

Package ‘CharAnalysis’

May 7, 2026

Type Package

Title Peak Detection and Fire History from Sediment-Charcoal Records

Version 2.0.3

Date 2026-04-29

Description A program for reconstructing local fire histories from high-resolution, continuously sampled lake-sediment charcoal records. ‘CharAnalysis’ decomposes a charcoal record into low- and high-frequency components and uses locally defined thresholds to separate fire signal from noise, following the approach of Higuera et al. (2009) <[doi:10.1890/07-2019.1](https://doi.org/10.1890/07-2019.1)>, with underlying assumptions and rationale described in Higuera et al. (2010) <[doi:10.1071/WF09134](https://doi.org/10.1071/WF09134)>. The package is designed for macroscopic charcoal records with contiguous sampling fine enough to resolve individual fire events, and is not appropriate for low-resolution or discontinuously sampled records. See the package URL for the User’s Guide and application examples.

License GPL-3

Encoding UTF-8

URL <https://github.com/phiguera/CharAnalysis>,
<https://doi.org/10.5281/zenodo.19304064>

BugReports <https://github.com/phiguera/CharAnalysis/issues>

Depends R (>= 4.0.0)

Imports MASS, graphics, stats, utils, zoo

Suggests ggplot2 (>= 3.4.0), patchwork, ggtext, readxl, knitr,
rmarkdown

VignetteBuilder knitr

RoxygenNote 7.3.3

NeedsCompilation no

Author Philip Higuera [aut, cre] (ORCID:
<<https://orcid.org/0000-0003-1278-4095>>)

Maintainer Philip Higuera <philip.higuera@umontana.edu>

Repository CRAN

Date/Publication 2026-05-03 15:00:09 UTC

Contents

CharAnalysis	2
char_lowess	3
char_parameters	5
char_peak_id	6
char_plot	7
char_pretreatment	9
char_smooth	10
char_thresh_global	12
char_thresh_local	13
char_validate_params	15
char_write_results	16
gaussian_mixture_em	18

Index **19**

CharAnalysis	<i>Run the full CharAnalysis pipeline</i>
--------------	---

Description

Top-level wrapper that calls each analytical stage in sequence and returns all intermediate and final results as a single named list.

Usage

```
CharAnalysis(file_name = NULL)
```

Arguments

file_name	Path to the *_charParams.csv (or .xlsx) parameter file. If omitted (or NULL), a system file-picker dialog is opened so you can navigate to the file interactively – matching the MATLAB behaviour of typing CharAnalysis with no argument. In non-interactive sessions (e.g. scripts, batch jobs) the argument is required and an error is thrown if it is missing.
-----------	---

Value

Named list with the following elements:

charcoal List of raw and resampled series. After Phase 2: includes accIS (smoothed background) and peak (C_peak, either residuals or ratios). After Phase 3: also includes charPeaks ($[N \times T]$ binary peak matrix), charPeaksThresh, peaksTotal, and threshFRI. After Phase 4: also includes peakInsig, peakMagnitude, smoothedFireFreque, peaksFreque.

- pretreatment** Pretreatment parameter list (possibly updated by [char_pretreatment()]-e.g. yrInterp auto-set, zoneDiv end-value corrected).
- smoothing** Smoothing parameter list.
- peak_analysis** Peak-analysis parameter list.
- results** Results / output parameter list.
- site** Character string: site name.
- gap_in** Integer matrix (nGaps x 2) of missing-value gap indices.
- char_thresh** Threshold list returned by [char_thresh_global()] or [char_thresh_local()]. Contains pos, neg, SNI, GOF, and (after Phase 3) minCountP – the $[N \times T]$ matrix of Shuie-Bain p-values.
- post** Post-processing list from [char_post_process()]: FRI series, smoothed FRI curve, per-zone Weibull statistics, and the assembled char_results output matrix ($N \times 33$).
- char_results** Numeric matrix ($N \times 33$) matching the MATLAB charResults output exactly (alias of post\$char_results).

