Package 'winn'

September 27, 2026

Title White Noise Normalization for Mass Spectrometry Profiling Data

Version 0.1.5

Description Provides a decision-guided workflow for correcting technical variability in chemical profiling data. Tests for white noise identify measured features that need correction while preserving those that already pass. The workflow combines robust outlier adjustment, adaptive drift detection, change-point segmentation, batch correction, and probabilistic quotient normalization in a single pipeline or as modular steps. It supports parameter tuning using pooled quality-control samples as well as operation for studies without pooled controls.

Depends R (>= 3.5.0)

Imports stats, lmtest, mgcv, splines

Suggests knitr, rmarkdown, sva, testthat (>= 3.0.0)

License GPL-3

Encoding UTF-8

VignetteBuilder knitr

Config/testthat/edition 3

URL https://github.com/ratschlab/winn

BugReports https://github.com/ratschlab/winn/issues

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Tanmay Tanna [aut, cre, cph]

Maintainer Tanmay Tanna <tanmay@tanmaytanna.com>

Repository CRAN

Date/Publication 2026-09-27 16:40:02 UTC

Contents

`adjust_outliers_mad` *Adjust Outliers in Data Matrix Using MAD*

Description

This function adjusts outliers in a data matrix on a per-metabolite basis using the median absolute deviation (MAD). For each metabolite, the median, MAD, thresholds, and upper/lower outlier masks are computed once from the original finite values. Upper- and lower-tail values are then shrunk independently in one non-iterative pass; adjusting one tail cannot change the threshold or membership of the other tail. Non-finite values are not eligible for threshold estimation and are returned unchanged by this standalone helper. The full `winn()` pipeline instead requires finite, non-negative input. Numeric zeros are treated as observed values and are never automatically converted to missing values.

Usage

```
adjust_outliers_mad(data)
```

Arguments

`data`	A numeric matrix or data frame where rows represent metabolites and columns represent samples.

Value

A numeric matrix with adjusted outliers.

Examples

```
set.seed(1)
your_data_matrix <- matrix(rnorm(200, mean = 100, sd = 15), nrow = 20)
adjusted_data <- adjust_outliers_mad(your_data_matrix)
```

anova_batch_correction *Perform ANOVA-based Mean-Only Batch Correction*

Description

This function runs an ANOVA test on each metabolite to detect batch effects, and then corrects significant batch effects by subtracting the estimated batch-specific shifts (while preserving the overall mean). With the default selective gate, features that do not pass the FDR threshold are unchanged; this differs from `combat_batch_correction()`, which applies an empirical Bayes location/scale model to every feature.

Usage

```
anova_batch_correction(
  data,
  batch,
  fdr_threshold = 0.05,
  gate = c("selective", "all", "none"),
  return_diagnostics = FALSE
)
```

Arguments

`data`	A numeric matrix (metabolites × samples).
`batch`	A factor or numeric vector indicating batch for each sample.
`fdr_threshold`	Significance threshold for FDR-adjusted p-values.
`gate`	Batch gate mode: `"selective"` (default), `"all"`, or `"none"`.
`return_diagnostics`	Logical; if `TRUE`, return feature-level test and correction diagnostics together with the corrected matrix.

Value

A numeric matrix of corrected intensities, or a list containing the matrix and diagnostics when `return_diagnostics = TRUE`.

Examples

```
set.seed(1)
mat <- matrix(rnorm(200, mean = 100, sd = 15), nrow=20)
batch <- rep(1:4, length.out=ncol(mat))
corrected <- anova_batch_correction(mat, batch, fdr_threshold=0.05)
```

`autocorrelation_correct` *Correct for Drift in Data Using Autocorrelation Correction*

Description

This function corrects for drift effects in metabolomics data by detrending based on run order within each batch segment. When `run_order` is supplied, every segment is sorted by that vector for both testing and smoothing, and the corrected values are then restored to the original matrix-column order.

