Package: appac (via r-universe)

July 15, 2026

Title Atmospheric Pressure Peak Area Correction for Gas Chromatography with Standard Detectors

Version 4.0.3

Description Corrects gas-chromatography peak areas for the influence of ambient air pressure on standard detectors open to the ambient atmosphere, such as the flame ionization detector, whose pressure sensitivity was characterised by Bocek, Novak and Janak (1969) <doi:10.1016/S0021-9673(00)99223-9>. Unlike the pressure compensation of Ayers and Clardy (1985) <https://patents.google.com/patent/US4512181A>, which is combined with a calibration and valid only for a single calibration period of a few days, per-cylinder peak areas are decomposed by principal components into a pressure-correlated component and per-peak drift; a common pressure-sensitivity coefficient (kappa) is estimated with a heavy-tail-robust fit on a drift-reduced signal, and slow drift plus a daily factor are removed. Returns the corrected areas together with a chi-square goodness-of-fit diagnostic. Structural-break detection (package 'strucchange', Zeileis and others (2002) <doi:10.18637/jss.v007.i02>) is provided for episode-level and variance breakpoint analysis.

License GPL-3

Encoding UTF-8

Depends R (>= 4.0.0)

Imports methods, stats, utils, data.table, dplyr, tibble, purrr, kza, robustbase, strucchange

Suggests ggplot2, patchwork, knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

LazyData true

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Ruediger Forster [aut, cre] (ORCID:
<https://orcid.org/0009-0006-4851-7430>)

Maintainer Ruediger Forster <meticulous.measurements@gmail.com>

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

Date/Publication 2026-07-15 18:00:02 UTC

RemoteUrl https://github.com/cran/appac

RemoteRef HEAD

RemoteSha 546b3d4f3b7cb6bde362fd9f1b223baa1d5d0d4f

Contents

---

appac *Fit the APPAC pressure correction*

Description

Fits the APPAC (Atmospheric Pressure Peak Area Correction) model and returns the corrected peak areas with diagnostics. The data must already be column-checked with `check_cols`. `appac` drives the whole pipeline: pivot the data, decompose it by principal components, estimate the common pressure sensitivity `kappa` with a heavy-tail-robust fit on a drift-reduced signal, estimate the drift / daily-factor model, and divide the assembled correction out of the raw areas.

Usage

```
appac(data, ct = NULL, P_ref = 1013.25)
```

Arguments

data	Long-format data frame with the canonical columns (Sample_Name, Peak_Name, Injection_Date, Air_Pressure, Raw_Area), e.g. the output of check_cols.
ct	Optional fixed per-sample centres (a list, named by sample, of per-peak reference values). NULL uses each peak's whole-series mean; pass the result of debias_ct for the de-biased centres.
P_ref	Reference pressure (hPa) at which the correction is unity.

Details

Missing areas are tolerated: a peak with up to 30% NA is kept and its gaps are imputed by low-rank reconstruction before the fit; whole missing injections (samples on staggered dates) are handled by the cross-sample reconstruction in the drift model.

Value

An object of class "Appac": the samples slot holds, per cylinder, the corrected.area; correction@coefficients the fitted kappa; and trend the drift / daily-factor model.

See Also

check_cols, debias_ct, goodness_of_fit

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat   <- check_cols(PLOT_FID, acn)
ap    <- as.numeric(dat[, "Air_Pressure"])
P_ref <- mean(range(ap, na.rm = TRUE))   # mid-range reference pressure

fit <- appac(dat, P_ref = P_ref)
unlist(fit@correction@coefficients)   # the fitted common kappa
```

Appac-class *Fitted APPAC model*

Description

The object returned by appac: the corrected areas plus the drift and pressure-correction models.

Usage

```
## S4 method for signature 'Appac'
show(object)

## S4 method for signature 'Appac'
print(x, ...)
```

Arguments

object	the object to display.
x	the "Appac" object to print.
...	ignored.

Slots

samples Per-cylinder list carrying raw.area, corrected.area, dates and pressure.

trend The drift / daily-factor model (a Compensation-class).

correction The pressure correction (a Correction-class).

---

check_cols *Validate and canonicalise the input columns*

Description

Validates that the input carries the required columns, renames them to the canonical internal names, and cleans the sample and peak names to valid R names with make.names. Call this before appac.

Usage