See Also

[char_parameters()], [char_validate_params()], [char_pretreatment()], [char_smooth()], [char_thresh_global()], [char_thresh_local()], [char_peak_id()], [char_post_process()]

Examples

```
# Run the full pipeline on the bundled example dataset:
params_file <- system.file("validation", "CO_charParams.csv",
                           package = "CharAnalysis")
out <- CharAnalysis(params_file)
# Phase 2 outputs
head(data.frame(ageTop_i = out$charcoal$ybpI,
                charAcc_i = out$charcoal$accI,
                charBkg_i = out$charcoal$accIS,
                charPeak_i = out$charcoal$peak))
# Phase 3 outputs
sum(out$charcoal$charPeaks[, ncol(out$charcoal$charPeaks)])
```

char_lowess	<i>Locally-weighted linear regression matching MATLAB charLowess.m / smooth()</i>
-------------	---

Description

Pure-R implementation of the shifted-window lowess algorithm from MATLAB's charLowess.m v2.0, which itself replicates MATLAB's smooth(y, k, 'lowess') from the Curve Fitting Toolbox. Replaces the previous wrapper around [lowess](#), which used a shrinking window at record boundaries and different tricubic weight normalisation, causing systematic differences of $\sim 1e-3$ in charBkg.

Usage

```
char_lowess(y, x = NULL, span = 0.1, iter = 0L)
```

Arguments

<code>y</code>	Numeric vector of values to smooth (NaN-free; bridge NaNs before calling and restore afterward, matching the MATLAB workflow).
<code>x</code>	Numeric vector of x-positions (same length as <code>y</code>). If <code>NULL</code> , <code>seq_along(y)</code> is used (index coordinates).
<code>span</code>	Smoothing span. <ul style="list-style-type: none"> • <code>span >= 1</code>: number of data points. • <code>span < 1</code>: fraction of data length. Converted to integer $k = \max(3, \text{round}(\text{span} * n))$ or $\max(3, \text{round}(\text{span}))$, matching <code>charLowess.m</code> lines 63-68.
<code>iter</code>	Number of bisquare robustness passes after the initial fit. <ul style="list-style-type: none"> • <code>iter = 0</code>: plain lowess (1 total WLS pass, no robustness). • <code>iter = 4</code>: robust lowess, matching MATLAB's 'r<code>lowess</code>' which uses <code>nIter = 5</code> total iterations (1 initial + 4 updates).

Details

Algorithm For each point i , a window of exactly k points is selected: the window is centred on i at interior points and SHIFTED (not shrunk) at record boundaries, maintaining k neighbours throughout. Each neighbour j receives a tricubic weight:

$$w_j = \left(1 - \left(\frac{|i - j|}{d_{\max}} \right)^3 \right)^3$$

where d_{\max} is the distance to the **furthest** point in the window (not the half-window radius). A weighted least-squares line is fitted and evaluated at i . This matches `charLowess.m` lines 109-147 exactly.

For `iter > 0` ('r`lowess`'), bisquare robustness weights are computed from the residuals after each WLS pass and multiplied into the tricubic weights for the next pass (matching `charLowess.m` lines 151-157).

Why not `stats::lowess()`? `stats::lowess()` trims (shrinks) the window at boundaries and normalises tricubic weights by the half-window radius hw . The two differences compound to $\sim 1e-3$ absolute error in `charBkg` for the CO dataset (500 yr / 15 yr = 33-point window, 17-point boundary zone).

Value

Numeric vector of smoothed values, same length as `y`.

See Also

[`char_smooth()`], [`char_thresh_local()`]

char_parameters	<i>Read CharAnalysis parameter and data files</i>
-----------------	---

Description

Reads the *_charParams.csv (or .xlsx) parameter file and the companion charcoal data file, then unpacks all analysis parameters into named lists that mirror the MATLAB struct layout.

Usage

```
char_parameters(file_name)
```

Arguments

file_name Path to the *_charParams.csv (or .xlsx) parameter file.

Details

****CSV convention**** (unchanged from MATLAB v1.1): the parameter file is named <site>_charParams.csv and the companion data file is <site>_charData.csv in the same directory. The site name is the basename with the trailing _charParams.csv (15 characters) removed, mirroring MATLAB's fileName(1:end-15) idiom.

****Parameter vector layout****: column 3 ("Parameters") of the CSV, rows 2-26 (25 data rows after the header), maps to positions 1-25 of the internal charParams vector exactly as in the MATLAB codebase. Unused zoneDiv slots are filled with -9999 in the CSV (sentinel) or NaN in Excel; both are stripped before the list is returned.