Usage

```
autocorrelation_correct(
  data,
  run_order = NULL,
  batch,
  lag = NULL,
  test = "Ljung-Box",
  ljung_box_fitdf = 0L,
  detrend = "mean",
  fdr_threshold = 0.05,
  spline_method = "conservative",
  gate = NULL,
  return_diagnostics = FALSE
)
```

Arguments

`data`	A numeric matrix with rows representing metabolites and columns representing samples.
`run_order`	An optional numeric vector representing the run order of the samples. If `NULL`, the current matrix-column order is treated as acquisition order.
`batch`	A vector indicating the batch (or segment) assignment for each sample.
`lag`	An optional integer specifying the lag to be used in the autocorrelation test. If `NULL`, the lag is selected adaptively for each batch segment using `max(min(10, floor(n / 5)), df + 3)` and then capped at `n - 1`.
`test`	A character string specifying the autocorrelation test to use ("Ljung-Box" or "DW").
`ljung_box_fitdf`	Non-negative integer passed to `stats::Box.test()` as `fitdf`. The default is `0` because WiNN does not fit an ARMA model before testing. Set this to `1` only to reproduce the legacy WiNN gate or for a documented sensitivity analysis.
`detrend`	A character string indicating the method for detrending ("mean" or "spline").
`fdr_threshold`	A numeric value specifying the FDR threshold for significance.

`spline_method` A character string specifying the spline method when detrend="spline" ("conservative" or "standard").

`gate` Optional gate mode: `"selective"`, `"all"`, or `"none"`. When `NULL`, the historical behavior is retained: `detrend = "mean"` is selective and `detrend = "spline"` corrects all testable profiles.

`return_diagnostics` Logical; if `TRUE`, return profile-level test and correction diagnostics together with the corrected matrix.

Value

A numeric matrix with drift corrected, or a list containing the matrix and diagnostics when `return_diagnostics = TRUE`.

Examples

```
set.seed(1)
your_data_matrix <- matrix(rnorm(200, mean = 100, sd = 15), nrow = 20)
batch <- rep(1:4, length.out = ncol(your_data_matrix))
run_order <- seq_len(ncol(your_data_matrix))
drift_corrected <- autocorrelation_correct(your_data_matrix, run_order, batch)
```

---

combat_batch_correction

Perform ComBat Batch Correction by batch

Description

This function applies the empirical Bayes ComBat method to correct batch effects by batch, adjusting both location and scale parameters across batch. Unlike the default selective ANOVA gate in `anova_batch_correction()`, ComBat is a global correction: every feature is passed to the empirical Bayes model.

Usage

```
combat_batch_correction(
  data,
  batch,
  par_prior = TRUE,
  mean_only = FALSE,
  ref_batch = NULL
)
```

Arguments

data	A numeric matrix (metabolites × samples).
batch	A factor or numeric vector indicating batch for each sample.
par_prior	Logical indicating whether to use parametric prior (default TRUE).
mean_only	Logical indicating mean-only adjustment (default FALSE).
ref_batch	Optional reference batch level for anchoring (default NULL).

Value

A numeric matrix of corrected intensities.

Examples

```
# Loading the optional Bioconductor dependency can take more than five seconds.
if (requireNamespace("sva", quietly = TRUE)) {
  set.seed(1)
  mat <- matrix(rnorm(200, mean = 100, sd = 15), nrow=20)
  batch <- rep(1:4, length.out=ncol(mat))
  corrected <- combat_batch_correction(mat, batch, par_prior = TRUE)
}
```

---

normalize_by_dilution_factor

Normalize Data Matrix by Dilution Factor

Description

This function normalizes a data matrix by dilution factors or alternatively shrinks sample measurements if any samples have dilution factors that are more than one standard deviation away from the mean dilution factor. Numeric zeros remain observed values. Features whose reference median is zero are omitted from quotient fitting, but retained when the estimated sample factors are applied. A sample with a non-positive dilution factor is rejected.

Usage

```
normalize_by_dilution_factor(
  data,
  processing = "shrink",
  control_samples = NULL,
  return_diagnostics = FALSE
)
```

Arguments

data	A numeric matrix or data frame where rows represent metabolites and columns represent samples.
processing	A character string specifying the processing method to use. Options are "shrink" (default) or "normalize".
control_samples	An optional numeric vector specifying which columns correspond to control samples. If provided, the reference spectrum is calculated using only these samples.
return_diagnostics	Logical; if TRUE, return the corrected matrix and sample-level dilution diagnostics in a list.