```
check_cols(data, appac_colnames, verbose = FALSE)
```

Arguments

data	The raw long-format data frame.
appac_colnames	Named list mapping the roles (sample_col, peak_col, date_col, pressure_col, area_col) to the actual column names in data.
verbose	If TRUE, report (via message) which column, peak and sample names were renamed. Off by default.

Details

The mapping is keyed by *role* (sample_col, peak_col, date_col, pressure_col, area_col), so the order in which appac_colnames lists them does not matter: each role's column is found by name and relabelled to its canonical name.

Value

The data frame with columns renamed to the canonical names and sample/peak names cleaned. Stops if a role or column is missing, or if `make.names()` would collapse two distinct sample or peak names into one.

See Also

appac

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat <- check_cols(PLOT_FID, acn)
head(dat)
```

Compensation-class *Drift / daily-factor model*

Description

The temporal-drift component of a fitted Appac-class object: the daily-factor and bias/trend decomposition of the area drift.

Usage

```
## S4 method for signature 'Compensation'
show(object)
```

Arguments

`object`	the object to display.

Slots

`date` Integer date axis the model is evaluated on.

`bias` Per-cylinder additive bias.

`trend` Per-cylinder linear trend.

`correlated.features` The common daily-factor signal.

`correlated.features.scaling` Scaling applied to the correlated features.

`bias.trend.scaling` Scaling applied to the bias/trend component.

`center` Per-sample, per-peak centre (reference) values.

`samples` Per-cylinder inputs to the drift model.

Correction-class *Pressure correction model*

Description

The pressure-correction component of a fitted Appac-class object.

Usage

```
## S4 method for signature 'Correction'
show(object)
```

Arguments

object	the object to display.

Slots

covariates Names of the correction covariates (e.g. air pressure).

coefficients Fitted coefficient (kappa) per covariate.

reference.values Reference value (e.g. P_ref) per covariate.

samples Per-cylinder correction inputs.

fit.data Stored kappa-fit input, used by plot_area_pressure_fit.

debias_ct *De-bias the per-peak centres*

Description

Refines ("de-biases") the per-peak centres by minimising the chi-square of the corrected-area residuals about the centre. Sweeps a scale factor cf around the current centre (the whole-series mean), re-runs appac at each cf, fits the per-peak chi-square as a parabola in cf and takes its (closed-form) minimum. Trades a little variance for reduced bias.

Usage

```
debias_ct(Appac, data, P_ref, npt = 20, quiet = FALSE)
```

Arguments

Appac	A fitted "Appac" object (from a first appac pass).
data	The same column-checked data passed to that appac() call.
P_ref	The same reference pressure (hPa).
npt	Number of sweep points for cf in [0.99, 1.01].
quiet	Suppress the progress dots.

Value

A list of de-biased centres (one numeric vector per sample), to feed back as `appac(..., ct = <this>)`.

See Also

`appac`

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat   <- check_cols(PLOT_FID, acn)
ap    <- as.numeric(dat[, "Air_Pressure"])
P_ref <- mean(range(ap, na.rm = TRUE))   # mid-range reference pressure

fit1 <- appac(dat, P_ref = P_ref)
ct   <- debias_ct(fit1, data = dat, P_ref = P_ref, quiet = TRUE)
fit  <- appac(dat, ct = ct, P_ref = P_ref)   # de-biased second pass
```

---

get_changepoints *Detect episode level breakpoints across cylinders*

Description

Detects abrupt level shifts ("episode" boundaries) in the area series. Per cylinder the change signal is the second principal-component score of a one-component PCA on the per-peak standardised areas; a structural-break model (`breakpoints`, gated by an OLS-MOSUM fluctuation test, `efp` / `sctest`) is fitted to the daily-averaged PC2 series and the BIC-optimal breakpoints are kept. The cross-cylinder breakpoints are the union of the per-cylinder breaks, merged when closer than `merge_within` days. Detection is deterministic.

Usage

```
get_changepoints(
  samples,
  alpha = 0.05,
  h = 0.15,
  max_grid = 250L,
  min_seg = 30L,
  merge_within = 3
)
```

Arguments

samples	Named list, each element with $date (vector) and $raw.area (peak matrix) – i.e. the samples slot of an "Appac" object.
alpha	Significance level of the OLS-MOSUM test that gates whether a cylinder has any break.
h	Minimum segment width, as a fraction of the (coarsened) series length, for the MOSUM bandwidth and the breakpoint search.
max_grid	Long daily series are coarsened onto at most this many bins before dating, so the search stays fast.
min_seg	Minimum daily-series length (days) required to attempt detection.
merge_within	Merge breakpoints from different cylinders that fall within this many days of each other.

Value

An IDate vector of breakpoint dates.

See Also

get_variance_changepoints, appac, plot_area_date

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat   <- check_cols(PLOT_FID, acn)
ap    <- as.numeric(dat[, "Air_Pressure"])
P_ref <- mean(range(ap, na.rm = TRUE))   # mid-range reference pressure

fit <- appac(dat, P_ref = P_ref)
get_changepoints(fit@samples)   # episode breakpoint dates
```