Value

A named list with elements:

char_data Numeric matrix (n_samples x 6+): cmTop, cmBot, ageTop, ageBot, charVol, charCount.

pretreatment List: zoneDiv, yrInterp, transform.

smoothing List: method, yr.

peak_analysis List: cPeak, threshType, threshMethod, threshValues, minCountP, peakFreq, bkgSens.

results List: saveFigures, save, allFigures.

site Character string: site name derived from the filename stem.

See Also

[char_validate_params()], [char_pretreatment()], [CharAnalysis()]

Examples

```
# Read a bundled example parameter file:
params_file <- system.file("validation", "CO_charParams.csv",
                           package = "CharAnalysis")
p <- char_parameters(params_file)
p$pretreatment$yrInterp # interpolation interval (yr)
p$smoothing$method      # smoothing method index (1-5)
```

char_peak_id	<i>Identify charcoal peaks and apply minimum-count screening</i>
--------------	--

Description

Mirrors CharPeakID.m from the MATLAB v2.0 codebase. For each threshold column, samples where C_{peak} exceeds the threshold are flagged, consecutive exceedances are collapsed to a single event (keeping the **last** sample of each run, i.e. the oldest, matching the MATLAB v1.1 / v2.0 convention), and each event is screened against a minimum-count criterion using the Shuie-Bain (1982) extension of the Detre-White (1970) test for unequal sediment volumes.

Usage

```
char_peak_id(charcoal, pretreatment, peak_analysis, char_thresh)
```

Arguments

charcoal	Named list containing: peak C_{peak} series (accI - accIS or accI / accIS), length N . ybpI Age at each interpolated sample (cal yr BP), length N . accI Interpolated charcoal accumulation rate, length N . countI Interpolated charcoal count, length N . volI Interpolated sediment volume (cm^3), length N .
pretreatment	Named list with element yrInterp (interpolation resolution, years).
peak_analysis	Named list with elements threshType (1 = global, 2 = local), threshValues (numeric vector, length T), and minCountP (alpha for minimum-count screen).
char_thresh	Named list returned by [char_thresh_global()] or [char_thresh_local()], containing possible (251-bin grid) and pos ($[N \times T]$ threshold matrix).

Details

Threshold value matrix For **global** thresholds (threshType == 1), char_thresh\$pos is a constant-row $[N \times T]$ matrix reused directly. For **local** thresholds (threshType == 2), char_thresh\$pos is already $[N \times T]$ (per-sample values).

Consecutive-peak removal After flagging all exceedances, a diff-based pass retains only the *last* sample of each consecutive run – the oldest sample within a group of contiguous above-threshold

values. This matches the MATLAB v1.1 algorithm (which the v2.0 comment documents correctly despite the v1.1 comment being misleading).

Minimum-count test For each identified peak i in column j , a time window of ± 150 yr is constructed around the peak, then narrowed to the adjacent peaks when they fall within the window. The test statistic is

$$d = \frac{|c_{\min} - (c_{\min} + c_{\max}) v_{\min} / (v_{\min} + v_{\max})| - 0.5}{\sqrt{(c_{\min} + c_{\max}) v_{\min} v_{\max} / (v_{\min} + v_{\max})^2}}$$

and the p-value is $1 - \Phi(d)$ (standard normal CDF; equivalent to MATLAB's $1 - \text{tcdf}(d, 1e10)$ because $t_{1 \times 10^{10}} \rightarrow z$). Peaks with $p > \alpha_{\text{peak}}$ are removed.

Value

A named list with two components:

charcoal Input charcoal list augmented with:

- charPeaks – $[N \times T]$ numeric: 1 at peak samples, 0 elsewhere.
- charPeaksThresh – $[N \times T]$ numeric: threshold value at each identified peak, 0 elsewhere.
- peaksTotal – numeric vector length T : total peaks per threshold column.
- threshFRI – numeric matrix ($\leq N \times T$): fire-return intervals derived from peak ages per threshold column.

char_thresh Input char_thresh list augmented with minCountP – $[N \times T]$ matrix of Shuie-Bain p-values (NaN where not computed).