Value

A numeric matrix of normalized intensities, or a list containing the matrix and diagnostics when return_diagnostics = TRUE.

Examples

```
# Example usage:
set.seed(1)
your_data_matrix <- matrix(rnorm(200, mean = 100, sd = 15), nrow = 20)
normalized_data <- normalize_by_dilution_factor(your_data_matrix, control_samples = 1:4)
```

---

scale_by_batch *Scale Data by batch*

Description

This function scales the values for each metabolite within each batch by subtracting the batch mean and dividing by the batch standard deviation.

Usage

```
scale_by_batch(data, batch)
```

Arguments

data	A numeric matrix with rows representing metabolites and columns representing samples.
batch	A numeric vector indicating the batch (or segment) assignment for each sample.

Value

A numeric matrix of scaled intensities.

Examples

```
set.seed(1)
your_data_matrix <- matrix(rnorm(200, mean = 100, sd = 15), nrow = 20)
batch <- rep(1:4, length.out = ncol(your_data_matrix))
scaled_data <- scale_by_batch(your_data_matrix, batch)
```

`winn` *Winn Correction for Metabolomics Data*

Description

This function performs a series of corrections on metabolomics data to adjust for dilution effects, outliers, drift, and batch effects. If batch information is not supplied, segments are automatically detected using an fkPELT-based approach and labeled as batch.

Usage

```
winn(
  data,
  batch = NULL,
  run_order = NULL,
  control_samples = NULL,
  parameters = "fixed",
  fdr_threshold = 0.05,
  median_adjustment = "shrink",
  detrend_non_autocorrelated = "mean",
  spline_method = "conservative",
  remove_batch_effects = "anova",
  test = "Ljung-Box",
  lag = NULL,
  ljung_box_fitdf = 0L,
  scale_by_batch = FALSE,
  pelt_penalty = NULL,
  return_details = FALSE
)
```

Arguments

`data`	A numeric matrix or data frame where rows represent metabolites and columns represent samples.
`batch`	An optional numeric vector indicating batch assignments for each sample. If NULL, segments will be auto-detected.
`run_order`	An optional numeric vector representing the run order of samples.
`control_samples`	An optional numeric vector representing the columns corresponding to control samples. If provided, these will be used for normalization and parameter tuning.

`parameters` A character string specifying fixed (`"fixed"`, the default) or QC-tuned (`"auto"`) parameters. Auto mode requires controls.

`fdr_threshold` Fixed-mode FDR threshold for drift and selective ANOVA batch correction. Auto mode searches 0.10, 0.05, and 0.01.

`median_adjustment`
Fixed-mode median adjustment (`"shrink"`, `"normalize"`, or `"none"`). Auto mode searches `"shrink"` and `"normalize"`.

`detrend_non_autocorrelated`
A character string specifying the method for detrending non-autocorrelated metabolites ("mean" or "spline").

`spline_method` Fixed-mode spline method (`"conservative"` or `"standard"`). Auto mode searches both.

`remove_batch_effects`
Batch method. `"anova"` selectively corrects features that pass its FDR gate; `"combat"` applies global empirical Bayes location/scale correction to all features.

`test` A character vector specifying the autocorrelation test(s) to use. Fixed mode requires a single value. Auto mode evaluates only the supplied tests, so `DW` is included only when explicitly requested.

`lag` An optional integer specifying the lag for the autocorrelation test. If `NULL`, `autocorrelation_correct()` selects the lag adaptively for each batch segment.

`ljung_box_fitdf`
Non-negative integer passed to `stats::Box.test()`. The default is `0` because no ARMA model is fitted. Use `1` only for legacy reproduction or a documented sensitivity analysis.

`scale_by_batch` Logical indicating whether to scale data by batch after corrections.

`pelt_penalty` Optional fkPELT penalty specification. Use a positive numeric value, `"bic"`, `"mbic"`, or `NULL`. When `NULL`, fkPELT uses the conservative MBIC default.

`return_details` Logical. The default `FALSE` returns the corrected matrix for backward compatibility. `TRUE` returns a `winn_result` list with the matrix, selected parameters, batch assignments, and stage decisions.

