentation rate curve by assigning 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 tracked cycle which is the result of 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
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
#in a wavelet spectra by assigning a duration of  405 kyr to the tracked cycle.
# the example uses the magnetic susceptibility data set of Pas et al., (2018)
# perform the CWT
mag_wt <- analyse_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
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 = FALSE
)

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

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

`curve2time` *Convert 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 = FALSE,
  keep_editable = FALSE
)
```

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=FALSE`.
`keep_editable`	Keep option to add extra features after plotting `Default=FALSE`

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 sedrate2time function of the 'astrochron' R package

References

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

Examples

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

#'# perform the CWT
mag_wt <- analyse_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
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 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 = FALSE
)

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

#convert period in meters to sedrate depth vs time
mag_track_time<- curve2time(tracked_cycle_curve=mag_track_complete,
```

```
tracked_cycle_period=405,
genplot=FALSE,
keep_editable=FALSE)
```

`curve2time_unc` *Convert the re-tracked curve results to a depth time space with uncertainty*

Description

Converts the re-tracked curve results from `retrack_wt_MC` function to a depth time space while also taking into account the uncertainty of the tracked astronomical cycle

Usage

```
curve2time_unc(
  tracked_cycle_curve = NULL,
  tracked_cycle_period = NULL,
  tracked_cycle_period_unc = NULL,
  tracked_cycle_period_unc_dist = "n",
  n_simulations = NULL,
  output = 1
)
```

Arguments

`tracked_cycle_curve`
: Curve of the cycle tracked using the `retrack_wt_MC` function
Any input (matrix or data frame) with 3 columns in which column 1 is the x-axis, column 2 is the mean tracked frequency (in cycles/metres) column 3 1 standard deviation

`tracked_cycle_period`
: Period of the tracked curve in kyr.

`tracked_cycle_period_unc`
: uncertainty in the period of the tracked cycle

`tracked_cycle_period_unc_dist`
: distribution of the uncertainty of the tracked cycle value need to be either "u" for uniform distribution or "n" for normal distribution `Default="n"`

`n_simulations`
: number of time series to be modeled

`output`
: If output = 1 a matrix with the predicted ages given the input for each run is given. If output = 2 a matrix with 6 columns is generated, the first column is depth/height, the other columns are the quantile (0.025,0.373,0.5,0.6827,0.975) age values of the runs if output = 3 a matrix with 4 columns is generated with the first column being depth/height, column 2 is the mean tracked duration (in kyrs) column 3 is mean duration + 1 standard deviation and column 4 is mean duration - 1 standard deviation

Value

If output = 1 a matrix with the predicted ages given the input for each run is given If output = 2 a matrix with 6 columns is generated, the first column is depth/height, the other columns are the quantile (0.02275, 0.373, 0.5, 0.6827, 0.97725) age values of the runs if output = 3 a matrix with 4 columns is generated with the first column being depth/height, column 2 is the mean tracked duration (in kyrs) column 3 is mean duration + 1 standard deviation and column 4 is mean duration - 1 standard deviation

Author(s)

Based on the sedrate2time function of the 'astrochron' R package

References

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

Examples

```
# Re-track the 110kyr eccentricity cycle in the wavelet scalogram
# from the XRF record of the Bisciaro data set of Arts (2014) and then
# add generate and age model including uncertainty

Bisciaro_al <- Bisciaro_XRF[, c(1, 61)]
Bisciaro_al <- astrochron::sortNave(Bisciaro_al,verbose=FALSE,genplot=FALSE)
Bisciaro_al <- astrochron::linterp(Bisciaro_al, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_al <- Bisciaro_al[Bisciaro_al[, 1] > 2, ]

Bisciaro_al_wt <-
 analyse_wavelet(
   data = Bisciaro_al,
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_al_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_al_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_al_wt_track <- completed_series(
#   wavelet = Bisciaro_al_wt,
#   tracked_curve = Bisciaro_al_wt_track,
#   period_up = 1.2,
```

```
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_al_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_al_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_ca <- Bisciaro_XRF[, c(1, 55)]
Bisciaro_ca <- astrochron::sortNave(Bisciaro_ca,verbose=FALSE,genplot=FALSE)
Bisciaro_ca <- astrochron::linterp(Bisciaro_ca, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_ca <- Bisciaro_ca[Bisciaro_ca[, 1] > 2, ]

Bisciaro_ca_wt <-
 analyse_wavelet(
   data = Bisciaro_ca,
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_ca_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_ca_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_ca_wt_track <- completed_series(
#   wavelet = Bisciaro_ca_wt,
#   tracked_curve = Bisciaro_ca_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_ca_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_ca_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
```

```
#     keep_editable = FALSE)

Bisciaro_sial <- Bisciaro_XRF[,c(1,64)]
Bisciaro_sial <- astrochron::sortNave(Bisciaro_sial,verbose=FALSE,genplot=FALSE)
Bisciaro_sial <- astrochron::linterp(Bisciaro_sial, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_sial <- Bisciaro_sial[Bisciaro_sial[, 1] > 2, ]

Bisciaro_sial_wt <-
 analyse_wavelet(
   data = Bisciaro_sial,
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_sial_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_sial_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_sial_wt_track <- completed_series(
#   wavelet = Bisciaro_sial_wt,
#   tracked_curve = Bisciaro_sial_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_sial_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_sial_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )


Bisciaro_Mn <- Bisciaro_XRF[,c(1,46)]
Bisciaro_Mn <- astrochron::sortNave(Bisciaro_Mn,verbose=FALSE,genplot=FALSE)
Bisciaro_Mn <- astrochron::linterp(Bisciaro_Mn, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_Mn <- Bisciaro_Mn[Bisciaro_Mn[, 1] > 2, ]

Bisciaro_Mn_wt <-
 analyse_wavelet(
```

```
   data = Bisciaro_Mn,
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_Mn_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mn_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mn_wt_track <- completed_series(
#   wavelet = Bisciaro_Mn_wt,
#   tracked_curve = Bisciaro_Mn_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
# Bisciaro_Mn_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mn_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_Mg <- Bisciaro_XRF[,c(1,71)]
Bisciaro_Mg <- astrochron::sortNave(Bisciaro_Mg,verbose=FALSE,genplot=FALSE)
Bisciaro_Mg <- astrochron::linterp(Bisciaro_Mg, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_Mg <- Bisciaro_Mg[Bisciaro_Mg[, 1] > 2, ]

Bisciaro_Mg_wt <-
 analyse_wavelet(
   data = Bisciaro_Mg,
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_Mg_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
```

```
#     wavelet = Bisciaro_Mg_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mg_wt_track <- completed_series(
#   wavelet = Bisciaro_Mg_wt,
#   tracked_curve = Bisciaro_Mg_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_Mg_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mg_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE)



wt_list_bisc <- list(Bisciaro_al_wt,
               Bisciaro_ca_wt,
               Bisciaro_sial_wt,
               Bisciaro_Mn_wt,
               Bisciaro_Mg_wt)

#Instead of tracking, the tracked solution data sets Bisciaro_al_wt_track,
#Bisciaro_ca_wt_track, Bisciaro_sial_wt_track, Bisciaro_Mn_wt_track,
# Bisciaro_Mn_wt_track and Bisciaro_Mg_wt_track are used

data_track_bisc <- cbind(Bisciaro_al_wt_track[,2],
                     Bisciaro_ca_wt_track[,2],
                     Bisciaro_sial_wt_track[,2],
                     Bisciaro_Mn_wt_track[,2],
                     Bisciaro_Mg_wt_track[,2])

x_axis_bisc <- Bisciaro_al_wt_track[,1]


bisc_retrack <- retrack_wt_MC(wt_list = wt_list_bisc,
             data_track = data_track_bisc,
             x_axis = x_axis_bisc,
             nr_simulations = 20,
             seed_nr = 1337,
             verbose = FALSE,
             genplot = FALSE,
```

```
                keep_editable = FALSE,
                create_GIF = FALSE,
                plot_GIF = FALSE,
                width_plt =  600,
                height_plt = 450,
               period_up  =  1.5,
                period_down = 0.5,
                plot.COI = TRUE,
                n.levels = 100,
                palette_name = "rainbow",
                color_brewer = "grDevices",
                periodlab = "Period (metres)",
                x_lab = "depth (metres)",
                add_avg = FALSE,
                time_dir = TRUE,
                file_name = NULL,
                run_multicore = FALSE,
                output = 5,
                n_imgs = 50,
                plot_horizontal = TRUE,
                empty_folder = FALSE)

bisc_retrack_age_incl_unc <- curve2time_unc(tracked_cycle_curve = bisc_retrack,
tracked_cycle_period = 110,
tracked_cycle_period_unc = ((135-110)+(110-95))/2,
tracked_cycle_period_unc_dist = "n",
n_simulations = 20,
output = 1)
```

`curve2time_unc_anchor` *Convert the re-tracked curve results to a depth time space with uncertainty*

Description

Converts the re-tracked curve results from `retrack_wt_MC` function to a depth time space using an anchor date while also taking into account the uncertainty of the tracked astronomical cycle

Usage

```
curve2time_unc_anchor(
  age_constraint = NULL,
  tracked_cycle_curve = NULL,
  tracked_cycle_period = NULL,
  tracked_cycle_period_unc = NULL,
  tracked_cycle_period_unc_dist = "n",
```

```
  n_simulations = 20,
  gap_constraints = NULL,
  proxy_data = NULL,
  cycles_check = NULL,
  uncer_cycles_check = NULL,
  max_runs = 1000,
  run_multicore = FALSE,
  n_cores = NULL,
  verbose = FALSE,
  genplot = FALSE,
  keep_nr = 2,
  keep_all_time_curves = FALSE,
  dj = 1/200,
  lowerPeriod = 1,
  upperPeriod = 4600,
  omega_nr = 6,
  seed_nr = 1337,
  dir = TRUE
)
```

Arguments

`age_constraint`	age constrains for the modelling run Input should be a data frame with 7 columns, the first columns are the ID names the second column are the ages (usually in kyr) the third column is the uncertainty (usually in kyr) given as the fourth column is the distribution which is either "n" for a normal distribution or "u" for a uniform distribution. The fifth column is the location in the depth domain of the age constraint. the sixth column is the location/thickness uncertainty of the age_constraint in the depth domain. The seventh column is the uncertainty distribution of the age_constrain in the depth domain
`tracked_cycle_curve`	Curve of the cycle tracked using the `retrack_wt_MC` function Any input (matrix or data frame) with 3 columns in which column 1 is the x-axis, column 2 is the mean tracked frequency (in cycles/metres) column 3 1 standard deviation
`tracked_cycle_period`	Period of the tracked curve in kyr.
`tracked_cycle_period_unc`	uncertainty in the period of the tracked cycle
`tracked_cycle_period_unc_dist`	distribution of the uncertainty of the tracked cycle value need to be either "u" for uniform distribution or "n" for normal distribution `Default="n"`
`n_simulations`	number of time series to be modeled `Default=20`
`gap_constraints`	gap parameters for the modelling run input should be a data frame with
`proxy_data`	proxy data to be tune and check preservation of astronomical cycles
`cycles_check`	astronomical cycles which are checked for their presence after tuning

uncer_cycles_check
: uncertainty of astronomical cycles to be check for after tunning

max_runs
: maximum runs before one of the age constraints is dropped Default=1000

run_multicore
: Run function using multiple cores Default="FALSE"

n_cores
: Number of cores used when 'run_multicore = TRUE' ('Default = NULL': an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set 'options(waverider.n_cores = ...)' to change the default. Only used when 'run_multicore = TRUE'. Results do not depend on the number of cores.

verbose
: Print text Default=FALSE.

genplot
: generate plot codeDefault=FALSE

keep_nr
: minimal number of age constraints to be kept Default=2

keep_all_time_curves
: weather to keep all the generated age curves including the ones rejected from the modelling run Default=FALSE

dj
: Spacing between successive scales. The CWT analyses analyses the signal using successive periods which increase by the power of 2 (e.g.2^0=1,2^1=2,2^2=4,2^3=8,2^4=16). To have more resolution in-between these steps the dj parameter exists, the dj parameter specifies how many extra steps/spacing in-between the power of 2 scaled CWT is added. The amount of steps is 1/x with a higher x indicating a smaller spacing. Increasing the increases the computational time of the CWT Default=1/200.

lowerPeriod
: Lowest period to be analysed Default=2. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyse the signal for a specific range of cycles. 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 analysed Default=1024. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyse the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.

omega_nr
: The omega number: central angular frequency of the Morlet wavelet

seed_nr
: The seed number of the Monte-Carlo simulations. Default=1337

dir
: time direction of tuning e.g. does time increase or decrease with depth

Value

The output is a list of 3 or 4 elements if keep_all_time_curves is set to TRUE then the list consist of the x-axis, all the fitted curves in a matrix format, the astrochronologically fitted age of the anchor, all the generated depth time curves if keep_all_time_curves is set to TRUE then the list consists of the x-axis, all the fitted curves in a matrix format and the astrochronologically fitted age of the anchor If genplot=TRUE then 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. #'

Author(s)

Part of the code is based on the sedrate2time function of the 'astrochron' R package

References

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

Examples

```
## Not run:


Bisciaro_al <- Bisciaro_XRF[, c(1, 61)]
Bisciaro_al <-
 astrochron::sortNave(Bisciaro_al, verbose = FALSE, genplot = FALSE)
Bisciaro_al <-
 astrochron::linterp(Bisciaro_al,
                     dt = 0.01,
                     verbose = FALSE,
                     genplot = FALSE)
Bisciaro_al <- Bisciaro_al[Bisciaro_al[, 1] > 2, ]

Bisciaro_al_wt <-
 analyse_wavelet(
   data = Bisciaro_al,
   dj = 1 / 200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )
# Bisciaro_al_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_al_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_al_wt_track <- completed_series(
#   wavelet = Bisciaro_al_wt,
#   tracked_curve = Bisciaro_al_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_al_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_al_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
```

```
#     keep_editable = FALSE
#   )



Bisciaro_ca <- Bisciaro_XRF[, c(1, 55)]
Bisciaro_ca <-
 astrochron::sortNave(Bisciaro_ca, verbose = FALSE, genplot = FALSE)
Bisciaro_ca <-
 astrochron::linterp(Bisciaro_ca,
                     dt = 0.01,
                     verbose = FALSE,
                     genplot = FALSE)
Bisciaro_ca <- Bisciaro_ca[Bisciaro_ca[, 1] > 2, ]

Bisciaro_ca_wt <-
 analyse_wavelet(
   data = Bisciaro_ca,
   dj = 1 / 200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )


#
# Bisciaro_ca_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_ca_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_ca_wt_track <- completed_series(
#   wavelet = Bisciaro_ca_wt,
#   tracked_curve = Bisciaro_ca_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_ca_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_ca_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )
```

```
Bisciaro_sial <- Bisciaro_XRF[, c(1, 64)]
Bisciaro_sial <-
 astrochron::sortNave(Bisciaro_sial, verbose = FALSE, genplot = FALSE)
Bisciaro_sial <-
 astrochron::linterp(Bisciaro_sial,
                     dt = 0.01,
                     verbose = FALSE,
                     genplot = FALSE)
Bisciaro_sial <- Bisciaro_sial[Bisciaro_sial[, 1] > 2, ]

Bisciaro_sial_wt <-
 analyse_wavelet(
   data = Bisciaro_sial,
   dj = 1 / 200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

#Bisciaro_sial_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_sial_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_sial_wt_track <- completed_series(
#   wavelet = Bisciaro_sial_wt,
#   tracked_curve = Bisciaro_sial_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_sial_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_sial_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )


Bisciaro_Mn <- Bisciaro_XRF[, c(1, 46)]
Bisciaro_Mn <-
```

```
 astrochron::sortNave(Bisciaro_Mn, verbose = FALSE, genplot = FALSE)
Bisciaro_Mn <-
 astrochron::linterp(Bisciaro_Mn,
                     dt = 0.01,
                     verbose = FALSE,
                     genplot = FALSE)
Bisciaro_Mn <- Bisciaro_Mn[Bisciaro_Mn[, 1] > 2, ]

Bisciaro_Mn_wt <-
 analyse_wavelet(
   data = Bisciaro_Mn,
   dj = 1 / 200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_Mn_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mn_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mn_wt_track <- completed_series(
#   wavelet = Bisciaro_Mn_wt,
#   tracked_curve = Bisciaro_Mn_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
# Bisciaro_Mn_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mn_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_Mg <- Bisciaro_XRF[, c(1, 71)]
Bisciaro_Mg <-
 astrochron::sortNave(Bisciaro_Mg, verbose = FALSE, genplot = FALSE)
Bisciaro_Mg <-
 astrochron::linterp(Bisciaro_Mg,
                     dt = 0.01,
                     verbose = FALSE,
                     genplot = FALSE)
```

```
Bisciaro_Mg <- Bisciaro_Mg[Bisciaro_Mg[, 1] > 2, ]

Bisciaro_Mg_wt <-
 analyse_wavelet(
   data = Bisciaro_Mg,
   dj = 1 / 200 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_Mg_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mg_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mg_wt_track <- completed_series(
#   wavelet = Bisciaro_Mg_wt,
#   tracked_curve = Bisciaro_Mg_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_Mg_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mg_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )




wt_list_bisc <- list(Bisciaro_al_wt,
                     Bisciaro_ca_wt,
                     Bisciaro_sial_wt,
                     Bisciaro_Mn_wt,
                     Bisciaro_Mg_wt)


data_track_bisc <- cbind(
 Bisciaro_al_wt_track[, 2],
 Bisciaro_ca_wt_track[, 2],
```

```
Bisciaro_sial_wt_track[, 2],
 Bisciaro_Mn_wt_track[, 2],
 Bisciaro_Mg_wt_track[, 2]
)

x_axis_bisc <- Bisciaro_al_wt_track[, 1]

bisc_retrack <- retrack_wt_MC(
wt_list = wt_list_bisc,
 data_track = data_track_bisc,
 x_axis = x_axis_bisc,
 nr_simulations = 500,
 seed_nr = 1337,
 verbose = TRUE,
 genplot = FALSE,
 keep_editable = FALSE,
 create_GIF = FALSE,
 plot_GIF = FALSE,
 width_plt =  600,
 height_plt = 450,
 period_up  =  1.5,
 period_down = 0.5,
 plot.COI = TRUE,
 n.levels = 100,
 palette_name = "rainbow",
 color_brewer = "grDevices",
periodlab = "Period (metres)",
x_lab = "depth (metres)",
add_avg = FALSE,
time_dir = TRUE,
file_name = "TEST",
run_multicore = TRUE,
output = 5,
 n_imgs = 50,
 plot_horizontal = TRUE,
 empty_folder = FALSE
)

proxy_list_bisc <- list(Bisciaro_al,
                    Bisciaro_ca,
                    Bisciaro_sial,
                    Bisciaro_Mn,
                    Bisciaro_Mg)



id <- c("CCT18_322", "CCT18_315", "CCT18_311")
ages <- c(20158, 20575, 20857)
ageSds <- c(28, 40, 34)
ages_unc_dist <- c("n", "n", "n")
position <- c(13.3, 7.25, 3.2)
anchor_thick <- c(0.2, 0.1, 0.1)
```

```
anchor_thick_unc_dist <- c("u", "u", "u")

ash_Bisc <-
 as.data.frame(
   cbind(
     id,
     ages,
     ageSds,
     ages_unc_dist,
     position,
     anchor_thick,
     anchor_thick_unc_dist
   )
 )

gap_dur = c(10, 20)
gap_unc = c(3, 10)
gap_depth = c(2.5, 9)
gap_unc_dist = c("n", "n")


gap_constraints_Bisc <-
 as.data.frame(cbind(gap_dur, gap_unc, gap_depth, gap_unc_dist))

cycles_checks <- c(110,40,22)
uncer_cycles_checks <- c(20,5,7)

curve2time_unc_anchor_res <-
 curve2time_unc_anchor(
 age_constraint =  ash_Bisc,
  tracked_cycle_curve =  bisc_retrack,
  tracked_cycle_period = 110,
   tracked_cycle_period_unc = ((135 - 110) + (110 - 95)) / 2,
  tracked_cycle_period_unc_dist = "n",
   n_simulations = 20,
   gap_constraints = gap_constraints_Bisc,
   proxy_data = proxy_list_bisc,
   cycles_check = NULL,
   uncer_cycles_check = NULL,
   cycles_check = cycles_checks,
   uncer_cycles_check = uncer_cycles_checks,
  max_runs = 1000,
   run_multicore = FALSE,
   verbose = FALSE,
   genplot = FALSE,
   keep_nr = 2,
   keep_all_time_curves = FALSE,
   dj = 1/200,
   lowerPeriod =1,
   upperPeriod =2500,
   omega_nr = 6,
   seed_nr=1337,
   dir=TRUE
```

```
)

## End(Not run)
```

---

curve2tune *Convert 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 = FALSE,
  keep_editable = FALSE
)
```

Arguments

data	Data set (matrix with 2 columns 1st column depth 2nd column proxy value) which was used as input for the analyse_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.
keep_editable	Keep option to add extra features after plotting Default=FALSE

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 data set. Plot 2: the depth time plot. Plot 3: the data set in the time domain.

Author(s)

Part of the code is based on the sedrate2time function of the 'astrochron' R 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 Pas et al., (2018).
# perform the CWT
mag_wt <- analyse_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
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 (meters)",
#                                  x_lab = "depth (meters)")

#Instead of tracking, the tracked solution data set 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 = FALSE
)

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

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

cvcoco *Cross-validated correlation coefficient (cvCOCO) analysis*

Description

The cvcoco function performs the bidirectional held-out ("cross-validated") correlation coefficient (cvCOCO) test of Acycle for astronomical forcing in a depth series. The cleaned record is split into two folds: two contiguous halves at the depth midpoint (`method = "blocked"`, Blocked cvCOCO, the Acycle public default) or the odd and even observations (`method = "interleaved"`, Interleaved cvCOCO). Each fold is linearly interpolated to its own median spacing. In each direction one fold is used for training: at every tested sedimentation rate the energies in four one-Rayleigh union bands (long eccentricity, short eccentricity, obliquity, precession) are leakage-corrected with an exact nonnegative least-squares solve of the finite-record 4x4 leakage matrix, and the rate with the highest spectral correlation gives four trained group amplitudes. These amplitudes are frozen and the held-out fold searches its own sedimentation rate (validation). The same-rate minimum of both directional validation curves is the consensus curve. A single outer stationary AR(1) Monte Carlo reproduces the complete bidirectional train/validate pipeline and gives global (rate-search corrected) and local p-values, the robust directional p-value max(pA, pB) and pCOCO. Two target designs are available: `"four-group-coherent-nine"` (the four group amplitudes expanded to nine zero-phase sinusoids summed coherently; Acycle cvcoco9B.m / interleavedcvcoco.m, the GUI default) and `"four-group"` (phase-averaged noncoherent four-group target; Acycle cvcoco2.m, for which Acycle issues the confirmatory conclusion report of cocoConclusionReport.m). The COCO method was introduced by Li et al. (2018). Translated from Acycle v3.1 cvcoco.m, cvcoco2.m, cvcoco9B.m, interleavedcvcoco.m, cocoConclusionReport.m and helpers (Mingsong Li).

This function calls the following helpers of the cocoR port (which must be loaded): `coco_orbit_groups`, `detrend_linear`, `ml_periodogram` and `theored_ar1_ml`; and, for `red = 3`, `swa_background`.

Usage

```
cvcoco(
  data = NULL,
 orbit9 = c(405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897, 23.682, 22.3758,
    18.9519),
  pad = 0,
  sr1 = 1,
  sr2 = 20,
  srstep = 0.1,
  red = 0,
  n_sim = 1000,
  cor_method = "Pearson",
  method = "blocked",
  target_model = "four-group-coherent-nine",
  max_frequency = NULL,
  batch_size = 100,
  seed = NULL,
  interleaved_phase = 0,
```

```
  genplot = FALSE,
  verbose = FALSE
)
```

Arguments

data	Matrix with two columns: depth (m) and proxy value. Unsorted, duplicated and irregularly spaced depths are allowed.
orbit9	Nine astronomical periods (kyr): one long eccentricity, four short eccentricity, one obliquity and three precession periods (groups are identified by descending period rank). Default=c(405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897,23.6820, 22.3758, 18.9519)
pad	Periodogram length (zero padding); each fold uses max(pad, n_fold). Default=0 (no padding, as in the Acycle GUI when padding is unchecked)
sr1	Minimum sedimentation rate (cm/kyr). Default=1
sr2	Maximum sedimentation rate (cm/kyr). Default=20
srstep	Sedimentation-rate step (cm/kyr). Default=0.1
red	Red-noise background removal: 0 none, 1 classical AR(1), 2 robust AR(1), 3 smoothed-window average (SWA). Default=0
n_sim	Number of Monte Carlo simulations (0 gives only the deterministic curves). Default=1000
cor_method	Correlation method, "Pearson" or "Spearman". Default="Pearson"
method	Split design: "blocked" (depth midpoint halves) or "interleaved" (odd/even observations). Default="blocked"
target_model	"four-group-coherent-nine" or "four-group" (only for method = "blocked"). Default="four-group-coh
max_frequency	Maximum temporal frequency (cycle/kyr) used in all spectral correlations. NULL uses 1.2 times the highest orbital frequency. Default=NULL
batch_size	Monte Carlo realisations processed simultaneously. Default=100
seed	Seed for the random-number generator; NULL leaves the generator untouched. Default=NULL
interleaved_phase	0: the first cleaned observation belongs to the Odd fold; 1: it belongs to the Even fold. Default=0
genplot	Plot the results with plot_cvcoco. Default=FALSE
verbose	Print preprocessing information and a progress bar. Default=FALSE

Value

A list with elements name ("Blocked cvCOCO" or "Interleaved cvCOCO"), target_model, sr_grid (tested rates), split_depth, fold_labels, data_clean, data_A, data_B (regularized folds), raw_index_A, raw_index_B, interpolation_A, interpolation_B, sampling_interval_A, sampling_interval_B, train_A, train_B (training curve, best rate, amplitudes, group amplitudes, frozen weights, leakage matrix at best rate), validate_AtoB, validate_BtoA (validation curve, best rate, score, directional p, Wilson interval, global and local p curves, participating periods), consensus (same-rate

minimum curve, best rate, global p, p curves, pCOCO curve and maximum), `score_symmetric`, `score_consensus`, `score_mean`, `p_sym`, `p_consensus`, `p_robust`, `p_A` (B-to-A, fold A held out), `p_B` (A-to-B, fold B held out), their confidence intervals, `p_curve_AtoB`, `p_curve_BtoA`, `p_local_curve_AtoB`, `p_local_curve_BtoA`, `p_curve_consensus`, `p_local_curve_consensus`, `pcoco`, null distributions (`null_symmetric`, `null_consensus`, `null_AtoB`, `null_BtoA`, `null_best_rate_AtoB`, `null_best_rate_BtoA`), AR(1) coefficients (`rho_A`, `rho_B`, `rho_full`), geometry diagnostics (`orbit_count_A/B`, `resolved_group_count_A/B`, `resolved_orbit_mask_A/B`, `valid_rate_mask_A/B`, `training_rate_mask_A/B`, `group_leakage_rcond_A/B`, `all_nine_rate_range_A/B/shared`, `active_orbit_count_AtoB/BtoA/ consensus`, `active_group_count_AtoB/BtoA`) `degraded_mode`, `training_completeness`, `spectra` (data and frozen-target temporal spectra at the best rates), `n_sim_completed`, `config`, and `conclusion` (confirmatory report; only for `target_model = "four-group"`, otherwise `NULL`).

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li.

References

Li, M., Kump, L.R., Hinnov, L.A., Mann, M.E., 2018. Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing. Earth and Planetary Science Letters 501, 165-179. doi:10.1016/j.epsl.2018.08.041

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22. doi:10.1016/j.cageo.2019.02.011

Examples

```
# synthetic ETP-like series in depth (5 cm/kyr) with AR(1) noise
set.seed(2)
t_kyr <- seq(0, 1600, by = 2)
etp <- sin(2 * pi * t_kyr / 405) + 0.6 * sin(2 * pi * t_kyr / 125) +
  0.5 * sin(2 * pi * t_kyr / 95) + 0.8 * sin(2 * pi * t_kyr / 41) +
  0.5 * sin(2 * pi * t_kyr / 23.7) + 0.4 * sin(2 * pi * t_kyr / 19)
noise <- as.numeric(stats::filter(stats::rnorm(length(t_kyr)), 0.5,
  method = "recursive"))
dat <- cbind(t_kyr * 5 / 100, etp + 0.5 * noise)
res <- cvcoco(data = dat, sr1 = 2, sr2 = 10, srstep = 0.5,
  n_sim = 50, seed = 1)
res$consensus$best_rate
res$p_robust
plot_cvcoco(res)
```

---

cycle_band *Period range implied by the frequency resolution of a record*

Description

Period range implied by the frequency resolution of a record

Usage

```
cycle_band(period, record_length = NULL, tol = NULL, mult = 1)
```

Arguments

period	the measured cycle period (kyr) or wavelength (m)
record_length	length of the analysed record, same units as 'period' (kyr for a tuned record, m for a depth series). The half-width used is 'mult / record_length' in frequency.
tol	alternative: a relative half-width (0.05 = +-5%)
mult	multiples of the Rayleigh resolution to use (1 = +-1 bandwidth)

Value

c(min, max) period range

---

cycles_to_k *Precession constant implied by an observed obliquity or precession cycle*

Description

Precession constant implied by an observed obliquity or precession cycle

Usage

```
cycles_to_k(period_kyr, term = "k+s3", g = NULL, s = NULL, data_dir = NULL)
```

Arguments

period_kyr	the cycle period read from the record (kyr)
term	which astronomical term it is: "k+s3" (obliquity), or one of the precession terms "k+g5", "k+g2", "k+g4", "k+g3"
g	Optional g frequencies (arcsec/yr) overriding the defaults.
s	Optional s frequencies (arcsec/yr) overriding the defaults.
data_dir	Directory with the data files ('NULL' = the data installed with the package).

Value

list: p in arcsec/yr and the precession constant k in years

days_per_year *Days in a year for a given solar day length*

Description

Days in a year for a given solar day length

Usage

```
days_per_year(lod_h)
```

Arguments

lod_h	Length of day in hours.

delpts_tracked_period_wt *Remove tracking points which were tracked in a wavelet spectra*

Description

Interactively select points for deletion With the track_period_wavelet function it is possible to track points in a wavelet spectra, however errors can be made and as such it is possible to delete these points with the delpts_tracked_period_wt function. This function allows one to select points for deletion. #'

Usage

```
delpts_tracked_period_wt(
  tracking_pts = NULL,
  wavelet = NULL,
  n.levels = 100,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  palette_name = "rainbow",
  color_brewer = "grDevices",
  colour_brewer = NULL
)
```

Arguments

tracking_pts	Points tracked using the track_period_wavelet function.
wavelet	Wavelet object created using the analyse_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)".
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

The results of the deletion of the tracking points 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

```
if (interactive()) { # needs mouse input (identify())
#Track the 405kyr eccentricity cycle in the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018)

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

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

#load the mag_track_solution data set to get an example data set from which
```

```
#data points can be deleted


mag_track_corr <- delpts_tracked_period_wt(tracking_pts = mag_track_solution,
                                    wavelet = mag_wt,
                                    n.levels = 100,
                                    periodlab = "Period (metres)",
                                    x_lab = "depth (metres)",
                                   palette_name ="rainbow",
                                   color_brewer ="grDevices")
}
```

`depth_rank_example` *An example depth rank series*

Description

The `depth_rank_example` example data set is a depth rank series which can be used as input for the `lithlog_disc` function which creates a discritzed record which can then be used as input in the `analyse_wavelet` function

Details

Column 1: depth (meters)
Column 2: depth rank

`detect_transients` *Transient (impulsive) features of a record by transient extraction*

Description

Finds the positions in a record where energy over many periods arrives at the same instant: the signature of an impulsive feature such as a hiatus (a phase jump of all cycles), an erosion surface, a turbidite or a spike. The wavelet coefficients are reassigned along the record to their group delay (the time-reassigned synchrosqueezing / transient-extracting transform, He et al. 2019; Yu et al. 2023), only the coefficients that lie on their own group delay are kept, and their energy is summed over the periods. The sum is compared with a running median and peaks above `threshold` times the median are reported, which can be passed as `jump_depths` to `track_astro_hmm` or `track_astro_auto`.

Usage

```
detect_transients(
  data,
  lowerPeriod,
  upperPeriod,
  dj = 1/20,
  omega_nr = 6,
  threshold = 4,
  min_distance = NULL,
  window = NULL
)
```

Arguments

`data`	Record: first column depth, second column proxy.
`lowerPeriod, upperPeriod, dj`	Period range and spacing of the transform.
`omega_nr`	Morlet omega_0 `Default=6` (short wavelets localise transients best).
`threshold`	Peaks above this multiple of the running median of the transient energy are reported `Default=4`.
`min_distance`	Minimum distance between reported transients (depth units) `Default=upperPeriod / 4`.
`window`	Length (depth units) of the running median `Default=2 * upperPeriod`.

Details

The transform cannot see a jump of the sedimentation rate itself (the cycles stay continuous); it sees the phase jumps that a gap in time causes.

Value

A list with `profile` (data frame: depth, energy, ratio to the running median) and `transients` (data frame: depth, ratio of the detected peaks, strongest first).

References

He, D., Cao, H., Wang, S. and Chen, X. (2019). Time-reassigned synchrosqueezing transform: the algorithm and its applications in mechanical signal processing. Mech. Syst. Signal Process. 117, 255-279.

Yu, G. et al. (2023). Synchro-transient-extracting transform for the analysis of signals with both harmonic and impulsive components. arXiv:2306.01320.

Examples

```
set.seed(2)
t <- seq(0, 2000, by = 0.5)
x_t <- sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 41) + 0.4 * cos(2 * pi * t / 405)
keep <- !(t > 800 & t < 900) & !(t > 1500 & t < 1550)       # two gaps in time (hiatuses)
```

```
z <- seq(0, by = 2, length.out = sum(keep))
x <- x_t[keep] + rnorm(length(z), sd = 0.5)
tr <- detect_transients(data.frame(z, x), lowerPeriod = 20, upperPeriod = 400)
tr$transients
```

---

dur_gaps *calculate the duration of stratigraphic gaps using astronomical cycles*

Description

calculate the duration of stratigraphic gaps using the duration of stable astronomical cycles

Usage

```
dur_gaps(
  proxies = NULL,
  retracked_period_1 = NULL,
  retracked_period_2 = NULL,
  min_max = NULL,
  n_simulations = 10,
  tracked_cycle_period = NULL,
  tracked_cycle_period_unc = NULL,
  tracked_cycle_period_unc_dist = "u",
  pts = 5,
  dj = 1/200,
  lowerPeriod = 1,
  upperPeriod = 3200,
  period_max = NULL,
  period_min = NULL,
  missing_cycle_dur = NULL,
  n_cycles_missing = 1,
  missing_cycle_unc = NULL,
  missing_cycle_unc_dist = "u",
  seed_nr = 1337,
  run_multicore = FALSE,
  n_cores = NULL
)
```

Arguments

proxies	list of proxies which were used to create a astrochronological age model and which are used to calculate the duration of the gap
retracked_period_1	A matrix of 3 columns in which the first column is depth/height.The second column is the period of the tracked cycle. The thirds column is uncertainty given as 1 standard deviation for the period of the tracked cycle. The gap to be modeled should be located in between retracked_period_1 and retracked_period_2

retracked_period_2
: A matrix of 3 columns in which the first column is depth/height.The second column is the period of the tracked cycle. The thirds column is uncertainty given as 1 standard deviation for the period of the tracked cycle. The gap to be modeled should be located in between retracked_period_1 and retracked_period_2

min_max
: list of "min" or "max" indicating whether time should be estimated between minima or maxima for each proxy

n_simulations
: number of gap duration to calculate

tracked_cycle_period
: period in time of the tracked cycle

tracked_cycle_period_unc
: uncertainty in the period of the tracked cycle

tracked_cycle_period_unc_dist
: distribution of the uncertainty of the tracked cycle value need to be either "u" for uniform distribution or "n" for normal distribution Default="u"

pts
: the pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm Default=5#'

dj
: Spacing between successive scales. The CWT analyses analyses the signal using successive periods which increase by the power of 2 (e.g.2^0=1,2^1=2,2^2=4,2^3=8,2^4=16). To have more resolution in-between these steps the dj parameter exists, the dj parameter specifies how many extra steps/spacing in-between the power of 2 scaled CWT is added. The amount of steps is 1/x with a higher x indicating a smaller spacing. Increasing the increases the computational time of the CWT Default=1/200.

lowerPeriod
: Lowest period to be analysed Default=2. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyse the signal for a specific range of cycles. 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 analysed Default=1024. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyse the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.

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.

missing_cycle_dur
: duration of the missing cycles

n_cycles_missing
: number of missing cycles Default=1

missing_cycle_unc
: duration uncertainty of the missing cycle

`missing_cycle_unc_dist`
: distribution of the uncertainty of the tracked cycle value need to be either "u" for uniform distribution or "n" for normal distribution `Default="u"`

`seed_nr`
: The seed number of the Monte-Carlo simulations. `Default=1337`

`run_multicore`
: Run function using multiple cores `Default="FALSE"`

`n_cores`
: Number of cores used when 'run_multicore = TRUE' ('Default = NULL': an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set 'options(waverider.n_cores = ...)' to change the default. Only used when 'run_multicore = TRUE'. Results do not depend on the number of cores.

Value

a vector with all the calculated gap durations

`dynamic_extraction` *Extract a signal in between tracked boundaries in a wavelet scalogram*

Description

Interactively select points in a wavelet scalogram to trace the upper and lower period of an cycle. The `dynamic_extraction` function plots a wavelet scalogram in which points peaks can selected allowing one to track the lower and upper period of a cycle. First track the upper or lower period of the to be extracted cycle and then track the other boundary. Tracking points can be selected in the Interactive interface and will be shown as white dots connected by a black line. When one wants to deselect a point the white dots can be re-clicked/re-selected and will turn red which indicates that the previously selected point is deselected. Deselecting points can be quite tricky. After tracking the lower and upper boundaries of the cycle the `dynamic_extraction` function will extract the signal in between the boundaries. the output can then used as input for the `minimal_tuning` function to create an age model.

Usage

```
dynamic_extraction(
  wavelet = NULL,
  n.levels = 100,
  add_peaks = FALSE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  palette_name = "rainbow",
  color_brewer = "grDevices",
  plot_horizontal = TRUE,
  smooth = FALSE,
  add_mean = TRUE,
  colour_brewer = NULL
)
```

Arguments

wavelet	Wavelet object created using the analyse_wavelet function.
n.levels	Number of color levels Default=100.
add_peaks	Setting which indicates whether spectral peaks should be added to the tracking plot Default=FALSE.
periodlab	label for the y-axis Default="Period (metres)".
x_lab	label for the x-axis Default="depth (metres)".
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
plot_horizontal	plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE
smooth	smooth the tracked period using the "loess_auto" function
add_mean	add the mean to the extracted signal
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

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 extracted tracked cycle The third column is upper tracked period The fourth column is lower tracked period

Author(s)

The function is based/inspired on the traceFreq function of the 'astrochron' R package

References

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

Examples

```
## Not run:
#Track the 405kyr upper and lower periods of the eccentricity cycle in the
#magnetic susceptibility record of the Sullivan core of Pas et al., (2018)

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

mag_ext <- dynamic_extraction(
 wavelet = mag_wt,
 n.levels = 100,
 add_peaks = FALSE,
 periodlab = "Period (metres)",
 x_lab = "depth (metres)",
 palette_name = "rainbow",
 color_brewer = "grDevices",
 plot_horizontal = TRUE,
 smooth = TRUE,
 add_mean = TRUE
)

## End(Not run)
```

`dynamic_filter` *Dynamic (depth-varying) Gaussian band-pass filter*

Description

The `dynamic_filter` function extracts a cycle whose frequency changes along the depth/time axis (e.g. because of a changing sedimentation rate) using a pass band that follows user-defined control points. The method is the dynamic filter of Nicolas Thibault and Giovanni Rizzi (2019), adapted by Mingsong Li (2020) for Acycle's "Frequency Stabilization" tool, in the deterministic form of Acycle's current core: the series is divided into sliding windows; in each window the pass band is obtained by linear interpolation of the control points at the window centre (held constant beyond the first/last control point), the window is mean-centred, tapered by the square root of a half-sample-shifted Hann window, Gaussian band-pass filtered (half-power edges at the lower/upper frequency, zero padded to `2^nextpow2(2 * window_samples - 1)`), tapered again and overlap-added. The result is divided by the summed Hann weights. In Acycle the control points are clicked on an evolutionary spectrum; here they are supplied as arguments, e.g. built with `dynamic_filter_controls` from a period curve tracked with WaverideR (`track_period_wavelet`, `completed_series`). Translated from Acycle v3.1 acycleDynamicFilter.m (strict core; legacy flow dynamic_filter.m / dynamic_filter_lang.m, GUI DynamicFilter.m) (Mingsong Li).

Usage

```
dynamic_filter(
  data = NULL,
  controls = NULL,
  window_length = NULL,
  step_length = NULL,
  genplot = FALSE,
  x_lab = "depth",
  y_lab = "proxy",
  verbose = FALSE
)
```

Arguments

data	Input data set, a matrix or data frame with 2 columns: the first column being depth/time (strictly increasing, regularly sampled to a relative spacing tolerance of 1e-5; interpolate beforehand otherwise) and the second column being the proxy value.
controls	Pass band control points. Either a matrix/data frame with columns depth, f_low, f_high and optionally f_centre (unnamed columns are taken in that order), or a list with elements lower and upper, each a 2-column matrix [depth, frequency] (Acycle's separate lower/upper boundary picks, which may be at different depths). Depths must be strictly increasing and lie within the data range. A single row gives a constant pass band. Because the Acycle Gaussian response is symmetric, a supplied f_centre must equal (f_low + f_high) / 2 (it is only checked).
window_length	Window length in depth units (first-to-last sample span). Default=NULL: 20 percent of the record length (Acycle GUI default). A window of about two times the longest period of the target cycle is recommended.
step_length	Step between successive windows in depth units. Default=NULL: 4 times the sample interval, or (record length - window_length) / 500 when that gives more than 500 steps (Acycle GUI default).
genplot	Generate a plot using plot_dynamic_filter Default=FALSE.
x_lab	label for the x-axis Default="depth".
y_lab	label for the y-axis Default="proxy".
verbose	Show a progress bar Default=FALSE.

Value

A list with elements

filtered matrix with columns depth, filtered, coverage_count (number of windows covering each sample) and coverage_weight (sum of Hann weights).

windows matrix with one row per window and columns centre, f_low, f_high, f_centre, start_depth, end_depth, start_index, end_index.

lower_controls, upper_controls the 2-column [depth, frequency] control points used.

sample_interval median sample spacing.

nyquist Nyquist frequency.

window_samples, step_samples window and step in samples.

window_length, step_length effective (sample-rounded) window and step in depth units.

fft_length zero-padded FFT length per window.

terminal_window_appended whether a final window ending on the last sample was added.

Author(s)

Michiel Arts (R translation); original MATLAB code: Nicolas Thibault and Giovanni Rizzi (2019), adapted by Mingsong Li (Acycle).

References

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22. <doi:10.1016/j.cageo.2019.02.011>

Thibault, N., Rizzi, G. (2019). Dynamic filtering routine, distributed with Acycle (Frequency Stabilization).

Examples

```
# cycle whose period grows linearly from 5 to 15 m (changing sed. rate)
depth <- seq(0, 300, by = 0.25)
period <- 5 + 10 * depth / 300
phase <- 2 * pi * cumsum(1 / period) * 0.25
set.seed(1)
proxy <- sin(phase) + 0.8 * sin(2 * pi * depth / 2.1) + rnorm(length(depth), sd = 0.3)
ctrl <- dynamic_filter_controls(tracked = cbind(depth, period),
  rel_bandwidth = 0.3, n_points = 30)
res <- dynamic_filter(data = cbind(depth, proxy), controls = ctrl,
  window_length = 40, step_length = 1, genplot = FALSE)
plot(depth, sin(phase), type = "l")
lines(res$filtered[, 1], res$filtered[, 2], col = "red")
```

`dynamic_filter_controls`

Build dynamic-filter control points from a tracked period curve

---

Description

The `dynamic_filter_controls` function converts a tracked period curve [depth, period] into pass band control points for `dynamic_filter`. This replaces the interactive clicking of lower/upper frequency boundaries on an evolutionary spectrum in Acycle (dynamic_filter_lang.m) by a reproducible, non-interactive step. The curve can come directly from the WaverideR functions `track_period_wavelet` (first two columns: depth, period) or `completed_series` (depth, period), so cycles already tracked

in a wavelet scalogram can be filtered with the Acycle dynamic filter. The centre frequency is `1 / period`; the pass band is `f_centre * (1 -/+ rel_bandwidth / 2)`, i.e. `rel_bandwidth` is the half-power bandwidth relative to the centre frequency (the band is symmetric in frequency, as the Acycle Gaussian response is). Written for the R translation of Acycle v3.1 dynamic filtering (Mingsong Li); no MATLAB equivalent.

Usage

```
dynamic_filter_controls(
  tracked = NULL,
  rel_bandwidth = 0.2,
  n_points = NULL,
  data = NULL
)
```

Arguments

`tracked`	Matrix or data frame whose first two columns are depth and period (in depth units) of the tracked cycle. Rows with non-finite values are dropped, rows are sorted by depth and duplicated depths are removed (first kept).
`rel_bandwidth`	Half-power bandwidth relative to the centre frequency, between 0 and 2 (exclusive) `Default=0.2`.
`n_points`	Optional maximum number of control points. When the tracked curve has more rows, it is linearly interpolated to `n_points` evenly spaced depths spanning the (clipped) curve `Default=NULL` (keep all rows).
`data`	Optional data set (2-column matrix) that will be filtered. When given, control points outside its depth range are removed (the Acycle core requires control depths inside the data range) `Default=NULL`.

Value

A matrix with columns `depth`, `f_low`, `f_high` and `f_centre`, directly usable as the `controls` argument of `dynamic_filter`.

Author(s)

Michiel Arts

References

Arts, M., 2023. WaverideR: Extracting signals from wavelet spectra. <https://CRAN.R-project.org/package=WaverideR>

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22. <doi:10.1016/j.cageo.2019.02.011>

Examples

```
depth <- seq(0, 300, by = 1)
period <- 5 + 10 * depth / 300
ctrl <- dynamic_filter_controls(tracked = cbind(depth, period),
```

```
  rel_bandwidth = 0.3, n_points = 20)
head(ctrl)
```

---

dynot *Dynamic noise after orbital tuning (DYNOT)*

Description

The dynot function runs the DYNOT sedimentary noise model of Li et al. (2018a,b) on an (orbitally tuned, usually unevenly sampled) time series. In each Monte Carlo realisation a sampling interval is drawn from a Weibull distribution fitted (maximum likelihood) to the sampling intervals of the data (values outside [sample_min, sample_max] are replaced by one shared uniform draw for all low values and one for all high values, as in Acycle), a window length is drawn uniformly from [window_min, window_max], a time-bandwidth product from nw_min + k/2 and, for each target orbital frequency f, the lower and upper band limits are f -/+ fz * nw / window with fz uniform in [fz_min, fz_max]. The series is pchip-interpolated at the drawn sampling interval, linearly detrended, padded by half a window (pad_type) and analysed with the legacy power decomposition (pdan.m, see pda_legacy). The power-ratio curves are interpolated onto a regular grid, rescaled by (grand mean / realisation mean) of the power ratio, normalised to a maximum of 1 if needed, and summarized by percentiles (median plus the selected confidence intervals). The noise is 1 - power ratio. Translated from Acycle v3.1 DYNOS.m (runDynot, loadDataReady, sanitizeData), dynotIterationWorker.m, dynotPercentilePlan.m, pdan.m and zeropad2.m (Mingsong Li).

Usage