See Also

[char_thresh_local()], [char_thresh_global()], [CharAnalysis()]

char_plot

CharAnalysis output figures

Description

Nine output figures mirroring the MATLAB CharAnalysis v2.0 plots. Function names follow R snake_case conventions.

Usage

char_plot_peaks(out)

char_plot_fire_history(out)

char_plot_cumulative(out)

char_plot_fri(out)

```
char_plot_zones(out)
```

```
char_plot_raw(out)
```

```
char_plot_thresh_diag(out)
```

```
char_plot_sni(out)
```

```
char_plot_all(out, save = FALSE, out_dir = NULL, width = 11, height = 8.5)
```

Arguments

out	Named list returned by [CharAnalysis()]. Must contain charcoal, pretreatment, peak_analysis, char_thresh, post, and site.
save	Logical. If TRUE, each figure is saved as a PDF in out_dir. Default FALSE.
out_dir	Directory for saved PDFs. Required when save = TRUE; no default. Use tempdir() for a transient location, or supply a path of your choosing. Ignored when save = FALSE.
width, height	PDF dimensions in inches. Defaults: 11 x 8.5.

Details

[char_plot_raw()] Figure 1 (allFigures only): C_raw, C_interpolated, and C_background smoothing options.

[char_plot_thresh_diag()] Figure 2 (allFigures only): Local threshold determination diagnostics (5x5 window grid).

[char_plot_peaks()] Figure 3: Resampled CHAR with background trend (top) and C_peak with thresholds and peak markers (bottom).

[char_plot_sni()] Figure 4: Sensitivity to alternative thresholds and signal-to-noise index.

[char_plot_cumulative()] Figure 5: Cumulative peaks through time.

[char_plot_fri()] Figure 6: FRI distributions by zone with Weibull model fits.

[char_plot_fire_history()] Figure 7: Peak magnitude (top), FRIs through time with smoothed FRI and CI ribbon (middle), and smoothed fire frequency (bottom).

[char_plot_zones()] Figure 8: Between-zone comparisons of raw CHAR distributions (CDF and box plots).

[char_plot_all()] Convenience wrapper: produces all figures and optionally saves them as PDF files.

Requires the **ggplot2** package. Multi-panel layout uses **patchwork** if available; otherwise panels are printed separately with a message.

Value

Individual figure functions each return a **patchwork** / **ggplot2** object. char_plot_all() returns a named list of all figure objects.

See Also

[CharAnalysis()], [char_post_process()]

Examples

```
# Run pipeline on the bundled example dataset, then plot:
params_file <- system.file("validation", "CO_charParams.csv",
                           package = "CharAnalysis")
out <- CharAnalysis(params_file)
char_plot_peaks(out)
char_plot_fire_history(out)
# Individual figures can also be called directly:
char_plot_sni(out)
char_plot_fri(out)
# Save all figures to PDF in a temporary directory:
char_plot_all(out, save = TRUE, out_dir = tempdir())
```

char_pretreatment *Interpolate and pretreat a charcoal time series*

Description

Resamples the raw charcoal data to equal time steps using a proportion-weighted scheme, fills missing-value gaps by linear interpolation, computes charcoal accumulation rates (CHAR), and optionally applies a log transformation.

Usage

```
char_pretreatment(
  char_data,
  site,
  pretreatment,
  results = NULL,
  plot_data = 1L
)
```

Arguments

char_data	Numeric matrix (n x 6+): cmTop, cmBot, ageTop, ageBot, charVol, charCount. Rows sorted youngest-first (ascending ageTop).
site	Character string; site name used in diagnostic messages.
pretreatment	Named list with elements: <ul style="list-style-type: none"> zoneDiv Numeric vector of zone boundaries (cal. yr BP), strictly ascending, at least 2 values. yrInterp Resampling interval (yr). 0 = use median raw resolution (auto). transform Integer: 0 = none; 1 = log₁₀(x+1); 2 = ln(x+1).

results	Named list; only allFigures is referenced. Pass NULL to suppress any plot-related behaviour (no plots are produced in the R implementation in Phase 1).
plot_data	0/1 integer flag. Ignored in R (no diagnostic plots); included for API symmetry with the MATLAB function signature.

Details

Mirrors CharPretreatment.m from the MATLAB v2.0 codebase. The vectorised proportion matrix (four broadcast cases) produces results numerically identical to the MATLAB implementation within floating-point tolerance ($\sim 1e-14$).