---

get_variance_changepoints

Detect variance breakpoints across cylinders

Description

Detects abrupt changes in the measurement *variance* (precision), the second-moment counterpart of get_changepoints. Per cylinder the signal is the daily mean of the per-injection noise energy – the sum of squared residuals after removing the leading drop principal component(s) (the level / pressure / drift) – and a change in its mean is a change in the noise variance. The same strucchange machinery (OLS-MOSUM gate + breakpoints) dates the breaks; results are merged and returned as for get_changepoints.

Usage

```
get_variance_changepoints(
  samples,
  drop = 1L,
  alpha = 0.05,
  h = 0.15,
  max_grid = 250L,
  min_seg = 30L,
  merge_within = 3
)
```

Arguments

samples	Named list, each element with $date and $raw.area, i.e. the samples slot of an "Appac" object.
drop	Number of leading principal components to remove (the level / pressure / drift) before forming the noise energy.
alpha, h, max_grid, min_seg, merge_within	As in get_changepoints.

Details

A pure mean (level) shift leaves the variance unchanged, so this detector does *not* flag the level breaks that get_changepoints finds – the two are complementary. Small variance steps sit near the detection floor: a change is only found when it is significant at alpha.

Value

An IDate vector of variance-breakpoint dates.

See Also

get_changepoints, appac

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat   <- check_cols(PLOT_FID, acn)
ap    <- as.numeric(dat[, "Air_Pressure"])
P_ref <- mean(range(ap, na.rm = TRUE))   # mid-range reference pressure

fit <- appac(dat, P_ref = P_ref)
get_variance_changepoints(fit@samples)   # precision-change dates
```

`goodness_of_fit` *Goodness of fit of the correction*

Description

Tests, per peak, the null hypothesis that the corrected areas are constant (equal to the centre) up to measurement noise. Residual structure left behind by the correction (drift, steps) inflates the statistic.

Usage

```
goodness_of_fit(Appac)
```

Arguments

`Appac`	A corrected `"Appac"` object (e.g. from a second `appac` pass).

Details

The chi-square sums the squared residuals about the centre, each scaled by the measurement-noise variance of its peak. That variance is estimated from successive injections – the von Neumann lag-1 estimator $\sigma^2 = mean(d^2)/2$ with d the date-ordered first differences – so it captures the irreducible short-term noise and is independent of any slow residual drift. Scaling by a noise estimate (rather than by the centre, as a Pearson form would, which assumes a Poisson variance) makes the statistic correct for continuous peak areas and dimensionless.

Use the **reduced** chi-square (chi-square / dof): about 1 means the corrected areas are down to the short-term noise floor (a good correction), above 1 flags residual structure. The p-value is returned too but is of little use here: with thousands of injections the distribution is so tight that any reduced chi-square a touch above 1 gives p near 0.

Value

A named list (per sample) of data frames, one row per peak, with columns `reduced.chisq`, `chisq`, `dof` and `p.value`.

See Also

`appac`

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat   <- check_cols(PLOT_FID, acn)
ap    <- as.numeric(dat[, "Air_Pressure"])
P_ref <- mean(range(ap, na.rm = TRUE))   # mid-range reference pressure
```

```
fit <- appac(dat, P_ref = P_ref)
goodness_of_fit(fit)[[1]]   # per-peak reduced chi-square, sample 1
```

`plot_area_date` *Plot peak area versus date*

Description

Scatter of raw and corrected peak area against injection date for one (sample, peak), with the reference (centre) line and, optionally, the robust episode change-points as dotted vertical lines.

Usage

```
plot_area_date(
  appac,
  sample = 1,
  peak = 1,
  show_changepoints = TRUE,
  size = 12
)
```

Arguments

`appac`	A fitted `"Appac"` object.
`sample`	Sample selector: a cylinder name or a positive index.
`peak`	Peak selector: a peak name or a positive index.
`show_changepoints`	Draw the detected change-points (see `get_changepoints`) as dotted vertical lines.
`size`	Base font size passed to the theme.