```
dynot(
  data = NULL,
  target_periods = c(405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897, 23.682,
    22.3758, 18.9519),
  window_min = 300,
  window_max = 500,
  nw_min = 2,
  nw_max = 2,
  pad = 1000,
  step = 5,
  n_out = 1000,
  pad_type = 1,
  fz_min = 0.9,
  fz_max = 1.2,
  f_min = 0.001,
  f_max = 1,
  sample_min = NULL,
  sample_max = NULL,
  intervals = c(50, 68, 80, 90, 95),
  n_sim = 1000,
```

```
  seed = NULL,
  verbose = FALSE,
  genplot = FALSE
)
```

Arguments

data	Matrix with two columns: time (kyr) and proxy value. Data are sorted, duplicated times are averaged and non-finite rows removed. Default=NULL
target_periods	Target orbital periods (same unit as time). Default are the nine Acycle default periods (405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897, 23.6820, 22.3758, 18.9519 kyr).
window_min	Minimum window length. Default=300
window_max	Maximum window length. Default=500
nw_min	Minimum time-bandwidth product. Default=2
nw_max	Maximum time-bandwidth product; 2*(nw_max-nw_min) must be an integer. Default=2
pad	FFT length of the MTM (zero-padding number). Default=1000
step	Step between PDA windows in samples. Default=5
n_out	Number of regular output grid points. Default=1000
pad_type	Half-window padding: 0 = none, 1 = zero, 2 = mirror, 3 = mean, 4 = random. Default=1
fz_min	Minimum bandwidth multiplier. Default=0.9
fz_max	Maximum bandwidth multiplier. Default=1.2
f_min	Lower cutoff frequency (clips target limits and total band). Default=0.001
f_max	Upper cutoff frequency. Default=1
sample_min	Lower limit of simulated sampling intervals. If NULL the 5th percentile of the data sampling intervals. Default=NULL
sample_max	Upper limit of simulated sampling intervals. If NULL the 95th percentile of the data sampling intervals. Default=NULL
intervals	Confidence intervals (in percent) to report, any of 50, 68, 80, 90, 95; the median is always reported. Default=c(50, 68, 80, 90, 95)
n_sim	Number of Monte Carlo simulations. Default=1000
seed	Seed for the random number generator (NULL = not set). Default=NULL
verbose	Show a progress bar. Default=FALSE
genplot	Plot the result with plot_dynot. Default=FALSE

Value

A list with elements percentile: matrix [time, noise percentiles] on the grid points where the median is defined (column names give the percentile levels); median: matrix [time, median noise]; percentiles: percentile levels; median_index; interval_count: number of confidence intervals; grid: full output grid; noise: matrix (n_out x n_sim) of adjusted noise (1 - adjusted power ratio) of every realisation; power_mean: mean power ratio of every realisation; draws: list with the simulated sampling intervals, windows, nw, band limits (matrix f3m) and the underlying uniform draws; weibull: fitted Weibull scale and shape; sample_range; data: the cleaned input data.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li (Acycle).

References

Li, M., Hinnov, L.A., Huang, C., Ogg, J.G., 2018a. Sedimentary noise and sea levels linked to land-ocean water exchange and obliquity forcing. Nature Communications 9, 1004. doi:10.1038/s4146701803454y

Li, M., Huang, C., Hinnov, L., Chen, W., Ogg, J., Tian, W., 2018b. Astrochronology of the Anisian stage (Middle Triassic) at the Guandao reference section, South China. Earth and Planetary Science Letters 482, 591-606. doi:10.1016/j.epsl.2017.11.042

Li, M., Huang, C., Hinnov, L., Ogg, J., Chen, Z.-Q., Zhang, Y., 2016. Obliquity-forced climate during the Early Triassic hothouse in China. Geology 44, 623-626. doi:10.1130/G37970.1

Examples

```
set.seed(2)
t <- cumsum(runif(900, 1, 3))
x <- sin(2 * pi * t / 405) + sin(2 * pi * t / 100) +
  sin(2 * pi * t / 41) + rnorm(900) * (0.2 + t / max(t))
res <- dynot(data = cbind(t, x), n_sim = 20, n_out = 200, seed = 1)
plot_dynot(res)
```

---

earth_moment_of_inertia

Earth's polar moment of inertia at spin rate omega (Farhat et al. 2022, A.3)

Description

Earth's polar moment of inertia at spin rate omega (Farhat et al. 2022, A.3)

Usage

```
earth_moment_of_inertia(omega)
```

Arguments

omega	Angular velocity of the Earth (rad/s).

earth_omega *Earth's angular velocity from the length of day*

Description

Earth's angular velocity from the length of day

Usage

```
earth_omega(lod_h, kind = c("solar", "sidereal"))
```

Arguments

lod_h	Length of day in hours.
kind	"solar" (AstroGeo22 lod column) or "sidereal"

earthmoon_angular_momentum *Earth-Moon system angular momentum (spin + lunar orbit), kg m^2 s^-1*

Description

Earth-Moon system angular momentum (spin + lunar orbit), kg m^2 s^-1

Usage

```
earthmoon_angular_momentum(a_RE, omega)
```

Arguments

a_RE	Earth-Moon distance in Earth radii.
omega	Angular velocity of the Earth (rad/s).

`earthmoon_from_precession` *Invert the AstroGeo22 precession curve: Earth-Moon state for a measured precession frequency / period.*

Description

The model curve is not strictly monotonic (tidal resonances), so every crossing is returned.

Usage

```
earthmoon_from_precession(
  p_kyr = NULL,
  p_arcsec = NULL,
  method = c("astrogeo22", "angular_momentum"),
  obliquity_deg = NULL,
  age_range = NULL,
  data_dir = NULL
)
```

Arguments

`p_kyr`	precession period (kyr); or give ‘p_arcsec‘
`p_arcsec`	Precession constant in arcsec/yr.
`method`	Method used: ‘"astrogeo22"‘ or ‘"angular_momentum"‘.
`obliquity_deg`	Obliquity in degrees.
`age_range`	optional c(min, max) Ma to restrict the search
`data_dir`	Directory with the data files (‘NULL‘ = the data installed with the package).

`earthmoon_state` *Full Earth-Moon state from any one observable*

Description

Give exactly one of: ‘p_arcsec‘/‘p_kyr‘/‘k_yr‘ (precession), ‘a_RE‘/‘a_km‘ (Earth-Moon distance), ‘lod_h‘ (solar day), ‘months_per_year‘, or an observed ‘cycle_kyr‘ together with its ‘term‘ (e.g. the obliquity cycle).

Usage

```
earthmoon_state(
  p_arcsec = NULL,
  p_kyr = NULL,
  k_yr = NULL,
  a_RE = NULL,
  a_km = NULL,
  lod_h = NULL,
  months_per_year = NULL,
  months_kind = "sidereal",
  cycle_kyr = NULL,
  term = "k+s3",
  method = c("angular_momentum", "astrogeo22"),
  obliquity_deg = NULL,
  age_ma = NULL,
  data_dir = NULL
)
```

Arguments

p_arcsec	Precession constant in arcsec/yr.
p_kyr	Precession period in kyr.
k_yr	Precession constant in arcsec/yr.
a_RE	Earth-Moon distance in Earth radii.
a_km	Earth-Moon distance in km.
lod_h	Length of day in hours.
months_per_year	Number of lunar months per year.
months_kind	Type of month, ‘"sidereal"‘ or ‘"synodic"‘.
cycle_kyr	Period of a cycle in kyr.
term	Frequency term, e.g. ‘"k+s3"‘.
method	how the Earth’s spin and the lunar distance are linked when inverting a precession constant: "angular_momentum" (Earth-Moon angular momentum conserved) or "astrogeo22" (read off the Farhat et al. 2022 curve)
obliquity_deg	obliquity to use; defaults to the AstroGeo22 value at ‘age_ma‘, else the present value
age_ma	Age(s) in Ma.
data_dir	Directory with the data files (‘NULL‘ = the data installed with the package).

Value

one-row-per-input data.frame with the precession constant, the Earth-Moon distance, the solar and sidereal day, days per year, sidereal and synodic months per year and the obliquity

ecoco_adaptive *Adaptive eCOCO*

Description

Adaptive COCO in physical-depth sliding windows with per-window AR(1) Monte Carlo nulls and ridge tracking (Acycle ecoco.m adaptive branch, ecocoAdaptiveCore.m; Mingsong Li).

Usage

```
ecoco_adaptive(
  data,
  orbit9 = orbit9_default(),
  window,
  sr = seq(0.5, 20, by = 0.1),
  step = 1,
  step_depth = NULL,
  pad = 5000,
  red = 1,
  nsim = 500,
  method = c("Pearson", "Spearman"),
  max_frequency = NULL,
  amplitude_mode = c("four-group-area", "per-orbit", "fixed"),
  seed = 1,
  center_limits = NULL,
  batch = NULL,
  verbose = TRUE,
  centre_limits = NULL
)
```

Arguments

data	two columns depth (m), value
orbit9	nine orbital periods (kyr)
window	sliding-window length (m)
sr	sedimentation-rate grid (cm/kyr)
step	sliding step in samples (used when step_depth is NULL: step_depth = step*dt)
step_depth	exact spacing (m) of window centres
pad, red, nsim, method, max_frequency, amplitude_mode, seed, batch, verbose	see coco_adaptive()
center_limits	optional c(first, last) window-centre limits
centre_limits	British spelling of center_limits; if given it overrides center_limits.

Value

list with matrices (rate-by-window): rho, p_global, p_local, norbit, pcoco, score; 'track' = ridge-tracked sedimentation-rate path

---

ecoco_best_per_window *Best sedimentation rate per eCOCO window*

Description

Row of maximum finite score in every window (bestPerWindowSummary in Acycle ecoco.m).

Usage

```
ecoco_best_per_window(sr, depth, rho, p, norbit, score)
```

Arguments

`sr`	sedimentation-rate grid (cm/kyr)
`depth`	window centres (m)
`rho`, `p`, `norbit`, `score`	rate-by-window matrices

Value

data.frame with depth, sr, rho, p_global, norbit, score, sr_low, sr_high

---

ecoco_blocked *Blocked eCOCO (cross-fitted evolutionary correlation coefficient)*

Description

Blocked eCOCO evaluates complete sliding validation windows with astronomical targets trained on strictly non-overlapping anchor windows. Four leakage-corrected orbital-group amplitudes (long eccentricity, short eccentricity, obliquity, precession; groups identified by descending period rank) are fitted at regularly spaced anchor windows (spacing `anchor_fraction * window`) using the complete sedimentation-rate grid, and expanded to a coherent nine-term sine target. Every validation window is evaluated with the nearest lower-depth anchor (forward) and the nearest higher-depth anchor (backward) with frozen weights. The consensus correlation at a rate is the smaller of the two directional correlations; at the record edges the available direction is used (`consensus_policy = "supported"`) or the window is left empty (`"strict"`). Stationary Gaussian AR(1) Monte Carlo nulls, one per unique window interval, repeat the complete anchor training and frozen validation. Global p-values use the null maximum over the complete rate grid within each window, local p-values the same rate. pCOCO = rho * |log10(p_global)| is the ridge score; the sedimentation-rate ridge is tracked as in Acycle (`trackEcocoRidge`). Translated from Acycle v3.1 ecocoCrossfitCore.m, ecoco.m (Blocked branch), ecocoWindowSpectra.m, ecocoEstimateAr1.m, ecocoPhysicalCenters.m, ecocoPhysicalWindowValues.m, cocoNonnegativeLeakageSolve.m and zeropad2.m (Mingsong Li). Uses the COCO helper functions of this package (helpers_coco.R and the ecoco_* functions).

Usage

```
ecoco_blocked(
  data = NULL,
 orbit9 = c(405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897, 23.682, 22.3758,
    18.9519),
  window = NULL,
  sr = NULL,
  step_depth = NULL,
  step = 1,
  pad = NULL,
  red = 0,
  n_sim = 2000,
  method = "Pearson",
  max_frequency = NULL,
  seed = 1,
  anchor_fraction = 0.5,
  window_mode = "physical-depth",
  edge_pad = TRUE,
  center_limits = NULL,
  compute_local_p = TRUE,
  consensus_policy = "supported",
  batch_size = 20,
  genplot = FALSE,
  verbose = FALSE,
  centre_limits = NULL
)
```

Arguments

data	Two-column matrix: depth (m) and proxy value. Irregular data are linearly interpolated at the median spacing (cocoPrepareRegularData).
orbit9	Nine orbital periods (kyr). Default=c(405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897, 23.6820, 22.3758, 18.9519)
window	Sliding-window length (m). Default=NULL: 25% of the record length (Acycle GUI default).
sr	Sedimentation-rate grid (cm/kyr), strictly increasing. Default=NULL: the Acycle GUI default range.
step_depth	Spacing of the output window centres (m). Default=NULL: the Acycle GUI default (dt, or dt*ceiling(n/300)).
step	Legacy integer sample step, used when window_mode = "legacy-count". Default=1
pad	Periodogram length (nfft, at least the window point count). Default=NULL: Acycle's defaultPad (5000 for up to 2500 points).
red	Background removal: 0 none, 1 classical AR(1), 2 robust AR(1), 3 smoothed-window average (SWA). Default=0
n_sim	Number of Monte Carlo simulations. Default=2000
method	Correlation, "Pearson" or "Spearman". Default="Pearson"

max_frequency	Maximum temporal frequency (cycles/kyr) in the correlation. Default=NULL: 1.2/min(orbit9).
seed	Random seed (set.seed); NULL leaves the RNG state. Default=1
anchor_fraction	Anchor spacing as a fraction of the window (0 < x <= 2). Default=0.5
window_mode	"physical-depth" (Acycle wrapper and GUI) or "legacy-count" (fixed sample count). Default="physical-depth"
edge_pad	Zero-pad half a window at both ends and restrict the output centres to the original record (Acycle GUI requirement for Blocked eCOCO). Default=TRUE
center_limits	Optional c(first, last) output-centre limits (overrides the limits set by edge_pad). Default=NULL
compute_local_p	Also compute same-rate local p-values. Default=TRUE
consensus_policy	"supported" (one-direction fallback at edges) or "strict" (bidirectional windows only). Default="supported"
batch_size	Monte Carlo simulations per batch (results do not depend on it). Default=20
genplot	Plot the result with plot_ecoco_cv. Default=FALSE
verbose	Print progress messages. Default=FALSE
centre_limits	British spelling of center_limits; if given it overrides center_limits.

Value

A list (class "ecoco_cv") with elements method, sr (rate grid), depth (window centres), rate-by-window matrices rho, p_global, p_local, norbit, pcoco and score (the published consensus surface), the lists forward, backward, consensus, strict (each with rho, p_global, p_local, norbit, pcoco, best_rate, ...), support (per-window "bidirectional", "forward-only", "backward-only" or "none"), windows and anchors (data frames with bounds, anchor links and trained rates/weights), geometry (rate masks), rho_m (per-window AR(1)), best (best rate per window) and track (tracked ridge: depth, sr, rho, p_global, norbit, score, sr_low, sr_high).

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li

References

Li, M., Kump, L.R., Hinnov, L.A., Mann, M.E., 2018. Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing. Earth and Planetary Science Letters 501, 165-179. doi:10.1016/j.epsl.2018.08.041

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22. doi:10.1016/j.cageo.2019.02.011

Examples

```
set.seed(2)
dt <- 0.05
depth <- seq(0, 30, by = dt)
tt <- depth * 100 / 5          # 5 cm/kyr
x <- sin(2 * pi * tt / 405.7) + 0.8 * sin(2 * pi * tt / 98.9) +
  0.5 * sin(2 * pi * tt / 40.99) + 0.4 * sin(2 * pi * tt / 23.68) +
  0.3 * rnorm(length(tt))
res <- ecoco_blocked(cbind(depth, x), window = 10, sr = seq(3, 8, 0.5),
                     step_depth = 1, pad = 400, n_sim = 20)
res$track[, c("depth", "sr")]
```

---

ecoco_classic *Evolutionary COCO (Li et al. 2018)*

Description

COCO in a sliding window (2019 Acycle ecoco.m driver) with best-rate summary and ridge tracking per window.

Usage

```
ecoco_classic(
  data,
  target,
  orbit = NULL,
  window,
  step = 1,
  sr = seq(0.2, 20, by = 0.2),
  dt = NULL,
  pad = 5000,
  red = 1,
  nsim = 0,
  slices = 1,
  adjust = 0,
  delinear = 1,
  method = c("Pearson", "Spearman"),
  mc = c("first-window", "each-window"),
  seed = NULL,
  track_ridge = TRUE,
  verbose = TRUE
)
```

Arguments

data, target, orbit, sr, dt, pad, red, nsim, slices, adjust, method, seed	see coco_classic()
window	sliding window length (m)
step	sliding step in number of samples
delinear	1 = remove the mean of each window (as in Acycle ecoco.m)
mc	"first-window" reproduces Acycle 2018-2019: the Monte Carlo null is simulated once (first window) and reused for all windows. "each-window" simulates a separate null for every window (slower).
track_ridge	track a smooth ridge (ecoco_track_ridge()) Default=TRUE
verbose	progress messages Default=TRUE

Value

list of matrices (rows = sr, columns = windows): rho, p_h0, norbit, pcoco, score (= norbit/norb * pcoco) plus ‘track‘ (best rate per window)

---

ecoco_interleaved *Interleaved eCOCO (odd/even cross-validated evolutionary COCO)*

Description

Interleaved eCOCO evaluates complete sliding windows of the cleaned (finite, sorted, duplicate-averaged) record. Within each window the globally odd observations train a frozen four-group coherent-nine astronomical target that is validated on the globally even observations, and vice versa (Interleaved cvCOCO applied per window). Each fold is linearly regularised at its own median spacing after the split. The consensus correlation is the same-rate minimum of the two directional curves; its global p-value is a plus-one Monte Carlo p against the null maximum over the complete rate grid. Each null realisation is one stationary Gaussian AR(1) process on the complete window order, split and interpolated like the data. Windows that cannot be resolved for the rate grid are returned empty. pCOCO = rho * |log10(p_global)| is the ridge score. Translated from Acycle v3.1 ecocoInterleavedCore.m, interleavedcvcoco.m and cvcoco.m (Mingsong Li). Requires the cocoR helpers (part of this package).

Usage

```
ecoco_interleaved(
  data = NULL,
 orbit9 = c(405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897, 23.682, 22.3758,
    18.9519),
  window = NULL,
  sr = NULL,
  step_depth = NULL,
  step = 1,
  pad = NULL,
```

```
  red = 0,
  n_sim = 2000,
  method = "Pearson",
  max_frequency = NULL,
  seed = 1,
  window_mode = "physical-depth",
  dt = NULL,
  batch_size = 20,
  genplot = FALSE,
  verbose = FALSE
)
```

Arguments

data	Two-column matrix: depth (m) and proxy value. Raw (possibly irregular) data; do not interpolate beforehand.
orbit9	Nine orbital periods (kyr). Default=c(405.6912, 130.6979, 123.8532, 98.8517, 94.8856, 40.9897, 23.6820, 22.3758, 18.9519)
window	Sliding-window length (m). Default=NULL: 25% of the record length.
sr	Sedimentation-rate grid (cm/kyr) with a constant step. Default=NULL: the Acycle GUI default range.
step_depth	Spacing of the window centres (m) in physical-depth mode. Default=NULL: the Acycle GUI default.
step	Integer sample step for window_mode = "legacy-count". Default=1
pad	Periodogram length; each fold uses max(pad, fold length). Default=NULL: Acycle's defaultPad.
red	Background removal: 0 none, 1 classical AR(1), 2 robust AR(1), 3 smoothed-window average (SWA). Default=0
n_sim	Number of Monte Carlo simulations per window. Default=2000
method	Correlation, "Pearson" or "Spearman". Default="Pearson"
max_frequency	Maximum temporal frequency (cycles/kyr). Default=NULL: 1.2/min(orbit9).
seed	Parent random seed. Each window uses a derived seed (Acycle deriveWindowSeed) passed to set.seed; NULL leaves the RNG state. Default=1
window_mode	"physical-depth" or "legacy-count". Default="physical-depth"
dt	Nominal sampling interval used by legacy-count mode and for the default step. Default=NULL: median spacing of the cleaned data.
batch_size	Monte Carlo simulations per batch. Default=20
genplot	Plot the result with plot_ecoco_cv. Default=FALSE
verbose	Print progress. Default=FALSE

Value

A list (class "ecoco_cv") with `method`, `sr`, `depth`, rate-by-window matrices `rho`, `p_global`, `p_local`, `norbit`, `pcoco`, `score`, lists `forward` (odd -> even), `backward` (even -> odd), `consensus` and `strict` (consensus curve with the larger directional p), `windows` (data frame: bounds, point counts, parity phase, derived seed, success, window-level p-values, fold spacings, all-nine rate ranges, AR(1) rho), `best` and `track`.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li

References

Li, M., Kump, L.R., Hinnov, L.A., Mann, M.E., 2018. Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing. Earth and Planetary Science Letters 501, 165-179. doi:10.1016/j.epsl.2018.08.041

Examples

```
set.seed(2)
dt <- 0.05
depth <- seq(0, 30, by = dt)
tt <- depth * 100 / 5
x <- sin(2 * pi * tt / 405.7) + 0.8 * sin(2 * pi * tt / 98.9) +
  0.5 * sin(2 * pi * tt / 40.99) + 0.4 * sin(2 * pi * tt / 23.68) +
  0.3 * rnorm(length(tt))
res <- ecoco_interleaved(cbind(depth, x), window = 15,
                         sr = seq(3, 8, 0.5), step_depth = 2, pad = 400,
                         n_sim = 20)
res$track[, c("depth", "sr")]
```

`ecoco_track` *Track sedimentation-rate solutions through eCOCO peak maps*

Description

Tracks sedimentation-rate paths through the per-window correlation peaks of an eCOCO result, following Acycle's legacy tracking routines. Peak locations (row indices of the `n_peaks` largest score values per window that pass the correlation, p-value and orbit-count criteria) are found as in eccpeaks.m. `mode = "slices"` reproduces ecocotrack.m: the windows are divided into `n_slices` depth slices (sharing boundary windows); in every slice a path is started from each trial rate between min(sr) and max(sr) in steps of `sr_threshold`, following the nearest peak within `sr_threshold` (reference = median of the last five tracked rates, ecocotrackn.m method 1). Candidate paths are summarised with eccount.m (mean/median rate, number of windows, mean rho, mean p, summed pCOCO, summed orbit count), sorted, filtered (mean rho >= `rho_min`, mean p <= `p_track_max`,

summed orbit count >= norbit_min, near-duplicate solutions within sr_threshold removed) and the solution with the largest summed pCOCO is kept. mode = "single" reproduces ecocotrackn.m for one path started at initial_sr (reference "median5" = method 1, "previous" = method 0). Translated from Acycle v3.1 ecocotrack.m, ecocotrackn.m, eccount.m and eccpeaks.m (Mingsong Li).

Usage

```
ecoco_track(
  ecoco = NULL,
  sr = NULL,
  depth = NULL,
  rho = NULL,
  p = NULL,
  pcoco = NULL,
  norbit = NULL,
  score = NULL,
  location = NULL,
  mode = "slices",
  n_peaks = 3,
  rho_min = 0.3,
  p_max = 0.05,
  p_track_max = p_max,
  norbit_min = 3,
  sr_threshold = 2.5,
  n_slices = 5,
  initial_sr = NULL,
  reference = "median5",
  genplot = FALSE
)
```

Arguments

ecoco	An eCOCO result list (from ecoco_blocked, ecoco_interleaved or cocoR ecoco_adaptive) with elements sr, depth, rho, p_global, pcoco and norbit. Default=NULL
sr, depth, rho, p, pcoco, norbit, score	Explicit inputs used when ecoco is NULL: rate grid, window depths and rate-by-window maps (score is the map searched for peaks; default pcoco). Default=NULL
location	Optional matrix of peak row indices (n_peaks x windows, NA = no peak), replacing the eccpeaks step. Default=NULL
mode	"slices" (ecocotrack) or "single" (ecocotrackn). Default="slices"
n_peaks	Number of peaks per window. Default=3
rho_min	Minimum correlation of a peak and of a path mean. Default=0.3
p_max	Maximum p-value (same units as p) of a peak. Default=0.05
p_track_max	Maximum mean p-value of a path. Default=p_max

norbit_min	Minimum orbit count of a peak and minimum summed orbit count of a path. Default=3
sr_threshold	Search radius and trial-rate spacing (cm/kyr). Default=2.5
n_slices	Number of depth slices. Default=5
initial_sr	Starting rate for mode = "single" (cm/kyr). Default=NULL: the best rate of the first window.
reference	"median5" (median of the previous five tracked rates) or "previous" (previous tracked rate) for mode = "single". Default="median5"
genplot	Plot the tracked rates against depth. Default=FALSE

Value

A list with depth, sr (tracked rate per window, NA where untracked), index (row index into the rate grid), location (peak row-index matrix); for mode = "slices" also slice_range (window indices of the slice boundaries), paths (rate-index paths of all trial rates, rows = trial rates) and summary (per slice matrix of the eccount statistics: mean index, median index, n, mean rho, mean p, summed pCOCO, summed orbit count; rejected solutions NA).

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li

References

Li, M., Kump, L.R., Hinnov, L.A., Mann, M.E., 2018. Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing. Earth and Planetary Science Letters 501, 165-179.

Examples

```
sr <- seq(1, 10, by = 0.25)
depth <- seq(0, 20, by = 0.5)
true_sr <- 4 + 2 * depth / 20
rho <- matrix(0, length(sr), length(depth))
for (j in seq_along(depth)){
  rho[, j] <- 0.8 * exp(-(sr - true_sr[j])^2 / 0.5)
}
p <- matrix(0.01, length(sr), length(depth))
nor <- matrix(9, length(sr), length(depth))
tr <- ecoco_track(sr = sr, depth = depth, rho = rho, p = p,
                  pcoco = 2 * rho, norbit = nor, n_slices = 2)
plot(tr$sr, tr$depth)
```

`ecoco_track_ridge` *Track a smooth sedimentation-rate ridge through an eCOCO score map*

Description

Within-window min-max normalised score, quadratic jump penalty (penalty 0.35, jump scale max(2, 4*sr step)), Viterbi back-tracking (trackEcocoRidge in Acycle ecoco.m).

Usage

```
ecoco_track_ridge(
  score,
  sr,
  depth,
  rho,
  p,
  norbit,
  fallback = NULL,
  jump_penalty = 0.35
)
```

Arguments

`score`, `rho`, `p`, `norbit`	rate-by-window matrices
`sr`	sedimentation-rate grid (cm/kyr)
`depth`	window centres (m)
`fallback`	result of ecoco_best_per_window() `Default=NULL`
`jump_penalty`	weight of the quadratic jump penalty `Default=0.35`

Value

data.frame with depth, sr, rho, p_global, norbit, score, sr_low, sr_high

`eha_log2` *Generate a log2 spaced EHA spectra*

Description

Compute a log2 spaced Evolutive Harmonic Analysis (EHA) & Evolutive Power Spectral Analysis This is a wrapper function for the "eha" function of the 'astrochron' R package

Usage

```
eha_log2(
  data = NULL,
  win = NULL,
  tbw = NULL,
  demean = NULL,
  detrend = NULL,
  upperPeriod = NULL,
  lowerPeriod = NULL,
  pad = NULL,
  padding = "noise"
)
```

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.
win	Window size for EHA, in units of space or time.
tbw	MTM time-bandwidth product (<=10)
demean	Remove mean from data series? (T or F)
detrend	Remove linear trend from data series? (T or F)
upperPeriod	Upper period to be analyzed. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyse the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.
lowerPeriod	Lowest period to be analyzed. The CWT analyses the signal starting from the lowerPeriod to the upperPeriod so the proper selection these parameters allows to analyse the signal for a specific range of cycles. scaling is done using power 2 so for the best plotting results select a value to the power or 2.
pad	with zeros to how many points? Must not factor into a prime number >23. Maximum number of points is 200,000.
padding	pad the edges of the data set with half a window length with the following, the "Mean", "noise" or "zero"

Value

The output is a list (analyze.eha object) which contain 12 objects which are the result of running the eha. Object 1: depth - depth/time of axis Object 2: log2_period - log2 scales period Object 3: pwr - Power values of the EHA run Object 4: amp - Amplitude values of the EHA run Object 5: prob - Probability values of the EHA run Object 6: f_test - F test values of the EHA run Scale size Object 7: Power.avg - Average power values Object 8: amp.avg - Average amplitude values Object 9: prob.avg - Average probability value Object 10: f_test.avg - Average f test values Object 11: x - x values of the data set Object 12: y - y values of the data set Object 13: window size

Author(s)

Code based on on the "eha" function of the 'astrochron' R package

References

S.R. Meyers, 2012, Seeing Red in Cyclic Stratigraphy: Spectral Noise Estimation for #'Astrochronology: Paleoceanography, 27, PA3228, <doi:10.1029/2012PA002307>

S.R. Meyers, 2019 Cyclostratigraphy and the problem of astrochronologic testing, Earth-Science Reviews, Volume 190, <doi.org/10.1016/j.earscirev.2018.11.015.>

Examples

```
#Example 1. Total Solar Irradiance
# data set of Steinhilber et al., (2012)

TSI_eha_log2 <- eha_log2(data = TSI,
win =  8192,
tbw = 4,
demean = TRUE,
detrend = TRUE,
upperPeriod = 8192,
lowerPeriod = 50,
pad = NULL,
padding = "noise")

#Example 2. Magnetic susceptibility data set of Pas et al., (2018)
mag_eha_log2 <- eha_log2(data = mag,
win =  50,
tbw = 4,
demean = TRUE,
detrend = TRUE,
upperPeriod = 50,
lowerPeriod = 2,
pad = NULL,
padding = "noise")

#Example 3. Greyscale data set of Zeeden et al., (2013)
grey_eha_log2 <- eha_log2(data = grey,
win =  20,
tbw = 4,
demean = TRUE,
detrend = TRUE,
upperPeriod = 20,
lowerPeriod = 2,
pad = NULL,
padding = "noise")
```

`em_state` *Full Earth-Moon state from a precession constant*

Description

Full Earth-Moon state from a precession constant

Usage

```
em_state(k_kyr, obl_deg = AC$obl0_deg)
```

Arguments

k_kyr	precession constant in kyr
obl_deg	obliquity to use in eq. A.7 (degrees)

---

example_pridoli *The Pridoli example: the four terms resolved at Cellon (Carnic Alps) on the astrochronozone time scale of Arts and Da Silva.*

Description

The Pridoli example: the four terms resolved at Cellon (Carnic Alps) on the astrochronozone time scale of Arts and Da Silva.

Usage

```
example_pridoli(file = NULL)
```

Arguments

file	Name of the example file.

---

expSedRamp *exponentially ramped sedimentation rate model to convert time to stratigraphy*

Description

Apply a exponentially increasing increasing sedimentation rate model to convert time to stratigraphy.

Usage

```
expSedRamp(data, sr_start, sr_end)
```

Arguments

data	Input data set should consist of a matrix with 2 columns with the first being time and the second being the proxy
sr_start	Initial sedimentation rate (in cm/kyr).
sr_end	Final sedimentation rate (in cm/kyr).

Value

Returns a list which contains 10 elements element 1: time series in the depth (m) domain element 2: sedimentation rate curve

Author(s)

Based on the sedRamp function of the 'astrochron' R package.

References

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

Examples

```
data_1 <- astrochron::etp(
tmin = 0,
tmax = 4000,
dt = 1,
eWt = 1.5,
oWt = 0.75,
pWt = 1,
esinw = TRUE,
solution = NULL,
standardize = TRUE,
genplot = FALSE,
verbose = FALSE
)
data_ramped_ls <- expSedRamp(data_1, sr_start = 1, sr_end = 5)
```

---

extract_amplitude *Extract the instantaneous amplitude (envelope) of a signal using the CWT*

Description

The wavelet equivalent of the Hilbert transform. A complex (analytic) Morlet wavelet is used, so the wavelet coefficients already are the analytic representation of a real signal; summing them back up gives the analytic signal $z(t)$ whose real part is the input and whose modulus is the instantaneous amplitude.

Called with only a signal, it returns the envelope of the whole record and behaves like a Hilbert transform (median ratio 1.000 on a ZB18a eccentricity window), but computed scale by scale, so a period range can be imposed and the ends of the record do not have to be wrapped around.

Given `cycle`, it instead returns the amplitude of that one cycle: the analytic signal is rebuilt only from the scales inside `band` around the given period, so other cycles, noise and trends do not enter the amplitude. This is the amplitude-modulation curve used in cyclostratigraphy.

Calibration is absolute in both modes, with no rescaling against peak statistics: a signal of constant amplitude 3 returns 3, independent of dt, dj, omega_nr and the scale range. In whole-record mode the reconstruction constant is fixed by requiring $\mathrm{Re}(z)$ to reproduce the input; in cycle mode it is the analytic response of the wavelet family to a unit cosine at the cycle's frequency.

Usage

```
extract_amplitude(
  signal = NULL,
  cycle = NULL,
  band = c(0.5, 2),
  periods = NULL,
  omega_nr = 6,
  dj = 1/100,
  pad = c("AR", "reflect", "zero"),
  mask_coi = FALSE,
  genplot = FALSE,
  ver_results = FALSE,
  keep_editable = FALSE,
  ...
)
```

Arguments

`signal`	Matrix/data frame: column 1 = depth/time, column 2 = proxy.
`cycle`	What to measure the amplitude of. `NULL` (default) = the whole signal, i.e. the Hilbert-equivalent envelope. A single number = the cycle of that period (e.g. `cycle = 405`). A 2-column matrix (depth/time, period) = a tracked period curve, e.g. from `track_period_wavelet()` + `completed_series()`, for records whose sedimentation rate varies. `"ridge"` = follow the strongest cycle at each depth (only for records dominated by one cycle).
`band`	Period band kept around `cycle`, as factors of it. `Default = c(0.5, 2)`. Narrow it (e.g. `c(0.8, 1.2)`) for noisy records or cycles close together; a wide band is needed when the cycle is strongly modulated. Ignored when `cycle = NULL`.
`periods`	Period range analysed, `c(lower, upper)`. Default: `2*dt` to half the record. Use it to keep noise or a trend out of the whole-record envelope, or to stop `cycle = "ridge"` wandering.
`omega_nr`	Cycles in the Morlet wavelet; >= 5 keeps it analytic. `Default = 6`.
`dj`	Scale spacing. `Default = 1/100`.

pad	End treatment: "AR" (default) extends the record with a forward/backward linear-prediction forecast so the wavelet does not see a step to zero, "reflect" mirrors it, "zero" is the classic zero padding. Falls back automatically if the AR fit fails.
mask_coi	Return NA inside the cone of influence. Default = FALSE.
genplot	Plot the signal with its +/- envelope, and the envelope.
ver_results	Print the median ratio to the Hilbert envelope (outside the cone of influence).
keep_editable	Keep plot parameters editable. Default = FALSE.
...	Ignored, with a note; lets old calls using pts, remean, tracked_cycle_curve, lowerPeriod etc. keep working (the old names are mapped onto the new ones where possible).

Value

Matrix, column 1 = depth/time, column 2 = amplitude. In cycle mode the period actually used is attached as attr(x, "period").

Examples

```
# envelope of the whole record - nothing else to set (insolation-like signal, time in kyr)
amp <- extract_amplitude(astrosignal_example, genplot = TRUE)

# amplitude of one cycle only: the precession band (~21 kyr) of the example signal;
# its amplitude is the eccentricity modulation
amp21 <- extract_amplitude(astrosignal_example, cycle = 21, band = c(0.75, 1.35),
                           genplot = TRUE)

# depth domain with a varying sedimentation rate: hand it the tracked 405-kyr period curve
# (magnetic susceptibility of Pas et al. 2018, tracked 405-kyr cycle in metres)
mag_wt <- analyse_wavelet(mag, dj = 1/100, lowerPeriod = 0.1, upperPeriod = 254,
                          verbose = FALSE, omega_nr = 10)
mag_track <- completed_series(wavelet = mag_wt, tracked_curve = mag_track_solution,
                              period_up = 1.2, period_down = 0.8, extrapolate = TRUE,
                              genplot = FALSE)
amp405 <- extract_amplitude(mag, cycle = mag_track, band = c(0.8, 1.25), genplot = TRUE)
```

---

extract_power *Extract power from a wavelet spectra*

Description

Extracts 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(
  tracked_period_curve = NULL,
  scalogram = NULL,
  period_up = 1.2,
  period_down = 0.8,
  tracked_cycle_period = NULL,
  extract_cycle_power = NULL,
  ...
)
```

Arguments

tracked_period_curve	Traced period result from the track_period_wavelet or track_period function completed using the completed_series. The input can be pre-smoothed using the the loess_auto function.
scalogram	Wavelet or superlet object created using the analyse_wavelet or or track_period 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).
...	older input previous versions

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.

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo. The assignment of the standard deviation of the uncertainty of the wavelet is based on the work of Gabor (1946) and Russell et al., (2016) The functionality of this function is is inspired by the integratePower function of the 'astrochron' R package.

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

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

Examples

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

mag_wt <- analyse_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
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 mag_track_solution
#is used
mag_track <- mag_track_solution

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

#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 = FALSE, print_span = FALSE)

#extract the spectral power of the 405 kyr eccentricity cycle
mag_power <- extract_power(
tracked_period_curve = mag_track_complete,
scalogram = 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 `analyse_wavelet` function for a given, `cycle`, `period_up` and `period_down`

Usage

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

Arguments

`scalogram`	Wavelet or superlet object created using the `analyse_wavelet` or or `track_period` function
`cycle`	Period of cycle for which the power will be extracted from the record.
`period_up`	Species the upper period of the to be extracted power `Default=1.2`.
`period_down`	specifies the lower period of the to be extracted power `Default=0.8`.
`...`	additional input parameters from older versions

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.

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998). The functionality of this function is is inspired by the integratePower function of the 'astrochron' R package

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

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

Examples

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

TSI_wt <-
 analyse_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )
TSI_wt_pwr_de_Vries_cycle <-  extract_power_stable(
 scalogram = 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 analyse_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

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998).

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

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 <- analyse_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
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 = FALSE
)

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

# extract the 405 kyr eccentricity 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 `analyse_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,
  keep_editable = FALSE
)
```

Arguments

`wavelet`	Wavelet object created using the `analyse_wavelet` function.
`cycle`	Period of the cycle which needs to be extracted.
`period_up`	Specifies the upper period as a factor of the to be extracted cycle `Default=1.2`.
`period_down`	Specifies the 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`.
`keep_editable`	Keep option to add extra features after plotting `Default=FALSE`

Value

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

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998).

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Examples

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

#Perform the CWT
TSI_wt <-
analyse_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = FALSE,
   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,
keep_editable=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,
  keep_editable = FALSE
)
```

Arguments

wavelet	wavelet object created using the analyse_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.
keep_editable	Keep option to add extra features after plotting 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.

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998).

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Examples

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

TSI_wt <-
analyse_wavelet(
data = TSI,
```

```
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = FALSE,
   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,
keep_editable=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 = FALSE,
  genplot_extracted = FALSE,
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  colour_brewer = NULL
)
```

Arguments

wavelet	Wavelet object created using the analyse_wavelet function.
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 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=FALSE curve.
genplot_extracted	Generates a plot with the data set and the extracted cycle on top Default=FALSE of it.
keep_editable	Keep option to add extra features after plotting Default=FALSE
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

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.

Author(s)

Code based on the reconstruct function of the 'WaveletComp' R package which is based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998). The assignment of the standard deviation of the uncertainty of the wavelet is based on the work of Gabor (1946) and Russell et al., (2016)

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

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.https://doi.org/10.1049/ji-3-2.1946.0074

Russell, Brian, and Jiajun Han. "Jean Morlet and the continuous wavelet transform. " CREWES Res. Rep 28 (2016): 115. https://www.crewes.org/Documents/ResearchReports/2016/CRR201668.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#Extract the 405 kyr eccentricity cycle from the magnetic susceptibility
#record of the Sullivan core of Pas et al., (2018) and use the Gabor
# uncertainty principle to define the mathematical uncertainty of the
# analysis and use a factor of that standard deviation to define
# boundaries

# perform the CWT
mag_wt <- analyse_wavelet(data = mag,
dj = 1/100,
```

```
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
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)",
#                                  palette_name="rainbow",
#                                  color_brewer="grDevices")

#Instead of tracking, the tracked solution data set 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 = FALSE
)

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

# extract the 405 kyr eccentricity cycle from the wavelet spectrum and use
# the Gabor uncertainty principle to define the mathematical uncertainty of
# 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 = FALSE,
genplot_extracted = FALSE,
keep_editable=FALSE,
palette_name="rainbow",
color_brewer="grDevices"
)
```

`fft_frequency_uncertainty` *Frequency uncertainty of the Fourier transform (FFT) for a record length*

Description

Calculator for the frequency uncertainty of a Fourier based spectrum. The practical resolution of an FFT is the bin spacing, the inverse of the record length ($\Delta f = 1/T$). The smallest frequency uncertainty that is possible according to the Heisenberg-Gabor uncertainty principle ($\Delta t \Delta f \geq 1/(4\pi)$) with $\Delta t = T$ is $\Delta f = 1/(4\pi T)$, a factor 4π smaller. The relative uncertainty at a frequency f is $\Delta f/f = 1/(fT)$. For a record of N samples with sampling frequency fs, $T = N/f_s$.

It is the Fourier counterpart of `wavelet_uncertainty`, which gives the uncertainty of the wavelet transform. Units are free: with the record length in metres (or kyr) the frequencies are in cycles per metre (or per kyr).

Usage

```
fft_frequency_uncertainty(
  T_length = NULL,
  N = NULL,
  fs = 1,
  f = NULL,
  period = NULL
)
```

Arguments

`T_length`	Length of the record (time or depth). Alternatively give `N` and `fs`.
`N`	Number of samples (used with `fs` when `T_length` is not given).
`fs`	Sampling frequency (samples per unit time or depth), `Default=1`.
`f`	Optional frequency (or frequencies) of interest in cycles per unit. Gives the relative uncertainty and the corresponding period range.
`period`	Optional period(s) of interest, an alternative to `f` (`f = 1 / period`).

Value

A data frame with the record length, the practical uncertainty `df_practical`, the Heisenberg-Gabor limit `df_gabor`, their ratio and, when `f` or `period` is given, the frequency, the relative uncertainties (`rel_practical`, `rel_gabor`) and the period ranges implied by `f - df` and `f + df` for both definitions.

References

Gabor, D. (1946). Theory of communication. Part 1: The analysis of information. Journal of the Institution of Electrical Engineers - Part III 93, 429-441.

Examples

```
# 2 s record, 100 Hz: practical 0.5 Hz, Heisenberg-Gabor limit 0.040 Hz, 0.5 percent
fft_frequency_uncertainty(T_length = 2, f = 100)
# 1000 samples every 0.5 kyr and the 405 kyr eccentricity cycle
fft_frequency_uncertainty(N = 1000, fs = 1 / 0.5, period = 405)
```

`fft_uncertainty` *Calculate the uncertainty associated with a tracked cycle based on the frequency uncertainty of the FFT*

Description

Fourier counterpart of `wavelet_uncertainty`. A tracked cycle (depth, period) gets an uncertainty in frequency that equals the bin spacing of an FFT of the analysed record ($1/T$, the practical resolution) or the Heisenberg-Gabor limit ($1/(4\pi T)$), see `fft_frequency_uncertainty`. In contrast to the wavelet the uncertainty does not depend on the frequency, only on the length of the record (or of the window of an evolutive spectrum). The uncertain frequencies are converted to curves of period (sedimentation rate) and with `curve2time` to uncertain time models.

Usage

```
fft_uncertainty(
  tracked_cycle = NULL,
  period_of_tracked_cycle = NULL,
  signal_length = NULL,
  type = c("practical", "gabor"),
  multi = 1,
  verbose = FALSE,
  genplot_time = FALSE,
  genplot_uncertainty = FALSE,
  keep_editable = FALSE
)
```

Arguments

`tracked_cycle`	Curve of the tracked cycle: a matrix or data frame whose first column is depth or time and the second column the period.
`period_of_tracked_cycle`	Period of the tracked cycle in time (e.g. 405 kyr).
`signal_length`	Length of the record (or window) that sets the frequency uncertainty, in the unit of the depth column. `Default=NULL`: the extent of `tracked_cycle`.
`type`	`"practical"` (1/T, default) or `"gabor"` (1/(4 pi T), the theoretical limit).
`multi`	Multiple of the uncertainty that is used `Default=1`.
`verbose`	Print text `Default=FALSE`.
`genplot_time`	Plot the time curves with the upper and lower uncertainty `Default=FALSE`.

genplot_uncertainty
: Plot the period curves with the upper and lower uncertainty Default=FALSE.

keep_editable
: Keep the plot settings after plotting Default=FALSE.

Value

A data frame with the columns "depth", "period", "frequency", "uncertainty in frequency", "uncertainty in frequency x_times", "time mean", "time plus x_times" and "time min x_times" (the same layout as wavelet_uncertainty, with the Fourier uncertainty), and the attribute "type".

Examples

```
tr <- data.frame(depth = seq(0, 100, by = 1), period = 10 + 0 * seq(0, 100, by = 1))
unc <- fft_uncertainty(tr, period_of_tracked_cycle = 100, type = "practical")
tail(unc, 2)
```

---

flmw *Fit linear models to spectral peaks extracted from the wavelet spectra to astronomical cycles multiplied by sedimentation rate x*

Description

The flmw function is used calculate the linear correlation for a list of astronomical cycles transformed using a range of sedimentation rates and then compared to spectral peaks of a wavelet spectra

Usage

```
flmw(
  wavelet = NULL,
  sedrate_low = NULL,
  sedrate_high = NULL,
  spacing = NULL,
  cycles = c(NULL),
  x_lab = "depth",
  y_lab = "sedrate",
  run_random = FALSE,
  rand_simulations = 1000,
  run_multicore = FALSE,
  n_cores = NULL,
  genplot = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  plot_res = 2,
  keep_editable = FALSE,
  verbose = FALSE,
  method = c("classic", "weighted", "template"),
```

```
  tolerance = 0.1,
  whiten = TRUE,
  colour_brewer = NULL
)
```

Arguments

wavelet	Wavelet object created using the analyse_wavelet function
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"
run_random	run multiple simulation to calculate percentile against the 0 hypothesis
rand_simulations	nr of simulations to calculate percentile against the 0 hypothesis
run_multicore	run simulation using multiple cores Default=FALSE the simulation is run at x-2 cores to allow the 2 remaining processes to run background processes
n_cores	Number of cores used when 'run_multicore = TRUE' ('Default = NULL': an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set 'options(waverider.n_cores = ...)' to change the default. A value > 1 also switches on parallel computing. Results do not depend on the number of cores.
genplot	Generate plot Default="FALSE"
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer,

	grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
plot_res	options 1-8 (only used with method = "classic"; the other methods always plot the score) option 1: slope coefficient, option 2: r squared, option 3: nr of components, option 4: difference to the origin , option 5: slope coefficient percentile option 6: r squared percentile, option 7: nr of components percentile, option 8: difference to the origin percentile Default=2
keep_editable	Keep option to add extra features after plotting Default=FALSE
verbose	Print text Default=FALSE.
method	Method used to compare the cycles with the spectrum. "classic" (default) is the original peak based method (output identical to earlier versions); "weighted" matches predicted cycles one-to-one to spectral peaks, weighting by peak power and using a relative (log period) misfit; "template" correlates the spectrum with Gaussian peaks at the predicted periods (no peak picking, fastest and usually most robust).
tolerance	Relative period tolerance (standard deviation in log period, 0.1 is about 10 percent) used by the "weighted" and "template" methods Default=0.1.
whiten	Divide the wavelet power of every period by its average over depth before comparing (removes the red background) for the "weighted" and "template" methods Default=TRUE.
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

Returns a list. For method = "classic" it has the 10 elements below plus score_mat (a heuristic combined score between 0 and 1), best_sedrate (sedimentation rate with the highest score per depth) and method. For the other methods the list has score_mat, best_sedrate, depth, y_axis and method. Use flmw_path to turn score_mat into a smooth sedimentation rate curve with uncertainty.