Proportion matrix For each resampled interval i and each raw sample j , prop_matrix[i, j] is the fraction of the raw sample's age span that falls within the resampled interval. Four mutually exclusive overlap geometries (Cases A-D) are evaluated via matrix broadcasting across the full $[N_{rs} \times N_{raw}]$ grid – no loops required.

| Case | Geometry | Overlap | |-----|-----|-----| | A | Raw straddles the ****bottom**** edge | rsAgeBot - ageTop | | B | Raw straddles the ****top**** edge | ageBot - rsAgeTop | | C | Raw lies ****entirely within**** resampled interval | ageBot - ageTop | | D | Resampled lies ****entirely within**** raw sample | yrInterp |

zoneDiv auto-correction If zoneDiv[end] exceeds the bottom age of the last raw sample, the terminal resampled intervals would have no overlapping raw data and accI would be NA. These NAs propagate into charBkg and can hang the GMM in Phase 2. The value is silently corrected to lastAgeBotInData and the user is notified (v2.0 behaviour, preserved here).

Value

Named list with three elements:

charcoal List of raw and resampled series: cm, count, vol, con, ybp, acc (raw); cmI, countI, volI, conI, accI, ybpI (resampled).

pretreatment Input list returned with yrInterp updated when it was 0 (auto), and zoneDiv[end] corrected if it exceeded the bottom age of the last raw sample.

gap_in Integer matrix (nGaps x 2) of gap row-index pairs, or matrix(NA_integer_, 0, 2) when no gaps exist.

See Also

[char_parameters()], [char_validate_params()], [CharAnalysis()]

char_smooth

Smooth charcoal record to estimate low-frequency C_{background}

Description

Applies one of five smoothing methods to the interpolated charcoal accumulation rate series (charcoal\$accI) and stores the result in charcoal\$accIS. Mirrors CharSmooth.m from the MATLAB v2.0 codebase.

Usage

```
char_smooth(charcoal, pretreatment, smoothing, results = NULL, plot_data = 0L)
```

Arguments

charcoal	Named list with elements accI (resampled CHAR, length N), ybpI (resampled ages), and ybp (raw ages).
pretreatment	Named list with element yrInterp.
smoothing	Named list with elements: method Integer 1-5 selecting the smoothing method. yr Window width in years.
results	Named list (not used in R; kept for API symmetry).
plot_data	0/1 flag; ignored in R (no diagnostic plots).

Details

```
## Smoothing methods | Index | Name | Implementation | |———|———|—————| | 1 | Lowess | [char_lowess()] with iter = 0 | 2 | Robust Lowess | [char_lowess()] with iter = 4 | 3 | Moving average | zoo::rollapply(..., mean, partial=TRUE) | 4 | Running median + Lowess | Shifted-window median loop, then [char_lowess()] | 5 | Running mode + Lowess | Shifted-window 100-bin mode loop, then [char_lowess()] |
```

```
## Span convention The smoothing window width in data-point units is s = smoothing$yr / pretreatment$yrInterp. This is passed to [char_lowess()] as span = s (number of points), which converts it to the fraction required by stats::lowess() via f = round(s) / N.
```

```
## NaN bridging NaN values in accI (from record gaps) cannot be passed to [char_lowess()] directly. They are bridged by linear interpolation before smoothing and restored afterward, matching the MATLAB fallback path in CharSmooth.m (used when the Curve Fitting Toolbox is absent). Methods 4 and 5 always use the bridged series.
```

```
## Window selection for methods 4 and 5 The boundary window is SHIFTED (not shrunk) to maintain exactly round(s) samples, matching MATLAB's loop logic. Note that MATLAB's round() rounds 0.5 away from zero while R uses banker's rounding (round half to even); this can cause single-sample differences in window boundaries for odd half-integers.
```

Value

The input charcoal list with an additional element accIS: the smoothed C_{background} series (length N).

See Also

[char_lowess()], [CharAnalysis()]

char_thresh_global *Calculate a global threshold value for charcoal peak identification*

Description

Determines a single threshold applied uniformly across the entire record. The noise component of C_peak is modelled as a zero-mean Gaussian (residuals), a one-mean Gaussian (ratios), or via a Gaussian mixture model (GMM). Mirrors CharThreshGlobal.m from the MATLAB v2.0 code-base.

