

Package: postexportKinetics (via r-universe)

September 26, 2026

Title Kinetic Model Comparison for Post-Export RNA Conversion

Version 0.99.0

Description Compartment-resolved four-state kinetic modelling of nuclear and cytoplasmic, unprocessed and processed RNA. Implements a constrained nested-model comparison between a null model without a post-export conversion rate and a full model with a non-negative post-export conversion rate, using trapezoidal interval balances, propagated and regularised measurement covariance, whitened non-negative least squares and a replicate-level generative bootstrap with an add-one p-value. Also provides simulation with the manuscript simulator and assay-noise models, diagnostic comparison of sampling designs with the manuscript benchmark, multiple-testing adjustment, exploratory candidate ranking, tidy result tables and plots. The post-export conversion rate is a phenomenological parameter and does not identify a specific molecular mechanism.

License MIT + file LICENSE

URL <https://github.com/bioinformatics-sannio/postexportKinetics>

BugReports <https://github.com/bioinformatics-sannio/postexportKinetics/issues>

Encoding UTF-8

LazyData false

Depends R (>= 4.1.0)

Imports deSolve, ggplot2 (>= 3.4.0), MASS, nnls, stats,
SummarizedExperiment, utils

Suggests expm, knitr, rmarkdown, testthat (>= 3.1.7)

VignetteBuilder knitr

Config/testthat/edition 3

biocViews Software, Transcriptomics, RNASeq, AlternativeSplicing,
TimeCourse, StatisticalMethod, MultipleComparison,
Visualization

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0
Config/pak/sysreqs zlib1g-dev
Repository https://biocstaging.r-universe.dev
Date/Publication 2026-09-26 11:05:44 UTC
RemoteUrl https://github.com/BiocStaging/postexportKinetics
RemoteRef HEAD
RemoteSha d099ac79d1bacc8edd24a5d9aa4f3ffaea67ee0f

Contents

adjust_postexport_pvalues	2
as.data.frame.postexport	4
check_operational_domain	5
fit_postexport_model	7
plot.postexport	9
postexport_control	11
postexport_data	13
postexport_data_from_se	14
postexport_example	16
postexport_example_truth	17
rank_postexport_candidates	18
simulate_postexport_kinetics	20
test_postexport_conversion	23
validate_postexport_data	25
Index	28

adjust_postexport_pvalues
<i>Adjust bootstrap p-values for multiple testing</i>

Description

Computes q-values for a set of events tested with `test_postexport_conversion()`, using `stats::p.adjust()`. Only valid tests (status "ok" and a finite p-value) enter the adjustment; other events keep their status and receive `q_value = NA`. Raw p-values are never modified.

Usage

`adjust_postexport_pvalues(x, method = "BH", groups = NULL)`

Arguments

x	A postexport_test_set (from <code>test_postexport_conversion()</code> with several events) or a single postexport_test.
method	Adjustment method passed to <code>stats::p.adjust()</code> ; default "BH" (Benjamini-Hochberg), as in the manuscript.
groups	Optional family labels: a vector with one label per event in the order of <code>x\$summary\$event</code> , or a vector named by event. NULL (default) uses a single family.

Value

x with a `q_value` column (and `adjustment_group`) added to `x$summary`, `q_value` added to each result's inference, and an adjustment element (method, families, numbers of adjusted and excluded events). Existing fields are unchanged. For a single postexport_test, `q_value` and `adjustment_method` are added to `x$inference` and the adjustment element is added to x.

Families

In the manuscript, Benjamini-Hochberg adjustment was applied separately within each dataset. By default all events of x form one family; with groups, the adjustment is performed separately within each group (for example one group per dataset). Combining events from different datasets in one family changes the q-values.

Single tests

A single postexport_test is also accepted. It is adjusted as a multiple-testing family of size one: `stats::p.adjust()` is applied normally, which under BH gives $q = p$. Adjustment is always an explicit step and is never performed inside `test_postexport_conversion()`. Multi-event studies should define their testing family (for example all events of one dataset) according to the experimental or dataset analysis plan, and adjust that family jointly rather than event by event.

See Also

`rank_postexport_candidates()`, `stats::p.adjust()`

Examples

```
# Synthetic example data (simulated; see ?postexport_example).
data(postexport_example)
ev <- c("alt_1", "alt_3", "null_2")
x <- postexport_data(postexport_example[postexport_example$event %in% ev, ],
  time_unit = "min")
# B = 49 is used only to keep the example fast and is not recommended for
# final scientific analysis: use the default B = 1999 or more.
res <- test_postexport_conversion(
  x, t_star = 332,
  control = postexport_control(B = 49, seed = c(alt_1 = 1, alt_3 = 2,
    null_2 = 3)))
res <- adjust_postexport_pvalues(res)
res$summary[, c("event", "status", "p_value", "q_value")]
```

```
# A single test is a family of size one (under BH, q = p).
one <- adjust_postexport_pvalues(res$results$alt_3)
c(p = one$inference$p_value, q = one$inference$q_value)
```

as.data.frame.postexport

Tidy result tables

Description

Flat, stable, user-facing tables of fit and test results, suitable for saving (for example with `utils::write.csv()`). Internal matrices are not included; the per-event result objects remain available unchanged.

Usage

```
## S3 method for class 'postexport_fit'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S3 method for class 'postexport_fit_set'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)

## S3 method for class 'postexport_test_set'
as.data.frame(x, row.names = NULL, optional = FALSE, ...)
```

Arguments

<code>x</code>	A <code>postexport_fit</code> , <code>postexport_test</code> , <code>postexport_fit_set</code> or <code>postexport_test_set</code> .
<code>row.names</code> , <code>optional</code>	Ignored (for compatibility with the generic).
<code>...</code>	Ignored.

Value

A data frame with one row per event.

Columns

identifier / status event, status, error_message.

inference p_value (bootstrap, one-sided), q_value (only after `adjust_postexport_pvalues()`; otherwise absent), adjustment_group, B, n_bootstrap_valid, bootstrap_failure_rate, atom_zero.

effect size sigma_c (phenomenological post-export conversion rate).

fit IR (relative RSS improvement), T_obs, RSS_null, RSS_full.

boundary at_boundary (frozen rule), sigma_c_zero, boundary_tolerance.

diagnostics condition_number, condition_number_nonzero_columns, min_singular_value, rank_full, rank_null, structurally_zero_columns.

design t_star, time_unit, n_time_points, min_replicates, design_warnings.

The role of every column is also given in the attribute "column_roles".