Elements of the classic result: element 1: slope coefficient element 2: r squared element 3: nr of components element 4: difference to the origin element 5: slope coefficient percentile element 6: r squared percentile element 7: nr of components percentile, element 8: difference to the origin percentile element 9: y-axis values of the matrices which is sedimentation rate element 10: x-axis values of the matrices which is depth

Choosing a method

"classic" keeps the earlier results reproducible. In a synthetic test with strong noise and a sedimentation rate that changes with depth, "weighted" and "template" followed by flmw_path recovered the true curve far better than the classic score (see the NEWS file). "template" needs no peak picking and is the fastest.

Author(s)

Based on the eAsm function of the 'astrochron' R package and the 'eCOCO' and 'COCO' function of the 'Acycle' software

References

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

Acycle: Time-series analysis software for paleoclimate research and education, Mingsong Li, Linda Hinnov, Lee Kump, Computers & Geosciences,Volume 127,2019,Pages 12-22,ISSN 0098-3004, <doi:10.1016/j.cageo.2019.02.011>

Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing, Mingsong Li, Lee R. Kump, Linda A. Hinnov, Michael E. Mann, Earth and Planetary Science Letters,Volume 501, T2018,Pages 165-179,ISSN 0012-821X,<doi:10.1016/j.epsl.2018.08.041>

Examples

```
#estimate sedimentation rate for the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018).

mag_wt <- analyse_wavelet(data = mag[1:200, ],
dj = 1/10,
lowerPeriod = 0.1,
upperPeriod = 32,
verbose = FALSE,
omega_nr = 10)

sedrates <- flmw(wavelet = mag_wt,
    sedrate_low = 0.5,
    sedrate_high = 4,
    spacing = 1,
    cycles = c(2376,1600,1180,696,406,110),
    x_lab = "depth",
    y_lab = "sedrate",
    run_random = FALSE,
    rand_simulations = 50, # increase to get better constrainted resutls
    run_multicore = FALSE,
    genplot = FALSE,
    palette_name = "rainbow",
    color_brewer = "grDevices",
    plot_res = 2,
    keep_editable=FALSE,
    verbose=FALSE)

# template method (no peak picking) followed by a smooth sedimentation rate curve
sed_t <- flmw(wavelet = mag_wt, sedrate_low = 0.5, sedrate_high = 4, spacing = 0.5,
    cycles = c(406, 110), method = "template")
sed_path <- flmw_path(sed_t)
```

flmw_path *Smooth sedimentation rate curve with uncertainty from a flmw score matrix*

Description

Turns the sedimentation rate x depth score matrix of `flmw` into a continuous sedimentation rate curve. The sedimentation rate is not allowed to jump around from one depth to the next: the function finds the most probable path through the score matrix (Viterbi algorithm) and also the probability of every sedimentation rate at every depth (forward-backward algorithm of a hidden Markov model). The probabilities give credible intervals, and prior bounds or known hiatuses can be supplied.

Usage

```
flmw_path(
  x,
  jump_sd = 0.2,
  sharpness = 3,
  normalize = TRUE,
  sedrate_range = NULL,
  hiatus = NULL,
  probs = c(0.025, 0.16, 0.84, 0.975),
  genplot = FALSE,
  x_lab = "depth",
  y_lab = "sedrate",
  normalise = NULL
)
```

Arguments

`x`	Result of `flmw` (any `method`).
`jump_sd`	Smoothness: standard deviation of the relative (log) change in sedimentation rate over 1 metre of depth, scaled with the square root of the depth step (random walk). Small values give a stiff curve `Default=0.2`.
`sharpness`	How strongly the score is trusted. The likelihood of a sedimentation rate is `exp(sharpness * score)` with scores between 0 and 1. Lower values give wider intervals `Default=3`.
`normalize`	Divide the scores by their 99th percentile (and cap at 1) so that `sharpness` means the same for every method `Default=TRUE`.
`sedrate_range`	Optional vector `c(min, max)` (cm/kyr): sedimentation rates outside this range are excluded.
`hiatus`	Optional depths of hiatuses or other known breaks. At these depths the sedimentation rate may change freely.
`probs`	Probabilities for the credible interval limits `Default=c(0.025, 0.16, 0.84, 0.975)`.

genplot	Plot the posterior probability with the path Default=FALSE.
x_lab	Label of the x-axis Default="depth".
y_lab	Label of the y-axis Default="sedrate".
normalise	British spelling of normalize; if given it overrides normalize.

Value

A list with depth, y_axis (trial sedimentation rates), path (most probable sedimentation rate curve), mean (posterior mean), interval (matrix with the credible interval limits for every depth), posterior (matrix of probabilities, rates x depths) and score_mat.

Examples

```
# synthetic score matrix with a gradual change in sedimentation rate
depth <- seq(0, 20, by = 0.1)
rates <- seq(1, 6, by = 0.25)
true_rate <- 2 + 3 * depth / 20
score <- outer(rates, true_rate, function(r, t) exp(-0.5 * ((r - t) / 0.4)^2))
set.seed(1)
score <- pmin(pmax(score + matrix(rnorm(length(score), 0, 0.2), nrow(score)), 0), 1)
fake <- list(score_mat = score, depth = depth, y_axis = rates)
res <- flmw_path(fake)
head(res$path)
```

---

gaussian_bandpass *Gaussian band-pass filter (Acycle)*

Description

The gaussian_bandpass function applies the Acycle Gaussian band-pass filter to a regularly sampled series. The value mean is removed, the series is zero padded to 10 * 2^nextpow2(n) samples and multiplied in the Fourier domain by a symmetric Gaussian response centred on (f_low + f_high) / 2 whose lower and upper frequencies are exact half-power (-3 dB, gain 1/sqrt(2)) points. The amplitude envelope is the modulus of the analytic signal (MATLAB hilbert) of the filtered series. The Gaussian parameterisation mimics the Gaussian filter of Analyseries 2.0 (Paillard et al., 1996), as implemented by K. Ramamurthy (2013) and corrected by L. Hinnov (2016). Translated from Acycle v3.1 acycleBandpassFilter.m (method "gaussian") and gaussfilter.m (Mingsong Li).

Usage

```
gaussian_bandpass(
  data = NULL,
  f_low = NULL,
  f_high = NULL,
  genplot = FALSE,
```

```
  x_lab = "depth",
  y_lab = "proxy"
)
```

Arguments

`data`	Input data set, a matrix or data frame with 2 columns: the first column being depth/time (strictly increasing and regularly sampled to a relative spacing tolerance of 1e-5) and the second column being the proxy value.
`f_low`	Lower half-power frequency of the pass band (cycles per unit of the first column).
`f_high`	Upper half-power frequency of the pass band; must be larger than `f_low` and strictly below the Nyquist frequency.
`genplot`	Generate a plot of the data, the filtered signal, its envelope and the filter response `Default=FALSE`.
`x_lab`	label for the x-axis `Default="depth"`.
`y_lab`	label for the y-axis `Default="proxy"`.

Value

A list with elements

filtered matrix with columns `depth`, `filtered` and `envelope`.

response matrix with columns `frequency` and `gain`: the one-sided Gaussian response (0 to Nyquist of the padded FFT grid).

f_low, f_high, f_centre the pass band used.

sample_interval median sample spacing.

nyquist Nyquist frequency `1 / (2 * sample_interval)`.

nfft length of the zero-padded FFT.

input_mean mean removed before filtering.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li (Acycle), Linda Hinnov and K. Ramamurthy (gaussfilter).

References

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22. <doi:10.1016/j.cageo.2019.02.011>

Paillard, D., Labeyrie, L., Yiou, P., 1996. Macintosh program performs time-series analysis. Eos 77, 379. <doi:10.1029/96EO00259>

Examples

```
depth <- seq(0, 200, by = 0.5)
proxy <- sin(2 * pi * depth / 20) + 0.5 * sin(2 * pi * depth / 5)
res <- gaussian_bandpass(data = cbind(depth, proxy), f_low = 0.04,
  f_high = 0.06, genplot = FALSE)
plot(res$filtered[, 1], res$filtered[, 2], type = "l")
```

---

geo_col *Generate standard color codes for the Geological Time Scale*

Description

Generates the R color code which corresponds its respective geological subdivision

Usage

```
geo_col(name = NULL)
```

Arguments

name	Name of the geologchronological subdivision

Value

Returns the color code of the geological subdivision

References

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

Examples

```
#generate the Silurian part of the GTS
plot.new()
plot(
 x = c(0, 1),
 y = c(419.2, 443.8),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(419, 444))
)            # Draw empty plot
```

```
polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Rhuddanian"),
 col =geo_col("Rhuddanian")
)

text(
 0.85,geo_mid("Rhuddanian"),
 "Rhuddanian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Aeronian"),
 col =geo_col("Aeronian")
)

text(
 0.85,geo_mid("Aeronian"),
 "Aeronian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Telychian"),
 col =geo_col("Telychian")
)

text(
 0.85,geo_mid("Telychian"),
 "Telychian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Sheinwoodian"),
 col =geo_col("Sheinwoodian")
)

text(
 0.85,geo_mid("Sheinwoodian"),
 "Sheinwoodian",
 cex = 1,
 col = "black",
```

```
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Homerian"),
 col =geo_col("Homerian")
)

text(
 0.85,geo_mid("Homerian"),
 "Homerian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Gorstian"),
 col =geo_col("Gorstian")
)

text(
 0.85,geo_mid("Gorstian"),
 "Gorstian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Ludfordian"),
 col =geo_col("Ludfordian")
)

text(
 0.85,geo_mid("Ludfordian"),
 "Ludfordian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Pridoli_Age"),
 col =geo_col("Pridoli_Age")
)
```

```
polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Pridoli"),
 col =geo_col("Pridoli")
)

text(
 0.5,geo_mid("Pridoli"),
 "Pridoli",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Ludlow"),
 col =geo_col("Ludlow")
)

text(
 0.5,geo_mid("Ludlow"),
 "Ludlow",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Wenlock"),
 col =geo_col("Wenlock")
)

text(
 0.5,geo_mid("Wenlock"),
 "Wenlock",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Llandovery"),
 col =geo_col("Llandovery")
)

text(
 0.5,geo_mid("Llandovery"),
```

```
 "Llandovery",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0, 0.33, 0.33, 0),
 y = geo_loc("Silurian"),
 col =geo_col("Silurian")
)

text(
 0.165,geo_mid("Silurian"),
 "Silurian",
 cex = 1,
 col = "black",
 srt = 0
)
```

---

geo_loc *Generates ages for the boundaries of a geochronological subdivision*

Description

Generates ages for the boundaries of a geochronological subdivision which is based on the Geological Time Scale

Usage

```
geo_loc(name = NULL)
```

Arguments

name	Name of the geologchronological subdivision

Value

Returns the ages of the boundary of a geochronological subdivision which can then be added to a polygon object

References

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

Examples

```
#generate the Silurian part of the GTS
plot.new()
plot(
 x = c(0, 1),
 y = c(419.2, 443.8),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(419, 444))
)            # Draw empty plot

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Rhuddanian"),
 col =geo_col("Rhuddanian")
)

text(
 0.85,geo_mid("Rhuddanian"),
 "Rhuddanian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Aeronian"),
 col =geo_col("Aeronian")
)

text(
 0.85,geo_mid("Aeronian"),
 "Aeronian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Telychian"),
 col =geo_col("Telychian")
)

text(
 0.85,geo_mid("Telychian"),
 "Telychian",
```

```
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Sheinwoodian"),
 col =geo_col("Sheinwoodian")
)

text(
 0.85,geo_mid("Sheinwoodian"),
 "Sheinwoodian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Homerian"),
 col =geo_col("Homerian")
)

text(
 0.85,geo_mid("Homerian"),
 "Homerian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Gorstian"),
 col =geo_col("Gorstian")
)

text(
 0.85,geo_mid("Gorstian"),
 "Gorstian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Ludfordian"),
 col =geo_col("Ludfordian")
```

```
)

text(
 0.85,geo_mid("Ludfordian"),
 "Ludfordian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Pridoli_Age"),
 col =geo_col("Pridoli_Age")
)



polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Pridoli"),
 col =geo_col("Pridoli")
)

text(
 0.5,geo_mid("Pridoli"),
 "Pridoli",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Ludlow"),
 col =geo_col("Ludlow")
)

text(
 0.5,geo_mid("Ludlow"),
 "Ludlow",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Wenlock"),
 col =geo_col("Wenlock")
)
```

```
text(
 0.5,geo_mid("Wenlock"),
 "Wenlock",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Llandovery"),
 col =geo_col("Llandovery")
)

text(
 0.5,geo_mid("Llandovery"),
 "Llandovery",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0, 0.33, 0.33, 0),
 y = geo_loc("Silurian"),
 col =geo_col("Silurian")
)

text(
 0.165,geo_mid("Silurian"),
 "Silurian",
 cex = 1,
 col = "black",
 srt = 0
)
```

`geo_mid` *Generate the mean age of a geological subdivision*

Description

Generates the mean age of a geological subdivision which is based on the Geological Time Scale

Usage

```
geo_mid(name = NULL)
```

Arguments

name	Name of the geologchronological subdivision

Value

Returns the mean age of the geochronological subdivision

References

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

Examples

```
#generate the Silurian part of the GTS
plot.new()
plot(
 x = c(0, 1),
 y = c(419.2, 443.8),
 col = "white",
 xlab = "",
 ylab = "Time (Ma)",
 xaxt = "n",
 xaxs = "i",
 yaxs = "i",
 ylim = rev(c(419, 444))
)            # Draw empty plot

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Rhuddanian"),
 col =geo_col("Rhuddanian")
)

text(
 0.85,geo_mid("Rhuddanian"),
 "Rhuddanian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Aeronian"),
 col =geo_col("Aeronian")
)

text(
 0.85,geo_mid("Aeronian"),
 "Aeronian",
 cex = 1,
```

```
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Telychian"),
 col =geo_col("Telychian")
)

text(
 0.85,geo_mid("Telychian"),
 "Telychian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Sheinwoodian"),
 col =geo_col("Sheinwoodian")
)

text(
 0.85,geo_mid("Sheinwoodian"),
 "Sheinwoodian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Homerian"),
 col =geo_col("Homerian")
)

text(
 0.85,geo_mid("Homerian"),
 "Homerian",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Gorstian"),
 col =geo_col("Gorstian")
)
```

```
text(
 0.85,geo_mid("Gorstian"),
 "Gorstian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Ludfordian"),
 col =geo_col("Ludfordian")
)

text(
 0.85,geo_mid("Ludfordian"),
 "Ludfordian",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.66, 1, 1, 0.66),
 y = geo_loc("Pridoli_Age"),
 col =geo_col("Pridoli_Age")
)



polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Pridoli"),
 col =geo_col("Pridoli")
)

text(
 0.5,geo_mid("Pridoli"),
 "Pridoli",
 cex = 1,
 col = "black",
 srt = 0
)


polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Ludlow"),
 col =geo_col("Ludlow")
)

text(
```

```
 0.5,geo_mid("Ludlow"),
 "Ludlow",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Wenlock"),
 col =geo_col("Wenlock")
)

text(
 0.5,geo_mid("Wenlock"),
 "Wenlock",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0.33, 0.66, 0.66, 0.33),
 y = geo_loc("Llandovery"),
 col =geo_col("Llandovery")
)

text(
 0.5,geo_mid("Llandovery"),
 "Llandovery",
 cex = 1,
 col = "black",
 srt = 0
)

polygon(
 x = c(0, 0.33, 0.33, 0),
 y = geo_loc("Silurian"),
 col =geo_col("Silurian")
)

text(
 0.165,geo_mid("Silurian"),
 "Silurian",
 cex = 1,
 col = "black",
 srt = 0
)
```

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 scalogram 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>

GTS_info *Information of the Geological timescale 2020*

Description

GTS_info data set consists the information of the Geological timescale 2020 including the color data of Ogg et al., (2021) The ages, durations, uncertainties and colors of the Geological timescale 2020 are included in the data set

Details

Column 1: name
Column 2: type
Column 1: top age
Column 1: top error
Column 1: bottom age
Column 1: bottom error
Column 1: Cyan value
Column 1: Magenta value
Column 1: Yellow value
Column 1: Key value
Column 1: Red Value
Column 1: Green value
Column 1: Blue value
Column 1: font style
Column 1: font color

References

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

k_from_bundling *Precession constant from bundling: precession cycles per eccentricity bundle*

Description

The scale-free way an eccentricity cycle does carry k. Count how many precession couplets make up one eccentricity bundle (or measure the ratio of the two periods, or the sideband spacing of the precession peak relative to its carrier) and invert p = n * (g_i - g_j) - f_term. No tuning, no sedimentation rate and no time scale are needed.

Usage

```
k_from_bundling(
  n,
  bundle_term = "e405",
  prec_term = "k+g5",
  age_ma,
  n_sd = 0.5,
  nsim = 20000,
  method = c("angular_momentum", "astrogeo22"),
  sampleorb = "U",
  g4g3_range_kyr = NULL,
  s4s3_range_kyr = NULL,
  seed = NULL,
  data_dir = NULL
)
```

Arguments

n	counted precession cycles per bundle (may be fractional)
bundle_term	the eccentricity term forming the bundle ("e405" by default; also "e125"/"e130", "e124", "e99", "e95", or any gi-gj)
prec_term	which precession (or obliquity) term is being bundled
age_ma	approximate age, for the secular frequencies and the obliquity
n_sd	1-sigma counting uncertainty on n (default 0.5 cycles); ‘nsim = 0‘ returns the plain point estimate
nsim	number of Monte Carlo draws
method	"angular_momentum" or "astrogeo22"
sampleorb, g4g3_range_kyr, s4s3_range_kyr	passed to the secular-frequency sampler
seed	optional RNG seed
data_dir	folder with the data tables

Value

list with the Earth-Moon ‘summary‘, ‘k_kyr‘, ‘k_sd_kyr‘, ‘k_ci95_kyr‘, ‘n‘, ‘terms‘, ‘samples‘, ‘predicted_n_model‘, ‘reference‘ and ‘method‘

---

k_from_cycles *Precession constant from one or more single-valued cycles*

Description

Each cycle is read as a named astronomical term and inverted directly: k = 360*3600 / (f_observed - f_term). Uncertainty is propagated by Monte Carlo over the cycle’s own uncertainty (frequency resolution or ‘tol‘) and over the secular frequencies at that age.

Usage

```
k_from_cycles(
  cycles,
  age_ma,
  record_length_kyr = NULL,
  tol = NULL,
  mult = 1,
  nsim = 20000,
  method = c("angular_momentum", "astrogeo22"),
  combine = c("multiply", "pool"),
  seed = NULL,
  tuned_to = NULL,
  anchor_term = "e405",
  sampleorb = "U",
  g4g3_range_kyr = NULL,
  s4s3_range_kyr = NULL,
  s3_uncertainty = c("published", "scaled"),
  data_dir = NULL
)
```

Arguments

cycles	named vector of cycle periods in kyr, names are terms ("k+s3", "k+g5", "k+g2", "k+g4", "k+g3"), e.g. c("k+s3" = 30.8, "k+g5" = 19.9)
age_ma	approximate age (Ma); single value or c(min, max)
record_length_kyr	length of the analysed record in kyr, for the frequency resolution; or give 'tol' (relative half-width)
tol, mult	see cycle_band()
nsim	number of Monte Carlo draws; 0 gives the plain point estimate
method	"angular_momentum" or "astrogeo22"
combine	"multiply" (product of the individual densities, as in Zeeden et al. 2023) or "pool" (all samples together)
seed	optional RNG seed
tuned_to	if the record was tuned, the period you assumed for the tuning target in kyr (e.g. 405). The true duration of that target at this age is uncertain, so every measured period is stretched by (true target period)/(tuned_to); set this to propagate that error. NULL (default) treats the input periods as true durations.
anchor_term	the tuning target term
sampleorb, g4g3_range_kyr, s4s3_range_kyr, s3_uncertainty	passed to the secular-frequency sampler
data_dir	folder with the data tables

Value

list: ‘per_term‘ (one row per cycle), ‘combined‘ (k and Earth-Moon state), ‘samples‘, and ‘consistent‘ (do the per-term 95% intervals overlap)

---

k_from_period *Precession constant implied by one measured cycle period*

Description

Precession constant implied by one measured cycle period

Usage

```
k_from_period(period_kyr, term = "k+s3", gs = LA2004)
```

Arguments

period_kyr	measured period, kyr
term	one of k+s3, k+g5, k+g2, k+g4, k+g3
gs	Vector of fundamental g frequencies (‘NULL‘ = LA2004 values).

---

k_from_ratio *Precession constant from the RATIO of two cycles in one section*

Description

Scale-free: the ratio of two cycle periods equals the ratio of the two wavelengths, so this works on an untuned depth series just as well as on a tuned record, and no sedimentation rate is needed. Solving T1/T2 = (p + f2)/(p + f1) for the precession frequency p gives p = (f2 - R f1) / (R - 1), R = T1/T2, T1 > T2.

Usage

```
k_from_ratio(
  long,
  short,
  term_long = "k+s3",
  term_short = "k+g5",
  age_ma,
  rel_sd = 0.03,
  record_length = NULL,
  mult = 1,
  nsim = 20000,
  method = c("angular_momentum", "astrogeo22"),
  seed = NULL,
  data_dir = NULL
)
```

Arguments

long, short	the two cycles, same units (kyr or m), long one first
term_long, term_short	which terms they are
age_ma	approximate age (Ma), for the secular frequencies and obliquity
rel_sd	relative 1-sigma uncertainty on each measured cycle (default 0.03); or give 'record_length' in the same units as the cycles to use the frequency resolution instead. 'nsim = 0' skips the uncertainty entirely and returns the plain point estimate.
record_length, mult	see cycle_band()
nsim	number of Monte Carlo draws
method	"angular_momentum" or "astrogeo22"
seed	optional RNG seed
data_dir	folder with the data tables

Details

Either cycle may be an eccentricity term ("e405", "e125", "e95", ...). Pairing an eccentricity cycle with a precession or obliquity cycle is often the most robust option in an untuned section, because the eccentricity terms carry no k and are the best known frequencies.

Value

list with the Earth-Moon 'summary', 'k_kyr', 'k_sd_kyr', 'k_ci95_kyr', 'ratio', 'samples', 'terms', 'reference' and 'method'

---

k_hypotheses *Which precession term is that peak? One k per hypothesis.*

Description

With only one or two precession cycles the assignment matters more than the measurement. This returns the precession constant and Earth-Moon state implied by reading a single cycle as each of the candidate terms.

Usage

```
k_hypotheses(
  period_kyr,
  age_ma,
  terms = c("k+g5", "k+g2", "k+g4", "k+g3", "k+s3"),
  method = c("angular_momentum", "astrogeo22"),
  data_dir = NULL
)
```

Arguments

period_kyr	the observed cycle period (kyr)
age_ma	approximate age, used for the obliquity and the model row
terms	candidate terms
method	"angular_momentum" or "astrogeo22"
data_dir	folder with the AstroGeo22 table

Value

data.frame, one row per candidate term

---

k_scan *Scan k and count how many measured windows a single k can satisfy*

Description

Scan k and count how many measured windows a single k can satisfy

Usage

```
k_scan(
  peaks,
  terms = c("k+s3", "k+g5", "k+g2", "k+g4"),
  k_grid = seq(12, 40, by = 0.02),
  tol = 0.04,
  gs = LA2004
)
```

Arguments

peaks	numeric vector of measured spectral peak periods, kyr
terms	Frequency terms to scan, e.g. ‘"k+s3"‘.
k_grid	Grid of precession constants (arcsec/yr).
tol	relative tolerance for a match (0.04 = 4%)
gs	Vector of fundamental g frequencies (‘NULL‘ = LA2004 values).

la2004_extrap *La2004 polynomial extrapolation (Laskar et al. 2004, eqs. 39-40) Strictly valid only over a few hundred Myr; shown for comparison.*

Description

La2004 polynomial extrapolation (Laskar et al. 2004, eqs. 39-40) Strictly valid only over a few hundred Myr; shown for comparison.

Usage

```
la2004_extrap(age_ma)
```

Arguments

age_ma	Age(s) in Ma.

lag_1 *lag-1 autocorrelation coefficient*

Description

The lag_1 function calculates the lag-1 autocorrelation coefficient using a windowed analysis monte carlo analysis

Usage

```
lag_1(
  data = NULL,
  n_sim = 10,
  run_multicore = FALSE,
  n_cores = NULL,
  win_max = NULL,
  win_min = NULL,
  verbose = FALSE
)
```

Arguments

data	Input data set should consist of a matrix with 2 columns with first column being depth and the second column being a proxy
n_sim	number of simulations to be ran
run_multicore	Run function using multiple cores Default="FALSE"

n_cores	Number of cores used when 'run_multicore = TRUE' ('Default = NULL': an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set 'options(waverider.n_cores = ...)' to change the default. A value > 1 also switches on parallel computing. Results do not depend on the number of cores.
win_max	maximum window size
win_min	minimum window size
verbose	print text

Value

Returns a matrix which contains 3 columns column 1: depth/time matrix column 2: mean autocorrelation coefficient column 3: sd autocorrelation coefficient

Author(s)

Michiel Arts

Examples

```
#The example uses the magnetic susceptibility data set of Pas et al., (2018).
# perform the CWT
mag_wt <- analyse_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
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 (meters)",
#                                  x_lab = "depth (meters)")

#Instead of tracking, the tracked solution data set 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 = FALSE
)

# smooth the tracking of the 405 kyr eccentricity cycle
mag_track_complete <- loess_auto(time_series = mag_track_complete,
```

```
genplot = FALSE, print_span = FALSE)
#convert period in meters to sedrate depth vs time
mag_track_time<- curve2tune(data=mag,
                            tracked_cycle_curve=mag_track_complete,
                            tracked_cycle_period=405,
                            genplot = FALSE,
                            keep_editable=FALSE)

mag_lag_1 <- lag_1(data = mag_track_time,n_sim = 10,
run_multicore = FALSE,
win_max = 505,
win_min = 150,
verbose=FALSE)
```

`lithlog_disc` *Discriticizes lithologs*

Description

Discriticizes lithologs to allow further time-series analysis first the Greatest common divisor/highest common factor is calculated which is then used to discriticize the litholog to an evenly sampled data series. The function is designed to place the boundary at the original depth level of the bed boundaries. The Greatest common divisor/highest common factor can be a very small number as such the discriticized data set can be large which impacts computational performance later on therefore a linear interpolation option is added to downscale the data to allow for computational efficiency later on. This is made to discriticize lithologs created using the 'StratigrapheR' package. as such the same data format for input is used. eg. column 1 is bottom of the bed, column 2 is top of bed, column is depth rank/proxy value

Usage

```
lithlog_disc(
  litholog = NULL,
  subset_fact = 10,
  lin_interp = FALSE,
  dt = NULL,
  genplot = FALSE,
  x_lab = "rank",
  y_lab = "depth (m)",
  keep_editable = FALSE
)
```

Arguments

`litholog`	litholog input matrix with 3 columns column 1 is bottom of the bed, column 2 is top of bed, column is depth rank/proxy value

subset_fact	subset factor which is x times the greatest common divider Default=10.
lin_interp	Linear interpolation of the data set Default=FALSE
dt	step size Default=NULL.
genplot	generate plot Default=FALSE
x_lab	label for the y-axis Default="rank"
y_lab	label for the y-axis Default="depth (m)"
keep_editable	Keep option to add extra features after plotting Default=FALSE

Value

Returns a matrix with 2 columns, the first column is depth the second columns is the depth/rank proxy If genplot is Default=TRUE then a plot of the discriticizes time series is plotted

References

Wouters, S., Da Silva, A.-C., Boulvain, F., and Devleeschouwer, X.. 2021. StratigrapheR: Concepts for Litholog Generation in R. The R Journal. <doi:10.32614/RJ-2021-039>

Examples

```
# Convert depth rank record to a discrete proxy record to allow for further
# analysis in which discrete time series are needed
depth_rank_example_disc <- lithlog_disc(litholog = depth_rank_example,
           subset_fact = 10,
           genplot = FALSE,
           x_lab = "rank",
           y_lab = "depth (m)",
           keep_editable=FALSE)
```

---

lod_from_moon_distance

Length of day implied by an Earth-Moon distance (angular momentum conserved)

Description

Length of day implied by an Earth-Moon distance (angular momentum conserved)

Usage

```
lod_from_moon_distance(a_RE, kind = "solar", L_total = NULL)
```

Arguments

a_RE	Earth-Moon distance in Earth radii.
kind	Type of day, ‘"solar"‘ or ‘"sidereal"‘.
L_total	Total angular momentum (‘NULL‘ = present-day Earth-Moon value).

---

lod_hours *Length of day from Earth's angular velocity*

Description

Length of day from Earth's angular velocity

Usage

```
lod_hours(omega, kind = c("solar", "sidereal"))
```

Arguments

omega	Angular velocity of the Earth (rad/s).
kind	Type of day, ‘"solar"‘ or ‘"sidereal"‘.

---

loess_auto *Perform an automatically loess based smoothing of a time series*

Description

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

Usage

```
loess_auto(
  time_series = NULL,
  genplot = FALSE,
  print_span = FALSE,
  keep_editable = FALSE
)
```

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.
keep_editable	Keep option to add extra features after plotting Default=FALSE

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' R package.

References

Cleveland, W. S. (1979) Robust locally weighted regression and smoothing scatter plots. Journal of the American Statistical Association. 74, 829–836. <doi:10.1080/01621459.1979.10481038> 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 <doi:10.1111/1467-9868.00125> Golub, G., Heath, M. and Wahba, G. (1979). Generalized cross validation as a method for choosing a good ridge parameter. Technometrics. 21, 215–224. <doi:10.2307/1268518>

Examples

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

mag_wt <- analyse_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
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 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 = FALSE,
  keep_editable=FALSE
)

#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 = FALSE, print_span = FALSE,keep_editable=FALSE)
```

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

Description

The magnetic susceptibility data set consists of the magnetic susceptibility measurements of 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 scalogram 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 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, pts)
```

Arguments

`data`	Matrix or data frame with the first column being depth or time and the second column being a proxy
`pts`	The pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm `Default=3`

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 Steinhilber et al., (2012)
#after which all maxima are extracted

TSI_wt <-
analyse_wavelet(
```

```
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = FALSE,
   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)
```

`mesa_average` *Average spectrum of an evolutive MESA*

Description

Averages the spectra of all window positions of an evolutive MESA (`mesa_evolutive`): the average amplitude spectrum, shown beside the spectral image in many cyclostratigraphic workflows. Amplitude is the square root of the power.

Usage

```
mesa_average(x, window = 1, normalise = TRUE)
```

Arguments

`x`	output of `mesa_evolutive`.
`window`	number (position in `x$windows`) or length of the window.
`normalise`	If TRUE (default) every window position is divided by its maximum before averaging, so that all positions weigh equally.

Value

data frame with `frequency`, `period` and `amplitude`.

`mesa_evolutive` *Evolutive maximum entropy spectral analysis (evolutive MESA)*

Description

Slides a window along a record and computes a maximum entropy (Burg) spectrum (`mesa_spectrum`) in every window position, giving the change of the spectrum with depth or time. Several window lengths can be given at once, because the best window depends on the cycle: a short window follows changes in the cycle period closely but resolves long cycles badly, a long window resolves them but smooths out changes. Near the top and bottom of the record the window extends beyond the data; this part is padded (see `pad`), so that the evolutive spectrum spans the entire record. The spectral peaks of every window position are extracted and can be followed with `track_mesa` and plotted with `plot_mesa_evolutive`.

Usage

```
mesa_evolutive(
  data = NULL,
  window = NULL,
  step = NULL,
  order = NULL,
  order_max = NULL,
  criterion = c("fpe", "aic", "mdl", "cat"),
  demean = TRUE,
  detrend = FALSE,
  regularise = TRUE,
  pad = c("reflect", "mean", "zero", "none"),
  min_cover = 0.5,
  nfreq = 512,
  fmin = 0,
  fmax = NULL,
  n_peaks = 8,
  min_prominence = 1.5
)
```

Arguments

`data`	Input is a time series with the first column being depth or time and the second column being a proxy.
`window`	Window length (in the units of the first column). A vector gives several windows, for example `c(30, 40, 50)`. The default is a fifth of the record.
`step`	Distance between the centres of successive windows. The default is a tenth of the shortest window.
`order`	AR order used in every window. The default NULL chooses the order in each window with `criterion`; a fixed order gives windows of the same spectral smoothness, which is easier to compare along the record.

`order_max`, `criterion`, `demean`, `detrend`, `regularise`
: see `mesa_spectrum`. The mean (or trend) is removed in every window.

`pad`
: Filling of the part of a window that lies outside the record: `"reflect"` (mirror the record at its edge; default), `"mean"`, `"zero"` or `"none"` (the window is cut at the edge of the record; windows with less than `min_cover` of their length inside the record are left empty).

`min_cover`
: For `pad = "none"`: smallest fraction of a window that must lie in the record.

`nfreq`
: Number of frequencies of the spectra.

`fmin`, `fmax`
: Frequency range. Defaults 0 and the Nyquist frequency.

`n_peaks`
: Maximum number of peaks stored per window position (the strongest).

`min_prominence`
: Peaks with a prominence (peak power divided by the higher of its two neighbouring minima) below this value are not stored.

Details

Powers are not normalised in the object; `plot_mesa_evolutive` and `track_mesa` normalise every window position by its maximum, so that weak and strong intervals of the record are comparable.

Value

A list of class `mesa_evolutive` with

windows one element per window length with `window`, `power` (matrix, frequencies in rows and window positions in columns), `order` (AR order of each position) and `peaks` (data frame with position, frequency, period, power, relative power and prominence).

frequency frequencies of the rows.

position centres of the windows.

dt, range sampling interval and the range of the record.

Examples

```
# a cycle that doubles its period half way
set.seed(2)
t <- seq(0, 800, by = 2)
y <- sin(2 * pi * cumsum(1 / ifelse(t < 400, 40, 80)) * 2) + rnorm(length(t), sd = 0.4)
ev <- mesa_evolutive(data.frame(t, y), window = c(150, 250), order = 20)
plot_mesa_evolutive(ev, vertical = FALSE)
```

`mesa_spectrum` *Maximum entropy spectral analysis (MESA, Burg)*

Description

Estimates the power spectrum of a time series with the maximum entropy method of Burg (1967, 1975). An autoregressive (AR) model is fitted to the series with the Burg recursion, which minimises the forward and backward prediction error, and the power spectrum is the spectrum of that AR model. Compared with the periodogram or the multi-taper method (`astrochron::mtm`), MESA gives very sharp spectral lines for short records and needs no window, which is useful for the few cycles typical of cyclostratigraphic data. The price is that the number of peaks and their width depend on the model order, so the order is chosen with an information criterion (default the final prediction error, FPE) or set by hand. Spurious peak splitting at high orders is a known property of the method; compare several orders or criteria (see `criteria` in the output) before interpreting a peak.

Usage

```
mesa_spectrum(
  data = NULL,
  order = NULL,
  order_max = NULL,
  criterion = c("fpe", "aic", "mdl", "cat"),
  demean = TRUE,
  detrend = FALSE,
  regularise = TRUE,
  nfreq = 2048,
  fmin = 0,
  fmax = NULL,
  n_peaks = 10,
  window = NULL,
  pad = c("reflect", "mean", "zero", "none")
)
```

Arguments

`data`	Input is a time series with the first column being depth or time and the second column being a proxy. It should be evenly sampled, see `regularise`.
`order`	Order of the AR model (number of coefficients). The default NULL selects the order with `criterion`.
`order_max`	Largest order considered when the order is selected. The default is `min(floor(N / 3), 300)` for `N` samples.
`criterion`	Order-selection criterion: `"fpe"` (final prediction error, Akaike 1969), `"aic"` (Akaike information criterion), `"mdl"` (minimum description length of Rissanen, the most conservative) or `"cat"` (criterion autoregressive transfer of Parzen). Ignored when `order` is given.

demean	Remove the mean of the series before the analysis (default TRUE).
detrend	Remove a linear trend before the analysis (default FALSE).
regularise	If TRUE (default) an unevenly sampled series is linearly interpolated to its median sampling interval (with a message); if FALSE an uneven series is an error.
nfreq	Number of frequencies at which the spectrum is evaluated between fmin and fmax (default 2048).
fmin, fmax	Frequency range of the spectrum. Defaults are 0 and the Nyquist frequency.
n_peaks	Maximum number of spectral peaks returned (the strongest ones); default 10.
window	Optional c(from, to) (in the units of the first column) selecting one window of the record. The window may extend beyond the record; the missing part is filled according to pad, so a spectrum that spans a window centred on the very top or bottom can be made. The default NULL uses the entire record.
pad	How samples of the window outside the record are filled: "reflect" (mirror the record at its edge, the default), "mean" (the mean of the window), "zero" (the mean of the record, i.e. zero after demeaning) or "none" (the window is cut at the edge of the record). Padding only adds samples outside the record; it never changes the record.

Details

The power spectral density is one-sided and scaled so that its integral over frequency equals the variance of the (demeaned) series. The AR coefficients are returned in the convention `x[t] = sum(ar[j] * x[t - j]) + e[t]`, the same as `stats::ar.burg`.

Value

A list of class `mesa_spectrum` with elements

spectrum data frame with `frequency`, `period` and `power`.

peaks data frame with the strongest local maxima: `frequency`, `period`, `power`, and `prominence` (peak power divided by the lower of the two neighbouring minima).

order the AR order used.

ar the AR coefficients.

noise_variance the prediction error variance of the final model.

criteria data frame with the criteria (fpe, aic, mdl, cat) for every order 0 to the maximum order.

dt sampling interval.

n number of samples.

References

Burg, J.P., 1967. Maximum entropy spectral analysis. 37th Annual International Meeting, Society of Exploration Geophysicists, Oklahoma City.

Burg, J.P., 1975. Maximum entropy spectral analysis. PhD thesis, Stanford University.

Akaike, H., 1969. Fitting autoregressive models for prediction. Annals of the Institute of Statistical Mathematics 21, 243-247.

Parzen, E., 1974. Some recent advances in time series modeling. IEEE Transactions on Automatic Control 19, 723-730.

Rissanen, J., 1978. Modeling by shortest data description. Automatica 14, 465-471.

Ulrych, T.J., Bishop, T.N., 1975. Maximum entropy spectral analysis and autoregressive decomposition. Reviews of Geophysics 13, 183-200.

Examples

```
# two cycles (100 and 41 units) plus noise in a short record
set.seed(1)
t <- seq(0, 600, by = 2)
y <- sin(2 * pi * t / 100) + 0.6 * sin(2 * pi * t / 41) + rnorm(length(t), sd = 0.5)
m <- mesa_spectrum(data.frame(t, y))
m$order
m$peaks
plot_mesa_spectrum(m)

# a fixed, low order gives a smoother spectrum
m8 <- mesa_spectrum(data.frame(t, y), order = 8)
plot_mesa_spectrum(m8, peaks = FALSE)
```

---

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 = NULL, pts = 3)
```

Arguments

`data`	Matrix or data frame with first column being depth or time and the second column being a proxy
`pts`	the pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm `Default=3`

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
#Irradiance data set of Steinhilber et al., (2012)
#after which all minima are extracted

TSI_wt <-
analyse_wavelet(
data = TSI,
dj = 1/200,
lowerPeriod = 16,
upperPeriod = 8192,
   verbose = FALSE,
   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)
```

---

minimal_tuning *Create an age model using minimal tuning*

Description

Create an age model using the minimal tuning technique. This means that the distance between 2 peaks of an extracted cycle are set to duration of the interpreted astronomical cycle

Usage

```
minimal_tuning(
  data = NULL,
  pts = 5,
  cycle = 405,
  tune_opt = "max",
  output = 0,
  genplot = FALSE,
  keep_editable = FALSE
)
```

Arguments

data	Input is an cycle extracted filtered in the depth domain
pts	The pts parameter specifies how many points to the left/right up/down the peak detect algorithm goes in detecting a peak. The peak detecting algorithm works by comparing the values left/right up/down of it, if the values are both higher or lower then the value a peak. To deal with error produced by this algorithm the pts parameter can be changed which can aid in peak detection. Usually increasing the pts parameter means more peak certainty, however it also means that minor peaks might not be picked up by the algorithm Default=5
cycle	duration in kyr of the filtered/extracted cycle
tune_opt	tuning options "min", "max" and "minmax" use minima, maxima or both of the cyclic signal to create the age model Default="max"
output	#'The output depends on the output setting If output = 0 output is a matrix of with 4 columns being; depth,proxy,sedimentation rate and time If output = 1 output is a matrix of with 2 columns being; depth and sedimentation rate #'If output = 2 output is a matrix of with 2 columns being; depth and time
genplot	Keep option to add extra features after plotting Default=FALSE
keep_editable	Keep option to add extra features after plotting Default=FALSE

Value

The output depends on the output setting If output = 0 output is a matrix of with 4 columns being (depth,proxy,sedimentation rate and time) If genplot = TRUE 4 plots are generated; depth vs proxy, depth vs sedimentation rate, depth vs time and time vs proxy If output = 1 output is a matrix of with 2 columns being (depth and sedimentation rate) If genplot = TRUE a plot of depth vs sedimentation rate is generated If output = 2 output is a matrix of with 2 columns being (depth and time) If genplot = TRUE a plot of depth vs time is generated

Author(s)

Part of the code is based on the sedrate2time function of the 'astrochron' R package

References

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

Examples

```
# Extract the 405kyr eccentricity cycle from the wavelet scalogram
# from the magnetic susceptibility record f the Sullivan core
# of Pas et al., (2018) and then create a age model using minimal tuning
# (e.g.) set the distance between peaks to 405 kyr

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

```
verbose = FALSE,
omega_nr = 10)


mag_405 <- extract_signal_stable_V2(
 wavelet = mag_wt,
 period_max = 4,
 period_min = 2,
 add_mean = FALSE,
 plot_residual = FALSE,
 keep_editable = FALSE
)

mag_405_min_tuning <- minimal_tuning(data = mag_405,
pts = 5,
cycle = 405,
tune_opt = "max",
output = 0,
genplot = FALSE,
keep_editable = FALSE)
```

---

modBchron *Run a modified version of the Bchronology algorithm from the R package "Bchron" (Haslett and Parnell, 2008)*

Description

Run a modified version of the Bchronology algorithm from the R package "Bchron" (Haslett and Parnell, 2008)

Usage

```
modBchron(
  ages,
  ageSds,
  positions,
  positionThicknesses = rep(0, length(positions)),
  ids,
  distTypes = rep("G", length(ages)),
  MC = 10000,
  burn = 2000,
  probability = 0.95,
  adapt = TRUE,
  mhSD = rep(0.001, length(unique(ids))),
  psiSD = 0.001,
```

```
  muSD = 0.001,
  mu0 = NULL,
  psi0 = NULL,
  thetas0 = NULL,
  mu_fixed = NULL,
  psi_fixed = NULL,
  predictPositions = seq(min(positions), max(positions), length = 500),
  truncateUp = 0,
  extrapUp = 100,
  truncateDown = 1e+10,
  extrapDown = 100
)
```

Arguments

ages	Vector of ages
ageSds	Vector of 1-sigma values for ages. Must be the same length and given in the same order as ages
positions	Vector of stratigraphic positions for ages. Must be the same length and given in the same order as ages. Must be input as distance above base of section.
positionThicknesses	Vector of stratigraphic uncertainties for each age. Specified as half thicknesses. Must be the same length and given in the same order as ages
ids	Vector of sample names for each age. All samples with the same ids will be combined into a single age PDF. Must be the same length and given in the same order as ages
distTypes	c('G','U') Vector of distribution types to model each age as. Choices are 'G' for Gaussian, and 'U' uniform. Must be the same length and given in the same order as ages
MC	Number of MCMC iteration to run for. Defaults to 10000
burn	Number of initial iterations to discard. Defaults to 2000
probability	Desired probability of the returned interval. Defaults to 0.95 (95 percent)
adapt	Should the proposal standard deviation for model parameters be determined using the adaptive proposal algorithm of Haario et al., 1998. Defaults to TRUE
mhSD	Metropolis-Hastings proposal standard deviation for the age parameters. If adapt = TRUE it will be determined by the model run. Otherwise a vector of standard deviations must be specified with length = unique(ids)
psiSD	Metropolis-Hastings proposal standard deviation for the Compound Poisson-Gamma scale parameter. If adapt = TRUE it will be determined by the model run.
muSD	Metropolis-Hastings proposal standard deviation for the Compound Poisson-Gamma mean parameter. If adapt = TRUE it will be determined by the model run.
mu0	Optional warm-start centre for the initial Compound Poisson-Gamma mean parameter mu. If NULL (default), mu is initialised the original way, from the raw

date/position spacing. If given, the chain instead starts at `abs(rnorm(1, mean = mu0, sd = muSD))` – lets you restart the chain near a previously-known-good region of parameter space (e.g. in a rejection-sampling/restart scheme) instead of re-exploring from scratch every time. Does not change the Metropolis-Hastings acceptance ratio at all, only where the chain begins.

`psi0` Optional warm-start centre for the initial Compound Poisson-Gamma scale parameter psi, analogous to `mu0`. If NULL (default), psi is initialised the original way (`abs(rnorm(1, 1, sd = psiSD))`).

`thetas0` Optional warm-start vector of starting ages for the dated horizons (one per unique id, in the same order as `unique(ids)`). If NULL (default) each is initialised from its own likelihood peak, as before. This matters most in a batch-RESTART scheme: the adaptive proposal SDs start at their `mhSD` default of 0.001 and grow by only about a factor of 10 per 200-iteration adaptation window, so they need roughly 1000 iterations to reach the right scale (measured on a 7500 +/- 15 date: mhSD goes 0.001 -> 0.012 at iter 200 -> 0.157 at iter 400 -> ~34 at iter 1000). A chain restarted COLD inside a short batch never gets there: its dated horizons stay pinned within a fraction of a kyr of their starting values, so the run silently reports near-ZERO age uncertainty at the dated horizons. Passing `thetas0` together with the previous run's adapted `mhSD`/`psiSD`/`muSD` lets a restarted chain resume properly mixed instead of paying that burn-in again. Values are still clipped into [truncateUp, truncateDown].

`mu_fixed`, `psi_fixed`

Optionally HOLD the Compound Poisson-Gamma mean/scale parameters at a supplied value, skipping their Metropolis-Hastings updates entirely. NULL (default) estimates them as before.

WHY THIS EXISTS. mu and psi are estimated from the gaps BETWEEN dated horizons, so n dates give only n-1 numbers to estimate both. With a handful of dates psi is effectively UNIDENTIFIED: on a 3-date section its posterior spanned a factor of 489 (95% range 28.9 to 14100). That matters enormously, because psi controls the Poisson event rate of the age-depth bridge, lambda = mu^(2-p)/(psi*(2-p)). At the posterior median psi of ~3450 that works out at about ONE event across a whole 3750-kyr interval – the age path between two dates is then essentially a single random gamma jump, and the interior uncertainty balloons to several hundred kyr regardless of how precise the dates are. Left "free", psi is not being estimated so much as allowed to wander somewhere unhelpful.

WHAT PINNING IT BUYS (measured, floating-time model, 3 dates at +/-30 kyr, 3 replicates each): psi free : interior SD 458 kyr, RMSE of the median vs the true curve 15.8 kyr psi = 100 : interior SD 245 kyr, RMSE 20.3 kyr psi = 10 : interior SD 82 kyr, RMSE 15.4 kyr psi = 1 : interior SD 35 kyr, RMSE 15.4 kyr So a 13-fold tightening with NO loss of accuracy – the free-psi width was not information, it was an unidentified nuisance parameter wandering.