Usage

```
char_thresh_global(
  charcoal,
  pretreatment,
  peak_analysis,
  site = NULL,
  results = NULL,
  plot_data = 0L,
  bkg_sens_in = 0L
)
```

Arguments

charcoal	Named list containing peak (C_peak series) and ybpI.
pretreatment	Named list with zoneDiv.
peak_analysis	Named list with cPeak, threshMethod, threshValues.
site	Character string (site name; unused in R, kept for API symmetry).
results	Named list (unused in R, kept for API symmetry).
plot_data	0/1 flag; ignored in R.
bkg_sens_in	0/1 flag; ignored in R (no sensitivity loop).

Details

Gaussian assumption (threshMethod = 2) For residuals (cPeak = 1): noise is zero-mean; sigma is estimated from the negative half of C_peak, mirrored and pooled. For ratios (cPeak = 2): noise is one-mean; values are shifted to zero, mirrored, shifted back, and sigma estimated from the pooled set.

GMM (threshMethod = 3) MATLAB's GaussianMixture(X, 2, 2, false) is replicated by gaussian_mixture_em(X) from utils_gaussian_mixture.R, using the same first/last-point initialisation and loose convergence criterion as the original Bowman CLUSTER EM. The noise component is identified as the Gaussian with the smaller mean (matching MATLAB's noiseIdx = find(mu == min(mu), 1)).

Bin-lookup for threshold values Percentile thresholds are mapped to the nearest bin in possible (251 equally-spaced values spanning C_peak range). The v2.0 bug fix is preserved: both sides of the abs() comparison use the CHAR-unit threshold value thresh[i], not the raw percentile.

Value

Named list char_thresh with elements:

possible Numeric vector of 251 candidate threshold bins.

pos Numeric matrix [N x 4]: positive threshold for each of the four threshValues.

neg Numeric matrix [N x 4] (method 1) or [N x 1] (methods 2-3): negative threshold.

noise_pdf Estimated noise PDF evaluated at possible (methods 2-3), or scalar -99 (method 1).

mu_hat Fitted noise-component mean.

sigma_hat Fitted noise-component standard deviation.

SNI Signal-to-noise index (scalar).

GOF Sentinel vector (-999, length N).

See Also

[char_thresh_local()], [char_smooth()], [CharAnalysis()]

char_thresh_local	<i>Calculate a per-sample local threshold for charcoal peak identification</i>
-------------------	--

Description

Determines a sliding-window threshold value for each sample, based on the distribution of C_peak within a window centred on that sample. The noise component within each window is modelled as a zero-mean Gaussian (residuals), a one-mean Gaussian (ratios), or via a Gaussian mixture model (GMM). Mirrors CharThreshLocal.m from the MATLAB v2.0 codebase.

Usage

```
char_thresh_local(
  charcoal,
  smoothing,
  peak_analysis,
  site = NULL,
  results = NULL,
  plot_data = 0L
)
```

Arguments

charcoal	Named list with peak (C_peak series), ybpI (resampled ages), and accI.
smoothing	Named list with yr (window width in years).
peak_analysis	Named list with cPeak, threshMethod, threshValues (length 4).
site	Character string (site name; unused in R, kept for API symmetry).
results	Named list (unused in R, kept for API symmetry).
plot_data	0/1 flag; ignored in R (no diagnostic plots).

Details

Window selection The half-window is $hw = \text{round}(0.5 * \text{smoothing\$yr} / r)$ samples, where $r = \text{mean}(\text{diff}(\text{charcoal\$ybpI}))$ is the record resolution. The window is shifted (not shrunk) at record boundaries, matching MATLAB's loop logic.

NaN bridging within windows NaN entries in `charcoal$peak` (from record gaps) are replaced with the neutral value (0 for residuals, 1 for ratios) before distribution fitting. This preserves window size while preventing gap samples from biasing the fit, matching the v2.0 bug fix documented in `CharThreshLocal.m`.

Small-sample fallback If fewer than 4 non-neutral samples exist in the window, a simple Gaussian is fitted (same as method 2) and the KS test is skipped.