Examples

```
data(postexport_example)
fits <- fit_postexport_model(
  postexport_data(postexport_example, time_unit = "min"), t_star = 332)
tab <- as.data.frame(fits)
tab[, c("event", "status", "sigma_c", "IR", "at_boundary")]
attr(tab, "column_roles")[c("sigma_c", "IR", "at_boundary")]
```

check_operational_domain

Compare a design with the manuscript benchmark (diagnostic only)

Description

Reports how configurations of the manuscript's corrected factorial benchmark that match (or are nearest to) an experimental design behaved: empirical Type-I error at $\alpha = 0.05$ with its Wilson 95% interval, whether the manuscript's operational criterion was met for those benchmark configurations, and power. This is an empirical comparison with simulated benchmark designs; it is not a calibration guarantee for any dataset, it never blocks or alters inference and it gives no valid/invalid verdict.

Usage

```
check_operational_domain(
  data = NULL,
  regime,
  t_star = NULL,
  platform = NULL,
  noise_level = NULL,
  n_time_points = NULL,
  n_replicates = NULL,
  sampling_interval = NULL,
  time_unit = NULL
)
```

Arguments

data	Optional postexport_data object. The number of time points, the minimum number of replicates per time point, the sampling interval (if regular) and the time unit are then taken from each event. Do not combine with n_time_points, n_replicates, sampling_interval or time_unit.
------	--

regime	"NONE", "SHUTOFF" or "PSEUDO_SHUTOFF"; required. Not inferred from the data.
t_star	Intervention time (for data with a shutoff regime); used to describe the sampling layout around t_star.
platform	Optional measurement platform ("gaussian", "rnaseq", "rtqpcr"); never inferred from abundances. NULL reports all platforms.
noise_level	Optional noise level ("very_low", "low", "medium", "high"); never inferred. NULL reports all levels.
n_time_points, n_replicates, sampling_interval	Explicit design (without data): number of distinct sampling times, replicates per time point and regular sampling interval (NA if irregular).
time_unit	Time unit of sampling_interval (without data).

Value

An object of class `postexport_domain_check`: a list with designs (one row per distinct design and the events sharing it), checks (per design: design, match_type = "exact", "nearest" or "not_benchmarked", matches (benchmark configurations), differences (for nearest designs: user value and benchmark levels per dimension), summary, statements and caveats) and provenance. For nearest designs, matches has an additional column `differs_in`.

Benchmark

1152 configurations of the corrected factorial benchmark (tag manuscript-revision-v1.0): regimes NONE (continuous transcription) and SHUTOFF (complete shutoff); platforms gaussian, rnaseq, rtqpcr; noise levels very_low, low, medium, high; 3, 5 or 10 replicates; sampling intervals 5, 10, 20, 50; nominal 3, 5, 10 or 20 time points (matching uses the effective number of distinct sampling times). Each configuration simulated 2000 genes (about 20% with $\sigma_c > 0$) with rates per minute, a common shutoff time 332, onset shifts between -100 and 100 and $B = 1999$ bootstrap replicates. SHUTOFF designs sampled one point at $t_{\text{star}} - \text{interval}$, one at t_{star} and the remaining points at spacing interval after t_{star} . Pseudo-shutoff designs are not part of this benchmark. The operational criterion of the manuscript (`analyze_benchmark_corrected_onset_final.R`) is that the Wilson 95% interval of the empirical Type-I error contains 0.05; the Type-I error lying in [0.025, 0.075] is reported as a secondary descriptive flag.

Matching

A benchmark configuration matches exactly when the regime, the platform and noise level (when supplied; otherwise all are included), the number of time points, the number of replicates and, for SHUTOFF, the sampling interval are equal. The sampling interval is compared only when it is exactly regular and the time unit is minutes (the benchmark time scale); units are never converted. Without an exact match, the set of nearest evaluated benchmark designs (not equivalent designs) is returned: among configurations with the same regime, platform and noise level, for every ordering of the design dimensions (sampling interval, number of time points, number of replicates) the configurations at the nearest evaluated level of each dimension in turn are kept, ties included, and the union over all orderings is reported. No single configuration is singled out as the closest, no calibration is interpolated and no score is computed. Each returned configuration lists the dimensions in which it differs (`differs_in`).

Vocabulary

Package labels map one to one to the manuscript benchmark labels: platforms gaussian = GAUSS, rnaseq = RNA-seq, rtqpcr = RT-qPCR; noise levels very_low = Very low, low = Low, medium = Medium, high = High.

See Also

[test_postexport_conversion\(\)](#)

Examples

```
check_operational_domain(regime = "SHUTOFF", platform = "rnaseq",
                          noise_level = "medium", n_time_points = 5,
                          n_replicates = 3, sampling_interval = 10,
                          time_unit = "min")
check_operational_domain(regime = "NONE", n_time_points = 5,
                          n_replicates = 3)
```

fit_postexport_model	<i>Fit the full and null post-export kinetic models</i>
----------------------	---

Description

Fits, for each event, the full model (post-export conversion rate $\sigma_c \geq 0$) and the constrained null model ($\sigma_c = 0$) to compartment-resolved time-course data, without bootstrap inference.

Usage

```
fit_postexport_model(
  data,
  t_star,
  control = postexport_control(),
  events = NULL
)
```

Arguments

data	A postexport_data object.
t_star	Intervention (transcriptional shutoff) time on the time axis of data, or NULL for continuous transcription. Must be supplied explicitly.
control	A postexport_control() object. Only the covariance and fitting settings are used.
events	Optional character vector of events to fit; default all.

Value

For one event, an object of class `postexport_fit` with elements

`event`, `status`, `error_message` Frozen status codes: "ok", "observed_system_failed" or "observed_fit_failed".

`estimates` `sigma_c` (effect-size estimate), `coef_full`, `coef_null` (named, original scale; `sigma_c = 0` in the null).

`fit` `RSS_null`, `RSS_full` (whitened), `T_obs`, `delta_RSS`, `IR` (relative RSS improvement $\text{delta_RSS} / \text{RSS_null}$).

`boundary` `tolerance`, `at_boundary` (frozen rule), `sigma_c_zero` ($|\text{sigma_c}| < 1\text{e-}12$).

`diagnostics` condition number and minimum singular value of the whitened full design, ranks of the full and null designs, pooled covariance, structurally zero design columns and the condition number over the remaining columns (diagnostic only).

`design` `t_star`, `time_unit`, `intervention`, time points and replicates per time point.

`system` the interval-balance system (`A`, `b`, `Sigma_b`, ...).

`data` the event's input rows (used by `plot()`).

`control`, `provenance` settings and frozen reference.