Value

A `ggplot` object.

See Also

`plot_area_pressure`, `get_changepoints`

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat   <- check_cols(PLOT_FID, acn)
ap    <- as.numeric(dat[, "Air_Pressure"])
P_ref <- mean(range(ap, na.rm = TRUE))   # mid-range reference pressure

fit <- appac(dat, P_ref = P_ref)
if (requireNamespace("ggplot2", quietly = TRUE))
  plot_area_date(fit, sample = 1, peak = 1, show_changepoints = FALSE)
```

---

plot_area_pressure *Plot peak area versus air pressure*

Description

Scatter of raw and corrected peak area against air pressure for one (sample, peak), with the reference (centre) line. The pressure dependence visible in the raw area should be flattened in the corrected area.

Usage

```
plot_area_pressure(appac, sample = 1, peak = 1, size = 12)
```

Arguments

appac	A fitted "Appac" object.
sample	Sample selector: a cylinder name or a positive index.
peak	Peak selector: a peak name or a positive index.
size	Base font size passed to the theme.

Value

A ggplot object.

See Also

plot_area_date, plot_residuals

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat   <- check_cols(PLOT_FID, acn)
ap    <- as.numeric(dat[, "Air_Pressure"])
P_ref <- mean(range(ap, na.rm = TRUE))   # mid-range reference pressure

fit <- appac(dat, P_ref = P_ref)
if (requireNamespace("ggplot2", quietly = TRUE))
  plot_area_pressure(fit, sample = 1, peak = 1)
```

---

plot_area_pressure_fit

Plot the kappa-fit input

Description

Shows the data that actually feeds the kappa fit: the reference-scaled correlated area, binned by pressure deviation, for each (sample, peak) and coloured by peak, with the fitted kappa-slope line overlaid. All series share the single common slope.

Usage

```
plot_area_pressure_fit(appac, covariate = 1, size = 12)
```

Arguments

appac	A fitted "Appac" object whose correction carries the stored kappa-fit input (populated by appac).
covariate	Covariate selector: a covariate name or a positive index (normally the pressure covariate, index 1).
size	Base font size passed to the theme.

Details

Pass the **de-biased** result – appac(ct = debias_ct(...)) – so this shows the chi-square-minimum fit (the correct kappa), not the first pass.

Value

A ggplot object.

See Also

appac, debias_ct

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat   <- check_cols(PLOT_FID, acn)
ap    <- as.numeric(dat[, "Air_Pressure"])
P_ref <- mean(range(ap, na.rm = TRUE))   # mid-range reference pressure

fit1 <- appac(dat, P_ref = P_ref)
ct   <- debias_ct(fit1, data = dat, P_ref = P_ref, quiet = TRUE)
fit  <- appac(dat, ct = ct, P_ref = P_ref)
if (requireNamespace("ggplot2", quietly = TRUE))
  plot_area_pressure_fit(fit)
```

PLOT_FID *Flame-ionization-detector peak areas for APPAC*

Description

An example set of gas-chromatography flame-ionization-detector (FID) injections from a real instrument, with expert-annotated peak integration, used to demonstrate the pressure correction. Long format: one row per injection and peak, from repeated analyses of several control cylinders.

Usage

```
PLOT_FID
```

Format

A data frame with 77365 rows and 6 columns:

sample.name Cylinder (control sample) identifier.

injection.date Date of the injection.

peak.name Component / peak identifier.

retention.time Peak retention time.

raw.area Raw integrated peak area.

air.pressure Ambient air pressure (hPa) at the injection.

plot_residuals *Residual diagnostic panel*

Description

A 2x2 panel of residual diagnostics for one (sample, peak): histogram with a fitted normal, normal Q-Q plot, residual versus date, and residual versus pressure. The residual is the corrected area minus the reference (centre).

Usage

```
plot_residuals(appac, sample = 1, peak = 1, size = 12)
```

Arguments

appac	A fitted "Appac" object.
sample	Sample selector: a cylinder name or a positive index.
peak	Peak selector: a peak name or a positive index.
size	Base font size passed to the theme.

Value

A patchwork object assembling the four panels (requires the **patchwork** package).

See Also

plot_area_date, goodness_of_fit

Examples