GMM (threshMethod = 3) MATLAB's `GaussianMixture(X, 2, 2, false)` is replicated by `gaussian_mixture_em(X)` from `utils_gaussian_mixture.R`. This uses the same first/last-point initialisation and loose convergence criterion ($\epsilon = 0.03 \log N$) as the original Bowman CLUSTER EM, substantially improving agreement with MATLAB threshold values compared to `mclust`. The noise component is identified as the Gaussian with the smaller mean. Poor-fit fallback (`mu1 == mu2`) re-fits with $K = 3$ and takes the two components with the smallest means, mirroring the MATLAB v2.0 bug fix (the fallback passes the local window `X`, not the full record).

KS goodness-of-fit MATLAB evaluates the fitted normal CDF at 101 equally-spaced bin centres and calls `kstest()` with a custom CDF table. R uses `stats::ks.test(noise, "pnorm", mu, sigma)`, which evaluates the continuous CDF at each observation. P-values may differ by a small amount for small samples; the statistic converges as sample size grows.

Post-loop smoothing After the per-sample loop, `pos`, `neg`, and `SNI` are smoothed with `char_lowess(span = smoothing$yr / r, iter = 0)`. `SNI` is then clamped to ≥ 0 .

Value

Named list `char_thresh` with elements:

pos Numeric matrix [N x 4]: per-sample positive threshold for each of the four `threshValues`, after Lowess smoothing.

neg Numeric matrix [N x 4]: per-sample negative threshold, after Lowess smoothing.

SNI Numeric vector [N]: signal-to-noise index time series, Lowess-smoothed and clamped to ≥ 0 .

GOF Numeric vector [N]: KS goodness-of-fit p-values per sample (NA where fewer than 4 noise samples exist).

See Also

[`char_thresh_global()`], [`char_smooth()`], [`CharAnalysis()`]

char_validate_params *Validate CharAnalysis input parameters*

Description

Checks all user-supplied parameters for internal consistency and plausible ranges. Throws a descriptive error and halts if any check fails. Emits warnings for non-fatal but unusual settings.

Usage

```
char_validate_params(  
    char_data,  
    pretreatment,  
    smoothing,  
    peak_analysis,  
    results  
)
```

Arguments

char_data	Numeric matrix (n x 6+).
pretreatment	List with zoneDiv, yrInterp, transform.
smoothing	List with method, yr.
peak_analysis	List with cPeak, threshType, threshMethod, threshValues, minCountP, peakFrequ.
results	List (not checked; included for API symmetry).

Details

Mirrors CharValidateParams.m from the MATLAB v2.0 codebase. All 15 checks and 2 non-fatal warnings are reproduced in the same order.

Value

NULL invisibly on success. Stops with an error on failure.

See Also

[char_parameters()], [CharAnalysis()]

char_write_results *Write the CharAnalysis results matrix to a CSV file*

Description

Writes the $N \times 33$ char_results matrix (assembled by [char_post_process()]) to a CSV file whose column headers and numeric format match the MATLAB CharAnalysis v2.0 *_charResults.csv output exactly.

Usage

```
char_write_results(char_results, site, out_dir, digits = 7L)
```

Arguments

char_results	Numeric matrix ($N \times 33$) returned in out\$char_results by [CharAnalysis()].
site	Character string: site name, used to build the output filename (<site>_charResults.csv).
out_dir	Directory in which to create the file. Required; no default. Created if it does not exist. Use tempdir() for a transient location, or supply a path of your choosing.
digits	Number of significant digits for numeric output. Default 7, matching MATLAB's fprintf default precision. Use NULL for R's full double precision (15 digits).

Details

Column order and headers Columns follow the MATLAB charResults layout:

1. cm Top_i (cm)
2. age Top_i (yr BP)
3. char Count_i (#)
4. char Vol_i (cm3)
5. char Con_i (# cm-3)
6. char Acc_i (# cm-2 yr-1)
7. charBkg (# cm-2 yr-1)
8. char Peak (# cm-2 yr-1)
9. thresh 1 (# cm-2 yr-1)
10. thresh 2 (# cm-2 yr-1)
11. thresh 3 (# cm-2 yr-1)
12. thresh FinalPos (# cm-2 yr-1)
13. thresh FinalNeg (# cm-2 yr-1)
14. SNI (index)
15. thresh GOF (p-val)