For several events, a `postexport_fit_set`: a list with summary (one row per event) and results (named list of `postexport_fit`). Events are processed sequentially; an unexpected error while processing one event is recorded as status "event_error" (with the message in `error_message`) and the remaining events are processed. See also [as.data.frame\(\)](#) for a tidy table.

Model

Four kinetic states are required: nuclear unprocessed `N`, nuclear processed `N_s`, cytoplasmic unprocessed `C` and cytoplasmic processed `C_s`, with

$$\begin{aligned} dN/dt &= R - (\sigma_n + \tau)N, & dN_s/dt &= \sigma_n N - \tau_s N_s, \\ dC/dt &= \tau N - (\sigma_c + \alpha)C, & dC_s/dt &= \tau_s N_s + \sigma_c C - \alpha_s C_s. \end{aligned}$$

Parameters are estimated in the order `R`, `tau`, `tau_s`, `sigma_c`, `sigma_n`, `alpha`, `alpha_s` from trapezoidal interval balances between consecutive time points. The measurement covariance is estimated from replicates within time points (shrinkage and eigenvalue floor as set by `postexport_control()`) and propagated to the interval balances; both models are fitted by non-negative least squares after whitening. The transcription input is active until `t_star` and zero afterwards (complete shutoff).

Interpretation

`sigma_c` is a phenomenological post-export conversion rate from `C` to `C_s`. A positive estimate does not identify a unique molecular mechanism and is not, by itself, evidence of cytoplasmic splicing. The fitted values depend on the experimental design; in particular the complete-shutoff assumption matters for pseudo-shutoff data, which are fitted as if transcription stopped completely at `t_star`.

Boundary

The statistic $T = \max(0, \text{RSS}_0 - \text{RSS}_1)$ is computed on the whitened scale. The fit is on the boundary when $T \leq 1\text{e-}10 * \max(1, |\text{RSS}_0|, |\text{RSS}_1|)$ (frozen rule); `test_postexport_conversion()` then reports `p = 1`.

See Also

[test_postexport_conversion\(\)](#), [postexport_data\(\)](#), [postexport_control\(\)](#)

Examples

```
# Synthetic example data (simulated; see ?postexport_example).
data(postexport_example)
x <- postexport_data(
  postexport_example[postexport_example$event == "alt_3", ],
  time_unit = "min")
fit <- fit_postexport_model(x, t_star = 332)
fit
summary(fit)
```

plot.postexport	<i>Plot post-export kinetic results</i>
-----------------	---

Description

ggplot2 plots of fit, test, simulation and operational-domain results. Plotting never modifies the object, never refits and never runs inference; each method returns a ggplot object.

Usage

```
## S3 method for class 'postexport_fit'
plot(x, type = "fit", show_means = TRUE, ...)

## S3 method for class 'postexport_test'
plot(x, type = c("fit", "bootstrap"), show_means = TRUE, ...)

## S3 method for class 'postexport_fit_set'
plot(x, type = c("status", "boundary", "sigma_IR"), ...)

## S3 method for class 'postexport_test_set'
plot(
  x,
  type = c("status", "boundary", "sigma_IR", "sigma_q", "IR_q", "score"),
  ...
)

## S3 method for class 'postexport_simulation'
plot(x, show_observed = TRUE, ...)

## S3 method for class 'postexport_domain_check'
plot(x, ...)
```

Arguments

<code>x</code>	A result object.
<code>type</code>	Plot type (see Details).
<code>show_means</code>	Logical; show time-point means (fits and tests).
<code>...</code>	Ignored.
<code>show_observed</code>	Logical; show noisy observations (simulations).

Value

A ggplot object.

Fits and tests

`type = "fit"` (default) shows, for the four states `N`, `N_s`, `C` and `C_s`, the replicate observations, the time-point means (`show_means`), the sampling times, `t_star` and the fitted full (`sigma_c >= 0`) and null (`sigma_c = 0`) models. The fitted curves are display reconstructions only: they are propagated with the ODE model from the observed replicate-mean state at the first sampled time using the fitted coefficients and the model's transcription schedule. The fit, the test statistic and the bootstrap/null calculations do not use these curves; they use the validated frozen interval-balance / Crank-Nicolson machinery. Plotting never refits or modifies the result. For `postexport_test`, `type = "bootstrap"` shows the bootstrap distribution of `T` with the observed `T`, the boundary atom, the p-value and the failure fraction.

Sets

`type = "status"` (default): counts by status; `"boundary"`: counts by boundary status; `"sigma_IR"`: `sigma_c` against `IR`. For test sets after `adjust_postexport_pvalues()` also `"sigma_q"` (`sigma_c` against $-\log_{10}(q)$), `"IR_q"` (`IR` against $-\log_{10}(q)$) and `"score"` (distribution of the exploratory score of `rank_postexport_candidates()`). The axes keep effect size, fit improvement and evidence separate.

Simulations

Latent trajectories per replicate (no smoothing), sampled latent states, noisy observations (`show_observed`), the transcriptional onset and the intervention time `t_star`.

Operational domain

Manuscript benchmark designs matched exactly or as nearest evaluated designs, with empirical Type-I error at $\alpha = 0.05$, Wilson 95% intervals and the nominal 0.05 reference. This describes benchmark simulations, not the calibration of a dataset.

Examples

```
p <- c(R = 100, tau = 0.03, tau_s = 0.015, sigma_c = 0.1, sigma_n = 0.1,
      alpha = 0.2, alpha_s = 0.08)
sim <- simulate_postexport_kinetics(
  p, times = c(90, 100, 110, 130, 160), onset_time = 0, t_star = 100,
```

```

    regime = "SHUTOFF", time_unit = "min", n_replicates = 3,
    param_cv = 0.05, noise = list(platform = "gaussian", level = "low"),
    seed = 1)
plot(sim)

data(postexport_example)
x <- postexport_data(
  postexport_example[postexport_example$event == "alt_3", ],
  time_unit = "min")
plot(fit_postexport_model(x, t_star = 332))
x <- postexport_data(sim$observed, time_unit = "min")
plot(fit_postexport_model(x, t_star = 100))

```

postexport_control *Numerical and inferential settings*

Description

Collects the settings passed to the frozen numerical core. All defaults are those of the frozen manuscript implementation (tag manuscript-revision-v1.0); results obtained with the defaults are the manuscript procedure. The intervention time `t_star` is a property of the experimental design and is therefore an explicit argument of `fit_postexport_model()` and `test_postexport_conversion()`, not a control setting.