```
acn <- list(sample_col = "sample.name", peak_col = "peak.name",
            date_col = "injection.date", pressure_col = "air.pressure",
            area_col = "raw.area")
dat   <- check_cols(PLOT_FID, acn)
ap    <- as.numeric(dat[, "Air_Pressure"])
P_ref <- mean(range(ap, na.rm = TRUE))   # mid-range reference pressure

fit <- appac(dat, P_ref = P_ref)
if (requireNamespace("ggplot2", quietly = TRUE) &&
    requireNamespace("patchwork", quietly = TRUE))
  plot_residuals(fit, sample = 1, peak = 1)
```

Synth_data *Synthetic stress-test data for APPAC*

Description

A compact, fully synthetic data set with a *known* ground truth, for unit tests and runnable examples. It is deliberately adversarial: three samples of largely different composition, ten peaks each spanning a factor-of-five dynamic range, measured over three fifty-run episodes separated by two planted breakpoints, with every classical adversary of the pressure fit stacked on top (a small kappa buried under brown (reddened, AR(1) $\phi = 0.85$) heavy-tailed Student-t noise, ~2% positive outlier contamination, and a slow per-sample drift confounded with the pressure signal). The AR(1) coefficient is calibrated to the real PLOT_FID noise colour (lag-1 autocorrelation ~0.87, spectral slope ~1.4).

Usage

```
Synth_data
```

Format

A data frame with 4500 rows (3 samples x 10 peaks x 150 runs) and 6 columns, matching PLOT_FID:

sample.name Sample identifier (S1, S2, S3).

injection.date Date of the injection (IDate).

peak.name Peak identifier (pk01..pk10).

retention.time Peak retention time (elution order).

raw.area Raw integrated peak area.

air.pressure Ambient air pressure (hPa), shared per run across samples.

Details

The forward model is

$$area = ref \cdot c_{ep} \cdot (1 + \kappa(P - P_{ref})) \cdot (1 + drift) \cdot (1 + noise),$$

with a shared per-run air pressure P (same instrument day, same ambient pressure across samples). The **composition is fixed** throughout – the same samples, the same relative peak amounts – and the two breakpoints shift only the per-episode *centre* c_{ep}, multiplicatively and predictably:

breakpoint 1 an abrupt $+5\%$ shift of the centre (episodes 2–3).

breakpoint 2 a further -2% centre step *and* a nominal $+200\%$ step in the measurement-noise variance (innovation sd $\times\sqrt{3}$, episode 3). The AR(1) memory and the short 50-run episodes pull the *realized* spread ratio to about $1.5\times$, so this break sits *near* the detection floor rather than clearing it (see 'Limitations' in appac-package).

Because the shifts are composition-preserving (common-mode) they are, by design, hard for the trend/PC2 change-point detector to recover – a naive whole-series appac fit is likewise *biased* by the unmodelled steps. That is the point: the set exercises robustness and motivates episode splitting rather than promising clean recovery.

Two moments, two detectors. Breakpoint 2 deliberately carries two *separable* signatures that live in different statistical moments of the per-sample PCA (standardise each peak, then decompose):

- the level steps (breakpoint 1, and breakpoint 2's -2% step) are a *first-moment* change – a shift in the *mean* of the dominant common-mode component (PC1). A trend / level change-point model recovers these (from PC1, not the PC2 that get_changepoints currently reads; see there).
- the variance expansion is a *second-moment* change – a shift in the *spread* of an uncorrelated (non-PC1) component, with its mean flat. A mean/trend model is blind to it by construction; it needs a *dispersion* change-point statistic (e.g. a change-point on the rolling variance of the uncorrelated subspace).

Which component carries the variance step is sample-dependent (an artefact of the whole-series PCA ordering: PC2 in one sample, PC3 or higher in another), so a variance detector must scan the uncorrelated subspace rather than a fixed PC. Physically the two moments are distinct instrument events: a calibration / level shift (mean) versus a loss of precision / repeatability (variance).

The planted ground truth is attached as `attr(Synth_data, "truth")`, a list with `kappa`, `P_ref`, the `breakpoints` (IDate), the `center_shift` / `center_shift_bp2` / `variance_expansion` magnitudes, `n_episodes`, `runs_per_episode`, `dynamic_range`, the per-sample per-peak `reference` composition, and the `noise` parameters. Regenerate with `data-raw/Synth_data.R`.

See Also

PLOT_FID, appac, get_changepoints

Examples

```
truth <- attr(Synth_data, "truth")
truth$kappa            # the planted common pressure sensitivity
truth$breakpoints      # the two planted episode boundaries
```

Index