16. peaks 1
17. peaks 2
18. peaks 3
19. peaks Final
20. peaks Insig.
21. peak Mag (# cm-2 peak-1)
22. smPeak Frequ (peaks lka-1)
23. smFRIs (yr fire-1)
24. nFRIs (#)
25. mFRI (yr fire-1)
26. mFRI_uCI (yr fire-1)
27. mFRI_lCI (yr fire-1)
28. WBLb (yr)
29. WBLb_uCI (yr)
30. WBLb_lCI (yr)
31. WBLc (unitless)
32. WBLc_uCI (unitless)
33. WBLc_lCI (unitless)

NA / empty handling NA values are written as empty fields (no quotes), matching MATLAB's blank-cell convention. This applies to:

- smFRIs rows beyond the smoothed-FRI coverage window;
- zone-statistics columns (24-33) for rows beyond the last zone;
- any column not computed for a given run configuration.

CI column convention MATLAB stores bootstrap CIs as [quantile(2.5%), quantile(97.5%)] in the columns labelled uCI / lCI respectively (i.e. uCI = lower bound, lCI = upper bound – MATLAB's own labelling is inverted). The R output follows the same convention so that column indices are identical to the MATLAB reference file.

Value

Invisibly returns the full path to the file written.

See Also

[char_post_process()], [CharAnalysis()]

Examples

```
# Run the pipeline on the bundled example and write results to tempdir:
params_file <- system.file("validation", "CO_charParams.csv",
                           package = "CharAnalysis")
out <- CharAnalysis(params_file)
char_write_results(out$char_results, out$site,
                  out_dir = tempdir())
```

gaussian_mixture_em *Gaussian Mixture EM – direct R port of GaussianMixture.m (Bowman CLUSTER)*

Description

Fits a K-component univariate Gaussian mixture using the same EM algorithm bundled with CharAnalysis v1.1 and v2.0 (Bowman CLUSTER implementation). Replicates three key behaviours that distinguish it from mclust:

Usage

```
gaussian_mixture_em(x, k = 2L)
```

Arguments

x Numeric vector of observations (the C_peak values in the window).
k Integer number of components (default 2).

Details

1. **First/last-point initialisation:** for $K = 2$, component means are seeded at the first and last elements of the input data vector (not at sorted min/max; the data are in time order). This is the `initMixture()` behaviour from MATLAB.
2. **Loose convergence criterion:** EM stops when the per-step log-likelihood gain falls to or below $\epsilon = 0.01 \times L_c \times \log N$ where $L_c = 3$ for univariate data. This is much looser than the mclust default (10^{-5}) and causes MATLAB to freeze closer to the initial configuration.
3. **Variance regularisation:** a small floor $R_{\min} = \bar{\sigma}^2/10^5$ is added to each component variance after every M-step, preventing degenerate (zero-variance) solutions.

Because CharAnalysis always calls `GaussianMixture(X, 2, 2, false)` (i.e. `\finalK = initK = 2`), the `MDLReduceOrder()` path is never exercised and is therefore not implemented here.

Value

Named list:

mu Numeric vector length k: component means sorted ascending.

sigma Numeric vector length k: component standard deviations (= sqrt of fitted variance), sorted to match mu.

prop Numeric vector length k: mixing proportions, sorted to match mu.

loglik Scalar: log-likelihood at convergence.

See Also

[char_thresh_local()], [char_thresh_global()]

Index

`char_lowess`, 3
`char_parameters`, 5
`char_peak_id`, 6
`char_plot`, 7
`char_plot_all` (`char_plot`), 7
`char_plot_cumulative` (`char_plot`), 7
`char_plot_fire_history` (`char_plot`), 7
`char_plot_fri` (`char_plot`), 7
`char_plot_peaks` (`char_plot`), 7
`char_plot_raw` (`char_plot`), 7
`char_plot_sni` (`char_plot`), 7
`char_plot_thresh_diag` (`char_plot`), 7
`char_plot_zones` (`char_plot`), 7
`char_pretreatment`, 9
`char_smooth`, 10
`char_thresh_global`, 12
`char_thresh_local`, 13
`char_validate_params`, 15
`char_write_results`, 16
`CharAnalysis`, 2

`gaussian_mixture_em`, 18

`lowess`, 3