Usage

```

postexport_control(
  B = 1999L,
  seed = NULL,
  lambda_time = 0.5,
  lambda_diag = 0.1,
  rel_floor = 1e-08,
  max_failure_rate = 0.05,
  scaling_A = TRUE,
  truncate_nonnegative_boot = FALSE
)

```

Arguments

- | | |
|------|---|
| B | Number of bootstrap replicates. Frozen default 1999. The bootstrap p-value is $(1 + \#\{T^* \geq T_{\text{obs}}\}) / (B_{\text{valid}} + 1)$, so B determines its resolution and Monte-Carlo error; it does not change the observed fit. Used by <code>test_postexport_conversion()</code> only. |
| seed | NULL (frozen default: the random-number generator is not reset) or an integer seed passed to <code>set.seed()</code> immediately before the observed fit, exactly as in the frozen implementation. For several events, a named integer vector with one seed per event. A fixed seed reproduces the bootstrap draws within a matched |

	numerical environment; across LAPACK/BLAS builds the draws can differ because the frozen bootstrap uses <code>MASS::mvrnorm()</code> . The frozen implementation calls <code>set.seed()</code> ; the public API saves the caller's random-number state before and restores it afterwards, so an explicit seed leaves the global state unchanged (the result is not affected). With <code>seed = NULL</code> the global stream is used and consumed, as in the frozen implementation.
<code>lambda_time</code>	Shrinkage of each time point's within-time covariance towards the pooled covariance, in $[0, 1]$. Frozen default 0.5. Changing it changes the estimated covariance, the whitening and hence the fit, the statistic and the p-value.
<code>lambda_diag</code>	Shrinkage of the pooled within-time covariance towards its diagonal, in $[0, 1]$. Frozen default 0.1. Changing it changes the statistical behaviour as for <code>lambda_time</code> .
<code>rel_floor</code>	Relative eigenvalue floor used to stabilise covariance matrices (eigenvalues are raised to <code>rel_floor</code> times the median positive diagonal element). Frozen default $1e-8$. It only matters for near-singular covariances; changing it can change the fit in such cases.
<code>max_failure_rate</code>	Largest acceptable fraction of failed bootstrap replicates, in $[0, 1]$. Frozen default 0.05. If exceeded, the p-value is NA with status "bootstrap_unstable". It does not change p-values that are reported.
<code>scaling_A</code>	Advanced option. Logical; scale the design-matrix columns to unit norm before the NNLS fits. Frozen default TRUE. This is a numerical conditioning step (coefficients are returned on the original scale), but it changes the floating-point path of the fits and can change results for ill-conditioned designs. Changing it changes the analysis relative to the manuscript procedure.
<code>truncate_nonnegative_boot</code>	Advanced option. Logical; set negative values of simulated bootstrap samples to zero. Frozen default FALSE. Changing it changes the bootstrap null distribution, hence the p-value, and therefore the analysis relative to the manuscript procedure.

Value

An object of class `postexport_control` (a named list).

Advanced options

`scaling_A` and `truncate_nonnegative_boot` exist in the frozen implementation and are exposed for completeness. Their frozen defaults define the manuscript procedure; non-default values change the analysis. `print()` flags every setting that differs from the frozen default.

See Also

`fit_postexport_model()`, `test_postexport_conversion()`

Examples

```
postexport_control()
# A smaller B shortens computation but coarsens the p-value (minimum
```

```
# 1 / (B + 1)); it is not recommended for final scientific analysis.
postexport_control(B = 199, seed = 1)
```

postexport_data	<i>Create a validated post-export kinetics data object</i>
-----------------	--

Description

Builds the canonical representation of compartment-resolved time-course measurements for the four kinetic states used by the model: nuclear unprocessed (N), nuclear processed (N_s), cytoplasmic unprocessed (C) and cytoplasmic processed (C_s) RNA.

Usage

```
postexport_data(x, time_unit, format = c("wide", "long"), columns = NULL)
```

Arguments

x	A data frame in wide or long format.
time_unit	Character string naming the unit of time (for example "min"). Required: kinetic rates, including sigma_c, are expressed per this unit.
format	Either "wide" (default, canonical) or "long".
columns	Optional named character vector mapping canonical column names to column names of x.

Value

An object of class postexport_data: a data frame with columns event (character), time (numeric), replicate (character), N, N_s, C, C_s (numeric), and attributes time_unit, format (the input format) and validation (a postexport_validation object).

Input formats

format = "wide" (**canonical**) One row per destructive biological sample, with columns event, time, replicate, N, N_s, C, C_s.

format = "long" One row per sample and state, with columns event, time, replicate, compartment, state, abundance. compartment must be "nuclear" or "cytoplasmic" and state must be "unprocessed" or "processed". The mapping is fixed: nuclear/unprocessed = N, nuclear/processed = N_s, cytoplasmic/unprocessed = C, cytoplasmic/processed = C_s. For retained-intron events quantified by rMATS, inclusion corresponds to unprocessed and skipping to processed; that assignment is the user's responsibility and is not inferred. Every sample must have exactly one value for each of the four states.

Column names are matched exactly. Different names can be declared with columns, a named character vector mapping canonical names to the names in x (for example c(event = "event_id", time = "minutes")). No other column is interpreted.

Design semantics

Each row is a destructive sample: replicate labels identify samples within a time point and are never used to pair samples across time points. Unequal numbers of replicates per time point are allowed. Row order is preserved.

Missing values

Missing states are an error. The constructor never converts missing values or missing states to zero and never imputes them. Samples with incomplete measurements must be removed (or completed) explicitly before construction.

See Also

[validate_postexport_data\(\)](#), [fit_postexport_model\(\)](#), [test_postexport_conversion\(\)](#)

Examples

```
# Wide format (synthetic example data; see ?postexport_example).
data(postexport_example)
x <- postexport_data(postexport_example, time_unit = "min")
x

# Long format: one row per sample and state.
long <- read.csv(system.file("extdata", "postexport_example_long.csv",
                           package = "postexportKinetics"))
head(long)
y <- postexport_data(long, time_unit = "min", format = "long")
```

postexport_data_from_se

Create post-export kinetics data from a SummarizedExperiment

Description

Converts an already state-resolved SummarizedExperiment (layout A) into the canonical [postexport_data\(\)](#) object. This is an interoperability input path only. All data validation is delegated to [postexport_data\(\)](#) (and so to [validate_postexport_data\(\)](#)), and the result is exactly the object that [postexport_data\(\)](#) returns for the equivalent wide table.