THE PRICE, AND IT IS REAL. psi encodes how much you trust the timescale you are fitting against. Pinning it small asserts that the cyclostratigraphic axis is right. Repeating the test with a deliberately WRONG axis (cycle misidentified as 25 kyr instead of 21, plus a spurious trend): psi free : interior SD 451 kyr, RMSE 94 kyr, 95% coverage 1.00 psi = 1 : interior SD 35 kyr, RMSE 94 kyr,

95% coverage 0.49 Both are equally WRONG, but the wide free-psi band still contained the truth while the pinned one did not – at half of all depths the true age fell outside the model's own 95% interval. The sprawling free-psi uncertainty was acting as an accidental safety net.

SO: pin psi only alongside the independent astronomical controls in spectral_filter.R (the spectral peak test and the cycle-count test). Those are what establish that the axis is right, and once psi is pinned they are no longer a refinement – they are the thing standing between you and a confidently wrong answer. Report a sensitivity range over psi rather than a single value.

`predictPositions`
: Vector of stratigraphic positions to evaluate the model at. Defaults to 500 evenly spaced points from the bottom to top of the section

`truncateUp`
: Truncation age for extrapolating above the top of the section. Defaults to 0. IMPORTANT: this is a HARD lower bound on the age of the TOPMOST DATED HORIZON as well, not just on extrapolated points above it – it is used directly as the 'low' limit of that horizon's Metropolis-Hastings proposal. So if your top date's own likelihood sits close to this bound (the common case being a "present day" date of 0 +/- sd with the default truncateUp = 0), roughly the lower half of that date's likelihood is excluded by the bound, and its posterior comes out correspondingly narrower and skewed. For a date of 0 +/- 15 with truncateUp = 0 the posterior is a half-normal: sd ~9.0 instead of 15, and a 95% width of ~33 kyr instead of ~59 kyr – verified against both the model output and the closed-form half-normal. That is the model correctly doing what you told it, NOT uncertainty being lost; the age-depth model's uncertainty at a dated horizon equals that date's POSTERIOR, which for any date far from a bound comes back essentially equal to the input (e.g. 7500 +/- 15 at the base returns a 95% width of ~58.6 kyr against the input's 58.8 kyr). If you did not intend the bound to bite, set truncateUp below the youngest age you want to permit (e.g. -1000) and the full date uncertainty is recovered. The same reasoning applies to `truncateDown` and the bottommost date.

`extrapUp`
: Number of extrapolations to perform above the top of the section. defaults to 100

`truncateDown`
: Truncation age for extrapolation below the bottom of the section. Defaults to 1e10 (a really big number). See the note on `truncateUp`: this is likewise a hard upper bound on the bottommost dated horizon's age, not only on extrapolated points.

`extrapDown`
: Number of extrapolations to perform below the bottom of the section. defaults to 100

Value

HDI = specified probability highest density interval for the model run

model = Raw model predictions for each MCMC iteration

thetas = The posterior values for each dated horizon from each MCMC run. burn is not removed.

predictPositions = Stratigraphic positions where the model was evaluated. Same as `predictPositions`

mu, psi = The posterior values for the Compound Poisson-Gamma mean and scale parameters. burn is not removed.

ageGrid = Grid that age PDFs were evaluated over. Useful for plotting

likelihoods = Age PDFs for each set of dated samples. Useful for plotting

nAges = Number of ages that were combined into each sample. Useful for plotting

masterPositions = Stratigraphic positions for each set of combined ages. Useful for plotting

ids = Vector of unique sample names. Useful for plotting

`modBchron_predict` *Predict ages at new positions from a modBchron (or modifiedBChron) model*

Description

This function is used to predict the age of new stratigraphic points using the output from an ageModel run.

Usage

```
modBchron_predict(
  model,
  newPositions,
  ids = 1:length(newPositions),
  newPositionThicknesses = rep(0, length(newPositions)),
  probability = 0.95
)
```

Arguments

`model`	Output of the `ageModel` function
`newPositions`	Vector of new stratigraphic positions to predict the age of. New positions must be within the bounds of the orignal model run
`ids`	optional vector of names for new positions. Defaults to 1, 2, 3, ...
`newPositionThicknesses`	Vector of stratigraphic uncertanties for each position. Specified as half thicknesses. Must be the same length and given in the same order as `newPositions`. Default is no positional uncertanty
`probability`	Probability of the returned interval. Defaults to 0.95

Value

HDI = 95 percent Highest Density Interval for each `newPosition`

raw = Age predictions for each `newPosition` for each MCMC run.

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,
  run_multicore = FALSE,
  n_cores = NULL,
  verbose = FALSE
)
```

Arguments

wavelet	Wavelet object created using the analyse_wavelet function.
n_simulations	Number of red noise simulations.
run_multicore	run simulation using multiple cores Default=FALSE the simulation is run at x-2 cores to allow the 2 remaining processes to run background processes.
n_cores	Number of cores used when 'run_multicore = TRUE' ('Default = NULL': an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set 'options(waverider.n_cores = ...)' to change the default. A value > 1 also switches on parallel computing. Results do not depend on the number of cores.
verbose	Print text Default=FALSE.

Value

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

Author(s)

Code based on the "analyze.wavelet" function of the 'WaveletComp' R package and "wt" function of the 'biwavelet' R package which are based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998).

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. "Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. " Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#'#generate average spectral power curves based on red noise curves which are
# based on the magnetic susceptibility record of the Sullivan core of Pas et al., (2018)

mag_wt <- analyse_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=10, # increase number for better constrained results
run_multicore=FALSE,
verbose=FALSE)
```

`months_per_year` *Lunar mean motion and months per year from the Earth-Moon distance (Kepler's third law, n^2 a^3 = G(M_E + M_M), as in Farhat et al. 2022 Table D.2)*

Description

Lunar mean motion and months per year from the Earth-Moon distance (Kepler's third law, n^2 a^3 = G(M_E + M_M), as in Farhat et al. 2022 Table D.2)

Usage

```
months_per_year(a_RE, kind = c("sidereal", "synodic"))
```

Arguments

a_RE	Earth-Moon distance in Earth radii.
kind	Type of month, ‘"sidereal"‘ or ‘"synodic"‘.

---

moon_distance_from_lod

Earth-Moon distance implied by a length of day (angular momentum conserved)

Description

Earth-Moon distance implied by a length of day (angular momentum conserved)

Usage

```
moon_distance_from_lod(lod_h, kind = "solar", L_total = NULL)
```

Arguments

lod_h	Length of day in hours.
kind	Type of day, ‘"solar"‘ or ‘"sidereal"‘.
L_total	Total angular momentum (‘NULL‘ = present-day Earth-Moon value).

---

moon_distance_from_months

Earth-Moon distance from the number of lunar months per year (the observable of tidal rhythmites)

Description

Earth-Moon distance from the number of lunar months per year (the observable of tidal rhythmites)

Usage

```
moon_distance_from_months(months, kind = c("sidereal", "synodic"))
```

Arguments

months	Number of months per year.
kind	Type of month, ‘"sidereal"‘ or ‘"synodic"‘.

`omega_from_moon_distance` *Earth's spin rate implied by an Earth-Moon distance, assuming the Earth-Moon angular momentum is conserved (solar tidal drain neglected).*

Description

Earth's spin rate implied by an Earth-Moon distance, assuming the Earth-Moon angular momentum is conserved (solar tidal drain neglected).

Usage

```
omega_from_moon_distance(a_RE, L_total = NULL)
```

Arguments

`a_RE`	Earth-Moon distance in Earth radii.
`L_total`	Total angular momentum ('NULL' = present-day Earth-Moon value).

`orbit9` *Nine principal astronomical terms for a given precession constant*

Description

Nine principal astronomical terms for a given precession constant

Usage

```
orbit9(k_kyr, gs = LA2004)
```

Arguments

`k_kyr`	precession constant (period of p) in kyr
`gs`	list with elements g and s, in arcsec/yr

Value

data frame: term, group, frequency (arcsec/yr), period (kyr)

orbit9_at_age *Nine orbital periods (kyr) at age_ma*

Description

The nine periods used by Acycle's COCO/eCOCO GUI (see calculate_orbit9()).

Usage

```
orbit9_at_age(age_ma, data_dir = NULL)
```

Arguments

age_ma	age in Ma.
data_dir	data folder Default=NULL.

Value

numeric vector of nine periods (kyr).

orbital_periods *All principal astronomical periods for a given precession constant*

Description

All principal astronomical periods for a given precession constant

Usage

```
orbital_periods(
  k_yr = NULL,
  p_arcsec = NULL,
  g = NULL,
  s = NULL,
  terms = c("nine", "all"),
  data_dir = NULL
)
```

Arguments

k_yr	precession constant (period of p) in years, e.g. 20700 for the Devonian estimate of Zeeden et al. (2023); or give 'p_arcsec'
p_arcsec	Precession constant in arcsec/yr.
g, s	secular frequencies in arcsec/yr (default: La2004, La04gisi.csv)
terms	"nine" (the Acycle ORBIT9 set) or "all" (every g_i-g_j and k+s_i / k+g_i combination for i = 1..5)
data_dir	Directory with the data files ('NULL' = the data installed with the package).

Value

data.frame: term, group, frequency (arcsec/yr), period (kyr)

---

pda *Power decomposition analysis (strict evolutionary MTM band-power ratio)*

Description

The pda function calculates, in moving windows, the ratio between the multitaper (MTM) power inside a set of target frequency bands and the power inside a total frequency band (power decomposition analysis, PDA; Li et al., 2016). Each window is linearly detrended, its adaptive-weighted MTM power spectral density is computed on an explicit FFT grid, and band powers are obtained by trapezoidal integration of the PSD with piecewise-linear interpolation at the exact band limits. The input is divided by one global scale (maximum absolute value) so that powers are comparable between windows. The data must already be regularly sampled (relative spacing tolerance 1e-5); no sorting, interpolation or padding is performed. Defaults for `step`, `nfft` and `total_band` follow the Acycle compatibility wrapper `pda.m` (one sample interval, next power of two of the window length, DC to Nyquist). Translated from Acycle v3.1 acyclePowerDecomposition.m and pda.m (Mingsong Li).

Usage

```
pda(
  data = NULL,
  target_bands = c(1/45, 1/25),
  window = 500,
  step = NULL,
  nw = 2,
  nfft = NULL,
  total_band = NULL,
  genplot = FALSE,
  verbose = FALSE
)
```

Arguments

`data`	Matrix with two columns: depth/time (strictly increasing, evenly spaced) and proxy value. `Default=NULL`
`target_bands`	Target frequency bands, either a B x 2 matrix [lower, upper] (rows ordered by increasing lower limit, non-overlapping) or a vector of paired limits c(lower1, upper1, lower2, upper2, ...). `Default=c(1/45, 1/25)`
`window`	Window length (first-to-last coordinate span), in units of the depth/time column. Rounded to a whole number of sample intervals. `Default=500`
`step`	Step between successive windows in depth/time units (rounded to whole samples). If NULL one sample interval is used. `Default=NULL`

nw	Time-bandwidth product of the DPSS tapers. Default=2
nfft	Even FFT length, at least the number of samples per window. If NULL the next power of two of the window sample count is used. Default=NULL
total_band	Total frequency band c(lower, upper). If NULL c(0, Nyquist) is used. Default=NULL
genplot	Plot the data and the power ratio. Default=FALSE
verbose	Show a progress bar. Default=FALSE

Value

A list with elements result: matrix with columns window center, ratio (target power / total power), target power (union of all target bands) and total power; band_powers: matrix with the window center followed by one power column per target band; meta: list with the resolved sampling interval, window and step samples, effective window/step lengths, nfft, frequency resolution, Nyquist frequency, value scale and number of unused trailing samples.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li and Linda Hinnov (Acycle).

References

Li, M., Huang, C., Hinnov, L., Ogg, J., Chen, Z.-Q., Zhang, Y., 2016. Obliquity-forced climate during the Early Triassic hothouse in China. Geology 44, 623-626. doi:10.1130/G37970.1

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22. doi:10.1016/j.cageo.2019.02.011

Examples

```
set.seed(1)
t <- seq(0, 2000, by = 2)
x <- sin(2 * pi * t / 40) + 0.5 * sin(2 * pi * t / 100) +
  0.5 * as.numeric(arima.sim(list(ar = 0.7), n = length(t)))
res <- pda(data = cbind(t, x), target_bands = c(1/45, 1/35),
  window = 300, step = 10, nw = 2, genplot = FALSE)
head(res$result)
```

---

pda_legacy *Legacy power decomposition analysis (Acycle PDA menu / DYNOT core)*

Description

The pda_legacy function reproduces the power decomposition analysis as run from the Acycle menu "Power Decomposition Analysis" and as used inside DYNOT (pdan.m; Li et al., 2016). Preprocessing as in the Acycle menu: rows with NA are removed, data are sorted, duplicated depths are replaced by their mean, unevenly sampled data (relative tolerance 1e-5) are linearly interpolated at the median sampling interval, and the series is optionally padded by half a window at both ends (zeropad2.m). In each window of trunc(window/dt) samples the data are linearly detrended and the adaptive MTM spectrum (pmtm(x, nw, pad)) is computed. Target and total powers are sums of spectral bins selected with the historical proportional bin mapping (lower = ceiling(nbin*f/fNyq), upper = trunc(nbin*f/fNyq)); overlapping target bands are counted once. Window centers are evenly spaced between first coordinate + window/2 and last coordinate - window/2. Translated from Acycle v3.1 pdan.m, zeropad2.m, interpolate.m, findduplicate.m and AC.m (menu_pda_Callback) (Mingsong Li).

Usage

```
pda_legacy(
  data = NULL,
  target_bands = c(1/45, 1/25),
  window = 500,
  nw = 2,
  f_min = 0,
  f_max = NULL,
  step = 1,
  pad = 5000,
  pad_type = 1,
  genplot = FALSE,
  verbose = FALSE
)
```

Arguments

data	Matrix with two columns: depth/time and proxy value. Default=NULL
target_bands	Vector of paired frequency limits (any order within a pair) or a B x 2 matrix. Default=c(1/45, 1/25)
window	Window length in depth/time units. Default=500
nw	Time-bandwidth product. Default=2
f_min	Lower cutoff frequency of the total band. Default=0
f_max	Upper cutoff frequency of the total band; values above the Nyquist frequency are clipped. If NULL the Nyquist frequency is used. Default=NULL
step	Step between windows in number of samples. Default=1
pad	FFT length passed to the MTM (zero-padding number). Default=5000
pad_type	Padding of the series by half a window before analysis: 0 = none, 1 = zero, 2 = mirror, 3 = mean, 4 = random. Default=1
genplot	Plot the data and the power ratio. Default=FALSE
verbose	Show a progress bar. Default=FALSE

Value

A matrix with columns window center, ratio (target / total), target power and total power.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li and Linda Hinnov (Acycle); zeropad2 padding options by Nicolas Thibault and Giovanni Rizzi.

References

Li, M., Huang, C., Hinnov, L., Ogg, J., Chen, Z.-Q., Zhang, Y., 2016. Obliquity-forced climate during the Early Triassic hothouse in China. Geology 44, 623-626. doi:10.1130/G37970.1

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22. doi:10.1016/j.cageo.2019.02.011

Examples

```
set.seed(1)
t <- seq(0, 2000, by = 2)
x <- sin(2 * pi * t / 40) + 0.5 * sin(2 * pi * t / 100) +
  0.5 * as.numeric(arima.sim(list(ar = 0.7), n = length(t)))
res <- pda_legacy(data = cbind(t, x), target_bands = c(1/45, 1/35),
  window = 300, step = 5, pad = 1000, pad_type = 1)
head(res)
```

---

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 `analyse_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

```
#'#generate average spectral power curves based on red noise curves which are
# based on the magnetic susceptibility record of the Sullivan core of Pas et al., (2018)

mag_wt <- analyse_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=10, # Increase number for a better constrained result
run_multicore=FALSE,
verbose=FALSE)


prob_curve <- percentile_from_red_noise(
red_noise = mag_wt_red_noise,
wavelet = mag_wt,
percentile = 0.9)
```

---

plot.astroMCBchron *Plot an astroMCBchron model*

Description

Three panels: (A) age model (95% range) with the dates and, if present, the modBchron model; (B) 2 sd age uncertainty of both models; (C) posterior and leave-one-out ages minus the dates. Oldest strata/ages on the left and bottom.

Usage

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

Arguments

x	an astroMCBchron object.
...	unused.

---

plot.coco_adaptive *Plot an adaptive COCO result*

Description

Correlation, global and local p, contributing periods and pCOCO against sedimentation rate.

Usage

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

Arguments

x	object returned by coco_adaptive()
...	unused

Value

invisible NULL

---

plot.coco_classic *Plot a COCO result (Li et al. 2018 method)*

Description

Correlation, H0 significance, number of contributing parameters and pCOCO against sedimentation rate.

Usage

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

Arguments

x	object returned by coco_classic()
...	unused

Value

invisible NULL

plot.ctest *Plot the CTest posterior distributions*

Description

Plot the CTest posterior distributions

Usage

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

Arguments

x	a ctest object
...	ignored

plot.ecoco_adaptive *Plot an adaptive eCOCO result*

Description

Maps of correlation, global and local p, contributing periods and normalised ridge score with the tracked sedimentation-rate ridge.

Usage

```
## S3 method for class 'ecoco_adaptive'
plot(x, ridge = TRUE, reverse_depth = FALSE, ...)
```

Arguments

x	object returned by ecoco_adaptive()
ridge	draw the tracked ridge Default=TRUE
reverse_depth	depth increasing downwards Default=FALSE
...	unused

Value

invisible NULL

plot.ecoco_classic *Plot an eCOCO result (Li et al. 2018 method)*

Description

Maps of correlation, H0 significance, contributing parameters and window-normalised score with the tracked sedimentation-rate ridge.

Usage

```
## S3 method for class 'ecoco_classic'
plot(x, ridge = TRUE, reverse_depth = FALSE, ...)
```

Arguments

x	object returned by ecoco_classic()
ridge	draw the tracked ridge Default=TRUE
reverse_depth	depth increasing downwards Default=FALSE
...	unused

Value

invisible NULL

plot_agemodel_check *Plot an age-model check: the de-chirped maps, the rate corrections the cycles ask for and the flagged intervals*

Description

Plot an age-model check: the de-chirped maps, the rate corrections the cycles ask for and the flagged intervals

Usage

```
plot_agemodel_check(
  x,
  transforms = names(x$dechirped),
  view = c("tuned", "depth"),
  rate_unit = c(1, "depth units / kyr"),
  main = NULL,
  ...
)
```

Arguments

x	Result of check_agemodel.
transforms	Which de-chirped maps to show (default all that were computed).
view	"tuned" (maps in the time of the track, cycles should be horizontal) or "depth" (mapped back to depth) Default="tuned".
rate_unit	Factor and label of the rate axis, e.g. c(100, "cm/kyr").
main	Title.
...	Passed to plot_dechirped.

Value

Invisibly x.

---

plot_amtm_spectrum *Plot an adaptive multitaper spectrum with background and detections*

Description

The plot_amtm_spectrum function plots the output of amtm_spectrum: (top) the aMTM PSD with all fitted background models, the selected model and its gamma-test threshold; (middle) the harmonic F-test with its threshold; (bottom) the gamma statistic S/B with its threshold and the gamma bands. Detected periodic signals are marked with vertical lines. The frequency axis is in cycles per unit of the input depth/time column. Translated/adapted from the plotting code of the amtm_signaldetect tutorial notebook (Hector A. Salinas).

Usage

```
plot_amtm_spectrum(
  amtm = NULL,
  xlim = NULL,
  log_freq = FALSE,
  freq_lab = "Frequency (cycles/unit)",
  show_all_models = TRUE
)
```

Arguments

amtm	Output list of amtm_spectrum
xlim	Frequency limits Default=NULL (full range)
log_freq	Use a logarithmic frequency axis Default=FALSE
freq_lab	Label of the frequency axis Default="Frequency (cycles/unit)"
show_all_models	Draw all fitted background models Default=TRUE

Value

No return value; a plot is drawn.

Author(s)

Michiel Arts (R translation); original Python code: Hector A. Salinas

References

Di Matteo, S., Viall, N. M., & Kepko, L. (2021). Journal of Geophysical Research: Space Physics, 126, e2020JA028748. doi:10.1029/2020JA028748

Examples

```
set.seed(2)
n <- 512
t <- 0:(n - 1)
x <- as.numeric(stats::filter(rnorm(n), 0.9, method = "recursive")) +
  0.5 * sin(2 * pi * 0.1 * t) + 0.3 * sin(2 * pi * 0.32 * t)
res <- amtm_spectrum(data = cbind(t, x))
plot_amtm_spectrum(res)
```

---

plot_astro *Six-panel summary of an inversion*

Description

Six-panel summary of an inversion

Usage

```
plot_astro(
  fit,
  peaks = NULL,
  age_ma = NULL,
  file = NULL,
  width = 11,
  height = 7,
  res = 300
)
```

Arguments

fit	result of astro_invert()
peaks	optional measured spectral peaks (kyr) for the k scan
age_ma	age of the record, for the AstroGeo22 comparison

file	optional output file; PNG if it ends in .png, else PDF
width	Width of the output file.
height	Height of the output file.
res	Resolution (dpi) of png output.

---

plot_astro_anchor *Plot proxy record anchored to an astronomical solution*

Description

Plot the results of the anchoring the extracted signal to an astronomical solution using which was conducted using the astro_anchor

Usage

```
plot_astro_anchor(
  astro_solution = NULL,
  proxy_signal = NULL,
  anchor_points = NULL,
  time_dir = TRUE,
  keep_editable = FALSE
)
```

Arguments

astro_solution	Input is an astronomical solution with with the the proxy record was 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 which was 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.
anchor_points	Anchor points generated using the astro_anchor function
time_dir	The direction of the proxy record which was assumed during anchoring if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then time_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then time_dir should be set to FALSE time_dir=TRUE
keep_editable	Keep option to add extra features after plotting Default=FALSE

Value

The output is a set of 2 plots connected by lines The top plot is the proxy record with anchor points on top of it The bottom plot is the astronomical solution The lines connect the anchor points

Examples

```
# Use the grey_track example tracking points to anchor the grey scale data set
# of Zeeden et al., (2013) to the p-0.5t la2004 solution

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

#Use the pretracked grey_track curve which traced the precession cycle
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,pts=3)

#use the astrosignal_example to tune to which is an \cr
# ETP solution (p-0.5t la2004 solution).

astrosignal_example <- na.omit(astrosignal_example)
astrosignal_example[,2] <- -1*astrosignal_example[,2]
astrosignal <- as.data.frame(astrosignal_example)

#anchor the synthetic insolation curve extracted from the
# grey scale record to the insolation curve.
#use the anchor_points_grey data set to plot the
#result of using the astro_anchor function

#anchor_points_grey <- 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,
#pts=3,
#verbose=FALSE,
#genplot=FALSE # set verbose to TRUE to allow for anchoring using text feedback commands
#)


plot_astro_anchor(astro_solution = astrosignal,
proxy_signal = insolation_extract,
anchor_points = anchor_points_grey,
time_dir = TRUE, # the anchor points of this example get older with increasing depth
keep_editable = FALSE)
```

`plot_astro_evidence` *Map of the astronomical evidence against sedimentation rate and depth*

Description

Shows what `track_astro_hmm` tracks: for every sedimentation rate of a (wide) range and every position, the weighted sum of the log evidence of all astronomical cycles at the depth periods they would have at that rate. Where the record carries the cycles, the map shows a ridge at the true rate, and changes of the rate (a ramp, a jump, an oscillation) are visible as the shape of that ridge, with the aliases (rates at which other cycles fall on the ridges) as weaker parallel bands. The map is the tool to look at before choosing `sedrate` for the tracking: on CIP Case 3 (sedimentation rate oscillating between 3 and 6 cm/kyr with a 1.2-Myr period, very red noise) the posterior map of a Morlet scalogram with `omega_nr = 16` over a rate range of 1-20 cm/kyr shows a ridge varying between 3 and 6 cm/kyr, without assuming anything about the form of the variation, whereas the synchrosqueezed maps show speckle and long horizontal streaks (the slow wander of the red noise reassigned to a few periods) with a bias towards the edges of the range.

Usage

```
plot_astro_evidence(
  wavelet = NULL,
  sedrate = NULL,
  track = NULL,
  reference = NULL,
  rate_unit = c(1, "depth units / kyr"),
  palette = "viridis",
  main = NULL,
  type = c("evidence", "posterior"),
  ...
)
```

Arguments

`wavelet`	A scalogram (`analyse_wavelet` with a high `omega_nr` works best for noisy records; `analyse_superlet_sst`, `analyse_chirplet_sst` etc. for clean ones).
`sedrate`	Range of the sedimentation rate to show (depth units per kyr); wide by default.
`track`	Optional result of `track_astro_hmm` or `track_astro_auto` to draw on the map.
`reference`	Optional data frame (depth, sedimentation rate) to draw, e.g. a known or published rate.
`rate_unit`	Factor and label of the rate axis, e.g. `c(100, "cm/kyr")` for a record in metres `Default=c(1, "depth units / kyr")`.
`palette`	Colour palette name for `hcl.colors` `Default="viridis"`.

main	Title.
type	"evidence" (default) or "posterior": the posterior of the sedimentation rate of track (the attribute posterior of a track_astro_hmm result; its square root is shown). The posterior combines the evidence with the continuity of the rate and is the cleaner picture of a weak ridge.
...	Arguments of track_astro_hmm that shape the evidence (astro_periods, weights, age_ma, k_yr, smooth_depth, normalise, consistent_cycles, ...).

Value

Invisibly, a list with depth, sedrate and the matrix shown (rates in rows).

Examples

```
set.seed(1)
t <- seq(0, 2000, by = 0.5)
x_t <- sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 19) + 0.8 * sin(2 * pi * t / 41)
sr <- 2 + 0.7 * sin(2 * pi * t / 600)                  # oscillating sedimentation rate
zt <- c(0, cumsum(sr[-1] * diff(t)))
z <- seq(0, max(zt), by = 1)
x <- approx(zt, x_t, z)$y + rnorm(length(z), sd = 1.5)
w <- analyse_wavelet(data.frame(z, x), lowerPeriod = 10, upperPeriod = 500, dj = 1 / 50,
    omega_nr = 12)
plot_astro_evidence(w, sedrate = c(0.5, 8), reference = data.frame(z, approx(zt, sr, z)$y))
```

---

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 analyse_wavelet function.

Usage

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

Arguments

wavelet	Wavelet object created using the analyse_wavelet function.
y_lab	Label for the y-axis Default="Power".
x_lab	Label for the x-axis Default="depth (metres)".
keep_editable	Keep option to add extra features after plotting Default=FALSE

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
# the Total Solar Irradiance data set of Steinhilber et al., (2012)
TSI_wt <-
 analyse_wavelet(
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )

plot_avg_wavelet(wavelet=TSI_wt,
                 y_lab= "power",
                 x_lab="period (years)",
                 keep_editable=FALSE)


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



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

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

---

plot_Awavelet *Plots a adaptive wavelet scalogram*

Description

Plot adaptive wavelet scalogram using the outcome of the analyse_Awavelet function.

Usage

```
plot_Awavelet(
  Awavelet = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  useRaster = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  keep_editable = FALSE,
  dev_new = wr_dev_new_default(),
  plot_dir = TRUE,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  add_MTM_peaks = FALSE,
  add_data = TRUE,
  add_avg = FALSE,
  add_MTM = FALSE,
  mtm_siglvl = 0.95,
  demean_mtm = TRUE,
  detrend_mtm = TRUE,
  padfac_mtm = 5,
  tbw_mtm = 3,
  plot_horizontal = TRUE,
  colour_brewer = NULL
)
```

Arguments

Awavelet	wavelet object created using the analyse_Awavelet function.
lowerPeriod	Lowest period value which will be plotted
upperPeriod	Highest period value which will be plotted
n.levels	Number of color levels Default=100.
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function The R package 'scico' has the color palette options: "acton", "bam","bamako", "bamO", "batlow", "batlowK","batlowW","berlin","bilbao","broc","br ","buda","bukavu","cork","CorkO","davos","devon","fes","Glasgow","grayC", "hawaii","imola","lajolla", "nuuk","oleron","oslo","roma","romaO","Tofino","Tokyo","turku","Vanimo", "vik","vikO" The R package 'Viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo"
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps,scico and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
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)".
keep_editable	Keep option to add extra features after plotting Default=FALSE
dev_new	Open a new plotting window for the plot. The wavelet plots contain a raster image that needs more room than the small RStudio plot pane, which gives "figure margins too large" errors or clipped text, so a separate window is opened by default in the RStudio GUI. In every other front end (plain R, ESS, Positron, knitr, Rscript) the plot is drawn on the current device and no window is opened. Set TRUE or FALSE to force either behaviour, or set options(waverider.dev_new = TRUE/FALSE). Default: TRUE in the RStudio GUI, FALSE elsewhere.
plot_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE plot_dir=TRUE

add_lines	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL
add_points	Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values Default=NULL
add_abline_h	Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_abline_v	Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_MTM_peaks	Add the MTM peak periods as horizontal lines Default=FALSE
add_data	Plot the data on top of the wavelet Default=TRUE
add_avg	Plot the average wavelet spectral power to the side of the wavelet Default=FALSE
add_MTM	Add the MTM plot next to the wavelet plot Default=FALSE
mtm_siglvl	select the significance level (0-1) for the MTM spectrum Default=0.95
demean_mtm	Remove mean from data before conducting the MTM analysis Default=TRUE
detrend_mtm	Remove mean from data before conducting the MTM analysis Default=TRUE
padfac_mtm	Pad factor for the MTM analysis Default=5
tbw_mtm	time bandwidth product of the MTM analysis Default=3
plot_horizontal	plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

The output is a plot of a cross wavelet scalogram.

Author(s)

plotting code based on the "wt.image" and "analyze.coherency" functions of the 'WaveletComp' R package

References

Roesch, A., & Schmidbauer, H. (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., & Mureșan, R. C. (2021). Time-frequency super-resolution with superlets. Nature Communications, 12(1), 337. doi:10.1038/s41467020205399

Examples

```
#Example 1. A plot of a wavelet spectra using the Total Solar Irradiance
# data set of Steinhilber et al., (2012)

TSI_wt <-
 analyse_Awavelet(
   data = TSI,
```

```
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
omega_min = 6,
omega_max = 12,
scaling = "log2",
alpha = 1
 )

plot_Awavelet(
 Awavelet = TSI_wt,
 lowerPeriod = 16,
 upperPeriod = 8192,
 n.levels = 100,
 palette_name = "rainbow",
color_brewer= "grDevices",
 useRaster = TRUE,
 periodlab = "Period (metres)",
 x_lab = "depth (metres)",
 keep_editable = FALSE,
 dev_new=TRUE,
 plot_dir = TRUE,
 add_lines = NULL,
 add_points= NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 add_MTM_peaks = FALSE,
 add_data = TRUE,
 add_avg = TRUE,
 add_MTM = FALSE,
 mtm_siglvl = 0.95,
 demean_mtm = TRUE,
 detrend_mtm = TRUE,
 padfac_mtm = 5,
 tbw_mtm = 3,
 plot_horizontal=TRUE)

#Example 2. A plot of a wavelet spectra using the magnetic susceptibility
#data set of Pas et al., (2018)
mag_wt <-
analyse_Awavelet(
data = mag,
dj = 1/100,
lowerPeriod = 1,
upperPeriod = 254,
verbose = FALSE,
omega_min = 6,
omega_max = 12,
scaling = "log2",
alpha = 1
)
```

```
plot_Awavelet(
Awavelet = mag_wt,
lowerPeriod = 1,
upperPeriod = 254,
n.levels = 100,
palette_name = "rainbow",
color_brewer= "grDevices",
useRaster = TRUE,
periodlab = "Period (metres)",
x_lab = "depth (metres)",
keep_editable = FALSE,
dev_new=TRUE,
plot_dir = TRUE,
add_lines= NULL,
add_points= NULL,
add_abline_h = NULL,
add_abline_v = NULL,
add_MTM_peaks = FALSE,
add_data = TRUE,
add_avg = TRUE,
add_MTM = FALSE,
mtm_siglvl = 0.95,
demean_mtm = TRUE,
detrend_mtm = TRUE,
padfac_mtm = 5,
tbw_mtm = 3,
plot_horizontal=TRUE)


#Example 3. A plot of a wavelet spectra using the greyscale
# data set of Zeeden et al., (2013)
grey_wt <-
 analyse_Awavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 2,
   upperPeriod = 25,
   verbose = FALSE,
omega_min = 6,
omega_max = 12,
scaling = "log2",
alpha = 1)

plot_Awavelet(
Awavelet = grey_wt,
lowerPeriod = 2,
upperPeriod = 25,
n.levels = 100,
palette_name = "rainbow",
color_brewer= "grDevices",
useRaster = TRUE,
periodlab = "Period (metres)",
x_lab = "depth (metres)",
```

```
keep_editable = FALSE,
dev_new=TRUE,
plot_dir = TRUE,
add_lines = NULL,
add_points= NULL,
add_abline_h = NULL,
add_abline_v = NULL,
add_MTM_peaks = FALSE,
add_data = TRUE,
add_avg = TRUE,
add_MTM = FALSE,
mtm_siglvl = 0.95,
demean_mtm = TRUE,
detrend_mtm = TRUE,
padfac_mtm = 5,
tbw_mtm = 3,
plot_horizontal=TRUE)
```

`plot_bayes_changepoint` *Plot the result of bayes_changepoint*

Description

Plots the data, the posterior-mean regression model with its pointwise credible interval, and the posterior probability of a change point (right axis), as in the Acycle change point figure (bayes_cpt.m), together with the posterior distribution of the number of change points.

Usage

```
plot_bayes_changepoint(
  bcp = NULL,
  x_lab = "depth",
  y_lab = "proxy",
  plot_k = TRUE
)
```

Arguments

bcp	Output of bayes_changepoint.
x_lab	Label of the x-axis Default="depth"
y_lab	Label of the proxy axis Default="proxy"
plot_k	Add a panel with the posterior distribution of the number of change points Default=TRUE

Value

No return value, called for its side effect (a base graphics plot).

Author(s)

Michiel Arts (R translation); original MATLAB code: Eric Ruggieri, Mingsong Li.

References

Ruggieri, E. (2013). A Bayesian approach to detecting change points in climatic records. International Journal of Climatology, 33, 520-528. <doi:10.1002/joc.3447>

Examples

```
set.seed(2)
y <- c(rep(0, 60), rep(1.5, 60)) + stats::rnorm(120, sd = 0.5)
bcp <- bayes_changepoint(data = cbind(1:120, y), k_max = 4, d_min = 5,
                         v_0 = 3, n_sim = 200, seed = 1)
plot_bayes_changepoint(bcp)
```

---

plot_bispectral *Plot the result of a bispectral analysis*

Description

The plot_bispectral function plots the result of bispectral_analysis. The "overview" layout follows Acycle: two aligned 2pi multitaper power spectra (NW = 2, 3 tapers, NFFT = 5N) above maps of the bispectrum magnitude (log10|B|) and the squared bicoherence, with the strongest fraction of each map retained, the significance boundary (if inference was calculated) drawn in black, the strongest (significant) bicoherence peaks annotated and period axes on top. Single maps of "bicoherence", "magnitude", "biphase", "real" or "imaginary" can also be drawn. Rendering is simplified with respect to Acycle (no interpolated display mesh or alpha ramp). Translated from Acycle v3.1 bispectralPlot.m (Mingsong Li).

Usage

```
plot_bispectral(
  result = NULL,
  quantity = "overview",
  keep_bispectrum = 0.5,
  keep_bicoherence = 0.5,
  color_grid = 32,
  freq_min = 0,
  freq_max = NULL,
  reference_periods = NULL,
  frequency_pairs = NULL,
```

```
  peak_count = 5,
  show_period_axes = TRUE,
  show_significance = TRUE,
  title = NULL,
  colour_grid = NULL
)
```

Arguments

result	Output of bispectral_analysis.
quantity	"overview", "bicoherence", "magnitude", "biphase", "real" or "imaginary". Default="overview"
keep_bispectrum	Fraction (0,1] of strongest |B| values shown. Default=0.5
keep_bicoherence	Fraction (0,1] of strongest b^2 values shown. Default=0.5
color_grid	Number of colours (4-256). Default=32
freq_min	Lower plotted frequency. Default=0
freq_max	Upper plotted frequency; NULL shows the full computed axis. Default=NULL
reference_periods	Periods drawn as guide lines f1 + f2 = 1/period. Default=NULL
frequency_pairs	Matrix [f1 f2] of coordinates marked with guides. Default=NULL
peak_count	Number of bicoherence peaks annotated. Default=5
show_period_axes	Add a period axis on top of the maps. Default=TRUE
show_significance	Draw the significance boundary. Default=TRUE
title	Figure title. Default=NULL
colour_grid	British spelling of color_grid; if given it overrides color_grid.

Value

Invisibly returns NULL; the plot is drawn on the current device.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li (Acycle bispectral module).

References

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22.

Examples

```
set.seed(4)
t <- 0:1023
y <- cos(2 * pi * 0.1 * t) + cos(2 * pi * 0.17 * t) +
  0.5 * cos(2 * pi * 0.27 * t) + 0.3 * rnorm(1024)
res <- bispectral_analysis(data = cbind(t, y), max_freq_bins = 128)
plot_bispectral(res)
plot_bispectral(res, quantity = "biphase")
```

`plot_chirp_eha` *Plot a chirp-matched EHA map, with or without the surrogate significance*

Description

Two styles of "enhanced spectrum" for a `analyse_chirp_eha` result: `"significance"` shows the line-to-continuum ratio divided by the 95th percentile of the warped surrogates (`Power_sig`, from `chirp_eha_significance`; above 1, i.e. above 0 on the log scale, is significant), `"outline"` shows the de-chirped map itself (`Power_norm`) with the significant cells outlined, and `"plain"` the de-chirped map alone.

Usage

```
plot_chirp_eha(
  ce,
  style = c("outline", "significance", "plain"),
  track = NULL,
  show = NULL,
  level = 0.95,
  palette = "viridis",
  main = NULL,
  xlab = "Depth",
  ylab = "Period",
  scale = "auto"
)
```

Arguments

`ce`	Result of `analyse_chirp_eha` (after `chirp_eha_significance` for the first two styles).
`style`	`"outline"` (default), `"significance"` or `"plain"`.
`track`	Optional track whose periods are drawn (`show` cycles in kyr at the tracked rate).
`show`	Astronomical periods (kyr) to draw with `track`.
`level`	Significance level for the outline `Default=0.95`.
`palette`	`hcl.colors` palette `Default="viridis"`.

main	Title.
xlab, ylab	Axis labels.
scale	Colour scale as in all image plots of the package (see plot_wavelet) Default="auto".

Value

Invisibly, the matrix shown.

---

plot_cvcoco *Plot a cvCOCO result*

Description

Plots the diagnostics of a cvcoco result in the style of Acycle's plotcvcoco.m, using base graphics. Four pages can be drawn: (1) the two regularized folds and the held-out validation spectra with their frozen targets, (2) the stacked result panels: directional and consensus correlation, global p, local p and the number of resolvable astronomical periods, (3) the consensus pCOCO curve and (4) the Monte Carlo audit (null distributions of the consensus and directional maxima). Translated from Acycle v3.1 plotcvcoco.m (Mingsong Li).

Usage

```
plot_cvcoco(result = NULL, pages = c("spectra", "result", "pcoco", "audit"))
```

Arguments

result	Output of cvcoco.
pages	Pages to draw, any of "spectra", "result", "pcoco", "audit". Default=c("spectra", "result", "pcoco", "audit")

Value

Invisibly returns NULL; called for its plots.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li.

References

Li, M., Kump, L.R., Hinnov, L.A., Mann, M.E., 2018. Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing. Earth and Planetary Science Letters 501, 165-179.

Examples

```
set.seed(3)
t_kyr <- seq(0, 1600, by = 2)
etp <- sin(2 * pi * t_kyr / 405) + 0.6 * sin(2 * pi * t_kyr / 125) +
  0.8 * sin(2 * pi * t_kyr / 41) + 0.5 * sin(2 * pi * t_kyr / 23.7)
dat <- cbind(t_kyr * 5 / 100, etp + 0.5 * stats::rnorm(length(t_kyr)))
res <- cvcoco(data = dat, sr1 = 2, sr2 = 10, srstep = 0.5, n_sim = 20,
  seed = 1)
plot_cvcoco(res, pages = "result")
```

---

plot_dechirped *Plot a de-chirped scalogram*

Description

Plot a de-chirped scalogram

Usage

```
plot_dechirped(
  x,
  style = c("outline", "significance", "plain", "tuned"),
  show = NULL,
  level = 0.95,
  palette = "viridis",
  main = NULL,
  xlab = "Depth",
  ylab = "Period",
  scale = "auto",
  outline_lwd = 0.6
)
```

Arguments

x	Result of analyse_dechirped.
style	"outline" (de-chirped map with the significant cells outlined, when surrogates were run), "significance" (ratio to the 95th percentile of the warped surrogates), "plain" (the de-chirped map) or "tuned" (the transform in time, periods in kyr).
show	Astronomical periods (kyr) to draw at the tracked rate.
level	Significance level of the outline Default=0.95.
palette	hcl.colors palette Default="viridis".
main	Title.
xlab, ylab	Axis labels.

scale	Colour scale as in all image plots of the package ("auto", "log", "quantile", "sqrt", "linear"; see plot_wavelet) Default="auto". The significance style always uses a fixed diverging scale centred on the 95th percentile of the surrogates.
outline_lwd	Line width of the significance outline Default=0.6.

Value

Invisibly, the matrix shown.

---

plot_dynamic_filter *Plot the result of a dynamic filter*

Description

The plot_dynamic_filter function plots the output of dynamic_filter: the (demeaned) data with the filtered signal, and the depth-varying pass band (window lower, upper and centre frequencies or periods, with the control points), reproducing the overlay Acycle draws on its evolutionary spectrum (dynamic_filter_lang.m). Written for the R translation of Acycle v3.1 dynamic filtering (Mingsong Li).

Usage

```
plot_dynamic_filter(
  data = NULL,
  dyn_filter = NULL,
  period = FALSE,
  x_lab = "depth",
  y_lab = "proxy"
)
```

Arguments

data	Input data set used in dynamic_filter (2-column matrix [depth, value]).
dyn_filter	Output list of dynamic_filter.
period	Plot the pass band as period instead of frequency Default=FALSE.
x_lab	label for the x-axis Default="depth".
y_lab	label for the y-axis of the data panel Default="proxy".

Value

No return value; a plot is produced.

Author(s)

Michiel Arts (R translation); original MATLAB code: Nicolas Thibault, Giovanni Rizzi and Mingsong Li (Acycle).

References

Li, M., Hinnov, L., Kump, L., 2019. Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences 127, 12-22. <doi:10.1016/j.cageo.2019.02.011>

Examples

```
depth <- seq(0, 300, by = 0.25)
period <- 5 + 10 * depth / 300
proxy <- sin(2 * pi * cumsum(1 / period) * 0.25)
ctrl <- dynamic_filter_controls(tracked = cbind(depth, period),
  rel_bandwidth = 0.3, n_points = 30)
res <- dynamic_filter(data = cbind(depth, proxy), controls = ctrl,
  window_length = 40, step_length = 1)
plot_dynamic_filter(data = cbind(depth, proxy), dyn_filter = res)
```

---

plot_dynot *Plot DYNOT results*

Description

The plot_dynot function plots the input data and the DYNOT noise model (median and confidence envelopes, reversed y-axis) as in the Acycle DYNOT window. Translated from the plotting part of Acycle v3.1 DYNOS.m (Mingsong Li).

Usage

```
plot_dynot(dynot_res = NULL, plot_data = TRUE, x_lab = "time (kyr)")
```

Arguments

dynot_res	Output of dynot. Default=NULL
plot_data	Also plot the input data in an upper panel. Default=TRUE
x_lab	Label of the x-axis. Default="time (kyr)"

Value

No return value; a plot is drawn.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li (Acycle).

References

Li, M., Hinnov, L.A., Huang, C., Ogg, J.G., 2018. Sedimentary noise and sea levels linked to land-ocean water exchange and obliquity forcing. Nature Communications 9, 1004. doi:10.1038/s4146701803454y

Examples

```
set.seed(2)
t <- cumsum(runif(900, 1, 3))
x <- sin(2 * pi * t / 405) + sin(2 * pi * t / 41) + rnorm(900)
res <- dynot(data = cbind(t, x), n_sim = 20, n_out = 200, seed = 1)
plot_dynot(res)
```

---

plot_ecoco_cv *Plot Blocked / Interleaved eCOCO maps*

Description

Plots the five eCOCO maps used by Acycle for Blocked and Interleaved eCOCO: consensus correlation, consensus SR-global p and same-rate local p (both as -log10(p) with their own 1-2-5 display range, limited by the Monte Carlo resolution 1/(n_sim+1), with the 0.05 and 0.01 contours), the number of contributing orbital periods, and the window-normalised ridge score with the tracked ridge (filled squares: global p <= 0.05; filled circles: local p <= 0.01; open circles: neither). Also accepts cocoR `ecoco_adaptive` results. Translated from Acycle v3.1 ecocoplot.m (Mingsong Li); base graphics only.

Usage

```
plot_ecoco_cv(
  ecoco = NULL,
  panels = 1:5,
  reverse_depth = FALSE,
  palette = grDevices::hcl.colors(64, "viridis")
)
```

Arguments

`ecoco`	Result of `ecoco_blocked`, `ecoco_interleaved` or `ecoco_adaptive`. `Default=NULL`
`panels`	Which panels to draw (1 rho, 2 global p, 3 local p, 4 orbit count, 5 ridge score). `Default=1:5`
`reverse_depth`	Plot depth increasing downwards. `Default=FALSE`
`palette`	Colour vector for the maps. `Default=grDevices::hcl.colors(64, "viridis")`

Value

Invisibly, the list of plotted matrices (rho, -log10 p maps, orbit count, normalised score) and the ridge.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li

References

Li, M., Kump, L.R., Hinnov, L.A., Mann, M.E., 2018. Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing. Earth and Planetary Science Letters 501, 165-179.

Examples

```
set.seed(2)
dt <- 0.05
depth <- seq(0, 30, by = dt)
tt <- depth * 100 / 5
x <- sin(2 * pi * tt / 405.7) + 0.8 * sin(2 * pi * tt / 98.9) +
  0.5 * sin(2 * pi * tt / 40.99) + 0.4 * sin(2 * pi * tt / 23.68) +
  0.3 * rnorm(length(tt))
res <- ecoco_blocked(cbind(depth, x), window = 10, sr = seq(3, 8, 0.5),
                     step_depth = 1, pad = 400, n_sim = 20)
plot_ecoco_cv(res)
```

---

plot_eha_log2 *Plots an log2 spaced EHA spectra*

Description

Plot wavelet scalogram using the outcome of the eha_log2 function.