Details

The correction pipeline is as follows:

1. Outlier adjustment using MAD.
2. Drift correction using autocorrelation-based detrending.
3. Batch effect correction using ANOVA-based mean-adjustment or ComBat.
4. Per-sample median adjustment using dilution factor.
5. Optional scaling by batch.

When control samples are provided and parameters="auto", the function performs comprehensive parameter optimization by testing multiple combinations of settings and selecting those that maximize control sample correlation while minimizing coefficient of variation. This searches spline

methods, FDR thresholds, and normalization approaches while respecting the supplied autocorrelation-test family and `scale_by_batch` setting. Auto mode requires at least two controls. In auto mode, `fdr_threshold`, `spline_method`, and `median_adjustment` are fixed-mode settings and are replaced by the documented internal candidate grids. Other arguments still constrain the candidate family. Set `return_details = TRUE` to retain the selected settings and stage decisions in machine-readable form.

The full pipeline expects finite, non-negative quantitative intensities. Zeros are treated as observed values and are not converted to missingness. Users should recode declared non-detection sentinels and impute missing values using a documented preprocessing policy before calling `winn()`.

Value

A numeric matrix of corrected intensities, or a structured `winn_result` when `return_details = TRUE`.

Examples

```
set.seed(1)
your_data_matrix <- matrix(rnorm(200, mean = 100, sd = 15), nrow = 20)
batch <- rep(1:4, length.out = ncol(your_data_matrix))
run_order <- seq_len(ncol(your_data_matrix))
corrected_data <- winn(your_data_matrix, batch = batch, run_order = run_order)
```

---

winn_ablation *Run WiNN as Explicit Ablation Stages*

Description

A thin, fixed-parameter orchestration layer around the package's existing outlier, drift, batch, and dilution-factor functions. It is intended for controlled mechanistic ablations and returns every intermediate matrix and gate diagnostic without changing the standard `winn()` API.

Usage

```
winn_ablation(
  data,
  batch,
  run_order = NULL,
  control_samples = NULL,
  parameters = "fixed",
  use_outlier_shrinkage = TRUE,
  drift_gate = c("selective", "none", "all"),
  batch_gate = c("selective", "none", "all"),
  pqn_mode = c("shrink", "none"),
  fdr_threshold = 0.05,
  test = "Ljung-Box",
  lag = NULL,
```

```
  ljung_box_fitdf = 0L,
  spline_method = "conservative",
  return_intermediates = TRUE,
  return_diagnostics = TRUE
)
```

Arguments

data	Numeric feature-by-sample matrix or data frame.
batch	Supplied batch or segment label for every sample.
run_order	Numeric acquisition order for every sample.
control_samples	Optional control-sample column indices. Use NULL for QC-agnostic ablations.
parameters	Must be "fixed"; automatic tuning is deliberately not available through this interface.
use_outlier_shrinkage	Whether to apply adjust_outliers_mad().
drift_gate	One of "none", "selective", or "all".
batch_gate	One of "none", "selective", or "all".
pqn_mode	One of "none" or "shrink".
fdr_threshold	Fixed FDR threshold for drift and batch gates.
test	Autocorrelation test, normally "Ljung-Box".
lag	Optional fixed lag; NULL uses the package's adaptive lag.
ljung_box_fitdf	Non-negative Ljung-Box fitted-model degrees of freedom. The default is 0; use 1 only for legacy reproduction.
spline_method	Drift smoother, normally "conservative".
return_intermediates	Whether to return all stage matrices.
return_diagnostics	Whether to return gate and magnitude diagnostics.

Value

A list containing the final matrix, intermediate matrices, diagnostics, stage runtimes, and the exact fixed configuration.

Examples

```
set.seed(1)
example_data <- matrix(abs(rnorm(120, mean = 100, sd = 15)), nrow = 10)
example_batch <- rep(1:3, each = 4)
example_result <- winn_ablation(
  example_data,
  batch = example_batch,
  run_order = seq_len(ncol(example_data)),
```

```
  drift_gate = "none",
  batch_gate = "none",
  pqn_mode = "none"
)
dim(example_result$data)
```

Index