Usage

```
postexport_data_from_se(
  se,
  time_unit,
  assays = c(N = "N", N_s = "N_s", C = "C", C_s = "C_s"),
  time = "time",
  replicate = "replicate"
)
```

Arguments

<code>se</code>	A SummarizedExperiment (or a subclass such as RangedSummarizedExperiment) with events as rows and destructive samples as columns.
<code>time_unit</code>	Character string naming the unit of time (for example "min"); passed to <code>postexport_data()</code> .
<code>assays</code>	Named character vector mapping the four states N, N_s, C and C_s to assay names of <code>se</code> .
<code>time</code>	Name of the <code>colData(se)</code> column holding the sampling time.
<code>replicate</code>	Name of the <code>colData(se)</code> column holding the replicate label within a time point.

Value

A `postexport_data` object, identical to `postexport_data(<wide table>, time_unit)` for the corresponding wide table.

Layout A

- rows are events; `rownames(se)` gives event;
- columns are destructive samples; `colData(se)[[time]]` gives time and `colData(se)[[replicate]]` gives replicate;
- four assays hold the state-resolved abundances, mapped directly by `assays` to N (nuclear unprocessed), N_s (nuclear processed), C (cytoplasmic unprocessed) and C_s (cytoplasmic processed).

Each (event, sample) cell becomes one row of the wide table, ordered by event (row order of `se`) and then by sample (column order of `se`). `rowData`, `metadata` and other `colData` columns and assays are ignored.

What the converter does not do

It expects an object that already holds the four states per destructive sample. It does **not** pair nuclear and cytoplasmic libraries, normalise counts, convert rMATS output, or decide which quantity is "unprocessed" or "processed" (for example inclusion versus skipping of a retained intron). These are analysis decisions made before conversion. Missing values are never zero-filled: they are errors, exactly as in `postexport_data()`.

See Also

`postexport_data()`, `validate_postexport_data()`, `SummarizedExperiment::SummarizedExperiment()`

Examples

```
# Build a layout-A SummarizedExperiment from the synthetic example data.
data(postexport_example)
ex <- postexport_example
events <- unique(ex$event)
samples <- unique(ex[, c("time", "replicate")])
rownames(samples) <- sprintf("%g_r%d", samples$time, samples$replicate)
```

```

cell <- cbind(match(ex$event, events),
              match(paste(ex$time, ex$replicate),
                    paste(samples$time, samples$replicate)))
state_matrix <- function(state) {
  m <- matrix(NA_real_, length(events), nrow(samples),
              dimnames = list(events, rownames(samples)))
  m[cell] <- ex[[state]]
  m
}
se <- SummarizedExperiment::SummarizedExperiment(
  assays = list(N = state_matrix("N"), N_s = state_matrix("N_s"),
                C = state_matrix("C"), C_s = state_matrix("C_s")),
  colData = samples)
x <- postexport_data_from_se(se, time_unit = "min")
x

```

postexport_example	<i>Small synthetic example dataset</i>
--------------------	--

Description

Simulated compartment-resolved time courses for eight events, in the canonical wide format accepted by [postexport_data\(\)](#). The data are synthetic: they were generated with [simulate_postexport_kinetics\(\)](#) (the ported manuscript simulator and assay-noise model). They contain no real measurements and are intended for examples, the vignette and smoke tests.

Usage

```
data(postexport_example)
```

Format

A data frame with 200 rows (8 events x 5 time points x 5 replicates) and 7 columns:

event event identifier (character).

time sampling time in minutes.

replicate replicate label within a time point (integer). Replicates are independent destructive samples and are not paired across time points.

N, N_s, C, C_s simulated abundances of nuclear unprocessed, nuclear processed, cytoplasmic unprocessed and cytoplasmic processed RNA (RNA-seq noise model, arbitrary scale).

Design

One design of the manuscript's corrected factorial benchmark: complete transcriptional shutoff ("SHUTOFF") at the common intervention time $t_{\text{star}} = 332$ min; sampling at $t_{\text{star}} - 10$, t_{star} , $t_{\text{star}} + 10$, $+ 20$ and $+ 30$ min (5 time points); 5 destructive replicates per time point; replicate-level rate variability $\text{param_cv} = 0.05$; RNA-seq assay noise at level "very_low"; integration grid

origin = 0, grid_step = 1, horizon = 1000, y0 = 0. The design was selected from the benchmark table before any data were generated. Among the benchmark's SHUTOFF designs with at most 10 time points and 5 replicates whose Type-I error interval contains 0.05, it had the highest power at $\alpha = 0.05$: about 0.24, with Type-I error 0.042. Most simulated alternative events are therefore not expected to reach significance. The example shows realistic uncertainty and does not guarantee detection.

Events and parameters

Events alt_1 to alt_4 were simulated with $\sigma_c > 0$ and events null_1 to null_4 with $\sigma_c = 0$. For each event, the kinetic parameters were drawn once from the manuscript benchmark ranges, and a transcriptional onset was drawn as a uniform integer in $[-100, 100]$ min. The script's seed, 20260925, was fixed a priori. Neither the design nor the data were chosen by searching over seeds. The true values are in [postexport_example_truth](#). The draws reproduce the benchmark's parameter distributions, not its per-gene random streams. Generation code (distributed with the package): `system.file("scripts", "generate_postexport_example.R", package = "postexportKinetics")`.

The same observations in long format are installed as `system.file("extdata", "postexport_example_long.csv", package = "postexportKinetics")`.

Source

Simulated with [simulate_postexport_kinetics\(\)](#); see `inst/scripts/generate_postexport_example.R`.

See Also

[postexport_example_truth](#), [postexport_data\(\)](#)

Examples

```
data(postexport_example)
head(postexport_example)
x <- postexport_data(postexport_example, time_unit = "min")
x
```

postexport_example_truth

Simulation truth for the example dataset

Description

The parameters used to simulate each event of [postexport_example](#).

Usage

```
data(postexport_example_truth)
```

Format

A data frame with 8 rows and 13 columns:

event event identifier, matching [postexport_example](#).

truth "alternative" ($\sigma_c > 0$) or "null" ($\sigma_c = 0$).

R, tau, tau_s, sigma_c, sigma_n, alpha, alpha_s kinetic parameters, with rates per minute. Replicates vary these rates by $\text{param_cv} = 0.05$, except R.

onset_time transcriptional onset (min).

t_star common intervention time (min).

simulation_seed seed passed to [simulate_postexport_kinetics\(\)](#).

Source

Simulated with [simulate_postexport_kinetics\(\)](#); see `inst/scripts/generate_postexport_example.R`.