Usage

```
plot_eha_log2(
  eha_log2 = NULL,
  plot_opt = "Amplitude",
  lowerPeriod = NULL,
  upperPeriod = NULL,
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  useRaster = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  keep_editable = FALSE,
  dev_new = wr_dev_new_default(),
  plot_dir = TRUE,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  add_MTM_peaks = FALSE,
```

```
  add_data = TRUE,
  add_avg = FALSE,
  pval_abline = c(0.1, 0.05),
  pval_cutoff = c(0.1),
  add_MTM = FALSE,
  mtm_siglvl = 0.95,
  demean_mtm = TRUE,
  detrend_mtm = TRUE,
  padfac_mtm = 5,
  tbw_mtm = 3,
  plot_horizontal = TRUE,
  colour_brewer = NULL,
  scale = "auto"
)
```

Arguments

eha_log2	eha_log2 object created using the eha_log2 function.
plot_opt	plot options are "Power", "Amplitude", "Probability" or "F_test"
lowerPeriod	Lowest period value which will be plotted
upperPeriod	Highest period value which will be plotted
n.levels	Number of color levels Default=100.
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function The R package 'scico' has the color palette options: "acton", "bam","bamako", "bamO", "batlow", "batlowK","batlowW","berlin","bilbao","broc","br ","buda","bukavu","cork","CorkO","davos","devon","fes","Glasgow","grayC", "hawaii","imola","lajolla", ","nuuk","oleron","oslo","roma","romaO","Tofino","Tokyo","turku","Vanimo", "vik","vikO" The R package 'Viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo"
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps,scico and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices

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)".
keep_editable	Keep option to add extra features after plotting Default=FALSE
dev_new	Open a new plotting window for the plot. The wavelet plots contain a raster image that needs more room than the small RStudio plot pane, which gives "figure margins too large" errors or clipped text, so a separate window is opened by default in the RStudio GUI. In every other front end (plain R, ESS, Positron, knitr, Rscript) the plot is drawn on the current device and no window is opened. Set TRUE or FALSE to force either behaviour, or set options(waverider.dev_new = TRUE/FALSE). Default: TRUE in the RStudio GUI, FALSE elsewhere.
plot_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE plot_dir=TRUE
add_lines	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL
add_points	Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values Default=NULL
add_abline_h	Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_abline_v	Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_MTM_peaks	Add the MTM peak periods as horizontal lines Default=FALSE
add_data	Plot the data on top of the wavelet Default=TRUE
add_avg	Plot the average wavelet spectral power to the side of the wavelet Default=FALSE
pval_abline	p value of the abline
pval_cutoff	p value cutoff below which no data is displayed
add_MTM	Add the MTM plot next to the wavelet plot Default=FALSE
mtm_siglvl	select the significance level (0-1) for the MTM spectrum Default=0.95
demean_mtm	Remove mean from data before conducting the MTM analysis Default=TRUE
detrend_mtm	Remove mean from data before conducting the MTM analysis Default=TRUE
padfac_mtm	Pad factor for the MTM analysis Default=5
tbw_mtm	time bandwidth product of the MTM analysis Default=3
plot_horizontal	plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.
scale	Colour scale as in plot_wavelet Default="auto".

Value

The output is a plot of a EHA spectra. if add_MTM_peaks = TRUE then the output of the MTM analysis will given as matrix

Author(s)

Code based on the wt.image" functions of the 'WaveletComp' R package The EHA and MTM analysis parts are from the astrochron R package of Meyers et al., (2012)

References

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

S.R. Meyers, 2012, Seeing Red in Cyclic Stratigraphy: Spectral Noise Estimation for Astrochronology: Paleoceanography, 27, PA3228, <doi:10.1029/2012PA002307>

S.R. Meyers, 2019 Cyclostratigraphy and the problem of astrochronologic testing, Earth-Science Reviews, Volume 190, <doi.org/10.1016/j.earscirev.2018.11.015.>

Examples

```
#Example 1. A plot of a wavelet spectra using the Total Solar Irradiance
# data set of Steinhilber et al., (2012)

TSI_eha_log2 <- eha_log2(data = TSI,
win =  8192,
tbw = 4,
demean = TRUE,
detrend = TRUE,
upperPeriod = 8192,
lowerPeriod = 50,
pad = NULL,
padding = "noise")

plot_eha_log2(
eha_log2 = TSI_eha_log2,
plot_opt = "Amplitude",
lowerPeriod = 50,
upperPeriod = 8192,
n.levels = 100,
palette_name = "rainbow",
color_brewer = "grDevices",
useRaster = TRUE,
periodlab = "Period (metres)",
x_lab = "depth (metres)",
keep_editable = FALSE,
dev_new = TRUE,
plot_dir = TRUE,
add_lines = NULL,
add_points = NULL,
add_abline_h = NULL,
```

```
add_abline_v = NULL,
add_MTM_peaks = FALSE,
add_data = TRUE,
add_avg = FALSE,
add_MTM = FALSE,
mtm_siglvl = 0.95,
demean_mtm = TRUE,
detrend_mtm = TRUE,
padfac_mtm = 5,
tbw_mtm = 3,
plot_horizontal = TRUE)

#Example 2. A plot of a wavelet spectra using the magnetic susceptibility
#data set of Pas et al., (2018)
mag_eha_log2 <- eha_log2(data = mag,
win =  50,
tbw = 4,
demean = TRUE,
detrend = TRUE,
upperPeriod = 50,
lowerPeriod = 1,
pad = NULL,
padding = "noise")

plot_eha_log2(
eha_log2 = mag_eha_log2,
plot_opt = "Amplitude",
lowerPeriod = 1,
upperPeriod = 50,
n.levels = 100,
palette_name = "rainbow",
color_brewer = "grDevices",
useRaster = TRUE,
periodlab = "Period (metres)",
x_lab = "depth (metres)",
keep_editable = FALSE,
dev_new = TRUE,
plot_dir = TRUE,
add_lines = NULL,
add_points = NULL,
add_abline_h = NULL,
add_abline_v = NULL,
add_MTM_peaks = FALSE,
add_data = TRUE,
add_avg = FALSE,
add_MTM = FALSE,
mtm_siglvl = 0.95,
demean_mtm = TRUE,
detrend_mtm = TRUE,
padfac_mtm = 5,
tbw_mtm = 3,
plot_horizontal = TRUE)
```

```
#Example 3. A plot of a wavelet spectra using the greyscale
# data set of Zeeden et al., (2013)
grey_eha_log2 <- eha_log2(data = grey,
win =  20,
tbw = 4,
demean = TRUE,
detrend = TRUE,
upperPeriod = 20,
lowerPeriod = 1,
pad = NULL,
padding = "noise")


plot_eha_log2(
eha_log2 = grey_eha_log2,
plot_opt = "Amplitude",
lowerPeriod = 1,
upperPeriod = 20,
n.levels = 100,
palette_name = "rainbow",
color_brewer = "grDevices",
useRaster = TRUE,
periodlab = "Period (metres)",
x_lab = "depth (metres)",
keep_editable = FALSE,
dev_new = TRUE,
plot_dir = TRUE,
add_lines = NULL,
add_points = NULL,
add_abline_h = NULL,
add_abline_v = NULL,
add_MTM_peaks = FALSE,
add_data = TRUE,
add_avg = FALSE,
add_MTM = FALSE,
mtm_siglvl = 0.95,
demean_mtm = TRUE,
detrend_mtm = TRUE,
padfac_mtm = 5,
tbw_mtm = 3,
plot_horizontal = TRUE)
```

`plot_mesa_evolutive` *Plot an evolutive MESA*

Description

Plots the spectral image of an evolutive maximum entropy spectrum (`mesa_evolutive`): one panel for every window length, side by side, with the position (depth or time) on one axis and the period

or frequency on the other, as in the spectral image columns of log-analysis software. The spectrum of every window position is divided by its maximum, so that the ridges of the spectral peaks show along the whole record. The peaks of every position, the tracked ridge (track_mesa) and the average amplitude spectrum (mesa_average) can be added.

Usage

```
plot_mesa_evolutive(
  x,
  vertical = TRUE,
  xaxis = c("period", "frequency"),
  period_lim = NULL,
  frequency_lim = NULL,
  log_scale = FALSE,
  peaks = FALSE,
  peak_min = 0.3,
  track = NULL,
  average = FALSE,
  colors = NULL,
  position_lim = NULL,
  main = NULL,
  log_axis = TRUE,
  scale = "auto",
  palette = "viridis",
  normalise = c("window", "period", "none"),
  ...
)
```

Arguments

x	output of mesa_evolutive.
vertical	If TRUE (default) the position is the vertical axis, increasing downwards (as for a depth log), and the windows are panels side by side. If FALSE the position is horizontal and the windows are stacked.
xaxis	"period" (default) or "frequency": the spectral axis.
period_lim	Limits of the period axis (default: from twice the sampling interval to the longest window).
frequency_lim	Limits of the frequency axis, used when xaxis = "frequency".
log_scale	Kept for compatibility: TRUE is scale = "log".
peaks	If TRUE the peaks of every position are drawn as white points.
peak_min	Smallest relative power (0 to 1) of a peak that is drawn (default 0.3).
track	A mesa_track from track_mesa or a list of them, drawn as lines.
average	If TRUE the average amplitude spectrum of every window is drawn next to its image, with the period of its strongest peaks marked.
colors	Optional vector of colours to interpolate (overrides palette).
position_lim	Limits of the position axis.

main	Overall title.
log_axis	Period or frequency axis on a log2 scale Default=TRUE: the spectrum of a cyclostratigraphic record spans several octaves, and on a linear axis the long periods take most of the plot.
scale	Colour scale as in all image plots of the package (see plot_wavelet) Default="auto"; a maximum-entropy spectrum spans many decades, so "auto" is logarithmic for it.
palette	hcl.colors palette Default="viridis".
normalise	"window" (each analysis window relative to its maximum, default), "period" (each period relative to its mean along the record, which removes the red-noise slope and shows where a cycle is stronger or weaker) or "none".
...	currently unused.

Value

Called for its plot; invisibly returns x.

Examples

```
set.seed(2)
t <- seq(0, 800, by = 2)
y <- sin(2 * pi * cumsum(1 / ifelse(t < 400, 40, 80)) * 2) + rnorm(length(t), sd = 0.4)
ev <- mesa_evolutive(data.frame(t, y), window = c(150, 250), order = 20)
plot_mesa_evolutive(ev, average = TRUE)
```

---

plot_mesa_spectrum *Plot a maximum entropy (MESA) spectrum*

Description

Plots the power spectrum of mesa_spectrum on a logarithmic power axis, against period or frequency, with the strongest peaks labelled.

Usage

```
plot_mesa_spectrum(
  x,
  xaxis = c("period", "frequency"),
  peaks = TRUE,
  n_label = 5,
  xlim = NULL,
  log_x = FALSE,
  col = "black",
  main = NULL,
  ...
)
```

Arguments

x	output of mesa_spectrum.
xaxis	"period" (default) or "frequency".
peaks	If TRUE (default) the peaks of x$peaks are marked and labelled with their period.
n_label	Number of strongest peaks to label (default 5).
xlim	Optional limits of the x axis (in the units of xaxis).
log_x	If TRUE the x axis is logarithmic (default FALSE).
col	Line colour.
main	Title. The default describes the order and criterion.
...	further arguments passed to plot.

Value

Called for its plot; invisibly returns x.

Examples

```
set.seed(1)
t <- seq(0, 600, by = 2)
y <- sin(2 * pi * t / 100) + rnorm(length(t), sd = 0.5)
plot_mesa_spectrum(mesa_spectrum(data.frame(t, y)))
```

---

plot_sed_model *Plot sedimentation modelling results*

Description

The plot_sed_model function is used plot/re-plot the results from the flmw and sum_power_sedrate functions

Usage

```
plot_sed_model(
  model_results = NULL,
  plot_res = 1,
  x_lab = "depth (m)",
  y_lab = "sed rate cm/kyr",
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  colour_brewer = NULL
)
```

Arguments

model_results	Wavelet object created using the analyse_wavelet function
plot_res	Numbers to be used as input form the flmwoutput options 1-8 option 1: slope coefficient, option 2: r squared, option 3: nr of components, option 4: difference to origin, option 5: slope coefficient percentile option 6: r squared percentile, option 7: nr of components percentile, option 8: difference to origin percentile. If the output of the sum_power_sedrate function is used then input should be option 1: sum max power option 2: nr of components
x_lab	Label for x-axis Default="depth (m)"
y_lab	Label for y-axis Default=""sed rate cm/kyr""
keep_editable	Keep option to add extra features after plotting Default=FALSE
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

Returns a plot of sedimentation rates vs depth and a value which was generated using the flmw or sum_power_sedrate functions

Examples

```
#estimate sedimentation rate for the the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018).

mag_wt <- analyse_wavelet(data = mag[1:200, ],
dj = 1/10,
lowerPeriod = 0.1,
upperPeriod = 32,
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 = 3, # increase for a better constrained result
run_multicore=FALSE,
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 = 1,
cycles = c(2376,1600,1180,696,406,110),
x_lab="depth",
y_lab="sedrate",
run_multicore=FALSE,
genplot = FALSE,
palette_name = "rainbow",
color_brewer= "grDevices",
verbose=FALSE)

plot_sed_model(model_results=sedrates,
plot_res=1,
x_lab = "depth (m)",
y_lab = "sed rate cm/kyr",
keep_editable=FALSE,
palette_name = "rainbow",
color_brewer= "grDevices")
```

`plot_spectral_moments` *Plot the result of spectral_moments*

Description

Plots the mean frequency and bandwidth (and, if available, the bandwidth trend and the sedimentation-rate estimate) returned by `spectral_moments`, as in the Acycle spectral moments GUI. Translated from the plotting part of Acycle v3.1 SpectralMomentsGUI.m (Mingsong Li).

Usage

```
plot_spectral_moments(
  sm = NULL,
  x_lab = "depth",
  y_lab = "frequency (cycles/unit)",
  sr_lab = "sed. rate",
```

```
  image = TRUE,
  yaxis = c("frequency", "period"),
  scale = c("log", "quantile", "linear"),
  palette = "viridis",
  freq_max = NULL
)
```

Arguments

sm	Output of spectral_moments.
x_lab	Label of the x-axis Default="depth"
y_lab	Label of the frequency axis Default="frequency (cycles/unit)"
sr_lab	Label of the sedimentation-rate axis Default="sed. rate"
image	Draw the periodograms of all windows as an image (frequency or period against depth, the evolutive spectrum the moments are computed from) above the moments Default=TRUE.
yaxis	"frequency" or "period" (log2 axis) for the image Default="frequency".
scale	Colour scale of the image: "log" (log10 power from the 2nd percentile), "quantile" (histogram-equalised) or "linear" Default="log".
palette	hcl.colors palette Default="viridis".
freq_max	Upper frequency shown in the image Default=NULL (all).

Value

No return value, called for its side effect (a base graphics plot).

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li.

References

Sinnesael, M., Zivanovic, M., De Vleeschouwer, D., Claeys, P. (2018). Spectral moments in cyclostratigraphy: advantages and disadvantages compared to more classic approaches. Paleoceanography and Paleoclimatology, 33, 493-510. <doi:10.1029/2017PA003293>

Examples

```
depth <- seq(0, 20, by = 0.01)
proxy <- sin(2 * pi * depth / (0.4 + 0.02 * depth))
sm <- spectral_moments(data = cbind(depth, proxy), window = 4, step = 0.1)
plot_spectral_moments(sm)
```

plot_superlet *Plots a superlet scalogram*

Description

Plot superlet scalogram using the outcome of the analyse_superlet function.

Usage

```
plot_superlet(
  superlet = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  useRaster = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  keep_editable = FALSE,
  dev_new = wr_dev_new_default(),
  plot_dir = TRUE,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  add_data = TRUE,
  add_avg = FALSE,
  plot_horizontal = TRUE,
  add_signif = NULL,
  colour_brewer = NULL,
  scale = "auto"
)
```

Arguments

superlet	superlet object created using the analyse_superlet function.
lowerPeriod	Lowest period value which will be plotted
upperPeriod	Highest period value which will be plotted
n.levels	Number of color levels Default=100.
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette

	options: “magma”, “plasma”, “inferno”, “viridis”, “mako”, and “rocket” and “turbo” To see the color palette options of the The R pacakge ’RColorBrewer’ run the wr_brewer_palettes() function The R package ’colorRamps’ has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package ’grDevices’ has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge ’grDevices’ run the grDevices::hcl.pals() function The R package ’scico’ has the color palette options: “acton”, “bam”,“bamako”, “bamO”, “batlow”, “batlowK”,“batlowW”,“berlin”,“bilbao”,”broc”,”br ”buda”,”bukavu”,”cork”,”CorkO”,”davos”,”devon”,”fes”,”Glasgow”,”grayC”, “hawaii”,”imola”,”lajolla”,” ”nuuk”,”oleron”,”oslo”,”roma”,”romaO”,”Tofino”,”Tokyo”,”turku”,”Vanimo”, ”vik”,”vikO” The R package ’Viridis’ has the color palette options: “magma”, “plasma”, “inferno”, “viridis”, “mako”, and “rocket” and “turbo”
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps,scico and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
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)".
keep_editable	Keep option to add extra features after plotting Default=FALSE
dev_new	Open a new plotting window for the plot. The wavelet plots contain a raster image that needs more room than the small RStudio plot pane, which gives "figure margins too large" errors or clipped text, so a separate window is opened by default in the RStudio GUI. In every other front end (plain R, ESS, Positron, knitr, Rscript) the plot is drawn on the current device and no window is opened. Set TRUE or FALSE to force either behaviour, or set options(waverider.dev_new = TRUE/FALSE). Default: TRUE in the RStudio GUI, FALSE elsewhere.
plot_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE plot_dir=TRUE
add_lines	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL
add_points	Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values Default=NULL
add_abline_h	Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_abline_v	Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_data	Plot the data on top of the wavelet Default=TRUE
add_avg	Plot the average wavelet spectral power to the side of the wavelet Default=FALSE

plot_horizontal
: plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE

add_signif
: Optional significance object created with wavelet_significance. The significant cells are outlined on the scalogram and the critical level of the global spectrum is drawn as a red dashed line on the average power panel Default=NULL.

colour_brewer
: British spelling of color_brewer; if given it overrides color_brewer.

scale
: Colour scale as in plot_wavelet Default="auto".

Value

The output is a plot of a superlet scalogram.

Author(s)

Code based on the "wt.image" functions of the 'WaveletComp' R package Whereas the "analyse_superlet" that generates the input for the plotting function is based on the matlab code in Moca et al. (2021)

References

Roesch, A., & Schmidbauer, H. (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., & Mureșan, R. C. (2021). Time-frequency superresolution with superlets. Nature Communications, 12(1), 337. doi:10.1038/s41467020205399

Examples

```
#Example 1. A plot of a wavelet spectra using the Total Solar Irradiance
# data set of Steinhilber et al., (2012)

TSI_super <-
 analyse_superlet(
   data = TSI,
   Nf = 128,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
   c1=1,
   o = c(1,5),
   mult = TRUE)

plot_superlet(
 superlet = TSI_super,
 lowerPeriod = 16,
 upperPeriod = 8192,
 n.levels = 100,
 palette_name = "rainbow",
color_brewer= "grDevices",
 useRaster = TRUE,
 periodlab = "Period (metres)",
```

```
 x_lab = "depth (metres)",
 keep_editable = FALSE,
 dev_new=TRUE,
 plot_dir = TRUE,
 add_lines = NULL,
 add_points= NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 add_data = TRUE,
 add_avg = TRUE,
 plot_horizontal = TRUE)

#Example 2. A plot of a wavelet spectra using the magnetic susceptibility
#data set of Pas et al., (2018)
mag_super <-
analyse_superlet(
   data = mag,
   Nf = 128,
   lowerPeriod = 0.1,
   upperPeriod = 254,
   verbose = FALSE,
   c1=1,
   o = c(1,5),
   mult = TRUE)


plot_superlet(
 superlet = mag_super,
 lowerPeriod = 0.1,
 upperPeriod = 254,
 n.levels = 100,
 palette_name = "rainbow",
color_brewer= "grDevices",
 useRaster = TRUE,
 periodlab = "Period (metres)",
 x_lab = "depth (metres)",
 keep_editable = FALSE,
 dev_new=TRUE,
 plot_dir = TRUE,
 add_lines = NULL,
 add_points= NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 add_data = TRUE,
 add_avg = TRUE,
 plot_horizontal = TRUE)



#Example 3. A plot of a wavelet spectra using the greyscale
# data set of Zeeden et al., (2013)
grey_super <-
```

```
analyse_superlet(
   data = grey,
   Nf = 128,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
   c1=1,
   o = c(1,5),
   mult = TRUE)

plot_superlet(
 superlet = grey_super,
 lowerPeriod = 0.02,
 upperPeriod = 256,
 n.levels = 100,
 palette_name = "rainbow",
color_brewer= "grDevices",
 useRaster = TRUE,
 periodlab = "Period (metres)",
 x_lab = "depth (metres)",
 keep_editable = FALSE,
 dev_new=TRUE,
 plot_dir = TRUE,
 add_lines = NULL,
 add_points= NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 add_data = TRUE,
 add_avg = TRUE,
 plot_horizontal = TRUE)
```

`plot_test_k` *Comparison figure of the precession-constant test*

Description

Plots the result of `test_tuned`: the MTM spectrum with the AstroGeo22 periods, search windows and the peaks used (A), k, Earth-Moon distance and length of day against AstroGeo22 and the Devonian composite (B-D), AstroGeo22 through time with the estimate and optional published constraints (E-G), and the pre-scan of the record.

Usage

```
plot_test_k(res, constraints = NULL, age_ma = res$spec$age_ma)
```

Arguments

res	output of test_tuned.
constraints	optional path to a CSV of published constraints, see test_tuned.
age_ma	age of the record (Ma).

Value

Called for its plot.

---

plot_track *Plot a continuous track on its scalogram*

Description

Plots the scalogram with the track(s) of track_ridge_hmm or track_astro_hmm and their credible bands, the confidence of the track and, when available, the sedimentation rate and the age model with its uncertainty.

Usage

```
plot_track(
  track,
  wavelet,
  show = NULL,
  flags = NULL,
  palette = "viridis",
  keep_editable = FALSE,
  scale = "auto"
)
```

Arguments

track	Output of track_ridge_hmm or track_astro_hmm.
wavelet	The scalogram the track was made from (or another one to display).
show	Periods of the astronomical cycles to draw (for track_astro_hmm) Default=NULL: all.
flags	Optional data frame of intervals (columns from, to) to shade, e.g. the flags of track_astro_auto.
palette	Colour palette of hcl.colors Default="viridis".
keep_editable	Keep the plot settings after plotting Default=FALSE.
scale	Colour scale as in all image plots of the package (see plot_wavelet) Default="auto": log for synchrosqueezed maps, quantiles of the power divided by its mean per period otherwise.

Value

Invisibly the track.

plot_wavelet *Plots a wavelet scalogram*

Description

Plot wavelet scalogram using the outcome of the analyse_wavelet function.

Usage

```
plot_wavelet(
  wavelet = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  useRaster = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  keep_editable = FALSE,
  dev_new = wr_dev_new_default(),
  plot_dir = TRUE,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  add_MTM_peaks = FALSE,
  add_data = TRUE,
  add_avg = FALSE,
  add_pval = FALSE,
  pval_abline = c(0.1, 0.05),
  pval_cutoff = c(0.1),
  add_MTM = FALSE,
  mtm_siglvl = 0.95,
  demean_mtm = TRUE,
  detrend_mtm = TRUE,
  padfac_mtm = 5,
  tbw_mtm = 3,
  plot_horizontal = TRUE,
  add_signif = NULL,
  colour_brewer = NULL,
  whiten = FALSE,
  scale = "auto"
)
```

Arguments

wavelet	wavelet object created using the analyse_wavelet function.

lowerPeriod	Lowest period value which will be plotted
upperPeriod	Highest period value which will be plotted
n.levels	Number of color levels Default=100.
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function The R package 'scico' has the color palette options: "acton", "bam","bamako", "bamO", "batlow", "batlowK","batlowW","berlin","bilbao","broc","br ","buda","bukavu","cork","CorkO","davos","devon","fes","Glasgow","grayC", "hawaii","imola","lajolla", "nuuk","oleron","oslo","roma","romaO","Tofino","Tokyo","turku","Vanimo", "vik","vikO" The R package 'Viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo"
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps,scico and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
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)".
keep_editable	Keep option to add extra features after plotting Default=FALSE
dev_new	Open a new plotting window for the plot. The wavelet plots contain a raster image that needs more room than the small RStudio plot pane, which gives "figure margins too large" errors or clipped text, so a separate window is opened by default in the RStudio GUI. In every other front end (plain R, ESS, Positron, knitr, Rscript) the plot is drawn on the current device and no window is opened. Set TRUE or FALSE to force either behaviour, or set options(waverider.dev_new = TRUE/FALSE). Default: TRUE in the RStudio GUI, FALSE elsewhere.
plot_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE plot_dir=TRUE
add_lines	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL

add_points Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values Default=NULL

add_abline_h Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL

add_abline_v Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL

add_MTM_peaks Add the MTM peak periods as horizontal lines Default=FALSE

add_data Plot the data on top of the wavelet Default=TRUE

add_avg Plot the average wavelet spectral power to the side of the wavelet Default=FALSE

add_pval add an transparent overlay on the wavelet scalogram based on the p-value and add the p-value curve to the average spectral power curve. The p-value is based on a Monte Carlo simulation of the analyse_wavelet function. The p-value is based on Monte Carlo modelling runs on surrogate data generated based on autocorrelated noise (red noise) the calculated using a windowed (the window is half the size of the data set) temporal autocorrelation and on shuffling the data set resulting in a random data sets which has similar spectral characteristics to the original data set.The shuffling of the data set creates white noise which ensures that high amplitude high frequency/short period cycles do not result in statistical significant peaks. The part of the data generated using the autocorrelated noise (red noise) based on the windowed (the window is half the size of the data set) temporal autocorrelation represent a spectral signature similar to to that of the original data. The original data might include spectral peaks which are the result of astronomical forcing. The result is that the spectral power profile is biased towards rejecting the 0-hypothesis (e.g. no astronomical forcing). By combining the shuffling of the data set with autocorrelated noise a surrogate data set is created which rejects high amplitude high frequency/short period cycles and a reduced biased towards towards rejecting the 0-hypothesis if the data was solely the result of autocorrelated noise. Default=FALSE

pval_abline Add straight lines to the average spectral power plot which indicate certain p-values Default=c(0.1,0.5)

pval_cutoff Cutoff p-value to be used in the transparent overlay of the wavelet scalogram Default=c(0.1)

add_MTM Add the MTM plot next to the wavelet plot Default=FALSE

mtm_siglvl select the significance level (0-1) for the MTM spectrum Default=0.95

demean_mtm Remove mean from data before conducting the MTM analysis Default=TRUE

detrend_mtm Remove mean from data before conducting the MTM analysis Default=TRUE

padfac_mtm Pad factor for the MTM analysis Default=5

tbw_mtm time bandwidth product of the MTM analysis Default=3

plot_horizontal

plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE

add_signif Optional significance object created with wavelet_significance. The significant cells are outlined on the scalogram and the critical level of the global spectrum is drawn as a red dashed line on the average power panel Default=NULL.

colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.
whiten	Divide the power of every period by its mean along the record before plotting Default=FALSE. Removes the red-noise background, so that weak ridges at short periods (precession in a noisy record) are shown with the same contrast as the strong long periods; on CIP Case 3 (very red noise) the oscillating precession and obliquity ridges only become visible this way. The average panel is not informative with whiten = TRUE.
scale	Colour scale, the same for every image plot of the package: "auto" (default: logarithmic for synchrosqueezed, superlet-SST, chirplet-SST and other mostly empty maps, histogram-equalised quantiles otherwise), "log", "quantile", "sqrt" or "linear".

Value

The output is a plot of a wavelet spectra. if add_MTM_peaks = TRUE then the output of the MTM analysis will given as matrix

Author(s)

Code based on the "analyze.wavelet" and "wt.image" functions of the 'WaveletComp' R package and "wt" function of the 'biwavelet' R package which are based on the wavelet MATLAB code written by Christopher Torrence and Gibert P. Compo (1998). The MTM analysis is from the astrochron R package of Meyers et al., (2012)

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

S.R. Meyers, 2012, Seeing Red in Cyclic Stratigraphy: Spectral Noise Estimation for Astrochronology: Paleoceanography, 27, PA3228, <doi:10.1029/2012PA002307>

Examples

```
#Example 1. A plot of a wavelet spectra using the Total Solar Irradiance
# data set of Steinhilber et al., (2012)

TSI_wt <-
 analyse_wavelet(
```

```
   data = TSI,
   dj = 1/200,
   lowerPeriod = 16,
   upperPeriod = 8192,
   verbose = FALSE,
   omega_nr = 6
 )

plot_wavelet(
 wavelet = TSI_wt,
 lowerPeriod = NULL,
 upperPeriod = NULL,
 n.levels = 100,
 palette_name = "rainbow",
color_brewer= "grDevices",
 useRaster = TRUE,
 periodlab = "Period (metres)",
 x_lab = "depth (metres)",
 keep_editable = FALSE,
 dev_new=TRUE,
 plot_dir = TRUE,
 add_lines = NULL,
 add_points= NULL,
 add_abline_h = NULL,
 add_abline_v = NULL,
 add_MTM_peaks = FALSE,
 add_data = TRUE,
 add_avg = TRUE,
 add_pval = FALSE,
 pval_abline = c(0.1,0.05),
 pval_cutoff = c(0.1),
 add_MTM = FALSE,
 mtm_siglvl = 0.95,
 demean_mtm = TRUE,
 detrend_mtm = TRUE,
 padfac_mtm = 5,
 tbw_mtm = 3,
 plot_horizontal=TRUE)

#Example 2. A plot of a wavelet spectra using the magnetic susceptibility
#data set of Pas et al., (2018)
mag_wt <-
analyse_wavelet(
data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
omega_nr = 10
)

plot_wavelet(
wavelet = mag_wt,
```

```
lowerPeriod = NULL,
upperPeriod = NULL,
n.levels = 100,
palette_name = "rainbow",
color_brewer= "grDevices",
useRaster = TRUE,
periodlab = "Period (metres)",
x_lab = "depth (metres)",
keep_editable = FALSE,
dev_new=TRUE,
plot_dir = TRUE,
add_lines= NULL,
add_points= NULL,
add_abline_h = NULL,
add_abline_v = NULL,
add_MTM_peaks = FALSE,
add_data = TRUE,
add_avg = TRUE,
add_pval = FALSE,
pval_abline = c(0.1,0.05),
pval_cutoff = c(0.1),
add_MTM = FALSE,
mtm_siglvl = 0.95,
demean_mtm = TRUE,
detrend_mtm = TRUE,
padfac_mtm = 5,
tbw_mtm = 3,
plot_horizontal=TRUE)


#Example 3. A plot of a wavelet spectra using the greyscale
# data set of Zeeden et al., (2013)
grey_wt <-
 analyse_wavelet(
   data = grey,
   dj = 1/200,
   lowerPeriod = 0.02,
   upperPeriod = 256,
   verbose = FALSE,
   omega_nr = 8
 )

plot_wavelet(
wavelet = grey_wt,
lowerPeriod = NULL,
upperPeriod = NULL,
n.levels = 100,
palette_name = "rainbow",
color_brewer= "grDevices",
useRaster = TRUE,
periodlab = "Period (metres)",
x_lab = "depth (metres)",
keep_editable = FALSE,
```

```
  dev_new=TRUE,
  plot_dir = TRUE,
  add_lines = NULL,
  add_points= NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  add_MTM_peaks = FALSE,
  add_data = TRUE,
  add_avg = TRUE,
  add_pval = FALSE,
  pval_abline = c(0.1,0.05),
  pval_cutoff = c(0.1),
  add_MTM = FALSE,
  mtm_siglvl = 0.95,
  demean_mtm = TRUE,
  detrend_mtm = TRUE,
  padfac_mtm = 5,
  tbw_mtm = 3,
  plot_horizontal=TRUE)
```

`plot_wavelet_coherence` *Plots a cross wavelet scalogram*

Description

Plot cross wavelet scalogram using the outcome of the `analyse_wavelet_coherence` function.

Usage

```
plot_wavelet_coherence(
  wavelet_coh = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  useRaster = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  keep_editable = FALSE,
  dev_new = wr_dev_new_default(),
  plot_dir = TRUE,
  plot_arrows = TRUE,
  pwr_quant = 0.85,
  n_arrows = 75,
```

```
  T_unit = NULL,
  P_unit = NULL,
  arrow_length = 0.075,
  arrow_lwd = 1.9,
  arrow_angle = 25,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  add_data = TRUE,
  add_avg = FALSE,
  plot_horizontal = TRUE,
  colour_brewer = NULL
)
```

Arguments

wavelet_coh	wavelet_coh object created using the analyse_wavelet_coherence function.
lowerPeriod	Lowest period value which will be plotted
upperPeriod	Highest period value which will be plotted
n.levels	Number of color levels Default=100.
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function The R package 'scico' has the color palette options: "acton", "bam","bamako", "bamO", "batlow", "batlowK","batlowW","berlin","bilbao","broc","bro "buda","bukavu","cork","CorkO","davos","devon","fes","Glasgow","grayC", "hawaii","imola","lajolla", "nuuk","oleron","oslo","roma","romaO","Tofino","Tokyo","turku","Vanimo", "vik","vikO" The R package 'Viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo"
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps,scico and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
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)".
keep_editable	Keep option to add extra features after plotting Default=FALSE
dev_new	Open a new plotting window for the plot. The wavelet plots contain a raster image that needs more room than the small RStudio plot pane, which gives "figure margins too large" errors or clipped text, so a separate window is opened by default in the RStudio GUI. In every other front end (plain R, ESS, Positron, knitr, Rscript) the plot is drawn on the current device and no window is opened. Set TRUE or FALSE to force either behaviour, or set options(waverider.dev_new = TRUE/FALSE). Default: TRUE in the RStudio GUI, FALSE elsewhere.
plot_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE plot_dir=TRUE
plot_arrows	add the phase arrows of the cross superlet transform Default=TRUE
pwr_quant	Power quantile above which the arrows are plotted Default= 0.85
n_arrows	number arrows width wise plotted on the time axis Default= 50
T_unit	length of the arrows in depth/time units Default=NULL: automatic, a fixed visual length of 0.3 inch whatever the record length
P_unit	length of the arrows in log2 period units Default=NULL: automatic (see T_unit)
arrow_length	length of the slanted part of the arrow Default= 0.1
arrow_lwd	thickness of the arrow Default= 2
arrow_angle	angle of the slanted part of the arrow Default= 25
add_lines	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL
add_points	Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values Default=NULL
add_abline_h	Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_abline_v	Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_data	Plot the data on top of the wavelet Default=TRUE
add_avg	Plot the average wavelet spectral power to the side of the wavelet Default=FALSE
plot_horizontal	plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

The output is a plot of a cross wavelet scalogram.

Author(s)

plotting code based on the "wt.image" and "analyze.coherency" functions of the 'WaveletComp' R package

References

Roesch, A., & Schmidbauer, H. (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., & Mureșan, R. C. (2021). Time-frequency super-resolution with superlets. Nature Communications, 12(1), 337. doi:10.1038/s41467020205399

Examples

```
#Example 1. A cross superlet plot of two etp solutions with noise overprint


etp_1 <- astrochron::etp(
 tmin = 0,
 tmax = 1500,
 dt = 1,
 eWt = 1.5,
 oWt = 0.75,
 pWt = 1,
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
 verbose = FALSE
)

etp_2 <- astrochron::etp(
 tmin = 0,
 tmax = 1500,
 dt = 1,
 eWt = 1,
 oWt = 0.5,
 pWt = 1.5,
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
verbose = FALSE
)

etp_1[, 2] <- etp_1[, 2] + wr_coloured_noise(
 nrow(etp_1),
 sd = sd(etp_1[, 2]) / 1.5,
 mean = mean(etp_1[, 2]),
 phi = 0.9
)
etp_2[, 2] <- etp_2[, 2] + wr_coloured_noise(
 nrow(etp_2),
 sd = sd(etp_2[, 2]) / 1.5,
```

```
 mean = mean(etp_2[, 2]),
 phi = 0.9
)

coh_etp <- analyse_wavelet_coherence(
 data_1 = etp_1,
 data_2  = etp_2,
 upperPeriod = 1024,
 lowerPeriod = 2
)
plot_wavelet_coherence(wavelet_coh = coh_etp, lowerPeriod = 2, upperPeriod = 1024,
              n.levels = 100, palette_name = "rainbow", color_brewer = "grDevices",
              useRaster = TRUE, periodlab = "Period (metres)", x_lab = "depth (metres)",
              keep_editable = FALSE, dev_new = TRUE, plot_dir = TRUE,
              plot_arrows = TRUE,
              pwr_quant = 0.85,
              n_arrows = 50,
              T_unit = NULL,
              P_unit  = NULL,
              arrow_length = 0.1,
              arrow_lwd = 2,
              arrow_angle = 25,add_lines = NULL, add_points = NULL, add_abline_h = NULL,
            add_abline_v = NULL, add_data = TRUE, add_avg = FALSE, plot_horizontal = TRUE)
```

`plot_win_fft` *Plot windowed fft based spectral analysis results*

Description

The `plot_win_fft` function allows for the (re)plotting of the results of the `win_fft`

Usage

```
plot_win_fft(
  win_fft = NULL,
  x_lab = c("depth (m)"),
  y_lab = c("frequency cycle/metre"),
  plot_res = 1,
  perc_vis = 0,
  freq_max = NULL,
  freq_min = NULL,
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  plot_horizontal = TRUE,
  dev_new = wr_dev_new_default(),
  colour_brewer = NULL
)
```

Arguments

win_fft	list which is the results of the win_fft
x_lab	label for the y-axis Default="depth"
y_lab	label for the y-axis Default="sedrate"
plot_res	plot 1 of 8 options option 1: Amplitude matrix, option 2: Power matrix, option 3: Phase matrix, option 4: AR1_CL matrix, option 5: AR1_Fit matrix , option 6: AR1_90_power matrix, option 7: AR1_95_power matrix, option 8: AR1_99_power matrix, Default=1
perc_vis	Cutoff percentile when plotting Default=0
freq_max	Maximum frequency to plot
freq_min	Minimum frequency to plot
keep_editable	Keep option to add extra features after plotting Default=FALSE
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function The R package 'scico' has the color palette options: "acton", "bam","bamako", "bamO", "batlow", "batlowK","batlowW","berlin","bilbao","broc","bro "buda","bukavu","cork","CorkO","davos","devon","fes","Glasgow","grayC", "hawaii","imola","lajolla","l "nuuk","oleron","oslo","roma","romaO","Tofino","Tokyo","turku","Vanimo", "vik","vikO" The R package 'Viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo"
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps,scico and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
plot_horizontal	plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE
dev_new	Open a new plotting window for the plot. The wavelet plots contain a raster image that needs more room than the small RStudio plot pane, which gives "figure margins too large" errors or clipped text, so a separate window is opened by default in the RStudio GUI. In every other front end (plain R, ESS, Positron, knitr, Rscript) the plot is drawn on the current device and no window is opened. Set TRUE or FALSE to force either behaviour, or set options(waverider.dev_new = TRUE/FALSE). Default: TRUE in the RStudio GUI, FALSE elsewhere.
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

Returns a plot of, which plot 1 of 8 options, option 1: Amplitude matrix option 2: Power matrix option 3: Phase matrix option 4: AR1_CL matrix option 5: AR1_Fit matrix option 6: AR1_90_power matrix option 7: AR1_95_power matrix option 8: AR1_99_power matrix

Examples

```
#Conduct a windowed fft on the magnetic susceptibility record \cr
# of the Sullivan core of Pas et al., (2018).

mag_win_fft <- win_fft(data= mag,
                   padfac = 5,
                   window_size = 12.5,
                   run_multicore = FALSE,
                   genplot = FALSE,
                   palette_name = "rainbow",
                   color_brewer="grDevices",
                   x_lab = c("depth (m)"),
                   y_lab = c("frequency cycle/meter"),
                   plot_res = 1,
                   perc_vis = 0.5,
                   freq_max = 5,
                   freq_min = 0.001,
                   keep_editable=FALSE,
                   verbose=FALSE)

# Plot the amplitude spectra
plot_win_fft(win_fft= mag_win_fft,
x_lab = c("depth (m)"),
y_lab = c("frequency cycle/meter"),
plot_res = 1,
perc_vis = 0.5,
freq_max = 5,
freq_min = 0.001,
keep_editable=FALSE,
palette_name = "rainbow",
color_brewer="grDevices",
plot_horizontal=TRUE,
dev_new=TRUE)
```

---

plot_win_timeOpt *plot the windowed timeOpt sedimentation rate estimation*

Description

The plot_win_timeOpt function plots a widowed timeOpt sedimentation rate estimation This function is based on the eTimeOptfunction

Usage

```
plot_win_timeOpt(
  win_timeOpt_result = NULL,
  proxy_name = NULL,
  abline_h = NULL,
  abline_v = NULL,
  add_lines = NULL,
  fig_lts = NULL,
  xlab = "depth (m)",
  ylab = "sedrate (cm/kyr)",
  sel_parameter = 3,
  n.levels = 100
)
```

Arguments

win_timeOpt_result	result of the win_timeOptfunction that needs to be used as input Default=NULL
proxy_name	the name of the used proxy record Default=NULL
abline_h	Add horizontal lines to the plot. Specify the lines as a vector e.g. 2,3,5,6 Default=NULL
abline_v	Add vertical lines to the plot. Specify the lines as a vector e.g. 2,3,5,6 Default=NULL
add_lines	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL
fig_lts	Add a text box Default=NULL
xlab	add a label to x-axis Default="depth (m)"
ylab	add a label to y-axis Default="sedrate (cm/kyr)"
sel_parameter	select one of the three returns of the win_timeOptfunction element 1: r_2_envelope matrix element 2: r_2_power matrix element 3: r_2_opt matrix Default=3
n.levels	Number of color levels Default=100.

Value

The output is a plot of the average spectral power of a windowed timeOpt

Author(s)

Based on the eTimeOpt function of the 'astrochron' R package.

References

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

Examples

```
#plot the windowed timeOpt of the magnetic susceptibility record
#of the Sullivan core of Pas et al., (2018).
mag_win_timeOpt <-win_timeOpt(
data = mag[1:250, ],
window_size = 6,
sedmin = 0.1,
sedmax = 1,
numsed = 5,
limit = FALSE,
fit = 2,
fitModPwr = TRUE,
flow = NULL,
fhigh = NULL,
roll = 10 ^ 6,
targetE = c(405.7, 130.7, 123.8, 98.9, 94.9),
targetP = c(20.9, 19.9, 17.1, 17.2),
detrend = TRUE,
normalize =TRUE,
linLog = 1,
run_multicore = FALSE,
verbose=FALSE)

plot_win_timeOpt(win_timeOpt_result = mag_win_timeOpt,
proxy_name= "mag",
abline_h=NULL,
abline_v = NULL,
add_lines=NULL,
fig_lts = NULL,
xlab="depth (m)",
ylab= "sedrate (cm/kyr)",
sel_parameter=3,
n.levels=100)
```

---

plot_Xsuperlet *Plots a cross superlet scalogram*

Description

Plot cross superlet scalogram using the outcome of the analyse_Xsuperlet function.

Usage

```
plot_Xsuperlet(
  Xsuperlet = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
```

```
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  useRaster = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  keep_editable = FALSE,
  dev_new = wr_dev_new_default(),
  plot_dir = TRUE,
  plot_arrows = TRUE,
  pwr_quant = 0.85,
  n_arrows = 75,
  T_unit = NULL,
  P_unit = NULL,
  arrow_length = 0.075,
  arrow_lwd = 1.9,
  arrow_angle = 25,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  add_data = TRUE,
  add_avg = FALSE,
  plot_horizontal = TRUE,
  colour_brewer = NULL
)
```

Arguments

`Xsuperlet`	Xsuperlet object created using the `analyse_Xsuperlet` function.
`lowerPeriod`	Lowest period value which will be plotted
`upperPeriod`	Highest period value which will be plotted
`n.levels`	Number of color levels `Default=100`.
`palette_name`	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages `Default=rainbow`. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the

	grDevices::hcl.pals() function The R package 'scico' has the color palette options: "acton", "bam","bamako", "bamO", "batlow", "batlowK","batlowW","berlin","bilbao","broc","br ","buda","bukavu","cork","CorkO","davos","devon","fes","Glasgow","grayC", "hawaii","imola","lajolla", "nuuk","oleron","oslo","roma","romaO","Tofino","Tokyo","turku","Vanimo", "vik","vikO" The R package 'Viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo"
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps,scico and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
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)".
keep_editable	Keep option to add extra features after plotting Default=FALSE
dev_new	Open a new plotting window for the plot. The wavelet plots contain a raster image that needs more room than the small RStudio plot pane, which gives "figure margins too large" errors or clipped text, so a separate window is opened by default in the RStudio GUI. In every other front end (plain R, ESS, Positron, knitr, Rscript) the plot is drawn on the current device and no window is opened. Set TRUE or FALSE to force either behaviour, or set options(waverider.dev_new = TRUE/FALSE). Default: TRUE in the RStudio GUI, FALSE elsewhere.
plot_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE plot_dir=TRUE
plot_arrows	add the phase arrows of the cross superlet transform Default=TRUE
pwr_quant	Power quantile above which the arrows are plotted Default= 0.85
n_arrows	number arrows width wise plotted on the time axis Default= 50
T_unit	length of the arrows in depth/time units Default=NULL: automatic, a fixed visual length of 0.3 inch whatever the record length
P_unit	length of the arrows in log2 period units Default=NULL: automatic (see T_unit)
arrow_length	length of the slanted part of the arrow Default= 0.1
arrow_lwd	thickness of the arrow Default= 2
arrow_angle	angle of the slanted part of the arrow Default= 25
add_lines	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL
add_points	Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values Default=NULL
add_abline_h	Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL

add_abline_v	Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_data	Plot the data on top of the wavelet Default=TRUE
add_avg	Plot the average wavelet spectral power to the side of the wavelet Default=FALSE
plot_horizontal	plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

The output is a plot of a cross superlet scalogram.

Author(s)

plotting code based on the "wt.image" function of the 'WaveletComp' R package Whereas the "analyse_Xsuperlet" that generates the input for the plotting function is based on the matlab code in Moca et al. (2021) and the the "analyze.coherency" function of the 'WaveletComp' R package

References

Roesch, A., & Schmidbauer, H. (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., & Mureșan, R. C. (2021). Time-frequency superresolution with superlets. Nature Communications, 12(1), 337. doi:10.1038/s41467020205399

Examples

```
#Example 1. A cross superlet plot of two etp solutions with noise overprint


etp_1 <- astrochron::etp(
 tmin = 0,
 tmax = 500,
 dt = 1,
 eWt = 1.5,
 oWt = 0.75,
 pWt = 1,
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
 verbose = FALSE
)

etp_2 <- astrochron::etp(
 tmin = 0,
 tmax = 500,
 dt = 1,
 eWt = 1,
 oWt = 0.5,
 pWt = 1.5,
```

```
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
verbose = FALSE
)

etp_1[, 2] <- etp_1[, 2] + wr_coloured_noise(
 nrow(etp_1),
 sd = sd(etp_1[, 2]) / 1.5,
 mean = mean(etp_1[, 2]),
 phi = 0.9
)
etp_2[, 2] <- etp_2[, 2] + wr_coloured_noise(
 nrow(etp_2),
 sd = sd(etp_2[, 2]) / 1.5,
 mean = mean(etp_2[, 2]),
 phi = 0.9
)

X_super_etp <- analyse_Xsuperlet(
 data_1 = etp_1,
 data_2  = etp_2,
 upperPeriod = 1024,
 lowerPeriod = 2,
 Nf = 32,
 c1 = 3,
 o = c(1, 10),
 mult = TRUE,
 verbose = FALSE
)
plot_Xsuperlet(Xsuperlet = X_super_etp, lowerPeriod = 2, upperPeriod = 1024,
               n.levels = 100, palette_name = "rainbow", color_brewer = "grDevices",
               useRaster = TRUE, periodlab = "Period (metres)", x_lab = "depth (metres)",
               keep_editable = FALSE, dev_new = TRUE, plot_dir = TRUE,
               plot_arrows = TRUE,
               pwr_quant = 0.85,
               n_arrows = 50,
               T_unit = NULL,
               P_unit  = NULL,
               arrow_length = 0.1,
               arrow_lwd = 2,
               arrow_angle = 25,add_lines = NULL, add_points = NULL, add_abline_h = NULL,
             add_abline_v = NULL, add_data = TRUE, add_avg = FALSE, plot_horizontal = TRUE)
```

`plot_Xwavelet` *Plots a cross wavelet scalogram*

Description

Plot cross wavelet scalogram using the outcome of the `analyse_Xwavelet` function.