See Also

[postexport_example](#)

Examples

```
data(postexport_example_truth)
postexport_example_truth[, c("event", "truth", "sigma_c")]
```

rank_postexport_candidates

Rank candidate events for exploratory follow-up

Description

Orders tested events by the manuscript's exploratory composite score

$$\text{score} = \sigma_c \times IR \times \min(-\log_{10}(\max(q, 10^{-10})), 6)$$

within each adjustment family. This is an exploratory prioritization; it is not inferential evidence and not an optimized discrimination score. Statistical evidence is given by the p- and q-values, σ_c is the effect-size estimate and IR the relative fit improvement.

Usage

```
rank_postexport_candidates(x)
```

Arguments

x A postexport_test_set processed by [adjust_postexport_pvalues\(\)](#).

Value

An object of class `c("postexport_ranking", "data.frame")`, one row per event, ordered by adjustment family and decreasing score, with columns `rank`, `event`, `status`, `score`, `evidence`, `p_value`, `q_value`, `sigma_c`, `IR`, `at_boundary`, `sigma_c_zero`, `adjustment_group`, `n_bootstrap_valid`, `bootstrap_failure_rate`, `condition_number`, `rank_full`.

Families and ties

Ranks are computed independently within each multiple-testing adjustment group; no global rank is formed across distinct families. A user who wants one common ranking must define one common multiple-testing family (`groups = NULL` in `adjust_postexport_pvalues()`) before adjustment and ranking. Tied scores share the same minimum (competition) rank (`ties.method = "min"`) and keep their input order for display.

Provenance

The score is `score_sigma_IR_q` of the frozen manuscript real-data analysis (`real_datasets/run_real_datasets_revision` lines 1453-1614; ordering `order(dataset, -score_sigma_IR_q)`, line 1841), the same composite as in the synthetic score ablation. `IR` is `NA` when the null residual sum of squares is not positive (frozen convention), which makes the score `NA`.

Handling

- `q`-values are required; raw `p`-values are never substituted.
- `q = 0` is floored at `1e-10`, so the evidence term is capped at 6.
- Events that are not rankable (`status` other than `"ok"`, missing `q`, `sigma_c` or `IR`) are kept with `score = NA` and `rank = NA` and are listed last.
- Boundary events (`sigma_c = 0`) have score 0 and are ranked.
- Ties share the same (minimum) rank; tied events keep their input order.

See Also

[adjust_postexport_pvalues\(\)](#)

Examples

```
# Synthetic example data (simulated; see ?postexport_example).
data(postexport_example)
ev <- c("alt_1", "alt_3", "null_2")
x <- postexport_data(postexport_example[postexport_example$event %in% ev, ],
  time_unit = "min")
# B = 49 is used only to keep the example fast and is not recommended for
# final scientific analysis: use the default B = 1999 or more.
res <- test_postexport_conversion(
  x, t_star = 332,
  control = postexport_control(B = 49, seed = c(alt_1 = 1, alt_3 = 2,
    null_2 = 3)))
rank_postexport_candidates(adjust_postexport_pvalues(res))
```

simulate_postexport_kinetics

Simulate compartment-resolved post-export RNA kinetics

Description

Simulates the four-state model (nuclear unprocessed N, nuclear processed N_s, cytoplasmic unprocessed C, cytoplasmic processed C_s) with the frozen manuscript simulator: deterministic latent trajectories, optional replicate-level biological variability and optional assay noise.

Usage

```
simulate_postexport_kinetics(
  params,
  times,
  onset_time,
  t_star,
  regime,
  time_unit,
  residual_fraction = NULL,
  n_replicates = 1L,
  param_cv = 0,
  noise = NULL,
  y0 = c(N = 0, N_s = 0, C = 0, C_s = 0),
  origin = 0,
  grid_step = 1,
  horizon = NULL,
  seed = NULL,
  event = "simulated"
)
```

Arguments

params	Named numeric vector or list with the seven kinetic parameters R, tau, tau _s , sigma _c , sigma _n , alpha _c , alpha _n (finite, non-negative). Names are required; unnamed vectors are not accepted.
times	Sampling times (finite, distinct, >= origin); returned in increasing order.
onset_time	Transcriptional onset time (single finite number).
t_star	Intervention time: a single finite number for "SHUTOFF" and "PSEUDO_SHUTOFF", NULL for "NONE". Must be supplied explicitly.
regime	Transcriptional regime: "NONE" (continuous transcription), "SHUTOFF" (complete shutoff at t _{star}) or "PSEUDO_SHUTOFF" (residual transcription after t _{star}). Must be supplied explicitly.
time_unit	Character string naming the time unit (rates are per this unit).
residual_fraction	For "PSEUDO_SHUTOFF" only: fraction of R remaining after t _{star} , in (0, 1] (the manuscript robustness analysis used 0.05 to 1).

n_replicates	Number of biological replicates (destructive samples per time point); default 1.
param_cv	Coefficient of variation of the replicate-level rate multipliers; default 0 (deterministic replicates).
noise	NULL (no assay noise; default) or a list with elements platform and level (see <i>Replicates and noise</i>).
y0	Named initial state c(N =, N_s =, C =, C_s =) (finite, non-negative); default all zero.
origin	Start of the simulation grid; default 0.
grid_step	Integration grid spacing; default 1.
horizon	End of the integration grid ($\geq \max(\text{times})$); default $\max(\text{times})$. A numerical-reproducibility parameter of the integration grid, not a biological parameter: it does not change the model, only where the grid ends (and hence, at the level of the solver tolerance, the integration intervals).
seed	NULL or a single whole number.
event	Event label used in the returned tables.

Value

An object of class `postexport_simulation`, a list with

`latent` deterministic (noise-free) states on the integration grid for each replicate: event, replicate, time, N, N_s, C, C_s and the transcription rate R in force.

`sampled` latent states at the sampling times.

`observed` noisy observations at the sampling times (event, time, replicate, N, N_s, C, C_s), or NULL without noise; can be passed to `postexport_data()`.

`params`, `replicate_parameters`, `steady_state` base parameters, per-replicate parameters and the closed-form steady state under constant transcription.

`timing` regime, `onset_time`, `t_star`, `residual_fraction`, post-intervention transcription fraction, `origin`, `grid_step`, sampling times.

`replication`, `noise`, `y0`, `rng`, `time_unit`, `provenance` design, noise settings, random-number meta-data and the frozen reference.

Model

$$dN/dt = R(t) - (\sigma_n + \tau)N, \quad dN_s/dt = \sigma_n N - \tau_s N_s,$$

$$dC/dt = \tau N - (\sigma_c + \alpha)C, \quad dC_s/dt = \tau_s N_s + \sigma_c C - \alpha_s C_s.$$

`sigma_c` is a phenomenological post-export conversion rate. Simulating with `sigma_c > 0` generates data from this kinetic model; it does not represent or establish any molecular mechanism.

Timing semantics

All times are on one experimental axis, in `time_unit`.

- `onset_time` is the (gene-specific) transcriptional onset: $R(t) = 0$ before it. It may lie before origin; the history from `onset_time` to origin is then integrated first.
- `t_star` is the common intervention (shutoff) time. It is not shifted by the onset: changing `onset_time` changes only the transcriptional history before the intervention. No time shift is drawn internally.
- Transcription schedule (right-continuous): $R(t) = 0$ for $t < \text{onset_time}$; R for $\text{onset_time} \leq t < t_star$; and, from `t_star` on, 0 ("SHUTOFF"), `residual_fraction * R("PSEUDO_SHUTOFF")`; "NONE" keeps R throughout.
- `y0` is the state at $\min(\text{onset_time}, \text{origin})$. The trajectory is integrated between consecutive points of the grid `seq(origin, horizon, by = grid_step)` (plus the sampling times and `t_star`) with `deSolve::ode()` (method `lsoda`, default tolerances), with onset and `t_star` as breakpoints. The manuscript benchmark used `origin = 0`, `grid_step = 1`, `horizon = 1000` and `y0 = 0`. The grid determines the integration intervals (and so values at the level of the solver tolerance) but not the model.

Replicates and noise

With `n_replicates > 1` and `param_cv > 0`, each replicate multiplies every kinetic rate by an independent log-normal factor with coefficient of variation `param_cv` (R is kept fixed), as in the frozen simulator; the manuscript benchmark used `param_cv = 0.05` and 3, 5 or 10 replicates. Replicates are independent destructive samples. `noise` adds assay noise to the sampled states with the benchmark presets: platform one of "gaussian", "rnaseq", "rtqpcr" and level one of "very_low", "low", "medium", "high" (benchmark labels GAUSS, RNA-seq, RT-qPCR and Very low, Low, Medium, High). Gaussian noise can give negative values (not truncated); RT-qPCR output is on the 2^{-Ct} scale.

Manuscript benchmark settings

The defaults (`n_replicates = 1`, `param_cv = 0`, `noise = NULL`) give a deterministic latent trajectory. The manuscript's corrected factorial benchmark used `param_cv = 0.05`, 3, 5 or 10 replicates, `origin = 0`, `grid_step = 1`, `horizon = 1000`, `y0 = 0`, a common `t_star = 332`, gene-specific onsets between -100 and 100 and the benchmark noise presets. Its per-gene and per-stream seeding is not reproduced by this function.

Vocabulary

Noise platforms and levels map one to one to the manuscript benchmark labels: `gaussian = GAUSS`, `rnaseq = RNA-seq`, `rtqpcr = RT-qPCR`; `very_low = Very low`, `low = Low`, `medium = Medium`, `high = High`.

Random numbers and reproducibility

Random numbers are drawn in this order: per replicate, one normal draw for each of `tau`, `tau_s`, `alpha`, `alpha_s`, `sigma` (the draw for R is discarded, as in the frozen simulator); then the assay noise for the states N , C , C_s , N_s . No draws are made when `param_cv = 0` and `noise = NULL`. With an explicit seed, `set.seed()` is

called first and the caller's random-number state is restored afterwards; with `seed = NULL` the global stream is used and consumed. ODE integration (compiled `lsoda`) can differ across platforms at the level of floating-point rounding; random draws are generated by R itself.

See Also

[postexport_data\(\)](#), [test_postexport_conversion\(\)](#)

Examples

```
p <- c(R = 100, tau = 0.03, tau_s = 0.015, sigma_c = 0.1, sigma_n = 0.1,
      alpha = 0.2, alpha_s = 0.08)
sim <- simulate_postexport_kinetics(
  p, times = c(90, 100, 110, 130, 160), onset_time = 0, t_star = 100,
  regime = "SHUTOFF", time_unit = "min", n_replicates = 3,
  param_cv = 0.05, noise = list(platform = "gaussian", level = "low"),
  origin = 0, grid_step = 1, seed = 1
)
sim
head(sim$observed)
```

test_postexport_conversion

Test for an additional post-export conversion component

Description

Compares, for each event, the constrained null model ($\sigma_c = 0$) with the full model ($\sigma_c \geq 0$); the scientific alternative is $\sigma_c > 0$. The observed statistic $T = \max(0, \text{RSS}_0 - \text{RSS}_1)$ (whitened non-negative least squares) is calibrated by a replicate-level generative bootstrap under the fitted null model.

Usage

```
test_postexport_conversion(
  data,
  t_star,
  control = postexport_control(),
  events = NULL
)
```

Arguments

<code>data</code>	A <code>postexport_data</code> object.
<code>t_star</code>	Intervention (transcriptional shutoff) time on the time axis of data, or <code>NULL</code> for continuous transcription. Must be supplied explicitly.
<code>control</code>	A postexport_control() object; <code>B</code> and <code>seed</code> control the bootstrap.
<code>events</code>	Optional character vector of events to fit; default all.

Value

For one event, an object of class `c("postexport_test", "postexport_fit")` with all elements of a `postexport_fit` (values taken from the frozen result) plus

`inference` `p_value` (bootstrap, one-sided), `B`, `n_bootstrap_valid`, `bootstrap_failure_rate`, `atom_zero` (fraction of T^* at the boundary), `T_boot`, and descriptions of the statistic, alternative and method.

`bootstrap_diagnostics` condition-number summaries and rank-deficient fraction over bootstrap replicates.

`null_means` null mean trajectory used by the bootstrap.

`rng` seed, `RNGkind()` and, for seed = NULL, the random-number state before the test.

`raw` the complete frozen result list.

Frozen status codes: "ok", "observed_system_failed", "observed_fit_failed", "null_trajectory_failed", "bootstrap_unstable". For several events, a `postexport_test_set` with summary and results; events are processed sequentially and an unexpected per-event error is recorded as status "event_error".

Use `adjust_postexport_pvalues()` for q-values, `rank_postexport_candidates()` for exploratory ranking and `as.data.frame()` for a tidy table.

Procedure

The procedure is the frozen manuscript implementation (tag `manuscript-revision-v1.0`), called unchanged:

1. interval-balance system and whitened NNLS fits of the full and null models (as in `fit_postexport_model()`);
2. null mean trajectory by Crank-Nicolson propagation of the fitted null model from the observed mean at the first time point;
3. `B` bootstrap datasets: at each time point, as many independent multivariate normal samples as observed replicates, around the null means, with the estimated within-time covariance;
4. in every bootstrap dataset the means, covariances, interval balances and propagated covariance are rebuilt and both models are refitted, giving T^* ;
5. if more than `max_failure_rate` of the replicates fail, the p-value is NA (status "bootstrap_unstable");
6. boundary rule: if $T \leq 1e-10 * \max(1, |RSS_0|, |RSS_1|)$ the p-value is 1; otherwise the add-one p-value $(1 + \#\{T^* \geq T\}) / (B_{\text{valid}} + 1)$.