Usage

```
plot_Xwavelet(
  Xwavelet = NULL,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  useRaster = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  keep_editable = FALSE,
  dev_new = wr_dev_new_default(),
  plot_dir = TRUE,
  plot_arrows = TRUE,
  pwr_quant = 0.85,
  n_arrows = 75,
  T_unit = NULL,
  P_unit = NULL,
  arrow_length = 0.075,
  arrow_lwd = 1.9,
  arrow_angle = 25,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  add_data = TRUE,
  add_avg = FALSE,
  plot_horizontal = TRUE,
  colour_brewer = NULL
)
```

Arguments

Xwavelet	Xwavelet object created using the analyse_Xwavelet function.
lowerPeriod	Lowest period value which will be plotted
upperPeriod	Highest period value which will be plotted
n.levels	Number of color levels Default=100.
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the

	color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function The R package 'scico' has the color palette options: "acton", "bam","bamako", "bamO", "batlow", "batlowK","batlowW","berlin","bilbao","broc","br ","buda","bukavu","cork","CorkO","davos","devon","fes","Glasgow","grayC", "hawaii","imola","lajolla", "nuuk","oleron","oslo","roma","romaO","Tofino","Tokyo","turku","Vanimo", "vik","vikO" The R package 'Viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo"
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps,scico and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
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)".
keep_editable	Keep option to add extra features after plotting Default=FALSE
dev_new	Open a new plotting window for the plot. The wavelet plots contain a raster image that needs more room than the small RStudio plot pane, which gives "figure margins too large" errors or clipped text, so a separate window is opened by default in the RStudio GUI. In every other front end (plain R, ESS, Positron, knitr, Rscript) the plot is drawn on the current device and no window is opened. Set TRUE or FALSE to force either behaviour, or set options(waverider.dev_new = TRUE/FALSE). Default: TRUE in the RStudio GUI, FALSE elsewhere.
plot_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE plot_dir=TRUE
plot_arrows	add the phase arrows of the cross superlet transform Default=TRUE
pwr_quant	Power quantile above which the arrows are plotted Default= 0.85
n_arrows	number arrows width wise plotted on the time axis Default= 50
T_unit	length of the arrows in depth/time units Default=NULL: automatic, a fixed visual length of 0.3 inch whatever the record length
P_unit	length of the arrows in log2 period units Default=NULL: automatic (see T_unit)
arrow_length	length of the slanted part of the arrow Default= 0.1
arrow_lwd	thickness of the arrow Default= 2
arrow_angle	angle of the slanted part of the arrow Default= 25
add_lines	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL

add_points	Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values Default=NULL
add_abline_h	Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_abline_v	Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_data	Plot the data on top of the wavelet Default=TRUE
add_avg	Plot the average wavelet spectral power to the side of the wavelet Default=FALSE
plot_horizontal	plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

The output is a plot of a cross wavelet scalogram.

Author(s)

plotting code based on the "wt.image" and "analyze.coherency" functions of the 'WaveletComp' R package

References

Roesch, A., & Schmidbauer, H. (2018). WaveletComp: Computational Wavelet Analysis. R package version 1.1. https://CRAN.R-project.org/package=WaveletComp

Moca, V. V., Bârzan, H., Nagy-Dăbâcan, A., & Mureșan, R. C. (2021). Time-frequency super-resolution with superlets. Nature Communications, 12(1), 337. doi:10.1038/s41467020205399

Examples

```
#Example 1. A cross superlet plot of two etp solutions with noise overprint


etp_1 <- astrochron::etp(
 tmin = 0,
 tmax = 1500,
 dt = 1,
 eWt = 1.5,
 oWt = 0.75,
 pWt = 1,
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
 verbose = FALSE
)

etp_2 <- astrochron::etp(
 tmin = 0,
 tmax = 1500,
```

```
 dt = 1,
 eWt = 1,
 oWt = 0.5,
 pWt = 1.5,
 esinw = TRUE,
 standardize = TRUE,
 genplot = FALSE,
verbose = FALSE
)

etp_1[, 2] <- etp_1[, 2] + wr_coloured_noise(
 nrow(etp_1),
 sd = sd(etp_1[, 2]) / 1.5,
 mean = mean(etp_1[, 2]),
 phi = 0.9
)
etp_2[, 2] <- etp_2[, 2] + wr_coloured_noise(
 nrow(etp_2),
 sd = sd(etp_2[, 2]) / 1.5,
 mean = mean(etp_2[, 2]),
 phi = 0.9
)

X_wavelet_etp <- analyse_Xwavelet(
 data_1 = etp_1,
 data_2  = etp_2,
 upperPeriod = 1024,
 lowerPeriod = 2,
  omega_nr = 8,
 verbose = FALSE
)
plot_Xwavelet(Xwavelet = X_wavelet_etp, lowerPeriod = 2, upperPeriod = 1024,
              n.levels = 100, palette_name = "rainbow", color_brewer = "grDevices",
              useRaster = TRUE, periodlab = "Period (metres)", x_lab = "depth (metres)",
              keep_editable = FALSE, dev_new = TRUE, plot_dir = TRUE,
              plot_arrows = TRUE,
              pwr_quant = 0.85,
              n_arrows = 50,
              T_unit = NULL,
              P_unit  = NULL,
              arrow_length = 0.1,
              arrow_lwd = 2,
              arrow_angle = 25,add_lines = NULL, add_points = NULL, add_abline_h = NULL,
            add_abline_v = NULL, add_data = TRUE, add_avg = FALSE, plot_horizontal = TRUE)
```

---

precession_frequency *Luni-solar precession frequency (Farhat et al. 2022, eq. A.7)*

Description

Luni-solar precession frequency (Farhat et al. 2022, eq. A.7)

Usage

```
precession_frequency(
  a_RE,
  lod_h = NULL,
  obliquity_deg = .AC$obliquity0_deg,
  Ed = .AC$Ed0,
  lod_kind = "solar"
)
```

Arguments

a_RE	Earth-Moon distance in Earth radii
lod_h	length of day; if NULL it is derived from ‘a_RE‘ by angular-momentum conservation
obliquity_deg	Earth’s obliquity
Ed	dynamical ellipticity at the present spin rate
lod_kind	Type of day, ‘"solar"‘ or ‘"sidereal"‘.

Value

list: p in arcsec/yr and the precession period in kyr

---

predict.astroMCBchron *Predict ages at new positions from an astroMCBchron model*

Description

Predict ages at new positions from an astroMCBchron model

Usage

```
## S3 method for class 'astroMCBchron'
predict(object, newPositions, side = c("upper", "lower"), ...)
```

Arguments

object	an astroMCBchron object.
newPositions	positions (same type and unit as the model).
side	for positions exactly at a hiatus: "upper" (younger side, default) or "lower".
...	unused.

Value

list with `raw` (draws x positions, in age units) and `HDI` (position, 2.5, 50 and 97.5 percentiles).

`print.astro_from_peaks` *Print an astro_from_peaks result*

Description

Print an astro_from_peaks result

Usage

```
## S3 method for class 'astro_from_peaks'
print(x, ...)
```

Arguments

`x`	an astro_from_peaks object
`...`	ignored

Value

x, invisibly

`print.ctest` *Print a CTest result*

Description

Print a CTest result

Usage

```
## S3 method for class 'ctest'
print(x, ...)
```

Arguments

`x`	a ctest object
`...`	ignored

Value

x, invisibly

rate_gram *Rate-gram: the astronomical template as a spectrum over sedimentation rate, per window*

Description

A purely spectral estimate of the sedimentation rate along a record, without any tracking: for every window of `analyse_chirp_eha` (`criterion = "template"`) the template score (the summed log line-to-continuum ratio at the periods of all astronomical cycles, the cyclostratigraphic counterpart of the harmonic product spectrum used for pitch detection in speech and music) is taken at its best gradient and shown against the rate; its maximum per window is the spectral rate estimate, optionally median-filtered over a few windows. On CIP Case 3 the maximum per window correlates 0.56 with the true rate, the median over 11 windows 0.73 (the tracked rate of `track_chirp_hmm`, which adds continuity, 0.97).

Usage

```
rate_gram(
  ce,
  k = 11,
  plot = TRUE,
  rate_unit = c(1, "depth units / kyr"),
  reference = NULL,
  main = NULL
)
```

Arguments

`ce`	Result of `analyse_chirp_eha` with `criterion = "template"`.
`k`	Width (windows) of the running median of the maximum `Default=11`.
`plot`	Draw the rate-gram `Default=TRUE`.
`rate_unit`	Factor and label of the rate axis, e.g. `c(100, "cm/kyr")`.
`reference`	Optional data frame (depth, rate) drawn for comparison.
`main`	Title.

Value

A list with `depth`, `sedrate` (rate grid), `map` (rates x windows), `best` (the maximum per window) and `smooth` (its running median).

resonance_state *Which secular resonance state does a record show?*

Description

Which secular resonance state does a record show?

Usage

```
resonance_state(ecc_kyr, obl_kyr, max_ratio = 4, data_dir = NULL)
```

Arguments

ecc_kyr	measured period of the long eccentricity modulation (the "2.4 Myr" cycle, g4-g3)
obl_kyr	measured period of the obliquity modulation (the "1.2 Myr" cycle, s4-s3)
max_ratio	largest integer ratio to consider
data_dir	Directory with the data files (‘NULL‘ = the data installed with the package).

Value

list: the implied frequencies, their ratio, the nearest simple resonance, the resonance argument (arcsec/yr and as a period), and the modern values for comparison

retrack_wt_MC *Re-track cycles using a Monte-Carlo simulation*

Description

When analysing multi-proxy records an age-model can be created for each proxy. These age-models can be in general agreement but might also indicate conflicting deposition rates. Picking one age-model out of the all multi-proxy age-models and stating that, that age-model is the best overlooks the information contained within the other proxies and hence a degree of error remains the age-model exists.To combine the multiple age-models all the age models can be averaged out and the uncertainty can be calculated by means of the standard deviation. The result is an age-model which takes into account all the age-models from the proxy records. The averaged out age-model does not take into account any small user errors during the creation of the individual age-models nor does the averaging take into account the differences between the age-models and how the initial age-model of a certain proxy might be off in certain intervals. the retrack_wt_MC mitigates these problems by re-tracking periods of astronomical cycles in the wavelet spectra. The re-tracking is based on the information provided by the age-models constructed from the different proxy records. First a synthetic tracked curve is created by adding up fractions (0-1) of the tracked periods of the different proxy records. This synthetic curve is then used to re-track the period/spectral peaks of an astronomical cycle in a randomly select wavelet scalogram. This process is repeated x times.

The result x tracked curves which take into account all the original age-models. From the re-tracked curves one can calculate the mean period and the standard deviation. The resulting standard deviation is a good indicator of the quality of the imprint of of astronomical cycles in the proxy records. A small standard deviation indicates that given the input of the different tracked cycles similar periods keep on being tracked indicating the an astronomical is well recorded in the proxy records and as such the age-model is very reliable in set interval. A high standard deviation on the other hand means that the tracking results in vastly different periods of the tracked astronomical cycle, as such the quality of the imprint of the astronomical cycle proxy records is poor and hence the age-model is less-reliable in this interval.

Usage

```
retrack_wt_MC(
  wt_list = NULL,
  data_track = NULL,
  x_axis = NULL,
  smoothing = c("auto"),
  nr_simulations = 50,
  seed_nr = 1337,
  verbose = FALSE,
  genplot = FALSE,
  keep_editable = FALSE,
  create_GIF = FALSE,
  plot_GIF = FALSE,
  width_plt = 600,
  height_plt = 450,
  period_up = 1.5,
  period_down = 0.5,
  plot.COI = TRUE,
  n.levels = 100,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  add_avg = FALSE,
  time_dir = TRUE,
  file_name = NULL,
  run_multicore = FALSE,
  n_cores = NULL,
  output = 1,
  n_imgs = 50,
  plot_horizontal = TRUE,
  empty_folder = FALSE,
  colour_brewer = NULL
)
```

Arguments

wt_list	a list containing all the wavelet objects created using the analyse_wavelet wavelet function To create a list use the list function
data_track	a matrix containing all the tracked period values. To create the matrix use the cbind function and only add the tracked period values so do not add the depth axis. When combining the tracked period values make sure that all curves have a similar depth spacing.
x_axis	The x-axis of the tracked period values
smoothing	setting the smoothing parameter and value to either "auto" which uses a automatic loess smoother,"loess" where one can specify Lowess smoothing parameter. or "window" where one can specific the window length of the moving average. one should specify the parameter and its value as vector #' @param wt_list a list containing all the wavelet objects created using the analyse_wavelet wavelet function To create a list use the list function
nr_simulations	The number of Monte-Carlo simulations which are to be conductedDefault=50
seed_nr	The seed number of the Monte-Carlo simulations. Default=1337
verbose	Print text when running the function Default=FALSE
genplot	Plot a plot with the mean period and + and - standard deviation Default=FALSE
keep_editable	Keep option to add extra features after plotting Default=FALSE
create_GIF	Create a GIF with the re-tracked lines in the wavelet scalograms Default=FALSE
plot_GIF	Plot a GIF with the re-tracked lines in the wavelet scalogramsDefault=FALSE
width_plt	width of the re-tracked plot Default=600
height_plt	width of the re-tracked plot Default=450
period_up	The period_up parameter is the factor with which the linear interpolated tracked_curve curve is multiplied by. This linear interpolated tracked_curve multiplied by the period_up factor is the upper boundary which is used for detecting the spectral peak nearest to the linear interpolated tracked_curve curve. If no spectral peak is detected within the specified boundary the interpolated value is used instead. between spectral peaks Default=1.5,
period_down	The period_down parameter is the factor with which the linear interpolated tracked_curve curve is multiplied by. This linear interpolated tracked_curve multiplied by the period_down factor is the lower boundary which is used for detecting the spectral peak nearest to the linear interpolated tracked_curve curve. If no spectral peak is detected within the specified boundary the interpolated value is used instead. between spectral peaks Default=0.5,
plot.COI	Option to plot the cone of influence Default=TRUE.
n.levels	Number of color levels Default=100.
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette

	options: “magma”, “plasma”, “inferno”, “viridis”, “mako”, and “rocket” and “turbo” To see the color palette options of the The R pacakge ’RColorBrewer’ run the wr_brewer_palettes() function The R package ’colorRamps’ has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package ’grDevices’ has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge ’grDevices’ run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
periodlab	Label for the y-axis Default="Period (metres)".
x_lab	Label for the x-axis Default="depth (metres)".
add_avg	Plot the average wavelet spectral power to the side of the wavelet Default=FALSE
time_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then time_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then time_dir should be set to FALSE time_dir=TRUE
file_name	Name of the images created using this function. Each file gets a number added to it which corresponds to which number of simulation it was the files are saved in a folder with a similar name created in the current directory
run_multicore	Run function using multiple cores Default="FALSE"
n_cores	Number of cores used when ‘run_multicore = TRUE‘ (‘Default = NULL‘: an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set ‘options(waverider.n_cores = ...)‘ to change the default. A value > 1 also switches on parallel computing. Results do not depend on the number of cores.
output	#’If output = 1, output is a list which contain 3 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. #’object 2 is a matrix with all the tracked periods. Object 3 is a GIF in which #’all the tracked periods are plotted. If output = 2, output is a list which contain 2 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. object 2 is a matrix with all the tracked periods. If output = 3, output is a list which contain 2 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. Object 2 is a GIF in which all the tracked periods are plotted. If output = 4, output is a list which contain 3 objects. Object 1 is a matrix with all the tracked periods. Object 2 is a GIF in which all the tracked periods are plotted. If output = 4 output is a list which contain 3 objects. Object 1 is a matrix with all the tracked periods. Object 2 is a GIF in which all the tracked periods are plotted. If output = 5 a matrix with the x-axis and the mean tracked frequency and standard deviation is returned. If output = 6, a matrix with all the tracked periods is returned. If output = 7, a GIF in which all the tracked periods are plotted is returned. Default=1

n_imgs	Number images used in creating the GIF a high number of images is computationally intensive and will create a large sized GIF Default=50
plot_horizontal	plot the wavelet horizontal or vertical eg y axis is depth or y axis power Default=TRUE
empty_folder	Empty the folder in which the images created using this function are saved Default=FALSE
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

The output depends on the output setting If genplot = TRUE a plot will be generated in which the mean period and standard deviation is plotted if plot_GIF = TRUE a GIF with n number of n_imgs will be plotted in which the retraced curve is plotted in a wavelet scalogram If output = 1, output is a list which contain 3 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. object 2 is a matrix with all the tracked periods. Object 3 is a GIF in which all the tracked periods are plotted. If output = 2, output is a list which contain 2 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. object 2 is a matrix with all the tracked periods. If output = 3, output is a list which contain 2 objects. object 1 is a matrix with the x-axis and the mean tracked frequency and standard deviation. Object 2 is a GIF in which all the tracked periods are plotted. If output = 4, output is a list which contain 3 objects. Object 1 is a matrix with all the tracked periods. Object 2 is a GIF in which all the tracked periods are plotted. If output = 4 output is a list which contain 3 objects. Object 1 is a matrix with all the tracked periods. Object 2 is a GIF in which all the tracked periods are plotted. If output = 5 a matrix with the x-axis and the mean tracked period and standard deviation is returned. If output = 6, a matrix with all the tracked periods is returned. If output = 7, a GIF in which all the tracked periods are plotted is returned

Examples

```
# Re-track the 110kyr eccentricity cycle in the wavelet scalogram
# from the XRF record of the Bisciaro data set of Arts (2014)

Bisciaro_al <- Bisciaro_XRF[, c(1, 61)]
Bisciaro_al <- astrochron::sortNave(Bisciaro_al,verbose=FALSE,genplot=FALSE)
Bisciaro_al <- astrochron::linterp(Bisciaro_al, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_al <- Bisciaro_al[Bisciaro_al[, 1] > 2, ]

Bisciaro_al_wt <-
 analyse_wavelet(
   data = Bisciaro_al,
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_al_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
```

```
#     wavelet = Bisciaro_al_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_al_wt_track <- completed_series(
#   wavelet = Bisciaro_al_wt,
#   tracked_curve = Bisciaro_al_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_al_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_al_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_ca <- Bisciaro_XRF[, c(1, 55)]
Bisciaro_ca <- astrochron::sortNave(Bisciaro_ca,verbose=FALSE,genplot=FALSE)
Bisciaro_ca <- astrochron::linterp(Bisciaro_ca, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_ca <- Bisciaro_ca[Bisciaro_ca[, 1] > 2, ]

Bisciaro_ca_wt <-
 analyse_wavelet(
   data = Bisciaro_ca,
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_ca_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_ca_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
# Bisciaro_ca_wt_track <- completed_series(
#   wavelet = Bisciaro_ca_wt,
#   tracked_curve = Bisciaro_ca_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
```

```
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_ca_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_ca_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE)

Bisciaro_sial <- Bisciaro_XRF[,c(1,64)]
Bisciaro_sial <- astrochron::sortNave(Bisciaro_sial,verbose=FALSE,genplot=FALSE)
Bisciaro_sial <- astrochron::linterp(Bisciaro_sial, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_sial <- Bisciaro_sial[Bisciaro_sial[, 1] > 2, ]

Bisciaro_sial_wt <-
 analyse_wavelet(
   data = Bisciaro_sial,
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_sial_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_sial_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_sial_wt_track <- completed_series(
#   wavelet = Bisciaro_sial_wt,
#   tracked_curve = Bisciaro_sial_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_sial_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_sial_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
```

```
#   )


Bisciaro_Mn <- Bisciaro_XRF[,c(1,46)]
Bisciaro_Mn <- astrochron::sortNave(Bisciaro_Mn,verbose=FALSE,genplot=FALSE)
Bisciaro_Mn <- astrochron::linterp(Bisciaro_Mn, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_Mn <- Bisciaro_Mn[Bisciaro_Mn[, 1] > 2, ]

Bisciaro_Mn_wt <-
 analyse_wavelet(
   data = Bisciaro_Mn,
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_Mn_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mn_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mn_wt_track <- completed_series(
#   wavelet = Bisciaro_Mn_wt,
#   tracked_curve = Bisciaro_Mn_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
# Bisciaro_Mn_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mn_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE
#   )

Bisciaro_Mg <- Bisciaro_XRF[,c(1,71)]
Bisciaro_Mg <- astrochron::sortNave(Bisciaro_Mg,verbose=FALSE,genplot=FALSE)
Bisciaro_Mg <- astrochron::linterp(Bisciaro_Mg, dt = 0.01,verbose=FALSE,genplot=FALSE)
Bisciaro_Mg <- Bisciaro_Mg[Bisciaro_Mg[, 1] > 2, ]

Bisciaro_Mg_wt <-
 analyse_wavelet(
   data = Bisciaro_Mg,
```

```
   dj = 1 /50 ,
   lowerPeriod = 0.01,
   upperPeriod = 50,
   verbose = FALSE,
   omega_nr = 8
 )

# Bisciaro_Mg_wt_track <-
#   track_period_wavelet(
#     astro_cycle = 110,
#     wavelet = Bisciaro_Mg_wt,
#     n.levels = 100,
#     periodlab = "Period (metres)",
#     x_lab = "depth (metres)"
#   )
#
#
# Bisciaro_Mg_wt_track <- completed_series(
#   wavelet = Bisciaro_Mg_wt,
#   tracked_curve = Bisciaro_Mg_wt_track,
#   period_up = 1.2,
#   period_down = 0.8,
#   extrapolate = TRUE,
#   genplot = FALSE,
#   keep_editable = FALSE
# )
#
# Bisciaro_Mg_wt_track <-
#   loess_auto(
#     time_series = Bisciaro_Mg_wt_track,
#     genplot = FALSE,
#     print_span = FALSE,
#     keep_editable = FALSE)



wt_list_bisc <- list(Bisciaro_al_wt,
               Bisciaro_ca_wt,
               Bisciaro_sial_wt,
               Bisciaro_Mn_wt,
               Bisciaro_Mg_wt)

#Instead of tracking, the tracked solution data sets Bisciaro_al_wt_track,
#Bisciaro_ca_wt_track, Bisciaro_sial_wt_track, Bisciaro_Mn_wt_track,
# Bisciaro_Mn_wt_track and Bisciaro_Mg_wt_track are used

data_track_bisc <- cbind(Bisciaro_al_wt_track[,2],
                     Bisciaro_ca_wt_track[,2],
                     Bisciaro_sial_wt_track[,2],
                     Bisciaro_Mn_wt_track[,2],
                     Bisciaro_Mg_wt_track[,2])
```

```
x_axis_bisc <- Bisciaro_al_wt_track[,1]


bisc_retrack <- retrack_wt_MC(wt_list = wt_list_bisc,
             data_track = data_track_bisc,
             x_axis = x_axis_bisc,
             nr_simulations = 20,
             seed_nr = 1337,
             verbose = FALSE,
             genplot = FALSE,
             keep_editable = FALSE,
             create_GIF = FALSE,
             plot_GIF = FALSE,
             width_plt =  600,
             height_plt = 450,
            period_up  =  1.5,
             period_down = 0.5,
             plot.COI = TRUE,
             n.levels = 100,
             palette_name = "rainbow",
             color_brewer = "grDevices",
             periodlab = "Period (metres)",
             x_lab = "depth (metres)",
             add_avg = FALSE,
             time_dir = TRUE,
             file_name = NULL,
             run_multicore = FALSE,
             output = 1,
             n_imgs = 50,
             plot_horizontal = TRUE,
             empty_folder = FALSE)
```

`rho1_evolutive` *Evolutionary lag-1 autocorrelation coefficient (rho1)*

Description

The `rho1_evolutive` function calculates the lag-1 autocorrelation coefficient of an AR(1) model in moving windows, as used in the sedimentary noise model of Li et al. (2018a,b). With `n_sim = 0` a single deterministic run is made (`acycleEvolutionaryRho1.m`): the series is linearly interpolated on a grid anchored at the first coordinate with interval `sample_min`, windows of `round(window_min/sample_min)` points are linearly detrended and rho1 = sum(z[t] z[t-1]) / sum(z[t-1]^2) of the demeaned residuals z is reported at the midpoint of each window. With `n_sim > 0` the Acycle Monte Carlo is run (AC.m menu "Lag-1 autocorrelation coefficient (rho1)"): for every simulation a sampling interval and a window are drawn uniformly from [`sample_min`, `sample_max`] and [`window_min`, `window_max`], the data are linearly interpolated (`interpolate.m`), evolutionary rho1 is computed with `erhoAR1.m`

(window of round(window/dt) points, linear detrending, rhoAR1ML.m), window centers are optionally shifted by a random offset uniform in [-window/2, window/2], and the curves are linearly interpolated on a regular grid and summarized by the percentiles 2.5, 5, 10, 15.865, 25, 50, 75, 84.135, 90, 95 and 97.5. Translated from Acycle v3.1 acycleEvolutionaryRho1.m, acycleInterpolateSeries.m, erhoAR1.m, rhoAR1ML.m, interpolate.m and AC.m (menu_rho_Callback) (Mingsong Li; rhoAR1 by Dorothee Husson).

Usage

```
rho1_evolutive(
  data = NULL,
  window_min = NULL,
  window_max = NULL,
  sample_min = NULL,
  sample_max = NULL,
  n_sim = 1000,
  n_out = NULL,
  shift_grid = TRUE,
  seed = NULL,
  verbose = FALSE,
  genplot = FALSE
)
```

Arguments

data	Matrix with two columns: depth/time and proxy value (sorted, non-finite rows removed). Default=NULL
window_min	Window length (single run) or minimum window length (Monte Carlo). If NULL: single run min(400, 0.3 * data length); Monte Carlo 400 if 0.3 * data length > 400, otherwise 0.3 * data length. Default=NULL
window_max	Maximum window length (Monte Carlo). If NULL: 500 if 0.3 * data length > 400, otherwise 0.4 * data length. Default=NULL
sample_min	Sampling interval (single run) or minimum sampling interval (Monte Carlo). If NULL the 95th percentile of the data sampling intervals. Default=NULL
sample_max	Maximum sampling interval (Monte Carlo). If NULL twice the 95th percentile of the data sampling intervals. Default=NULL
n_sim	Number of Monte Carlo simulations (at least 50, as in Acycle); 0 gives a single deterministic run. Default=1000
n_out	Number of grid points of the Monte Carlo output. If NULL min(1000, number of data points). Default=NULL
shift_grid	Randomly shift the window centers of every simulation (Acycle "shift grids" > 1). Default=TRUE
seed	Seed for the random number generator (NULL = not set). Default=NULL
verbose	Show a progress bar. Default=FALSE
genplot	Plot the result. Default=FALSE

Value

For `n_sim = 0` a list with `rho1` (matrix [window center, rho1]), `window_points`, `sample_interval` and `window`. For the Monte Carlo a list with `percentile` (matrix [x, 11 percentiles] on the grid points where they are defined), `median` (matrix [x, median]), `percentiles`, `grid`, `rho1` (n_out x n_sim matrix of all realisations) and `draws` (sampling intervals, windows, shifts).

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li (Acycle), rhoAR1 by Dorothee Husson.

References

Li, M., Hinnov, L.A., Huang, C., Ogg, J.G., 2018a. Sedimentary noise and sea levels linked to land-ocean water exchange and obliquity forcing. Nature Communications 9, 1004. doi:10.1038/s4146701803454y

Li, M., Huang, C., Hinnov, L., Chen, W., Ogg, J., Tian, W., 2018b. Astrochronology of the Anisian stage (Middle Triassic) at the Guandao reference section, South China. Earth and Planetary Science Letters 482, 591-606. doi:10.1016/j.epsl.2017.11.042

Mudelsee, M., 2002. TAUEST: a computer program for estimating persistence in unevenly spaced weather/climate time series. Computers & Geosciences 28, 69-72.

Examples

```
set.seed(3)
t <- cumsum(runif(800, 0.5, 1.5))
x <- as.numeric(arima.sim(list(ar = 0.8), n = 800))
single <- rho1_evolutive(data = cbind(t, x), n_sim = 0)
mc <- rho1_evolutive(data = cbind(t, x), n_sim = 50, seed = 1,
  genplot = TRUE)
```

---

sedrate2tune *Use a sedimentation curve to convert data to the time domain*

Description

Convert a proxy record from the depth to time domain using a sedimentation rate curve

Usage

```
sedrate2tune(
  data = NULL,
  sed_curve = NULL,
  genplot = FALSE,
  keep_editable = FALSE
)
```

Arguments

data	Input should be a matrix of 2 columns with first column being depth and the second column is a proxy value
sed_curve	Input should be a matrix of 2 columns with first column being depth and the second column is the sedimentation rate is cm/kyr
genplot	Generates a plot of the proxy record in the time domain Default=FALSE.
keep_editable	Keep option to add extra features after plotting Default=FALSE

Value

The output is a matrix with 2 columns. The first column is time The second column is the proxy value If genplot=TRUE then a time vs proxy value plot will be plotted.

Author(s)

Part of the code is based on the sedrate2time function of the 'astrochron' R package

References

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

Examples

```
# Extract the 405kyr eccentricity cycle from the wavelet scalogram
# from the magnetic susceptibility record of the Sullivan core
# of Pas et al., (2018) and then create a age model using minimal tuning
# (e.g.) set the distance between peaks to 405 kyr. The age model
# (sedimentation rate curve) is then used to convert the data
# from the depth to the time domain

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


mag_405 <- extract_signal_stable_V2(
 wavelet = mag_wt,
 period_max = 4,
 period_min = 2,
 add_mean = TRUE,
 plot_residual = FALSE,
 keep_editable = FALSE
)

mag_405_min_tuning <- minimal_tuning(data = mag_405,
pts = 5,
```

```
cycle = 405,
tune_opt = "max",
output = 1,
genplot = FALSE,
keep_editable = FALSE)

mag_time <- sedrate2tune(
data=mag,
sed_curve=mag_405_min_tuning,
genplot=FALSE,
keep_editable=FALSE)
```

`spectral_moments` *Sliding-window spectral moments (mean frequency and bandwidth)*

Description

The `spectral_moments` function calculates the first two spectral moments, the mean frequency (μ_f) and the bandwidth (B), of a uniformly sampled series in a sliding window (Sinnesael et al., 2018). Because both moments scale with the sedimentation rate, their evolution along a stratigraphic series can be used to detect changes in sedimentation rate.

Within every window the (globally mean-centred) segment is linearly detrended, its one-sided rectangular-window periodogram is computed on an even NFFT grid from DC up to the Nyquist frequency and

$$\mu_f = \sum f P(f) / \sum P(f), \quad B = \sqrt{\sum (f - \mu_f)^2 P(f) / \sum P(f)}$$

are calculated.

Two computational paths are available, mirroring the Acycle GUI (SpectralMomentsGUI.m):

- the strict deterministic core (default; used whenever `sr_mean=NULL` and `edge_padding` is `"none"` or `"zero"`). Translated from Acycle v3.1 acycleSpectralMoments.m (Mingsong Li). The window length is rounded (half away from zero) to an integer number of sampling intervals, the step to an integer number of samples, NFFT = pad x window samples (+1 if odd), and window centres are the midpoints of the actual window coordinates.
- the legacy path (used when `sr_mean` is given or when `edge_padding` is `"mirror"` or `"mean"`). Translated from Acycle v3.1 spectralmoments.m and zeropad2.m (Mingsong Li, based on SpectralMoments_with_SR_estimation.m and DetrendSignalPP.m by Matthias Sinnesael). When `sr_mean` is given the long-term bandwidth trend is estimated with the overlap-add polynomial baseline model of DetrendSignalPP.m (Acycle's 'poly' option) and converted into a relative sedimentation-rate curve scaled to `sr_mean` ($sr = sr_{mean} B_1 / B_{trend}$, re-centred on `sr_mean`). As in Acycle, the trend window is `round(window/step_samples)` bandwidth values, i.e. `window` expressed in the unit of the depth column is used as a number of samples, so the smoothing depends on the depth unit.

Usage

```
spectral_moments(
  data = NULL,
  window = NULL,
  step = NULL,
  pad = 5,
  edge_padding = "none",
  sr_mean = NULL,
  freq_range = NULL,
  noise_floor = FALSE,
  anchor = NULL,
  moment = c("bandwidth", "mean_frequency"),
  trend = c("poly", "lowess"),
  trend_window = NULL,
  trend_order = 2,
  genplot = FALSE,
  verbose = FALSE
)
```

Arguments

data	Input data set, a matrix with 2 columns: first column depth or time (strictly increasing, uniformly sampled), second column the proxy.
window	Length of the running window in the unit of the first column. Default=NULL which equals 25% of the record length (the Acycle GUI default).
step	Step of the running window in the unit of the first column. Default=NULL which equals the sampling interval.
pad	Zero-padding factor: NFFT is pad times the number of samples in a window. Default=5
edge_padding	Padding of the record edges by half a window: "none", "zero" (strict core), or "mirror", "mean" (legacy path, as in the Acycle GUI). Default="none"
sr_mean	Mean sedimentation rate (e.g. cm/kyr). When given the legacy path is used and the bandwidth trend and a sedimentation-rate curve are returned. Default=NULL
freq_range	Frequency band (cycles per depth unit) over which the moments are taken; by default the whole spectrum up to the Nyquist frequency, as in the paper. The moments of a record with white noise are biased towards the cutoff frequency (section 2.3.3 of the paper), so a band around the astronomical lines makes them follow the sedimentation rate rather than the noise. Default=NULL
noise_floor	Subtract the median of every window's periodogram (an estimate of the white-noise floor) before the moments are taken Default=FALSE. In the synthetic test of the examples (precession, obliquity, eccentricity, white noise 0.3 times the signal, rate varying 1-3 cm/kyr, Nyquist 25 cycles/m) the anchored bandwidth trend correlates 0.00 with the true rate over the whole spectrum (the noise plateau up to the Nyquist frequency dominates the second moment), 0.97 over freq_range = c(0.1, 8), and 0.99 with the floor removed as well (91 % of the record within 20 %); the mean frequency is less sensitive (0.98 over the whole spectrum).

anchor	The sedimentation rate of Sinnesael et al. (2018): the smoothed trend of the bandwidth (or of the mean frequency, see moment) is anchored at one depth where an astronomical component is identified, since every spatial frequency, and so every moment, scales with 1 / sedimentation rate (their equation 12): c(depth, period_kyr, frequency) (the component with period period_kyr sits at frequency cycles per depth unit at depth, so the rate there is 1 / (period_kyr * frequency)), or list(depth = , rate =) with the rate in depth units per kyr. The result then holds moment_trend and sedrate_anchored. This is the procedure of the paper (U1437: the 0.45 cycle/m precession line read as 21 kyr; ODP 982 and the synthetic insolation: the obliquity line at the top of the record), as opposed to the Acycle sr_mean path, which rescales by the first raw bandwidth value and re-centres the curve additively on a mean rate. Default=NULL
moment	Which moment carries the sedimentation rate: "bandwidth" (the paper's choice, less sensitive to the noise bias than the mean frequency) or "mean_frequency" Default="bandwidth".
trend	Smoothing of the moment before anchoring: "poly" (the overlap-add polynomial baseline of Verhelst 2000 used in the paper) or "lowess" Default="poly".
trend_window	Length (depth units) of the smoothing window of the trend; by default three analysis windows. The paper takes a long-term trend: the result is the first-order change of the rate, not its detail.
trend_order	Polynomial order of the baseline Default=2.
genplot	Generate a plot of the result (plot_spectral_moments) Default=FALSE
verbose	Print a progress bar Default=FALSE

Value

A list with elements

- result: matrix with columns depth (window centre), mean_frequency, bandwidth and, when sr_mean was given, bandwidth_trend and sedrate.
- method: "strict" or "legacy".
- meta: list with the effective settings (sampling interval, window/step in samples and physical units, NFFT, padding, Nyquist, number of windows).
- with anchor: result also holds moment_trend and sedrate_anchored, and anchor records the anchor used.
- spectra: the periodogram of every window (frequency, depth of the window centres, power matrix frequency x window), the evolutive spectrum that plot_spectral_moments draws as an image with the two moments on top.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li (Acycle), Matthias Sinnesael (spectral moments and DetrendSignalPP), Milan Zivanovic.

References

Sinnesael, M., Zivanovic, M., De Vleeschouwer, D., Claeys, P. (2018). Spectral moments in cyclostratigraphy: advantages and disadvantages compared to more classic approaches. Paleoceanography and Paleoclimatology, 33, 493-510. <doi:10.1029/2017PA003293>

Li, M., Hinnov, L., Kump, L. (2019). Acycle: Time-series analysis software for paleoclimate research and education. Computers & Geosciences, 127, 12-22. <doi:10.1016/j.cageo.2019.02.011>

Examples

```
# Two cycles (40 and 10 cm period in "age" units) recorded with a
# sedimentation rate that doubles halfway the record (depth in cm): the
# mean frequency halves in the upper part.
dt <- 0.5
depth <- seq(0, 1000, by = dt)
age <- ifelse(depth <= 500, depth, 500 + (depth - 500) / 2)
set.seed(1)
proxy <- sin(2 * pi * age / 40) + 0.5 * sin(2 * pi * age / 10) +
  stats::rnorm(length(depth), sd = 0.2)
sm <- spectral_moments(data = cbind(depth, proxy), window = 200,
                       step = 5, pad = 5, genplot = FALSE)
plot_spectral_moments(sm, x_lab = "depth (cm)")

# with a sedimentation-rate estimate (legacy path, 'poly' trend)
sm_sr <- spectral_moments(data = cbind(depth, proxy), window = 200,
                          step = 5, pad = 5, sr_mean = 1.5)
plot_spectral_moments(sm_sr, x_lab = "depth (cm)")
# the sedimentation rate of Sinnesael et al. (2018): the obliquity line is identified at
# 5 m (41 kyr at f_obl cycles/m), the bandwidth trend is anchored there; moments over a band
set.seed(3); t <- seq(0, 3000, 0.5)
x <- 0.8 * sin(2 * pi * t / 41) + sin(2 * pi * t / 23.7) + 0.5 * sin(2 * pi * t / 405) +
  0.3 * rnorm(length(t))
rate <- 0.02 + 0.01 * sin(2 * pi * t / 1500)
z <- cumsum(c(0, rate[-1] * diff(t))); zg <- seq(0, max(z), 0.02)
d <- data.frame(zg, approx(z, x, zg)$y)
f_obl <- 1 / (41 * approx(z, rate, 5)$y)     # the obliquity line at 5 m depth
sm <- spectral_moments(d, window = 5, step = 0.1, anchor = c(5, 41, f_obl),
                       freq_range = c(0.1, 8), noise_floor = TRUE)
plot_spectral_moments(sm, freq_max = 8)
```

`sst_significance` *Significance of synchrosqueezed (and other) scalograms against autocorrelated noise*

Description

Tests which parts of a synchrosqueezed transform (`analyse_wavelet_sst`, `analyse_superlet_sst`, `analyse_conceft`) are unlikely to be produced by noise. The noise of the record is modelled by an

autoregressive process (order chosen by AIC, ar_order_max); optionally its variance may change along the record (a moving standard deviation of the innovations, as in the time-varying AR bootstrap of Chen and Wu, 2025). n_sim noise records are simulated and transformed with exactly the same settings as the original. For every period the (1 - alpha) quantile of the noise power, pooled over all positions, gives the threshold; with local_variance = TRUE it is scaled by the local noise variance. Coefficients below the threshold are set to zero in the returned object, so only significant ridges remain in plots and tracking.

Under the null hypothesis (no cycles) the record itself is used as the noise model; this is conservative when strong cycles are present, because they increase the variance attributed to noise.

Usage

```
sst_significance(
  wavelet,
  n_sim = 50,
  alpha = 0.05,
  ar_order_max = 5,
  local_variance = FALSE,
  window = NULL,
  seed = 1,
  verbose = FALSE
)
```

Arguments

wavelet	Object made by analyse_wavelet_sst, analyse_superlet_sst or analyse_conceft.
n_sim	Number of simulated noise records Default=50.
alpha	Significance level Default=0.05.
ar_order_max	Maximum order of the AR noise model Default=5; 1 gives the classical red noise (AR1).
local_variance	Scale the threshold with the local variance of the noise Default=FALSE.
window	Length (in depth/time units) of the moving window of the local variance Default=NULL: twice the longest period of the transform.
seed	Seed of the simulations Default=1.
verbose	Print messages.

Value

The input object with Power, Wave, Ampl and Power.avg restricted to the significant coefficients, and the extra elements significant (logical matrix), Power_all (the unmasked power), threshold_period (noise power threshold per period) and alpha.

References

Chen, Z., Wu, H.-T., 2025. Uncertainty quantification of synchrosqueezing transform under complicated nonstationary noise. arXiv:2506.00779.

Torrence, C., Compo, G.P., 1998. A practical guide to wavelet analysis. Bulletin of the American Meteorological Society 79, 61-78.