This is not a likelihood-ratio test: no likelihood is evaluated, and the statistic is a difference of whitened residual sums of squares.

Interpretation

`sigma_c` is a phenomenological post-export conversion rate. A small p-value indicates kinetic evidence consistent with an additional post-export conversion component within the model; `sigma_c > 0` does not identify a unique molecular mechanism and is not, by itself, evidence of cytoplasmic splicing. Calibration of the bootstrap is design dependent (sampling times relative to `t_star`, replication, noise), and pseudo-shutoff data are fitted as if transcription stopped completely at `t_star`. The p-value is the inferential quantity; `sigma_c` is an effect-size estimate; neither is a ranking score. No multiple-testing adjustment is applied: for several events, adjust the p-values explicitly, e.g. with `stats::p.adjust()`.

Reproducibility

With a fixed seed, the bootstrap draws are reproducible within a matched numerical environment (operating system, LAPACK/BLAS, R and package versions). Across LAPACK/BLAS builds the draws, and hence Monte-Carlo p-values, can differ even under the same seed, because the frozen bootstrap uses `MASS::mvrnorm()`. With an explicit seed, the frozen implementation calls `set.seed()` before the observed fit; the caller's random-number state is saved before and restored after each event, so the call does not alter it. With `seed = NULL`, the global random-number stream is used and consumed, as in the frozen implementation.

See Also

`fit_postexport_model()`, `postexport_control()`

Examples

```
# Synthetic example data (simulated; see ?postexport_example).
data(postexport_example)
x <- postexport_data(
  postexport_example[postexport_example$event == "alt_3", ],
  time_unit = "min")
# B = 99 is used only to keep the example fast and is not recommended for
# final scientific analysis: use the default B = 1999 or more (the
# smallest attainable p-value is 1 / (B + 1)).
res <- test_postexport_conversion(
  x, t_star = 332, control = postexport_control(B = 99, seed = 1)
)
res
summary(res)
```

validate_postexport_data

Validate post-export kinetics data

Description

Checks compartment-resolved time-course data before model fitting and reports findings at three levels. Validation never modifies the data.

Usage

```
validate_postexport_data(
  x,
  format = c("wide", "long"),
  columns = NULL,
  t_star = NULL
)
```

Arguments

<code>x</code>	A data frame (wide or long format) or a <code>postexport_data</code> object.
<code>format</code>	Either "wide" or "long"; ignored for <code>postexport_data</code> .
<code>columns</code>	Optional named character vector mapping canonical column names to column names of <code>x</code> ; ignored for <code>postexport_data</code> .
<code>t_star</code>	Optional intervention (transcriptional shutoff) time, on the time axis of the data, used for design diagnostics.

Value

An object of class `postexport_validation`: a list with data frames errors, warnings and info (columns event, check, message) and design (one row per event: number of samples, time points, and minimum/maximum replicates per time point).

Errors

- missing identifier columns (event, time, replicate) or state columns (N, N_s, C, C_s), or malformed long-format structure;
- missing identifier values; non-numeric or non-finite time;
- non-numeric state columns;
- missing (NA) or non-finite state observations;
- duplicated event/time/replicate samples;
- an event with fewer than two distinct time points (the frozen core requires at least two);
- an event without any time point having at least two replicates (the frozen core cannot estimate the within-time covariance);
- an invalid intervention time `t_star` (not NULL and not a single finite number).

Warnings

Negative abundances (accepted and used unchanged). An intervention time `t_star` at or before the first sampled time of an event (the transcription rate R is then not estimable), or at or after the last sampled time (no post-intervention interval is observed; the fit equals continuous transcription). Warnings never block or alter inference.

Diagnostic information

Unsorted time points; unequal replication across time points; time points with a single replicate (the frozen core then uses the pooled covariance for that time point); events with exactly two time points. These are reported for information only and are not thresholds.

See Also

[postexport_data\(\)](#)

Examples

```
tab <- data.frame(  
  event = "e1", time = rep(c(0, 30, 60), each = 2), replicate = 1:2,  
  N = c(10, 11, 8, 9, 6, 7), N_s = c(5, 6, 5, 5, 4, 5),  
  C = c(3, 3, 3, 2, 2, 2), C_s = c(9, 8, 9, 9, 8, -0.1)  
)  
validate_postexport_data(tab, t_star = 0)
```

Index

* datasets

- postexport_example, 16
- postexport_example_truth, 17
- adjust_postexport_pvalues, 2
- adjust_postexport_pvalues(), 4, 10, 18, 19, 24
- as.data.frame(), 8, 24
- as.data.frame.postexport, 4
- as.data.frame.postexport_fit
 - (as.data.frame.postexport), 4
- as.data.frame.postexport_fit_set
 - (as.data.frame.postexport), 4
- as.data.frame.postexport_test_set
 - (as.data.frame.postexport), 4
- check_operational_domain, 5
- deSolve::ode(), 22
- fit_postexport_model, 7
- fit_postexport_model(), 11, 12, 14, 24, 25
- MASS::mvrnorm(), 12, 25
- plot(), 8
- plot.postexport, 9
- plot.postexport_domain_check
 - (plot.postexport), 9
- plot.postexport_fit(plot.postexport), 9
- plot.postexport_fit_set
 - (plot.postexport), 9
- plot.postexport_simulation
 - (plot.postexport), 9
- plot.postexport_test(plot.postexport), 9
- plot.postexport_test_set
 - (plot.postexport), 9
- postexport_control, 11
- postexport_control(), 7–9, 23, 25
- postexport_data, 13

- postexport_data(), 9, 14–17, 21, 23, 26
- postexport_data_from_se, 14
- postexport_example, 16, 17, 18
- postexport_example_truth, 17, 17
- rank_postexport_candidates, 18
- rank_postexport_candidates(), 3, 10, 24
- set.seed(), 11, 12, 22, 25
- simulate_postexport_kinetics, 20
- simulate_postexport_kinetics(), 16–18
- stats::p.adjust(), 2, 3, 24
- SummarizedExperiment::SummarizedExperiment(), 15
- test_postexport_conversion, 23
- test_postexport_conversion(), 2, 3, 7–9, 11, 12, 14, 23
- utils::write.csv(), 4
- validate_postexport_data, 25
- validate_postexport_data(), 14, 15