Examples

```
z <- 0:999
x <- sin(2 * pi * z / 40) + as.numeric(arima.sim(list(ar = 0.7), 1000))
s <- analyse_wavelet_sst(data.frame(z, x), lowerPeriod = 10, upperPeriod = 200, dj = 1 / 40)
sig <- sst_significance(s, n_sim = 20)
plot_wavelet(sig)
```

`sum_power_sedrate` *Calculate sum of maximum spectral power for sedimentation rates 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 normalised 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",
  run_multicore = FALSE,
  n_cores = NULL,
  genplot = FALSE,
  plot_res = 1,
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  verbose = FALSE,
  colour_brewer = NULL
)
```

Arguments

`red_noise`	Red noise curves generated using the `model_red_noise_wt` function
`wavelet`	Wavelet object created using the `analyse_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 normalised 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"
run_multicore	run simulation using multiple cores Default=FALSE the simulation is run at x-2 cores to allow the 2 remaining processes to run background processes
n_cores	Number of cores used when 'run_multicore = TRUE' ('Default = NULL': an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set 'options(waverider.n_cores = ...)' to change the default. A value > 1 also switches on parallel computing. Results do not depend on the number of cores.
genplot	Generate plot Default="FALSE"
plot_res	plot options are 1: sum max power or 2: nr of components Default=2
keep_editable	Keep option to add extra features after plotting Default=FALSE
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
verbose	Print text Default=FALSE.
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

Returns a list which contains 4 elements element 1: sum of maximum spectral power element 2: number of cycles used in the sum of maximum spectral power element 3: y-axis values of the matrices which is sedimentation rate element 4: x-axis values of the matrices which is depth

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 or nr of cycles 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.

Author(s)

Based on the asm and eAsm functions of the 'astrochron' R package and the 'eCOCO' and 'COCO' functions of the 'Acycle' software

References

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

Acycle: Time-series analysis software for paleoclimate research and education, Mingsong Li, Linda Hinnov, Lee Kump, Computers & Geosciences,Volume 127,2019,Pages 12-22,ISSN 0098-3004, <doi:10.1016/j.cageo.2019.02.011>

Tracking variable sedimentation rates and astronomical forcing in Phanerozoic paleoclimate proxy series with evolutionary correlation coefficients and hypothesis testing, Mingsong Li, Lee R. Kump, Linda A. Hinnov, Michael E. Mann, Earth and Planetary Science Letters,Volume 501, T2018,Pages 165-179,ISSN 0012-821X,<doi:10.1016/j.epsl.2018.08.041>

Examples

```
#estimate sedimentation rate for the the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018).

mag_wt <- analyse_wavelet(data = mag[1:200, ],
dj = 1/10,
lowerPeriod = 0.1,
upperPeriod = 32,
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 = 3,
run_multicore=FALSE,
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 = 1,
cycles = c(2376,1600,1180,696,406,110),
x_lab="depth",
y_lab="sedrate",
run_multicore=FALSE,
genplot = FALSE,
plot_res=1,
keep_editable=FALSE,
palette_name = "rainbow",
color_brewer="grDevices",
verbose=FALSE)
```

---

swa_background *Smoothed Window Average (SWA) spectral background*

Description

Estimates the background of a power spectrum with the Smoothed Window Average (SWA) method of Kemp (2016), as refined by Weedon (2020, 2022). The log10 power spectrum is averaged in non-overlapping windows of odd width (11 to 99 ordinates, always keeping at least three complete windows). The window averages are linearly interpolated, with two alternative extrapolations towards zero frequency. Among all candidate widths/extrapolations the fit with the fewest non-decreasing fitted steps (and, in case of a tie, the smallest RMSE) is selected. A constant (white-noise) and a quadratic high-frequency tail are then tested, and finally the fit is repeatedly Hanning smoothed as long as the RMSE degrades by less than 1%. Translated from Acycle v3.1 specswa.m (Mingsong Li, after the FORTRAN code of Graham Weedon).

Usage

```
swa_background(freq = NULL, log_power = NULL, n_data = NULL, verbose = FALSE)
```

Arguments

freq	Strictly increasing frequency vector (at least 33 values).
log_power	log10 power values belonging to freq.
n_data	Number of data points of the analysed series. Retained for compatibility with Acycle; it does not influence the fitted background.
verbose	Print fit diagnostics. Default=FALSE

Value

A list with elements swa (SWA background in linear power units, same length as freq), n_windows (selected window width in ordinates; second output of specswa.m) and diagnostics (list with initial_variant, initial_nondecreasing_count, initial_rmse, selected_window, tail_model ("extrapolated", "constant" or "quadratic"), n_smoothing, final_rmse, n_data, n_frequency).

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li, after the SWA FORTRAN program of Graham Weedon.

References

Kemp, D.B., 2016. Optimizing significance testing of astronomical forcing in cyclostratigraphy. Paleoceanography 31, 1516-1531.

Weedon, G.P., 2020. Spectral background estimation for cyclostratigraphy (SWA). In: Weedon, G.P. 2022, Problems with the current practice of spectral analysis in cyclostratigraphy: Avoiding false detection of regular cyclicity. Earth-Science Reviews 235, 104261.

Examples

```
set.seed(1)
n <- 512
x <- stats::filter(stats::rnorm(n), 0.7, method = "recursive")
x <- as.numeric(x) + sin(2 * pi * (1:n) / 20)
sp <- stats::spec.pgram(x, taper = 0, detrend = FALSE, plot = FALSE)
bg <- swa_background(freq = sp$freq, log_power = log10(sp$spec),
  n_data = n)
plot(sp$freq, sp$spec, type = "l", log = "y")
lines(sp$freq, bg$swa, col = "red", lwd = 2)
```

---

swa_spectrum *Power spectrum with SWA background, chi-square and FDR confidence levels*

Description

Calculates a power spectrum, locates the spectral background with the Smoothed Window Average (SWA) method (Kemp, 2016; Weedon, 2020, 2022) and derives chi-square confidence levels and False Discovery Rate (FDR) confidence levels (Benjamini and Hochberg, 1995) following the procedure of Miller et al. (2001), with the correlation of neighbouring spectral estimates accounted for as in Hopkins et al. (2002).

Two spectral estimators are available:

`method = "mtm"` Multi-taper spectrum of the (evenly spaced, zero-mean) series, SWA background from `swa_background`. Degrees of freedom = 2K (K = number of tapers). This is the SWA option of the Acycle spectral analysis window. Translated from Acycle v3.1 spectralswafdr.m (Mingsong Li).

`method = "lomb"` The program of Weedon (2020, 2022) used by the Acycle SWA window: lag-1 check, linear detrending, 5% cosine taper, Lomb-Scargle transform (routine PERIOD of Press et al., 1992; allows unevenly spaced data or data with gaps), three passes of a Hanning spectral window (8 degrees of freedom), SWA background, chi-square and FDR confidence levels. Translated from Acycle v3.1 specswafdr.m (Graham Weedon, Mingsong Li).

Usage

```
swa_spectrum(
  data = NULL,
  method = "mtm",
  nw = 2,
  pad_times = 1,
  background = "specswa",
  genplot = FALSE,
  verbose = FALSE
)
```

Arguments

data	Matrix or data frame with two columns: depth/time and proxy value. Rows are sorted on the first column and non-finite rows removed. For method = "mtm" the data must be evenly spaced; as in Acycle the series is used as is (remove the mean/trend beforehand).
method	"mtm" or "lomb" (see Description). Default="mtm"
nw	Time-bandwidth product of the multi-taper spectrum (method = "mtm" only). Default=2
pad_times	Zero-padding factor; nfft = max(n, round(pad_times * n)) (method = "mtm" only). Default=1
background	Background fit used for method = "lomb": "specswa" uses swa_background (current specswa.m algorithm, also used by method = "mtm"); "legacy" reproduces the older SWA fit coded inside specswafdr.m, i.e. exactly what the Acycle SWA window reports. Default="specswa"
genplot	Plot the spectrum, background and confidence levels. Default=FALSE
verbose	Print progress information. Default=FALSE

Value

A list with elements

freq Frequencies.

power Spectral power used for testing (MTM PSD, or the Hanning-smoothed Lomb-Scargle spectrum normalised by the variance).

periodogram ("lomb" only) raw normalised Lomb-Scargle periodogram before Hanning smoothing.

swa SWA background.

chi2_factor Multiplication factors of the background for the 90, 95, 99, 99.9 and 99.99% chi-square confidence levels.

chi2_cl Matrix with the chi-square confidence levels (columns chi90, chi95, chi99, chi999, chi9999).

fdr_cl Matrix with the FDR confidence levels (columns fdr10, fdr5, fdr1, fdr01, fdr001 for 10, 5, 1, 0.1 and 0.01% FDR). As in Acycle a column is NA when the level is not reached or does not differ from the next less strict level; the 10% column is always NA in Acycle (kept for compatibility).

fdr Data frame per FDR level: `n_rejected` (number of frequencies declared significant, -1 = none), `factor` (multiplication factor of the background), `chi2_cl_percent` (equivalent chi-square confidence level) and `drawn` (whether the level is present in `fdr_cl`).

alpha Probability (p-value on the Acycle grid) of every spectral estimate.

variance_ratio power / swa.

dof Degrees of freedom.

cn Hopkins et al. (2002) correction constant.

bandwidth, rayleigh Resolution bandwidth and Rayleigh frequency.

n_windows, diagnostics Selected SWA window width and fit diagnostics.

rho1 (`"lomb"` only) lag-1 autocorrelation of the input.

irregular (`"lomb"` only) TRUE if the spacing varies by more than 5%.

Author(s)

Michiel Arts (R translation); original MATLAB code: Mingsong Li, translated from the FORTRAN programs of Graham P. Weedon.

References

Benjamini, Y., Hochberg, Y., 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society B 57, 289-300.

Hopkins, A.M., Miller, C.J., Connolly, A.J., Genovese, C., Nichol, R.C., Wasserman, L., 2002. A new source detection algorithm using the false-discovery rate. The Astronomical Journal 123, 1086-1094.

Kemp, D.B., 2016. Optimizing significance testing of astronomical forcing in cyclostratigraphy. Paleoceanography 31, 1516-1531.

Miller, C.J., Genovese, C., Nichol, R.C., Wasserman, L., Connolly, A., Reichart, D., Hopkins, A., Schneider, J., Moore, A., 2001. Controlling the false-discovery rate in astrophysical data analysis. The Astronomical Journal 122, 3492-3505.

Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. Numerical Recipes in FORTRAN, 2nd ed. Cambridge University Press.

Weedon, G.P., 2020. Confirmed detection of Palaeogene and Jurassic orbitally-forced sedimentary cycles in the depth domain using False Discovery Rates and Bayesian probability spectra. Boletin Geologico y Minero 131, 207-230.

Weedon, G.P., 2022. Problems with the current practice of spectral analysis in cyclostratigraphy: avoiding false detection of regular cyclicity. Earth-Science Reviews 235, 104261.

Examples

```
set.seed(1)
n <- 600
t <- (0:(n - 1)) * 0.5
noise <- as.numeric(stats::filter(stats::rnorm(n), 0.7,
  method = "recursive"))
x <- noise + 1.5 * sin(2 * pi * t / 20) + sin(2 * pi * t / 7.3)
```

```
dat <- cbind(t, x - mean(x))
res_mtm <- swa_spectrum(data = dat, method = "mtm", nw = 2,
  genplot = TRUE)
res_lomb <- swa_spectrum(data = dat, method = "lomb", genplot = TRUE)
res_lomb$fdr
```

`term_sensitivity` *Sensitivity of every cycle to each fundamental frequency*

Description

Sensitivity of every cycle to each fundamental frequency

Usage

```
term_sensitivity(
  age_ma = 0,
  delta = 0.01,
  k_yr = NULL,
  fundamentals = c("g1", "g2", "g3", "g4", "g5", "s1", "s3", "s4", "s6", "k"),
  data_dir = NULL
)
```

Arguments

`age_ma`	Age(s) in Ma.
`delta`	fractional change applied to each fundamental (0.01 = +1%)
`k_yr`	Precession constant in arcsec/yr ('NULL' = taken from 'astrogeo22(age_ma)$p_arcsec').
`fundamentals`	Fundamental frequencies to vary.
`data_dir`	Directory with the data files ('NULL' = the data installed with the package).

Value

matrix of percent change in each term's period (rows) per 'delta' change in each fundamental (columns), including the precession constant k

`test_tuned` *Precession-constant test for a tuned record*

Description

Tests a tuned (time-domain) record with the precession constant k. Significant multi-taper (MTM) spectral peaks of the record, optionally pooled with other proxies on the same time scale, are assigned to the obliquity (k+s3, k+s6) and precession (k+g5, k+g2, k+g4) terms. Every combination of candidate peaks is tried over a wide range of k (no window is centred on a reference value), and the combinations in which all terms give one consistent k are kept. Because the terms share one k, a correct assignment gives the same k from every term. The result is compared with AstroGeo22 (Farhat et al. 2022) and a Devonian range, and combined with two further checks of the tuned record: the spectral peaks of the eccentricity, obliquity and precession terms, and the amplitude modulation (405 kyr from the 100 kyr envelope, 100 kyr from the precession envelope) against phase-randomised surrogates. This is the updated version of the test (version 3), which works better for Silurian/Devonian data than a search around the AstroGeo22 value.

Usage

```
test_tuned(
  URAN_time,
  others = NULL,
  tbw = 2,
  pad = 5,
  siglevel = 0.9,
  ftest = FALSE,
  tol = 0.04,
  scan = TRUE,
  scan_band = c(15, 23),
  scan_win = 200,
  scan_thr = 0.5,
  scan_min = 100,
  n_surr = 200,
  seed = 42,
  plot = TRUE,
  verbose = FALSE,
  age_ma = 386,
  k_ref = NULL,
  k_ref_range = NULL,
  k_dev = c(20.4, 21.6),
  k_search = c(19.5, 23.5),
  constraints = NULL
)
```

Arguments

`URAN_time`	data.frame, time in kyr and value; e.g. the output of a tuning function.

`others` optional list of other tuned proxies on the same time scale; their peaks are pooled with those of `URAN_time` (a term found in several proxies adds support).

`tbw`, `pad` MTM settings of `astrochron::mtm`: time-bandwidth product and zero-padding factor. `tbw` sets the resolution: the precession terms k+g5 and k+g2 only separate when `tbw <= (record length in kyr) / 1600` (about 3 for 5.5 Myr). A larger `tbw` merges them into one broad peak and k cannot be measured (a warning is printed).

`siglevel` significance level for peaks (AR1 confidence level at the peak).

`ftest` if TRUE also require the harmonic F-test confidence level to reach `siglevel`. The default FALSE uses the AR1 level only, because the F-test tests for a pure sinusoid and rejects broad, strong peaks of a tuned record.

`tol` half-width of the search window around each predicted period (fraction of the period).

`scan` if TRUE intervals with little precession imprint are removed (set to zero) before the spectrum is built. `scan_band` precession band (kyr), `scan_win` running window (kyr), `scan_thr` an interval is weak when its precession-band variance is below `scan_thr` times its median, `scan_min` shortest kept or removed piece (kyr).

`scan_band`, `scan_win`, `scan_thr`, `scan_min` see `scan`.

`n_surr` number of phase-randomised surrogates for the amplitude-modulation test.

`seed` random seed.

`plot` draw the comparison figure with `plot_test_k`.

`verbose` also print the peak tables and all peak assignments.

`age_ma` age of the record (Ma); the AstroGeo22 reference k and its envelope are taken at this age unless `k_ref` and `k_ref_range` are given.

`k_ref`, `k_ref_range` reference precession constant and its range (kyr); default AstroGeo22 at `age_ma`.

`k_dev` plausible range of k (kyr) for the test verdict (default: the Devonian range).

`k_search` range of k (kyr) in which peaks are searched, independent of AstroGeo22. The search window of each term runs from its period at `k_search[1]` to its period at `k_search[2]` (plus or minus `tol`). NULL gives the old windows around the AstroGeo22 period, which can only return k within about 1 kyr of AstroGeo22 and so cannot test it.

`constraints` optional path to a CSV of published constraints (columns set, observable, age_ma, a_RE, a_sd, lod_h, lod_sd, k_arcsec, k_sd) drawn in panels E-G of the figure.

Value

Invisibly a list with the combined k (`k`, `k_obliquity`, `k_precession`, `k_ranges`), the peak assignment tables (`k_terms`, `solutions`), the verdicts and score of the three tests, and the spectra used for plotting.

References

Zeeden et al. (2023), see zeeden2023_table1. Farhat, M., Auclair-Desrotour, P., Boue, G., Laskar, J. (2022). The resonant tidal evolution of the Earth-Moon distance. Astronomy & Astrophysics 665, L1.

Examples

```
# synthetic tuned record with a known precession constant (k = 21 kyr)
test_tuned_check(k_true = 21, tbw = 2, len = 3000)
```

`test_tuned_check` *Check the precession-constant test on a synthetic record*

Description

Builds synthetic records (five orbital terms, a 405 kyr cycle and AR(1) noise) with a known precession constant and prints the k recovered by `test_tuned`. A check that the chosen `tbw` resolves the terms.

Usage

```
test_tuned_check(
  k_true = c(20.7, 21, 21.2, 21.4),
  tbw = 2,
  noise = 1.5,
  seed = 1,
  len = 5500,
  ...
)
```

Arguments

`k_true`	known precession constants (kyr) to test.
`tbw`	MTM time-bandwidth product.
`noise`	standard deviation of the AR(1) noise.
`seed`	random seed.
`len`	length of the synthetic record (kyr).
`...`	passed on to `test_tuned`.

Value

Prints one line per `k_true`.

`track_astro_auto` *Recommended workflow for continuous astronomical tracking of noisy records*

Description

Runs the workflow that worked best in stress tests (changing sedimentation rates with jumps, red noise):

1. a scalogram that keeps weak cycles: `analyse_wavelet_sst` with `omega_nr = 12` and four reassignment iterations (`scalogram = "msst"`, default), or the sharper superlet synchrosqueezing (`analyse_superlet_sst`, `scalogram = "superlet_sst"`), which was more accurate in records with moderate noise but drops weak cycles in very noisy records;
2. astronomically constrained tracking with all cycles (`track_astro_hmm`), with deep-time periods from `age_ma` or `k_yr`, a realistic sedimentation-rate range, and optionally a rough guide and the depths of known hiatuses or abrupt facies changes;
3. an independent check: the sedimentation rate is also estimated from long windows only (an eCOCO-like correlation of red-noise-corrected spectra with the astronomical template), and intervals where the posterior of the track is spread are flagged. Check flagged intervals by eye, with `ecoco_classic` or with amplitude modulation tests.

Tests on real records (ODP 1119, Orcadian Basin 12/13-1, KTG-01, Klonk, Bisciaro, Pod Barrandovem) showed that the tracking is only reliable with an approximate mean sedimentation rate (e.g. from the approximate duration known from biostratigraphy or radiometric ages), given as `guide` with a wide `guide_tolerance`; without it the track often settled on a sedimentation rate that was about twice too high or too low (the wrong cycles). In records where only the eccentricity cycles are resolved, the result followed such a prior closely, i.e. the record itself did not constrain the mean rate. The `duration` argument sets such a prior directly.

Usage

```
track_astro_auto(
  data,
  sedrate,
  age_ma = 0,
  k_yr = NULL,
  astro_periods = NULL,
  scalogram = c("msst", "superlet_sst", "cwt", "chirplet_sst"),
  omega_nr = 6,
  guide = NULL,
  guide_tolerance = 0.4,
  duration = NULL,
  duration_tolerance = 0.5,
  constraints = NULL,
  jump_depths = NULL,
  long_weight = 0,
  flag_ratio = 0.25,
```

```
  min_confidence = 0.5,
  flag_long = FALSE,
  dj = 1/50,
  method = c("both", "chirp", "hmm"),
  jump_prob = 0,
  ...
)
```

Arguments

data	Record: first column depth, second column proxy.
sedrate	Range c(lower, upper) of the sedimentation rate (depth units per kyr).
age_ma, k_yr	Age (Ma) or precession constant for the astronomical periods (see astro_periods_tracking) Default=0 (present-day periods).
astro_periods	Optional data frame (term, family, period, weight) or vector of periods replacing the ones from age_ma.
scalogram	"msst" (default), "superlet_sst", "chirplet_sst" (matched chirped wavelets, analyse_chirplet_sst; sharper where the sedimentation rate changes quickly, about 4 times slower) or "cwt" (a plain Morlet scalogram with omega_nr). In records in which only the eccentricity cycles are resolved, a CWT with a low omega_nr (5-6) followed the ridges more continuously than the synchrosqueezed maps (Klonk, Pridoli).
omega_nr	Morlet omega_nr for scalogram = "cwt" Default=6.
guide, guide_tolerance, jump_depths	Optional prior information, see track_astro_hmm.
duration	Approximate duration of the record (kyr), e.g. from biostratigraphy or radiometric ages. It is turned into a guide with the constant mean sedimentation rate and a tolerance of duration_tolerance (default 0.5, i.e. the rate may differ by a factor 1.5 from the mean). Strongly recommended for real records.
duration_tolerance	Relative tolerance of the duration guide Default=0.5.
constraints	Optional duration constraints (data frame from, to, duration in kyr and optionally tolerance, default 0.025), e.g. from radiometric ages or well-dated stage boundaries, see track_astro_hmm. Recommended whenever available: with the Pridoli duration (Klonk) the published age model was reproduced within about 0.3 Myr; without it the track settled on other cycles.
long_weight	Weight of the long-window evidence inside the main track Default=0; set to 1 for records in which the cycles are barely visible.
flag_ratio	Relative difference between the track and the long-window estimate above which an interval is flagged when flag_long = TRUE Default=0.25.
min_confidence	Intervals where less than this posterior probability lies within 10 percent of the tracked sedimentation rate (confidence10) are flagged Default=0.5.
flag_long	Also flag intervals where the long-window estimate disagrees Default=FALSE. In real records the long-window estimate was itself often biased, so its disagreement is reported in check but not used for the flags by default.

dj	Spacing of the period axis Default=1/50.
method	"both" (default): the scalogram is tracked with track_astro_hmm (track_hmm) and, in addition, chirp-matched windowed spectra (analyse_chirp_eha, automatic windows and gradient grid) are tracked with track_chirp_hmm with the scalogram evidence added for locality; this chirp-matched track is returned as track, check holds both rates and their disagreement, and agreement the fraction of the record where they are within 10 percent. "chirp" and "hmm" run only one of them. In the tests the chirp-matched track was better or equal everywhere: CIP Case 3 with the rate unknown over 1-20 cm/kyr, correlation with the true rate 0.97 against 0.08 for the HMM track (92 against 40 percent within 10 percent); CIP Case 2, both 0.93-0.95; synthetic records with three abrupt jumps, 0.86-0.94 against 0.00-0.92. Where the two agree both are probably right; where they disagree, look at rate_gram and plot_astro_evidence. The chirp-matched tracking adds 1-2 minutes.
jump_prob	Jump probability of the chirp-matched tracker, see track_chirp_hmm Default=0. Tested on all records: with 1e-3 the age errors of two of the four synthetic records with jumps halved (57 and 70 against 123 and 215 kyr) and those of the other two doubled, and Case 3 collapsed (0.97 to 0.10), so it stays off unless hiatuses or abrupt facies changes are known.
...	Further arguments for track_astro_hmm.

Value

A list with scalogram, track (see track_astro_hmm), check (data frame: depth, sedrate of the track, long-window sedrate, ratio, confidence, flagged) and flags (flagged intervals: from, to). Plot with plot_track(x$track, x$scalogram, flags = x$flags).

Examples

```
set.seed(1)
t <- seq(0, 1500, by = 0.5)
ecc <- 0.4 * cos(2 * pi * t / 405) + 0.25 * cos(2 * pi * t / 125)
x_t <- (1 + ecc) * (sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 19)) +
  0.7 * sin(2 * pi * t / 41)
sr <- ifelse(t < 700, 1.5, 2.6)                      # a jump of the sedimentation rate
zt <- c(0, cumsum(sr[-1] * diff(t)))
z <- seq(0, max(zt), by = 1)
x <- approx(zt, x_t, z)$y + rnorm(length(z), sd = 1.5)
res <- track_astro_auto(data.frame(z, x), sedrate = c(0.8, 5))
plot_track(res$track, res$scalogram, show = c(405, 41, 23.7), flags = res$flags)
```

---

track_astro_hmm *Astronomically constrained continuous tracking of the sedimentation rate in a scalogram*

Description

Tracks the sedimentation rate (and with it the periods of all astronomical cycles) continuously through a scalogram, using all cycles at once. At every position the hidden state is the sedimentation rate s; the evidence for a value of s is the (normalised, smoothed) power of the scalogram at the depth periods `P_k * s` of all astronomical cycles `P_k` together (e.g. eccentricity, obliquity and precession). Where one cycle is weak (a node of the eccentricity modulation of precession, a gap in a synchrosqueezed ridge) the others keep the track on course, and the fixed ratios between the periods prevent the confusion of one cycle with another (e.g. an obliquity ridge read as precession) that single ridge tracking suffers from. The sedimentation rate follows a random walk with an optional small probability of abrupt changes. The forward-backward algorithm gives a continuous most probable sedimentation rate curve with a credible band, and paths sampled from the posterior give an age model with uncertainty.

Performance in synthetic tests (3000 kyr; eccentricity-modulated precession, obliquity and eccentricity; three jumps of the sedimentation rate by factors 1.7, 0.5 and 1.6 plus gradual changes; AR1 noise; scalogram from `analyse_wavelet_sst(omega_nr = 12, n_iter = 4)`): the largest age error was about 2 percent of the duration with noise half as strong as the signal, 3-4 percent with equally strong noise and 6-9 percent with noise twice as strong (always about half of the error of picking the strongest precession power in a band). With noise three times as strong as the signal the track only succeeded with a rough guide (about 7 percent); without one it locked onto the wrong cycles. The credible bands then also become too narrow: they describe the uncertainty given that the track follows the right cycles, which the method cannot guarantee at such noise levels. Check weak intervals (low `confidence`) by eye and with independent methods (e.g. `ecoco_classic`, amplitude modulation tests).

The astronomical periods must fit the age of the record: for deep time use precession and obliquity periods that account for the recession of the Moon (e.g. from `astro_terms`, a k value from `test_tuned` or a published solution).

Usage

```
track_astro_hmm(
  wavelet,
  astro_periods = c(405, 125, 95, 53.6, 41, 39.7, 23.7, 22.4, 19),
  weights = NULL,
  sedrate = NULL,
  guide = NULL,
  guide_tolerance = 0.25,
  floor = 0.5,
  whiten = FALSE,
  background = c("none", "ar1"),
  sd_change = 0.15,
  jump_prob = 1e-04,
  smooth_period = 0.02,
  smooth_depth = NULL,
  fallback = "cwt",
  fallback_weight = 0.2,
  sharpness = 0.5,
  n_positions = 1500,
  n_samples = 200,
```

```
  level = 0.95,
  jump_depths = NULL,
  jump_prob_at = 0.05,
  long_window = NULL,
  long_weight = 0,
  local_weight = 1,
  guide_penalty = 50,
  normalise = c("median", "column", "local", "loglog"),
  local_octaves = 1,
  consistent_cycles = "partial",
  constraints = NULL,
  smooth_path = c("none", "tv", "trend"),
  smooth_lambda = 1,
  age_ma = NULL,
  k_yr = NULL,
  seed = 1
)
```

Arguments

wavelet	Scalogram object (class analyze.wavelet), see track_ridge_hmm.
astro_periods	Periods of the astronomical cycles in time Default=c(405, 125, 95, 53.6, 41, 39.7, 23.7, 22.4, 19) (kyr, Cenozoic values). Listing all the main terms (including the weaker obliquity and precession terms) helps to tell the cycles apart.
weights	Weights of the cycles in the evidence, best proportional to their expected amplitude. Default: for the default periods c(0.5, 0.3, 0.3, 0.3, 1, 0.4, 1, 0.8, 0.6), otherwise 1 for all.
sedrate	Range c(lower, upper) of the sedimentation rate (depth units per time unit of astro_periods) in which the track is searched.
guide	Optional rough sedimentation rate curve (data frame: depth, sedimentation rate), e.g. from a manual tracking divided by the period of the tracked cycle; the search is restricted to guide * (1 +- guide_tolerance). In very noisy records this prevents the track from locking onto the wrong cycle.
guide_tolerance	Relative width of the search band around the guide Default=0.25.
floor	Value added to the normalised evidence (average one) before taking the logarithm. It limits the penalty for a cycle that is absent at a position Default=0.5.
whiten	Divide the power of every period by its average along the record before tracking (removes the red-noise slope) Default=FALSE (in tests this made the tracking worse, because noise at the long periods is amplified).
background	"none" or "ar1": evidence relative to an AR1 red-noise background Default="none".
sd_change	Standard deviation of the change of log2(sedimentation rate) over a distance of one cycle of the shortest astronomical period Default=0.15.
jump_prob	Probability of an abrupt change of the sedimentation rate between neighbouring positions Default=1e-4; small, because a jump of the track by the ratio of two

astronomical periods (e.g. from precession to obliquity) is the typical failure of tracking.

`smooth_period`, `smooth_depth`, `fallback`, `fallback_weight`, `sharpness`, `n_positions`, `n_samples`, `level`, `seed`
: See `track_ridge_hmm`; `smooth_depth` defaults to twice the depth period of the shortest cycle at the (geometric) mean sedimentation rate. For this tracker a scalogram in which weak cycles are kept (the Morlet scalogram with omega_nr around 12, or `analyse_wavelet_sst` with `n_iter = 4`) worked better in tests than the consensus of `analyse_superlet_sst`, which removes weak cycles.

`jump_depths`, `jump_prob_at`, `constraints`
: See `track_ridge_hmm`.

`long_window`, `long_weight`
: Long-window evidence (similar to eCOCO): the wavelet power averaged over windows of `long_window` depth units (default eight cycles of the shortest period at the mean sedimentation rate), with the red-noise slope removed, is correlated with the astronomical template and added with weight `long_weight` `Default=0` (off). In stress tests this term kept the track on the right cycles when the noise was three times the signal (largest age error about 400 instead of 700 kyr in 3000 kyr) but made the age model less accurate at lower noise, because the averaged spectra are biased where the sedimentation rate changes within a window. Use it (e.g. `long_weight = 1`) for very noisy records, or through `track_astro_auto` as an independent check.

`local_weight`
: Weight of the local (scalogram) evidence `Default=1`; 0 with `long_weight > 0` gives a pure long-window estimate.

`guide_penalty`
: Penalty (log-likelihood per octave outside the band, plus 10 percent) for sedimentation rates outside `guide * (1 +- guide_tolerance)` `Default=50`, i.e. practically a hard limit; smaller values make the guide soft.

`normalise`
: How the scalogram is turned into evidence: `"median"` (default) divides the power by its running median along the period axis (over `+- local_octaves`), which removes the red-noise slope without being pulled up by the cycles themselves, and scores every cycle relative to the average evidence at that position, so that cycles outside the period range of the scalogram neither help nor hurt; `"column"` (the behaviour before) divides by the mean power at each position, `"local"` by a Gaussian average and `"loglog"` by a quadratic fit of log power against log period. In synthetic tests `"column"` and `"median"` were best; in six real records `"column"` was biased towards sedimentation rates that put the long cycles on the red-noise slope or the short cycles outside the resolved range, and `"median"` gave the smallest errors when an approximate mean sedimentation rate was given.

`local_octaves`
: Half width (octaves) of the background window of `normalise` `Default=1`.

`consistent_cycles`
: `TRUE`: use only the cycles that lie inside the period range of the scalogram at every sedimentation rate of `sedrate`; `"partial"` (default): cycles that leave the period range at one end of the rate range (typically precession below the sampling resolution at the lowest rates) are kept, but only the changes of their evidence along the record are used (row means removed), so they show where the rate changes without favouring the rates at which they are resolved; `FALSE`:

all cycles, all evidence. In CIP Case 3 (precession at 3-5 samples per cycle) `TRUE` silently dropped the precession cycles and the 1.2-Myr oscillation of the rate was lost; `"partial"` keeps them. If the set of usable cycles changed with the sedimentation rate, rates at which more cycles fall inside the resolved range would be favoured: in the Klonk core, which resolves only eccentricity, the ~110-kyr cycle was read as obliquity and the 405-kyr cycle as ~125 kyr. Give a realistic `sedrate` range (or a guide) so that the cycles you expect stay in.

`smooth_path` Optional L1 smoothing of the tracked path (Viterbi path) with weights from the posterior (majorisation-minimisation; Wang et al. 2023): `"tv"` penalises the first differences of the log sedimentation rate (total variation: a piecewise constant rate with sharp jumps is favoured), `"trend"` the second differences (piecewise linear). `Default="none"`.

`smooth_lambda` Strength of the smoothing: the penalty per octave of change of the log2 sedimentation rate (`"tv"`) or per octave of change of its slope (`"trend"`), against the weighted squared distance from the tracked path `Default=1`.

`age_ma`, `k_yr` Optional age (Ma) or precession constant: the astronomical periods and weights are then taken from `astro_periods_tracking` (deep-time obliquity and precession periods), replacing `astro_periods`.

Value

A data frame of class `wr_track` with `depth`, `sedrate` (Viterbi path), `sedrate_mean`, `sedrate_lower`, `sedrate_upper`, `confidence`, `time`, `time_lower`, `time_upper`, and a column `period_<P>` with the depth period of every astronomical cycle. The attribute `astro_periods` holds the periods.

References

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257-286.

Meyers, S.R., 2015. The evaluation of eccentricity-related amplitude modulation and bundling in paleoclimate data: an inverse approach for astrochronologic testing and time scale optimization. Paleoceanography 30, 1625-1640.

Examples

```
set.seed(1)
t <- seq(0, 1500, by = 0.5)
ecc <- 0.4 * cos(2 * pi * t / 405) + 0.25 * cos(2 * pi * t / 125)
x_t <- (1 + ecc) * (sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 19)) +
  0.7 * sin(2 * pi * t / 41)
sr <- 2 + 0.8 * sin(2 * pi * t / 600)                # sedimentation rate (cm/kyr)
zt <- c(0, cumsum(sr[-1] * diff(t)))
z <- seq(0, max(zt), by = 1)
x <- approx(zt, x_t, z)$y + rnorm(length(z), sd = 1)
s <- analyse_superlet_sst(data.frame(z, x), lowerPeriod = 15, upperPeriod = 400, dj = 1 / 50)
tr <- track_astro_hmm(s, astro_periods = c(405, 125, 41, 23.7, 19), sedrate = c(0.8, 5))
plot_track(tr, s)
```

`track_astro_iterative` *Iterative (demodulated) astronomical tracking of the sedimentation rate*

Description

Tracks the sedimentation rate, converts the record to approximate time with the result, transforms it again in that (nearly stationary) time domain with sharper wavelets, tracks the residual stretch of the time axis, composes the age models and repeats. Demodulating a signal by its own estimated instantaneous frequency before the synchrosqueezing makes the ridges much sharper than any higher-order synchrosqueezing of the raw signal (Meignen et al. 2017; Jiang et al. 2022), and in cyclostratigraphy the demodulation is the age model itself: in the time domain the astronomical cycles have (nearly) fixed periods, so wavelets with many more cycles can be used, the eccentricity quartet and the precession bands separate, and the tracked residual converges to the true age model.

Usage

```
track_astro_iterative(
  data,
  sedrate,
  n_iter = 1,
  residual = 1.6,
  scalogram = c("chirplet_sst", "superlet_sst", "msst", "cwt"),
  omega_nr = 6,
  omega_nr_warped = c(12, 18, 24),
  age_ma = 0,
  k_yr = NULL,
  astro_periods = NULL,
  guide = NULL,
  guide_tolerance = 0.4,
  duration = NULL,
  duration_tolerance = 0.5,
  constraints = NULL,
  jump_depths = NULL,
  dj = 1/50,
  n_samples = 100,
  verbose = TRUE,
  ...
)
```

Arguments

`data`	Record: first column depth, second column proxy.
`sedrate`	Range `c(lower, upper)` of the sedimentation rate (depth units per kyr).
`n_iter`	Maximum number of warped iterations `Default=1`. In synthetic tests (three jumps of the sedimentation rate, AR1 noise) the first warped iteration lowered

	the largest age error by about 10-15 percent and a second one gave nothing more (it sometimes regressed), so one iteration is the default.
residual	Range of the residual stretch of the time axis that is searched in the warped iterations (c(1/residual, residual)); the first warped iteration uses residual, later ones sqrt(residual) Default=1.6.
scalogram	Scalogram of the first (depth) iteration: "chirplet_sst" (default; in the synthetic tests its tracks had about 35 percent lower age errors than those of "superlet_sst", at four times the cost), "superlet_sst", "msst" or "cwt" (best for records that resolve only the eccentricity, e.g. Klonk; see track_astro_auto).
omega_nr	Morlet omega_nr for scalogram = "cwt" Default=6.
omega_nr_warped	Morlet omega_0 values of the superlet-SST orders in the warped (time) iterations Default=c(12, 18, 24).
age_ma, k_yr, astro_periods	Astronomical periods, see track_astro_auto.
guide, guide_tolerance, duration, duration_tolerance, constraints, jump_depths	Prior information, see track_astro_auto and track_astro_hmm. Constraints and jump depths are given in depth and are carried into the warped iterations.
dj	Spacing of the period axis Default=1/50.
n_samples	Posterior paths Default=100.
verbose	Print the progress Default=TRUE.
...	Further arguments for track_astro_hmm (e.g. sd_change, seed).

Details

Every iteration is checked with the posterior of the tracking (confidence10, the probability within 10 percent of the track): when the confidence falls, the previous iteration is kept and the loop stops. The demodulation sharpens the ridges but cannot repair a first track that follows the wrong cycles over a long interval (the warped record then shows the same ambiguity at a different stretch), so the prior information of track_astro_auto (duration, constraints, jump_depths) matters more than the iterations. The credible band of an iteration is the band of the residual widened in quadrature by the band of the previous iteration.

Value

A wr_track data frame in depth as from track_astro_hmm (columns depth, sedrate, confidence, confidence10, time, time_lower, time_upper, ...), with attributes iterations (the per-iteration summaries: mean confidence10, duration), tracks (the track of every iteration in depth), warped (the last warped scalogram with its period axis in kyr, i.e. the tuned scalogram, and the record in time) and first (the result of the depth iteration, a track_astro_auto object).

References

Meignen, S., Pham, D.-H. and McLaughlin, S. (2017). On demodulation, ridge detection, and synchrosqueezing for multicomponent signals. IEEE Trans. Signal Process. 65, 2093-2103.

Jiang, Q., Prater-Bennette, A., Suter, B.W. and Zeyani, A. (2022). Instantaneous frequency-embedded synchrosqueezing transform for signal separation. Front. Appl. Math. Stat. 8, 830530.

Examples

```
set.seed(1)
t <- seq(0, 1500, by = 0.5)
ecc <- 0.4 * cos(2 * pi * t / 405) + 0.25 * cos(2 * pi * t / 125)
x_t <- (1 + ecc) * (sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 19)) +
  0.7 * sin(2 * pi * t / 41)
sr <- ifelse(t < 700, 1.5, 2.6)
zt <- c(0, cumsum(sr[-1] * diff(t)))
z <- seq(0, max(zt), by = 1)
x <- approx(zt, x_t, z)$y + rnorm(length(z), sd = 1.5)
res <- track_astro_iterative(data.frame(z, x), sedrate = c(0.8, 5))
plot(res$depth, res$sedrate, type = "l"); lines(z, approx(zt, sr, z)$y, col = 2)
```

`track_chirp_hmm` *Chirp-steered tracking of the sedimentation rate (rate and rate gradient as the hidden state)*

Description

Hidden Markov tracking on the chirp-matched template scores of `analyse_chirp_eha` (with `criterion = "template"`): the hidden state is the pair (sedimentation rate, relative rate gradient). The evidence of a window is the match of the astronomical template with the spectrum de-chirped by that gradient, at that rate; the transition moves the rate by the gradient (log r' = log r + g step) and lets the gradient drift slowly. The gradient therefore steers the track along the direction the de-chirped evidence points to, and a change of the rate has to be announced by the chirp of the cycles before the track follows it: a wiggle of the rate that the spectra support in both its level and its slope is followed, one that the slopes contradict is not.

Usage

```
track_chirp_hmm(
  ce,
  weight = 10,
  sd_rate = 0.01,
  sd_gradient = 1.5,
  jump_prob = 0,
  gradient_prior = Inf,
  scalogram = NULL,
  scalogram_weight = 1,
  scalogram_args = list(),
  n_samples = 100,
  level = 0.95,
  seed = 1,
  debias = TRUE,
  chirp = seq(-0.05, 0.05, by = 0.005),
  ...
)
```

Arguments

ce	Result of analyse_chirp_eha with criterion = "template" or of analyse_chirp_wavelet; or any scalogram (analyse_wavelet, analyse_superlet_sst, ...): the evidence of track_astro_hmm is then tracked with the gradient as a hidden state (arguments of track_astro_hmm such as astro_periods and sedrate in ..., gradient grid chirp); the evidence then does not depend on the gradient, only the continuity of the rate's slope steers the track.
weight	Weight of the template score of one window against the transition prior Default=10. On CIP Case 3 (window 20 m, step 0.6 m) the result was the same for 5-20; below 3 the track no longer follows the evidence.
sd_rate	Standard deviation (octaves per window step) of the rate beyond the move given by the gradient Default=0.01.
sd_gradient	Standard deviation of the change of the gradient per window step, in units of the gradient grid spacing Default=0.7.
jump_prob	Probability per window step of a jump of the rate to any value (an abrupt change: a hiatus, an erosion surface, a facies change), after which the gradient restarts at zero Default=0. A jump is not a gradient: without this the gradient-steered track can only reach a new rate through a ramp. Any value above zero lets the track jump wherever the evidence gains more than the (log) probability: on the synthetic records with three jumps 1e-3 halved the age errors, on CIP Case 3 (no jumps) it introduced a few spurious jumps (correlation 0.97 to 0.88), so it is off by default and should be switched on for records with hiatuses or abrupt facies changes.
gradient_prior	Standard deviation (in steps of the gradient grid) of a zero-centred prior on the gradient, applied at every window Default=Inf (none; a prior of 3 grid steps lowered the Case 3 result from 0.97 to 0.55 and did not help the records with jumps).
scalogram	Optional scalogram (analyse_wavelet etc.) whose track_astro_hmm evidence is added as complementary, local evidence (same astro_periods in scalogram_args): the windows of the chirp-matched spectra need 15-25 cycles and straddle abrupt changes of the rate, the short wavelets see them; the sum has the sharp lines of the one and the locality of the other.
scalogram_weight	Weight of that evidence relative to the template scores (both scaled by their standard deviation) Default=1.
scalogram_args	Arguments of track_astro_hmm for the scalogram evidence.
n_samples	Posterior paths for the credible band Default=100.
level	Credible level Default=0.95.
seed	Seed Default=1.
debias	Remove, for every gradient, the mean score over all rates and windows Default=TRUE (the warped windows of a strong gradient have a slightly different noise continuum, which otherwise favours or penalises that gradient everywhere).
chirp	Gradient grid for a plain scalogram.
...	Arguments of track_astro_hmm for a plain scalogram.

Value

A `wr_track` data frame (depth, sedrate, gradient, sedrate_lower, sedrate_upper, confidence10, time, time_lower, time_upper) with the attribute `posterior` (depth, sedrate grid, posterior over the rate) and `gradient_grid`.

Examples

```
set.seed(1)
t <- seq(0, 3000, by = 1)
x_t <- sin(2 * pi * t / 23.7) + 0.6 * sin(2 * pi * t / 19) + 0.8 * sin(2 * pi * t / 41)
sr <- 2 + 0.7 * sin(2 * pi * t / 600)
zt <- c(0, cumsum(sr[-1] * diff(t)))
z <- seq(0, max(zt), by = 1)
x <- approx(zt, x_t, z)$y + rnorm(length(z), sd = 1.5)
ce <- analyse_chirp_eha(data.frame(z, x), window = 250, step = 10, chirp = seq(-0.004,
    0.004, by = 0.0005),
                        lowerPeriod = 10, upperPeriod = 150, criterion = "template",
                        astro_periods = c(41, 23.7, 19), sedrate = c(0.8, 5))
tr <- track_chirp_hmm(ce)
plot(tr$depth, tr$sedrate, type = "l"); lines(z, approx(zt, sr, z)$y, col = 2)
```

`track_ecoco` *Track the sedimentation rate through an eCOCO map*

Description

Interactively select points in the map of an eCOCO result (sedimentation rate against depth) to trace how the sedimentation rate changes along the record. This is the eCOCO counterpart of `track_period`: the map of one of the eCOCO statistics (by default the combined correlation and significance score `pcoco`) is plotted, the local maxima (ridges) are marked, and the sedimentation rate is tracked by clicking ridge points.

Selected points are shown as white dots. Clicking a selected point again deselects it (red dot). Press the right mouse button or Esc (depending on the graphics device) to finish. An interactive device that supports `identify` is needed (on Windows this works out of the box, on macOS XQuartz or the Quartz device, on Linux an X11 device).

The tracking can also be done without a mouse: pass the points as `points`. They are snapped to the nearest ridge point, which makes a tracking reproducible and scriptable.

Usage

```
track_ecoco(
  ecoco = NULL,
  map = c("pcoco", "rho", "score", "p_global", "norbit"),
  track_peaks = TRUE,
  n_peaks = 3,
  rho_min = NULL,
```

```
  p_max = NULL,
  norbit_min = NULL,
  points = NULL,
  interpolate = FALSE,
  n.levels = 100,
  plot_horizontal = TRUE,
  plot_dir = TRUE,
  x_lab = "depth (metres)",
  y_lab = "sedimentation rate (cm/kyr)",
  palette_name = "viridis",
  genplot = FALSE
)
```

Arguments

ecoco	An eCOCO result from ecoco_adaptive, ecoco_blocked or ecoco_interleaved (a list with sr, depth and rate-by-window matrices).
map	The map used for tracking and plotting: "pcoco", "rho", "score", "p_global" (shown as minus log10 p) or "norbit". Default="pcoco".
track_peaks	If TRUE only the local maxima along the sedimentation rate axis can be selected, otherwise every grid cell Default=TRUE.
n_peaks	Maximum number of peaks per window that can be selected (the highest ones) Default=3.
rho_min, p_max, norbit_min	Optional criteria a peak must meet: minimum correlation, maximum p-value (global) and minimum number of contributing orbital periods. Default=NULL (not used).
points	Optional data frame or matrix with columns depth and sedimentation rate. If given, no mouse input is needed: the points are snapped to the nearest selectable point.
interpolate	Linearly interpolate the tracked sedimentation rate to all windows of the eCOCO result (constant outside the tracked interval) Default=FALSE.
n.levels	Number of colour levels Default=100.
plot_horizontal	Plot depth on the x-axis (TRUE) or on the y-axis (FALSE) Default=TRUE.
plot_dir	Logical. TRUE if depth increases along the axis, FALSE to reverse the depth axis (for example for borehole data with depth increasing downwards) Default=TRUE.
x_lab	Label of the depth axis Default="depth (metres)".
y_lab	Label of the sedimentation rate axis Default="sedimentation rate (cm/kyr)".
palette_name	Name of the hcl.colors palette Default="viridis".
genplot	Plot the result (only needed when points is used; the interactive mode always plots) Default=FALSE.

Value

A data frame with the tracked windows: `depth`, `sedrate` (cm/kyr) and, when available in `ecoco`, `rho`, `p_global`, `norbit` and the value of the map. The result can be used like the output of `track_period` (for example to build an age model).

Author(s)

Michiel Arts. Inspired by `track_period` and the tracking routines of Acycle (Mingsong Li); see `ecoco_track` for the automatic version.

See Also

`ecoco_track`, `ecoco_track_ridge`, `track_period`

Examples

```
# synthetic eCOCO-like map: sedimentation rate increases with depth
sr <- seq(1, 10, by = 0.25)
depth <- seq(0, 20, by = 0.5)
true_sr <- 4 + 2 * depth / 20
rho <- sapply(true_sr, function(s) 0.8 * exp(-(sr - s)^2 / 0.5))
ecoco <- list(sr = sr, depth = depth, rho = rho, pcoco = 2 * rho,
              p_global = matrix(0.01, length(sr), length(depth)),
              norbit = matrix(9, length(sr), length(depth)))
# scripted tracking: points are snapped to the nearest ridge point
trk <- track_ecoco(ecoco, points = data.frame(depth = c(2, 10, 18), sr = c(4.2, 5, 5.8)))
trk

# interactive tracking (needs a graphics device with mouse input)
if (interactive()) trk <- track_ecoco(ecoco)
```

`track_mesa` *Track a spectral peak through an evolutive MESA*

Description

Follows one spectral ridge (for example the 100 ka cycle) through the window positions of an evolutive MESA (`mesa_evolutive`). A dynamic-programming search finds the path through the spectral image that has the highest summed log relative power, with the frequency allowed to change by at most `max_jump` (as a fraction of the frequency) from one position to the next and a penalty for large changes, so that the track does not hop between neighbouring peaks. Optionally the track is read off the stored peaks instead of the image.

Usage

```
track_mesa(
  x,
  window = 1,
  period = NULL,
  frequency = NULL,
  max_jump = 0.1,
  penalty = 0.3,
  start = NULL,
  start_at = c("top", "end"),
  snap = TRUE
)
```

Arguments

x	output of mesa_evolutive.
window	number or length of the window to track in.
period	Period (wavelength) band c(lower, upper) in which the ridge is searched, in the units of the first column of the record. Give period or frequency.
frequency	Frequency band c(lower, upper), alternative to period.
max_jump	Largest relative change of the frequency between neighbouring window positions (default 0.1, i.e. 10 percent).
penalty	Penalty (in units of log10 relative power) for a change of the frequency by max_jump; smaller changes cost proportionally less. Default 0.3.
start	Optional period at the first position (or last, if start_at = "end") that fixes where the track begins; the search then starts at the peak nearest to it.
start_at	"top" (first position, default) or "end".
snap	If TRUE (default) every position of the track is moved to the nearest local maximum of the spectrum within one frequency step, so that the track follows the spectral peak and not the shoulder of a broad one.

Value

A data frame of class mesa_track with position, frequency, period and rel_power (power relative to the maximum at that position), and the attribute window giving the window length.

Examples

```
set.seed(2)
t <- seq(0, 800, by = 2)
y <- sin(2 * pi * cumsum(1 / ifelse(t < 400, 40, 80)) * 2) + rnorm(length(t), sd = 0.4)
ev <- mesa_evolutive(data.frame(t, y), window = 200, order = 20)
tr <- track_mesa(ev, period = c(25, 120))
plot_mesa_evolutive(ev, track = tr, vertical = FALSE)
```

track_period *Track the period of a cycle in a wavelet or superlet scalogram*

Description

Interactively select points in a wavelet or superlet power spectrum to trace the evolution of a cycle with changing period.

The track_period function plots a time-frequency spectrum in which spectral peaks can be selected to track a ridge through time or depth. This allows the user to follow a cycle whose period varies along the record.

Tracking points are selected interactively and displayed as white dots. Previously selected points can be deselected by clicking them again, after which they are shown as red dots. Because points may be closely spaced, de-selection can be difficult. In such cases, delpts_tracked_period_wt can be used to remove points that were previously selected.

Usage

```
track_period(
  scalogram = NULL,
  astro_cycle = 405,
  n.levels = 100,
  track_peaks = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  palette_name = "rainbow",
  color_brewer = "grDevices",
  plot_horizontal = TRUE,
  plot_dir = TRUE,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  colour_brewer = NULL
)
```

Arguments

scalogram	A wavelet or superlet object created using analyse_wavelet or analyse_superlet.
astro_cycle	Duration (in kyr) of the astronomical cycle that is being tracked.
n.levels	Number of colour levels used for plotting. Default is 100.
track_peaks	Logical flag indicating whether tracking is restricted to spectral peaks (TRUE) or whether any point within the spectrum can be selected (FALSE). Default is TRUE.
periodlab	Label for the period axis. Default is "Period (metres)".

x_lab	Label for the x-axis. Default is "depth (metres)".
palette_name	Name of the colour palette used for plotting.
color_brewer	Name of the R package from which the colour palette is selected. Supported packages are RColorBrewer, grDevices, ColorRamps, and viridis. Default is "grDevices".
plot_horizontal	Logical flag indicating whether the spectrum is plotted horizontally or vertically. Default is TRUE.
plot_dir	Logical flag defining the direction of the record. If time increases with increasing depth (e.g. borehole data), set to TRUE. If time decreases with increasing depth, set to FALSE. Default is TRUE.
lowerPeriod	Lowest period value to be displayed.
upperPeriod	Highest period value to be displayed.
add_lines	Optional matrix of additional lines to overlay on the spectrum. The first column must be depth or time, and subsequent columns contain period values.
add_points	Optional matrix of additional points to overlay on the spectrum. The first column must be depth or time, and subsequent columns contain period values.
add_abline_h	Optional numeric vector specifying horizontal reference lines.
add_abline_v	Optional numeric vector specifying vertical reference lines.
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

A data frame with three columns: first - depth - Depth or time of the tracked points second - period - Tracked period of the cycle third - sedrate - Estimated sedimentation rate based on the cycle duration

Author(s)

The function is based on and inspired by the traceFreq function from 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

```
if (interactive()) { # needs mouse input (identify())
## Track the 405 kyr eccentricity cycle in a magnetic susceptibility record
mag_wt <- analyse_wavelet(
  data = mag,
  dj = 1/100,
  lowerPeriod = 0.1,
  upperPeriod = 254,
  verbose = FALSE,
  omega_nr = 10
)
```

```
mag_track <- track_period(
  scalogram = mag_wt,
  astro_cycle = 405,
  track_peaks = TRUE
)
}
```

`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. Tracking points can be selected in the Interactive interface and will be shown as white dots when one wants to deselect a point the white dots can be re-clicked/re-selected and will turn red which indicates that the previously selected point is deselected. Deselecting points can be quite tricky due to the close spacing of points and such the `delpts_tracked_period_wt` can be used to delete points were previously selected using the `track_period_wavelet` function.

Usage

```
track_period_wavelet(
  wavelet = NULL,
  astro_cycle = 405,
  n.levels = 100,
  track_peaks = TRUE,
  periodlab = "Period (metres)",
  x_lab = "depth (metres)",
  palette_name = "rainbow",
  color_brewer = "grDevices",
  plot_horizontal = TRUE,
  plot_dir = TRUE,
  lowerPeriod = NULL,
  upperPeriod = NULL,
  add_lines = NULL,
  add_points = NULL,
  add_abline_h = NULL,
  add_abline_v = NULL,
  colour_brewer = NULL
)
```

Arguments

wavelet	Wavelet object created using the analyse_wavelet function.
astro_cycle	Duration (in kyr) of the cycle which traced.
n.levels	Number of color levels Default=100.
track_peaks	Setting which indicates whether tracking is restricted to spectral peaks (track_peaks=TRUE) or whether any point within the wavelet spectra can be selected (track_peaks=FALSE) Default=TRUE. With track_peaks=TRUE a click selects the nearest local power maximum (black circles); with track_peaks=FALSE any point of the scalogram can be clicked: the click is snapped to the nearest depth of the wavelet and its period is used as clicked. In both modes clicking a selected point again removes it; finish with right-click/Esc (or "Finish" in RStudio).
periodlab	label for the y-axis Default="Period (metres)".
x_lab	label for the x-axis Default="depth (metres)".
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
plot_horizontal	TRUE: depth on the x-axis and period on the y-axis; FALSE: period on the x-axis and depth on the y-axis. The output is the same in both layouts Default=TRUE
plot_dir	The direction of the proxy record which is assumed for tuning if time increases with increasing depth/time values (e.g. bore hole data which gets older with increasing depth) then plot_dir should be set to TRUE if time decreases with depth/time values (eg stratospheric logs where 0m is the bottom of the section) then plot_dir should be set to FALSE plot_dir=TRUE
lowerPeriod	Lowest period value which will be plotted
upperPeriod	Highest period value which will be plotted
add_lines	Add lines to the wavelet plot input should be matrix with first axis being depth/time the columns after that should be period values Default=NULL

add_points	Add points to the wavelet plot input should be matrix with first axis being depth/time and columns after that should be period values Default=NULL
add_abline_h	Add horizontal lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
add_abline_v	Add vertical lines to the plot. Specify the lines as a vector e.g. c(2,3,5,6) Default=NULL
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

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

Author(s)

The function is based/inspired on the traceFreq function of the 'astrochron' R package

References

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

Examples

```
if (interactive()) { # needs mouse input (identify())
#Track the 405kyr eccentricity cycle in the magnetic susceptibility record
# of the Sullivan core of Pas et al., (2018)

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

mag_track <- track_period_wavelet(wavelet = mag_wt,
astro_cycle = 405,
n.levels = 100,
track_peaks = TRUE,
periodlab =  "Period (metres)",
x_lab = "depth (metres)",
palette_name = "rainbow",
color_brewer = "grDevices",
plot_horizontal = TRUE,
plot_dir = TRUE,
lowerPeriod = NULL,
upperPeriod = NULL,
add_lines = NULL,
add_points = NULL,
add_abline_h = NULL,
```

```
add_abline_v = NULL)
}
```

`track_ridge_hmm` *Continuous tracking of a cycle in a (synchrosqueezed) scalogram with a hidden Markov model*

Description

Tracks the period of one cycle continuously through a scalogram made by `analyse_wavelet`, `analyse_wavelet_sst`, `analyse_superlet_sst`, `analyse_conceft` or any object of class `analyze.wavelet`. Sharp synchrosqueezed scalograms can have gaps where the cycle is weak (low eccentricity, noise, a facies change), which makes manual or ridge-by-ridge tracking discontinuous. Here the period at every position is a hidden state that follows a random walk (the sedimentation rate changes gradually, with an optional small probability of an abrupt change, e.g. at a hiatus) and the scalogram is the evidence. The forward-backward algorithm combines the evidence of the whole record: where the cycle is clear it fixes the period, and through gaps the track is bridged by the neighbouring evidence and by the fallback scalogram (the ordinary wavelet power, which is never empty). The result is a continuous curve with

- the most probable path (Viterbi),
- the posterior mean and a credible band of the period at every position,
- a confidence value (the posterior probability within 2 percent of the tracked period), which shows where the track is constrained by data and where it is interpolated,
- optionally (`period_of_tracked_cycle`) the sedimentation rate and an age model with uncertainty from paths sampled from the posterior.

The first two columns (depth, period) can be used directly by `curve2time` and the other functions that take a tracked curve.

Usage

```
track_ridge_hmm(
  wavelet,
  period = NULL,
  guide = NULL,
  guide_tolerance = 0.25,
  sd_change = 0.1,
  jump_prob = 0.001,
  smooth_period = 0.02,
  smooth_depth = NULL,
  fallback = "cwt",
  fallback_weight = 0.2,
  sharpness = 0.5,
  n_positions = 1500,
  period_of_tracked_cycle = NULL,
```

```
  n_samples = 200,
  level = 0.95,
  jump_depths = NULL,
  constraints = NULL,
  jump_prob_at = 0.05,
  seed = 1
)
```

Arguments

wavelet	Scalogram object (class analyze.wavelet).
period	Band c(lower, upper) in which the cycle is searched (in depth units). Alternatively give guide.
guide	Optional rough track (data frame: depth, period), e.g. from track_period_wavelet or a previous run; the search is restricted to guide * (1 +- guide_tolerance).
guide_tolerance	Relative width of the search band around the guide Default=0.25.
sd_change	Standard deviation of the change of log2(period) over a distance of one period Default=0.1 (about 7 percent per cycle). Larger values follow faster changes of the sedimentation rate.
jump_prob	Probability of an abrupt change of the period between neighbouring positions Default=1e-3.
smooth_period	Smoothing of the scalogram along the period axis before tracking (sd in octaves) Default=0.02.
smooth_depth	Moving average of the scalogram along the record before tracking (in depth units) Default=NULL: the lower limit of the band (one shortest period).
fallback	"cwt" (default) to mix in the normalised Morlet wavelet power (from the object or computed), another scalogram object, or "none".
fallback_weight	Weight of the fallback evidence Default=0.2.
sharpness	Weight of the evidence Default=0.5. Because neighbouring positions of a scalogram are not independent, the log evidence is scaled so that one period of the cycle counts as one observation; larger values trust the scalogram more relative to the smoothness of the track and give narrower credible bands; with 0.5 the 95 percent bands of the age model contained the true age 95 to 100 percent of the time in synthetic tests.
n_positions	Number of positions at which the track is computed (it is interpolated to all positions) Default=1500; fewer are used when the positions would be closer than a quarter of the (reference) period.
period_of_tracked_cycle	Optional duration of the tracked cycle (e.g. 405 kyr): adds the sedimentation rate and an age model with uncertainty.
n_samples	Number of posterior paths used for the uncertainty of the age model Default=200.
level	Width of the credible bands Default=0.95.

`jump_depths` Optional depths where an abrupt change of the sedimentation rate is plausible (a hiatus, erosion surface, abrupt facies change or tectonic contact known from the field or from a change-point analysis such as `bayes_changepoint`). Near these depths the jump probability is raised to `jump_prob_at`. Abrupt changes are hard to detect from the scalogram alone in noisy records, so giving them helps the track to follow the jump instead of smoothing it.

`constraints` Optional geological constraints: a data frame with the columns `from`, `to` (depths) and `duration` (time, in the units of the astronomical periods, e.g. kyr), for instance a biostratigraphic bracket, the time between two dated ash beds or the duration of a stage known from another section. The track is then the most probable one that satisfies them: the evidence is tilted by a Lagrange multiplier times the time spent in each interval, with the multiplier found by bisection so that the interval has the required duration. The achieved durations and multipliers are returned in the attribute `constraints`.

`jump_prob_at` Jump probability near `jump_depths` `Default=0.05`.

`seed` Seed of the posterior sampling `Default=1`.

Value

A data frame of class `wr_track` with the columns `depth`, `period` (Viterbi path), `period_mean`, `period_lower`, `period_upper`, `confidence` and, with `period_of_tracked_cycle`, `sedrate` (depth units per time unit), `sedrate_lower`, `sedrate_upper`, `time`, `time_lower` and `time_upper` (time since the top/start of the record).

References

Rabiner, L.R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE 77, 257-286.

Carmona, R.A., Hwang, W.L., Torresani, B., 1997. Characterization of signals by the ridges of their wavelet transforms. IEEE Transactions on Signal Processing 45, 2586-2590.

Examples

```
set.seed(1)
z <- 0:1999
Pt <- 40 + 10 * sin(2 * pi * z / 1500)
amp <- 0.3 + abs(sin(2 * pi * z / 600))
x <- amp * sin(2 * pi * cumsum(1 / Pt)) + rnorm(length(z), sd = 0.5)
s <- analyse_superlet_sst(data.frame(z, x), lowerPeriod = 15, upperPeriod = 120, dj = 1 / 50)
tr <- track_ridge_hmm(s, period = c(25, 70), period_of_tracked_cycle = 20)
plot_track(tr, s)
```

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>

upb_grain_loglik — *Eruption-age likelihoods from single-zircon U-Pb dates (for astroMCBchron)*

Description

`upb_grain_loglik()` turns single-zircon dates of one or more ash beds into one eruption-age log-likelihood function per bed, for the `date_loglik` argument of `astroMCBchron`. Instead of a weighted mean, every grain is modelled as a draw from a mixture of

- a crystallisation interval: the grain crystallised uniformly between the eruption age *te* and *te* + *D* (antecrystic growth before eruption), with the duration *D* marginalised over a uniform prior between 0 and `D_max`;
- inherited or recycled grains (probability `p_inherit`): uniform over `W_inherit` before the eruption;
- grains with residual Pb loss (probability `p_pbloss`): uniform over `W_pbloss` after the eruption,

each convolved with the analytical uncertainty of the grain. The likelihood of *te* is computed on a grid, so no grain has to be rejected beforehand. With `D_max = 0` and no mixture components the result is the likelihood of a weighted mean; with a large `D_max` the eruption age approaches the youngest concordant grains (as in the Bayesian eruption-age estimator of Keller et al., 2018). Which regime applies depends on the ratio of the crystallisation interval to the analytical uncertainty of the grains, which is why the choice is left to the data rather than fixed beforehand.

The functions return a log-likelihood (not a posterior): no prior is placed on *te*, so the stratigraphic order and the astrochronology in `astroMCBchron` act as the prior. `astroMCBchron` normalises each function numerically and combines it with its outlier mixture.

Usage

```
upb_grain_loglik(
  grains,
  D_max = 0.6,
  p_inherit = 0.1,
  p_pbloss = 0.05,
  W_inherit = 20,
  W_pbloss = 3,
  step = 0.002,
  nD = 61
)
```

Arguments

grains	data.frame with columns id (bed), age (single-grain date, in age_unit) and either sd (1 sigma) or sd2 (2 sigma, analytical).
D_max	maximum crystallisation interval (age units). Default 0.6 (Ma).
p_inherit, p_pbloss	probabilities of inherited and Pb-loss grains. Defaults 0.1 and 0.05.
W_inherit, W_pbloss	widths of the inherited and Pb-loss components (age units). Defaults 20 and 3 (Ma).
step	grid step of the eruption age (age units). Default 0.002 (2 kyr for Ma).
nD	number of grid values for the marginalisation over *D*. Default 61.

Value

A list with loglik (named list of functions, one per bed, of the eruption age in age units) and summary (data.frame per bed: number of grains, maximum- likelihood eruption age, mean and 2 sd of the normalised likelihood, youngest grain and weighted mean with MSWD for comparison).

References

Keller, C.B., Schoene, B., Samperton, K.M., 2018. A stochastic sampling approach to zircon eruption age interpretation. Geochemical Perspectives Letters 8, 31-35.

Examples

```
set.seed(2)
g <- data.frame(id = rep(c("ashA", "ashB"), each = 8),
           age = c(430 + runif(8, 0, 0.1) + rnorm(8, 0, 0.04), 431 + runif(8, 0, 0.1) + rnorm(8, 0, 0.04)),
                sd = 0.04)
g$age[3] <- 433.2                                    # one inherited grain
gl <- upb_grain_loglik(g)
gl$summary
```

`wavelet_band_significance` *Significance of scale-averaged wavelet power (Torrence and Compo, section 4d)*

Description

Averages the wavelet power over a band of periods (a "scale-averaged wavelet power" time series, eq. 24 of Torrence and Compo, 1998) and tests it against a white or AR(1) red noise background. This is the test to use when one asks "is there significant power in this Milankovitch band at this depth?" without looking at every scale separately: the band average is much less noisy than the power at one scale.

Usage

```
wavelet_band_significance(
  object,
  period_range,
  background = c("ar1", "white"),
  siglvl = 0.95,
  method = c("exact", "torrence"),
  lag1 = NULL,
  delta_j0 = NULL,
  genplot = FALSE
)

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

Arguments

`object`	an object returned by `analyse_wavelet`.
`period_range`	numeric vector of length 2, lowest and highest period (in the units of the depth/time axis) of the band to be averaged, for example `c(0.3, 0.5)` for an obliquity band.
`background`	`"ar1"` (default, red noise) or `"white"`.
`siglvl`	significance level, default 0.95.
`method`	`"exact"` (default) or `"torrence"`, see Details.
`lag1`	lag-1 autocorrelation of the background. By default estimated from the record as in `wavelet_significance`.
`delta_j0`	decorrelation length of `method = "torrence"`. Default `0.6 * 6 / omega_nr`.
`genplot`	logical, plot the result.
`x`	an object returned by `wavelet_band_significance`.
`...`	passed to `plot`.

Details

Two ways of obtaining the null distribution are offered. Both are analytic and run instantaneously.

`method = "exact"` (default) The scale-average of the squared modulus of a linear transform of a Gaussian process is a weighted sum of independent exponential variables. The weights are the eigenvalues of a small (number of scales x number of scales) matrix that is computed from the wavelet and the background spectrum with the same Gaussian integrals that define the Morlet wavelet. This is exact for any `omega_nr`, any `dj`, any band and any background spectrum, and it needs no tabulated constants. The distribution of the weighted sum is approximated by a three-moment (shifted gamma) fit, which was checked against Monte Carlo simulations with the package's own `analyse_wavelet` (see the package tests).

`method = "torrence"` The published closed form: the band average is a scaled chi-square variable with $\nu = 2n_a(S_{avg}/S_{mid})\sqrt{1+(n_a\delta j/\delta j_0)^2}$ degrees of freedom (eq. 28), where n_a is the number of scales, S_{avg} the harmonic-type mean scale and S_{mid} the geometric mid-scale of the band, and δj_0 an empirical decorrelation length. Torrence and Compo only tabulated $\delta j_0 = 0.60$ for a Morlet wavelet with `omega_nr = 6`. Because the Morlet wavelet becomes narrower in log-scale when `omega_nr` increases, that constant under-estimates the degrees of freedom for larger values, so the test becomes too conservative (at `omega_nr = 10` only about 2% instead of 5% of white noise points exceed the 95% level). Here the default is rescaled as $\delta j_0 = 0.60 \cdot 6/\omega_0$, with ω_0 the value of `omega_nr`,, which was fitted against the exact method for `omega_nr` between 4 and 12 and is within about 10% of the exact degrees of freedom for bands wider than one octave. Use `method = "exact"` when the exact answer matters.

Only points whose whole period band lies outside the cone of influence are tested.

The band average is $\delta j \delta t / C_\delta \sum_j |W_j|^2/s_j$ (eq. 24). The factor C_δ is evaluated numerically for the `omega_nr` and `dj` of the object, normalised so that the average over all scales of unit-variance white noise is 1. It only sets the units of `power`; the significance decision does not depend on it.

The test is for a single point in depth. The band average is strongly autocorrelated along depth, so the number of significant points expected by chance is 5% of the points, but they occur in runs.

Value

An object of class `wr_band_significance`, a list with: `x` (depth/time axis), `power` (the scale-averaged power, in units of variance when the record is standardised), `crit` (significance level of the band average), `pvalue`, `significant` (logical, only points outside the cone of influence), `valid`, `dof` (equivalent chi-square degrees of freedom), `background_mean` (expected band average under the null), `period_range`, `n_scales`, `lag1`, `Cdelta`, `delta_j0`, `method`, `background`, `siglvl`. There are `print` and `plot` methods.

References

Torrence, C., Compo, G.P., 1998. A practical guide to wavelet analysis. Bulletin of the American Meteorological Society 79, 61-78.

See Also

`wavelet_significance`

Examples

```
set.seed(1)
z <- seq(0, 60, by = 0.1)
d <- data.frame(z, sin(2 * pi * z / 4) + as.numeric(arima.sim(list(ar = 0.5), length(z))))
wt <- analyse_wavelet(d, dj = 1 / 10, lowerPeriod = 0.4, upperPeriod = 30,
                      verbose = FALSE, omega_nr = 8)
bs <- wavelet_band_significance(wt, period_range = c(3, 5))
bs
```

---

wavelet_significance *Significance of wavelet and superlet power: analytic and surrogate (Monte Carlo) levels*

Description

Calculates the significance level of the spectral power in a wavelet or superlet scalogram against a noise background, and which cells of the scalogram are significant.

Two ways are available:

`method = "analytic"` (wavelets only) the near instantaneous method of Torrence and Compo (1998): the theoretical Fourier spectrum of a lag-1 autoregressive (red noise) process times a chi-square distribution with 2 degrees of freedom for the local wavelet power (their equations 16 and 18), and for the global (time averaged) spectrum the chi-square distribution with the degrees of freedom of their equation 23. No simulations are needed.

`method = "surrogate"` (wavelets and superlets) noise surrogates of the record are transformed in exactly the same way as the data. For every period the `siglvl` quantile of the power outside the cone of influence is taken, and averaged over the surrogates (scale wise critical level, after Weesp et al.). The `siglvl` quantile of the time averaged power of the surrogates gives the critical level of the global spectrum. The background can be white noise, AR(1) red noise, AR(1) noise estimated after removing the periods of the candidate cycles (so that the cycles that are tested do not inflate the noise background), or a power law ($1/f^{\beta}$) noise.

The result can be added to the plots with the argument `add_signif` of `plot_wavelet` and `plot_superlet`.

Usage

```
wavelet_significance(
  object,
  method = NULL,
  background = c("ar1", "white", "ar1_corrected", "powerlaw"),
  siglvl = 0.95,
  n_sim = 200,
  lag1 = NULL,
  notch_periods = NULL,
  notch_tol = 0.2,
  candidates = NULL,
  candidate_tol = 0.2,
```

```
  gamma = NULL,
  run_multicore = FALSE,
  n_cores = NULL,
  seed = NULL,
  verbose = FALSE
)
```

Arguments

object	A wavelet (analyse_wavelet) or superlet (analyse_superlet) object.
method	"analytic" (Torrence and Compo, wavelets only) or "surrogate". Default="analytic" for wavelets and "surrogate" for superlets.
background	Noise background for the surrogates: "ar1", "white", "ar1_corrected" or "powerlaw". The analytic method supports "ar1" and "white". Default="ar1".
siglvl	Significance level (0-1) Default=0.95.
n_sim	Number of surrogates (method = "surrogate") Default=200. Use 1000 or more for publication.
lag1	Lag-1 autocorrelation of the red noise. Default=NULL: estimated from the record (maximum likelihood for the surrogates, the lag-1 and lag-2 autocorrelation $(\alpha_1 + \sqrt{\alpha_2})/2$ of Torrence and Compo for the analytic method).
notch_periods	Periods of the candidate cycles that are removed before the AR(1) coefficient or the power law exponent is estimated ("ar1_corrected" and "powerlaw") Default=NULL.
notch_tol	Relative width of the removed bands, 0.2 removes periods within 20 percent of every notch period Default=0.2.
candidates	Optional periods of candidate cycles. For each, the p-value of the largest time averaged power within candidate_tol of the period is calculated against the surrogates (the largest value in the band is compared with the largest value of every surrogate, which corrects for searching a band).
candidate_tol	Relative width of the candidate bands Default=0.2.
gamma	Decorrelation factor of the wavelet used in the degrees of freedom of the global spectrum. The default NULL takes the value of Torrence and Compo (1998, table 2) for the wavelet of the object: 2.32 (Morlet), 1.17 (Paul), 1.43 (DOG of order 2); 2.32 is used for the other wavelets Default=2.32 (Torrence and Compo, 1998, for the Morlet wavelet).
run_multicore	Run the surrogates on several cores Default=FALSE.
n_cores	Number of cores (see analyse_wavelet) Default=NULL.
seed	Optional random seed. Results do not depend on the number of cores.
verbose	Print messages Default=FALSE.

Value

A list of class "wr_significance" with period, crit (critical power level per period, in the units of object$Power), gws, gws_crit (time averaged power outside the cone of influence and its critical level, per period), significant (logical matrix with the same layout as object$Power: power above the critical level and outside the cone of influence), valid (outside the cone of influence), candidates (data frame of p-values), and the settings and estimated noise parameters (lag1, beta).

References

Torrence, C., and G. P. Compo, 1998. A practical guide to wavelet analysis. Bulletin of the American Meteorological Society 79, 61-78.

Weesp analysis: surrogate significance with white, AR(1), signal corrected AR(1) and power law backgrounds.

Liu, Y., San Liang, X., Weisberg, R. H., 2007. Rectification of the bias in the wavelet power spectrum. Journal of Atmospheric and Oceanic Technology 24, 2093-2102.

Examples

```
set.seed(1)
z <- seq(0, 60, by = 0.1)
d <- data.frame(z, sin(2 * pi * z / 4) + as.numeric(arima.sim(list(ar = 0.5), length(z))))
wt <- analyse_wavelet(d, dj = 1 / 10, lowerPeriod = 0.4, upperPeriod = 30,
                      verbose = FALSE, omega_nr = 6)
sig <- wavelet_significance(wt)                    # analytic, instantaneous
sig$period[which(sig$gws > sig$gws_crit)]        # periods with significant global power

sig2 <- wavelet_significance(wt, method = "surrogate", background = "ar1", n_sim = 20, seed = 1)
```

---

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 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

```
wavelet_uncertainty(
  tracked_cycle = NULL,
  period_of_tracked_cycle = NULL,
  wavelet = NULL,
  multi = 1,
  verbose = FALSE,
  genplot_time = FALSE,
  genplot_uncertainty = FALSE,
  genplot_uncertainty_wt = FALSE,
  keep_editable = FALSE,
  palette_name = "rainbow",
  color_brewer = "grDevices",
```

```
  colour_brewer = NULL
)
```

Arguments

tracked_cycle	Curve of the cycle tracked using the track_period_wavelet function Any input (matrix or data frame) 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 analyse_wavelet function.
multi	multiple of the standard deviation to be used for defining uncertainty Default=1.
verbose	Print text Default=FALSE.
genplot_time	plot time curves with a upper and lower uncertainty based on Gabor uncertainty principle applied to the continuous wavelet transform using a Morlet wavelet, which uses the omega number (the central angular frequency of the Morlet 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 Morlet wavelet, which uses the omega number (the central angular frequency of the Morlet 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 Morlet 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 black curve is the curve tracked using the 'Default=tracked_cycle_curve function Default=TRUE
keep_editable	Keep option to add extra features after plotting Default=FALSE
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function

color_brewer — Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices

colour_brewer — British spelling of color_brewer; if given it overrides color_brewer.

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 deviation 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.

Author(s)

Code based on the "analyze.wavelet" function of the 'WaveletComp' R package and "wt" function of the 'biwavelet' R package which are based on the wavelet 'MATLAB' code written by Christopher Torrence and Gibert P. Compo (1998). The assignment of the standard deviation of the uncertainty of the wavelet is based on the work of Gabor (1946) and Russell et al., (2016)

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. https://paos.colorado.edu/research/wavelets/bams_79_01_0061.pdf

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.https://doi.org/10.1049/ji-3-2.1946.0074

Russell, Brian, and Jiajun Han.Jean Morlet and the continuous wavelet transform. CREWES Res. Rep 28 (2016): 115. https://www.crewes.org/Documents/ResearchReports/2016/CRR201668.pdf

Morlet, Jean, Georges Arens, Eliane Fourgeau, and Dominique Glard. Wave propagation and sampling theory—Part I: Complex signal and scattering in multilayered media. Geophysics 47, no. 2 (1982): 203-221.

J. Morlet, G. Arens, E. Fourgeau, D. Giard; Wave propagation and sampling theory; Part II, Sampling theory and complex waves. Geophysics 1982 47 (2): 222–236.

Examples

```
#calculate the Gabor uncertainty derived mathematical uncertainty of the
#magnetic susceptibility record of the Sullivan core of Pas et al., (2018)

mag_wt <- analyse_wavelet(data = mag,
dj = 1/100,
lowerPeriod = 0.1,
upperPeriod = 254,
verbose = FALSE,
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)",
#                                palette_name="rainbow",
#                                color_brewer= "grDevices")

#Instead of tracking, the tracked solution data set 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 = FALSE,
  genplot = FALSE,
  keep_editable=FALSE
)
```

```
mag_track_complete <- loess_auto(time_series = mag_track_complete,
 genplot = FALSE, print_span = FALSE,keep_editable=FALSE)

uncertainty <- wavelet_uncertainty(
  tracked_cycle = mag_track_complete,
  period_of_tracked_cycle = 405,
  wavelet = mag_wt,
  multi=1,
  verbose = FALSE,
  genplot_time = FALSE,
  genplot_uncertainty = FALSE,
  genplot_uncertainty_wt = FALSE,
  keep_editable=FALSE,
  palette_name="rainbow",
  color_brewer= "grDevices"
)
```

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

Description

Data sets for testing the 'WaverideR' R package:
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 `astrosignal_example` data set consists of pre-generated ETP (eccentricity-tilt-precession)
data set based on the p-0.5t la2004 solution and was generated using
the etp function of the 'astrochron' R package

The `depth_rank_example` data set is synthetic succession of sedimentary
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 `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)

The `mag` data set is the magnetic susceptibility record of Pas et al. (2018)

The `mag_track_solution` is the period of the 405 kyr eccentricity cycle in
the magnetic susceptibility record of from Pas et al. (2018)

The `TSI` data set is the Total Solar Irradiance record of Steinhilber et al. (2012)

The `Bisciaro_Mg_wt_track` data set is the 110-kyr (short eccentricity)
cycle tracked in the wavelet scalogram of the Magnesium (XRF) record of Arts (2014)

The Bisciaro_Mn_wt_track data set is the 110-kyr (short eccentricity) cycle tracked in the wavelet scalogram of the Manganese (XRF)record of Arts (2014)

The Bisciaro_al_wt_track data set is the 110-kyr (short eccentricity) cycle tracked in the wavelet scalogram of the Aluminum (XRF) record of Arts (2014)

The Bisciaro_ca_wt_track data set is the 110-kyr (short eccentricity) cycle tracked in the wavelet scalogram of the Calcium (XRF) record of Arts (2014)

The Bisciaro_sial_wt_track data set is the 110-kyr (short eccentricity) cycle tracked in the wavelet scalogram of the Silicon/Aluminum (XRF) record of Arts (2014)

The Bisciaro_XRF is the XRF data set of Arts (2014)

The anchor_points_Bisciaro_al data set consist of the tie points between the Bisciaro_al record of Arts (2014) and the la2011 solution of laskar et al. (20111)

The GTS_info data set contains the color coding and ages and uncertainties of Geologic Time Scale 2020 of Ogg et al. (2021)

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>

Stephen R. Meyers,Cyclostratigraphy and the problem of astrochronologic testing, Earth-Science Reviews,Volume 190,2019,Pages 190-223,ISSN 0012-8252 <doi:10.1016/j.earscirev.2018.11.015>

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. <doi:10.1051/0004-6361:20041335>

Laskar, J., M. Gastineau, J. B. Delisle, A. Farrés, and A. Fienga (2011b), Strong chaos induced by close encounters with Ceres and Vesta, Astron. Astrophys., 532, L4,<doi:10.1051/0004-6361/201117504>

M.C.M. Arts, 2014, Magnetostratigrpahy and geochemical analysis of the early Miocene Bisciaro Formation in the Contessa Valley (Northern Italy). Unpublished Bsc. thesis

Ogg, Gabi & Ogg, James & Gradstein, Felix. (2021). Recommended color coding of stages - Appendix 1 from Geologic Time Scale 2020.

`win_fft` *Windowed fft based spectral analysis*

Description

The `win_fft` function conducts a windowed spectral analysis based using the fft

Usage

```
win_fft(
  data = NULL,
  padfac = 5,
  window_size = NULL,
  run_multicore = FALSE,
  n_cores = NULL,
  genplot = FALSE,
  x_lab = c("depth (m)"),
  y_lab = c("frequency cycle/metre"),
  plot_res = 1,
  perc_vis = 0,
  freq_max = NULL,
  freq_min = NULL,
  palette_name = "rainbow",
  color_brewer = "grDevices",
  keep_editable = FALSE,
  verbose = FALSE,
  dev_new = FALSE,
  colour_brewer = NULL
)
```

Arguments

`data`	Input data set should consist of a matrix with 2 columns with first column being depth and the second column being a proxy
`padfac`	Pad record with zero, zero padding smooths out the spectra
`window_size`	size of the running window
`run_multicore`	Run function using multiple cores `Default="FALSE"`

n_cores	Number of cores used when 'run_multicore = TRUE' ('Default = NULL': an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set 'options(waverider.n_cores = ...)' to change the default. A value > 1 also switches on parallel computing. Results do not depend on the number of cores.
genplot	Generate plot Default="FALSE"
x_lab	label for the y-axis Default="depth"
y_lab	label for the y-axis Default="sedrate"
plot_res	plot 1 of 8 options option 1: Amplitude matrix, option 2: Power matrix, option 3: Phase matrix, option 4: AR1_CL matrix, option 5: AR1_Fit matrix , option 6: AR1_90_power matrix, option 7: AR1_95_power matrix, option 8: AR1_99_power matrix, Default=1
perc_vis	Cutoff percentile when plotting Default=0
freq_max	Maximum frequency to plot
freq_min	Minimum frequency to plot
palette_name	Name of the color palette which is used for plotting. The color palettes than can be chosen depends on which the R package is specified in the color_brewer parameter. The included R packages from which palettes can be chosen from are; the 'RColorBrewer', 'grDevices', 'ColorRamps' and 'Viridis' R packages. There are many options to choose from so please read the documentation of these packages Default=rainbow. The R package 'viridis' has the color palette options: "magma", "plasma", "inferno", "viridis", "mako", and "rocket" and "turbo" To see the color palette options of the The R pacakge 'RColorBrewer' run the wr_brewer_palettes() function The R package 'colorRamps' has the color palette options:"blue2green", "blue2green2red", "blue2red", "blue2yellow", "colorRamps", "cyan2yellow", "green2red", "magenta2green", "matlab.like", "matlab.like2" and "ygobb" The R package 'grDevices' has the built in palette options:"rainbow", "heat.colors", "terrain.colors","topo.colors" and "cm.colors" To see even more color palette options of the The R pacakge 'grDevices' run the grDevices::hcl.pals() function
color_brewer	Name of the R package from which the color palette is chosen from. The included R packages from which palettes can be chosen are; the RColorBrewer, grDevices, ColorRamps and Viridis R packages. There are many options to choose from so please read the documentation of these packages. "Default=grDevices
keep_editable	Keep option to add extra features after plotting Default=FALSE
verbose	Print text Default=FALSE.
dev_new	Opens a new plotting window to plot the plot, this guarantees a "nice" looking plot however when plotting in an R markdown document the plot might not plot Default=FALSE
colour_brewer	British spelling of color_brewer; if given it overrides color_brewer.

Value

Returns a list which contains 10 elements element 1: Amplitude matrix element 2: Power matrix element 3: Phase matrix element 4: AR1_CL matrix element 5: AR1_Fit matrix element 6:

AR1_90_power matrix element 7: AR1_95_power matrix element 8: AR1_99_power matrix element 9: depth element 10: y_axis If genplot is `Default=TRUE` then a plot of one of the elements 1:8 is plotted

Author(s)

Based on the periodogram function of the 'astrochron' R package.

References

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

Examples

```
#Conduct a windowed ftt on the magnetic susceptibility record
#of the Sullivan core of Pas et al., (2018).

mag_win_fft <- win_fft(data= mag,
                   padfac = 5,
                   window_size = 12.5,
                   run_multicore = FALSE,
                   genplot = FALSE,
                   x_lab = c("depth (m)"),
                   y_lab = c("frequency cycle/metre"),
                   plot_res = 1,
                   perc_vis = 0.5,
                   freq_max = 5,
                   freq_min = 0.001,
                   palette_name ="rainbow",
                   color_brewer= "grDevices",
                   keep_editable=FALSE,
                   verbose=FALSE,
                   dev_new=FALSE)
```

`win_timeOpt` *Windowed timeOpt sedimentation rate estimation*

Description

The `win_timeOpt` function for conducts a widowed timeOpt sedimentation rate estimation This function is based on the `eTimeOpt` but allows for multithreaded analysis speeding up the process of conducting a Windowed timeOpt sedimentation rate estimation

Usage

```
win_timeOpt(
  data = NULL,
  window_size = 10,
  sedmin = 0.5,
  sedmax = 2,
  numsed = 100,
  limit = FALSE,
  fit = 2,
  fitModPwr = TRUE,
  flow = NULL,
  fhigh = NULL,
  roll = 10^6,
  targetE = c(405.7, 130.7, 123.8, 98.9, 94.9),
  targetP = c(20.9, 19.9, 17.1, 17.2),
  detrend = TRUE,
  normalize = TRUE,
  linLog = 1,
  run_multicore = FALSE,
  n_cores = NULL,
  verbose = FALSE,
  normalise = NULL
)
```

Arguments

data	Input data set should consist of a matrix with 2 columns with the first column being depth and the second column being a proxy Default=NULL
window_size	size of the moving window in metres Default=15
sedmin	Minimum sedimentation rate for investigation (cm/ka). Default=0.1
sedmax	Maximum sedimentation rate for investigation (cm/ka). Default=1
numsed	Number of sedimentation rates to investigate in optimisation grid. Default=100
limit	Limit evaluated sedimentation rates to region in which full target signal can be recovered? .Default=FALSE
fit	Test for (1) precession amplitude modulation or (2) short eccentricity amplitude modulation? Default=2
fitModPwr	Include the modulation periods in the spectral fit? Default=TRUE
flow	Low frequency cut-off for Taner bandpass (half power point in cycles/ka) Default=TRUE
fhigh	High frequency cut-off for Taner bandpass (half power point; in cycles/ka) Default=NULL
roll	Taner filter roll-off rate, in dB/octave. Default=c(10^6)
targetE	A vector of eccentricity periods to evaluate (in ka). These must be in order of decreasing period, with a first value of 405 ka. Default= "c(405.7, 130.7, 123.8, 98.9, 94.9)"
targetP	A vector of precession periods to evaluate (in ka). These must be in order of decreasing period. Default=c(20.9, 19.9, 17.1, 17.2)

detrend	Remove linear trend from data series? Default=TRUE
normalize	normalize the r2 curves of individual timeOpt runs Default=TRUE
linLog	Use linear or logarithmic scaling for sedimentation rate grid spacing? (0=linear, 1=log; default value is 1) Default=1
run_multicore	Run function using multiple cores Default=FALSE
n_cores	Number of cores used when 'run_multicore = TRUE' ('Default = NULL': an automatic, safe number: physical cores minus one, at most 4, and limited by the free memory). Set 'options(waverider.n_cores = ...)' to change the default. A value > 1 also switches on parallel computing. Results do not depend on the number of cores.
verbose	print text Default=FALSE
normalise	British spelling of normalize; if given it overrides normalize.

Value

Returns a list which contains 10 elements element 1: r_2_envelope matrix element 2: r_2_power matrix element 3: r_2_opt matrix element 4: r_2_envelope_avg element 5: r_2_opt_avg element 6: depth element 7: y_axis element 8: linLog value

Author(s)

Based on the eTimeOpt function of the 'astrochron' R package.

References

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

Examples

```
#Conduct a windowed timeOpt on the magnetic susceptibility record
#of the Sullivan core of Pas et al., (2018).
mag_win_timeOpt <-win_timeOpt(
data = mag[1:250, ],
window_size = 6,
sedmin = 0.1,
sedmax = 1,
numsed = 5,
limit = FALSE,
fit = 2,
fitModPwr = TRUE,
flow = NULL,
fhigh = NULL,
roll = 10 ^ 6,
targetE = c(405.7, 130.7, 123.8, 98.9, 94.9),
targetP = c(20.9, 19.9, 17.1, 17.2),
detrend = TRUE,
normalize =TRUE,
linLog = 1,
```

```
run_multicore =FALSE,
verbose=FALSE)
```

`wr_coloured_noise` *Colour noise and palette helpers*

Description

`wr_coloured_noise` generates AR(1) red noise (replaces `colorednoise::colored_noise`); `wr_brewer_palettes` lists the available ColorBrewer palette names and their maximum number of colours.

Usage

```
wr_coloured_noise(timesteps, mean, sd, phi)

wr_brewer_palettes()

wr_colored_noise(timesteps, mean, sd, phi)
```

Arguments

`timesteps`	Number of values to generate.
`mean`	Mean of the noise.
`sd`	Standard deviation of the noise.
`phi`	Lag-1 autocorrelation (0-1).

Value

`wr_coloured_noise`: numeric vector. `wr_brewer_palettes`: data frame with the palettes.

Examples

```
set.seed(1)
x <- wr_coloured_noise(100, mean = 0, sd = 1, phi = 0.8)
head(wr_brewer_palettes())
```

zeeden2023_table1 *Devonian reference values of Zeeden et al. (2023), EPSL 621, 118348, Table 1*

Description

Devonian reference values of Zeeden et al. (2023), EPSL 621, 118348, Table 1

Usage

```
zeeden2023_table1()
```

Index

add_wavelet, 10
add_wavelet_avg, 15
age_model_zeeden, 19
amtm_prewhiten, 20, *20*, *22*
amtm_spectrum, 21, *21*, *260*
analyse_Awavelet, 24, *268, 269*
analyse_chirp_eha, 28, *31*, *33*, *37*, *110*, *276*, *325*, *357*, *364, 365*
analyse_chirp_wavelet, 31, *365*
analyse_chirplet_sst, *30*, 33, *33*, *265*, *356*
analyse_conceft, 35, *342, 343*, *375*
analyse_dechirped, 37, *66*, *107–109*, *278*
analyse_superlet, 39, *45*, *58*, *294*, *370*, *383*
analyse_superlet_sst, *35*, 43, *265*, *342, 343*, *355*, *360*, *365*, *375*
analyse_wavelet, *10*, *15*, *25*, *35*, *41*, *45*, 46, *53–55*, *119*, *143*, *149*, *151*, *156*, *184*, *186*, *188*, *190*, *192*, *194*, *200*, *247*, *255*, *265, 266*, *291*, *300*, *302*, *328*, *344*, *365*, *370*, *373*, *375*, *380, 381*, *383*, *385*
analyse_wavelet_coherence, 50, *306, 307*
analyse_wavelet_sst, *34*, *37*, *43*, *45*, 53, *342, 343*, *355*, *360*, *375*
analyse_Xsuperlet, 56, *314, 315*
analyse_Xwavelet, 60, *318, 319*
analyze_Awavelet *(*analyse_Awavelet*)*, 24
analyze_chirp_eha *(*analyse_chirp_eha*)*, 28
analyze_chirp_wavelet *(*analyse_chirp_wavelet*)*, 31
analyze_chirplet_sst *(*analyse_chirplet_sst*)*, 33
analyze_conceft *(*analyse_conceft*)*, 35
analyze_superlet *(*analyse_superlet*)*, 39
analyze_superlet_sst *(*analyse_superlet_sst*)*, 43
analyze_wavelet *(*analyse_wavelet*)*, 46
analyze_wavelet_coherence *(*analyse_wavelet_coherence*)*, 50
analyze_wavelet_sst *(*analyse_wavelet_sst*)*, 53
analyze_Xsuperlet *(*analyse_Xsuperlet*)*, 56
analyze_Xwavelet *(*analyse_Xwavelet*)*, 60
anchor2time, 65, *76*, *78*
anchor_points_Bisciaro_al, 64
anchor_points_grey, 64
as_wr_track, 66, *107*, *109*
asm, *346*
astro_anchor, *64, 65*, 67, *67*, *75–78*, *91*, *262*
astro_anchor_app, 74
astro_anchor_interactive, *74*, 76, *76*
astro_ctest, 79
astro_from_peaks, 80
astro_from_periods, 81
astro_invert, 82
astro_mc, 83
astro_obliquity, 84
astro_periods_tracking, 84, *108*, *356*, *361*
astro_term_search, 85
astro_terms, *84*, 86, *358*
astrochron-package, *92*
astrogeo22, 86
astroMCBchron, 87, *378*
astrosignal_example, *64*, 91, *91*

bayes_changepoint, 92, *92*, *94*, *273*, *377*
bayes_changepoint_r2, 94, *94*
bayes_spectrum, 96
Bisciaro_al_wt_track, 97
Bisciaro_ca_wt_track, 98
Bisciaro_Mg_wt_track, 98
Bisciaro_Mn_wt_track, 99
Bisciaro_sial_wt_track, 99
Bisciaro_XRF, 100
bispectral_analysis, 100, *100*, *274, 275*
bispectral_surrogate, 104, *104*
bundling_ratios, 105

calculate_orbit9, 106
cbind, *328*
chaos_scenario, 106
check_agemodel, *66*, 107, *260*
chirp_eha_significance, 110, *276*
coco_adaptive, 111
coco_adaptive_evaluate, 112
coco_classic, 113
coco_fixed_target_weights, 114
coco_prepare_regular_data, 114
coco_target_fixed, 115
coco_target_from_periods, 116
coco_target_from_series, 117
compare_published, 118
completed_series, *87*, 118, *118*, *184*
ctest, 120
ctest_combine, 121
ctest_earthmoon, 122
ctest_input, 123
curve2sedrate, 124
curve2time, 125, *198*, *375*
curve2time_unc, 127
curve2time_unc_anchor, 133
curve2tune, 143
cvcoco, 145, *145*, *277*
cycle_band, 147
cycles_to_k, 148

days_per_year, 149
delpts_tracked_period_wt, 149, *149*, *370*, *372*
depth_rank_example, 151, *151*
detect_transients, 151
dur_gaps, 153
dynamic_extraction, 155, *155*
dynamic_filter, 157, *157*, *159*, *160*, *279*
dynamic_filter_controls, *157*, 159, *159*
dynot, 161, *161*, *280*

earth_moment_of_inertia, 163
earth_omega, 164
earthmoon_angular_momentum, 164
earthmoon_from_precession, 165
earthmoon_state, 165
eAsm, *201*, *346*
ecoco_adaptive, 167, *367*
ecoco_best_per_window, 168
ecoco_blocked, 168, *367*
ecoco_classic, 171, *355*, *358*
ecoco_interleaved, 172, *367*
ecoco_track, 174, *368*
ecoco_track_ridge, 177, *368*
eha_log2, 177, *282*, *283*
em_state, 179
eTimeOpt, *312*, *313*, *392*, *394*
etp, *69*, *91*, *92*, *388*
example_pridoli, 180
expSedRamp, 180
extract_amplitude, 181
extract_power, 183
extract_power_stable, 186
extract_signal, *53*, *55*, *77*, 187
extract_signal_stable, 190
extract_signal_stable_V2, 191
extract_signal_standard_deviation, 193

fft_frequency_uncertainty, 197, *198*
fft_uncertainty, 198
flmw, 199, *199*, *203*, *290*, *291*
flmw_path, *201*, 203

gaussian_bandpass, 204, *204*
geo_col, 206
geo_loc, 210
geo_mid, 214
getLaskar, *67*, *74*
grey, *64*, *91*, 219
grey_track, 219
GTS_info, 220

hcl.colors, *299*, *367*

identify, *366*
integratePower, *184*, *186*

k_from_bundling, 220
k_from_cycles, 221
k_from_period, 223
k_from_ratio, 223
k_hypotheses, 224
k_scan, 225

la2004_extrap, 226
lag_1, 226, *226*
list, *328*
lithlog_disc, *151*, 228
locator, *78*
lod_from_moon_distance, 229
lod_hours, 230

loess_auto, *184*, 230

mag, 232
mag_track_solution, 232
max_detect, *75*, *77*, 233, *233*
mesa_average, 234, *288*
mesa_evolutive, *234*, 235, *287, 288*, *368, 369*
mesa_spectrum, *235, 236*, 237, *289, 290*
min_detect, 239, *239*
minimal_tuning, *155*, 240
modBchron, *87*, *90*, 242
modBchron_predict, 246
model_red_noise_wt, 247, *247*, *255*, *344, 345*
months_per_year, 248
moon_distance_from_lod, 249
moon_distance_from_months, 249

omega_from_moon_distance, 250
orbit9, 250
orbit9_at_age, 251
orbital_periods, 251

pda, 252, *252*
pda_legacy, *161*, 253, *254*
percentile_from_red_noise, 255, *255*
periodogram, *392*
plot.astroMCBchron, 256
plot.coco_adaptive, 257
plot.coco_classic, 257
plot.ctest, 258
plot.ecoco_adaptive, 258
plot.ecoco_classic, 259
plot.wr_band_significance *(*wavelet_band_significance*)*, 380
plot_agemodel_check, *109*, 259
plot_amtm_spectrum, *22*, 260, *260*
plot_astro, 261
plot_astro_anchor, *76*, *78*, 262
plot_astro_evidence, *30*, 265, *357*
plot_avg_wavelet, 266
plot_Awavelet, 268
plot_bayes_changepoint, 273
plot_bispectral, *103*, 274, *274*
plot_chirp_eha, 276
plot_cvcoco, *146*, 277
plot_dechirped, *260*, 278
plot_dynamic_filter, *158*, 279, *279*
plot_dynot, *162*, 280, *280*
plot_ecoco_cv, 281
plot_eha_log2, 282
plot_mesa_evolutive, *235, 236*, 287
plot_mesa_spectrum, 289
plot_sed_model, 290, *290*
plot_spectral_moments, 292, *341*
plot_superlet, 294, *382*
plot_test_k, 298, *353*
plot_track, 299
plot_wavelet, *30*, *35*, *38*, *45*, *53*, *277*, *279*, *284*, *289*, *296*, *299*, 300, *382*
plot_wavelet_coherence, 306
plot_win_fft, 310, *310*
plot_win_timeOpt, 312, *312*
plot_Xsuperlet, 314
plot_Xwavelet, 318
precession_frequency, 322
predict.astroMCBchron, 323
print.astro_from_peaks, 324
print.ctest, 324

rate_gram, 325, *357*
resonance_state, 326
retrack_wt_MC, *87, 88*, *127*, *133, 134*, 326, *326*
rho1_evolutive, 335, *335*
runApp, *76*

sedRamp, *181*
sedrate2time, *126*, *128*, *135*, *144*, *241*, *338*
sedrate2tune, 337
spectral_moments, *292, 293*, 339, *339*
sst_significance, 342
sum_power_sedrate, *290, 291*, 344, *344*
swa_background, 347, *348, 349*
swa_spectrum, 348

taner, *75*
term_sensitivity, 351
test_tuned, *84*, *298, 299*, 352, *354*, *358*
test_tuned_check, 354
traceFreq, *156*, *371*, *374*
track_astro_auto, *30*, *37, 38*, *107–109*, *151*, *265*, *299*, 355, *360*, *363*
track_astro_hmm, *30*, *33*, *35*, *37*, *84*, *151*, *265, 266*, *299*, *355, 356*, 357, *357*, *363*, *365*
track_astro_iterative, 362

track_chirp_hmm, *31*, *33*, *37*, *109*, *325*, *357*, 364
track_ecoco, 366
track_mesa, *235, 236*, *288*, 368
track_period, *118, 119*, *184*, *186*, *366*, *368*, 370
track_period_wavelet, *53*, *66*, *87*, *97–99*, *118*, *124, 125*, *143*, *149*, *183, 184*, *194*, *232*, 372, *372*, *376*, *385*
track_ridge_hmm, *299*, *359, 360*, 375
TSI, 378

upb_grain_loglik, 378

wavelet_band_significance, 380
wavelet_significance, *296*, *302*, *380, 381*, 382
wavelet_uncertainty, *197–199*, 384, *384*
WaverideR *(*WaverideR-package*)*, 6
WaverideR-package, 6
WaverideR_Datasets, 388
win_fft, *310*, *311*, 390, *390*
win_timeOpt, *313*, 392, *392*
wr_brewer_palettes *(*wr_coloured_noise*)*, 395
wr_colored_noise *(*wr_coloured_noise*)*, 395
wr_coloured_noise, 395

zeeden2023_table1, *354*, 396