Package 'toxdrc'

July 28, 2026

Title Pipeline for Dose-Response Curve Analysis

Version 2.0.0

Description Provides a variety of tools for assessing dose response curves, with an emphasis on toxicity test data. The main feature of this package are modular functions which can be combined through the namesake pipeline, 'runtoxdrc', to automate the analysis for large and complex datasets. This includes optional data preprocessing steps, like outlier detection, solvent effects, blank correction, averaging technical replicates, and much more. Additionally, this pipeline is adaptable to any long form dataset, and does not require specific column or group naming to work.

Maintainer Jack Salole <salolej@mcmaster.ca>

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

URL https://github.com/jsalole/toxdrc

BugReports https://github.com/jsalole/toxdrc/issues

Imports dplyr, magrittr, outliers, purrr, rlang, stats, drc

Suggests testthat (>= 3.1.5)

Config/testthat/edition 3

Depends R (>= 4.1)

LazyData true

NeedsCompilation no

Author Jack Salole [aut, cre]

Repository CRAN

Date/Publication 2026-07-28 21:20:02 UTC

Contents

acutetox 2
averageresponse 4

blankcorrect 5
cellglow 6
checktoxicity 7
config_runtoxdrc 8
flagCV 9
getECx 9
getmetadata 11
interpolateECx 11
modelcomp 13
normalizeresponse 15
pctl 16
print.toxdrc_preset 17
removeoutliers 17
runtoxdrc 18
toxdrc_endpoint 20
toxdrc_modelling 21
toxdrc_normalization 23
toxdrc_output 24
toxdrc_preset 24
toxdrc_qc 26
toxdrc_toxicity 27
toxresult 28

Index **29**

`acutetox` *Simulated acute lethality test data with a quantal endpoint*

Description

A simulated acute lethality study, for demonstrating and testing the quantal path through [runtoxdrc()]. Organisms are scored affected or unaffected at each concentration, so the response is a count out of a known group size rather than a measurement.

Usage

```
acutetox
```

Format

‘acutetox‘ A data frame with 54 rows and 8 columns:

TestID Combination of Test_Number, Substance and Replicate

Test_Number Identifying number of each test substance

Substance Substance label, named for the response pattern it shows

Replicate Experimental replicate, A to C

Conc Concentration in arbitrary units; 0 is the control

Affected Number of organisms affected in the group

Total Number of organisms exposed in the group, 20 throughout

Prop Affected fraction, ‘Affected / Total‘

Details

These data are simulated, not measured. Counts are computed deterministically from known log-logistic curves rather than sampled at random, so the dataset is identical across R versions and the generating values can be recovered exactly. Replicate variation is a fixed offset of 0, +1 and -1 affected organisms. The generating script is in ‘data-raw/acutetox.R‘.

Three substances are included, each exercising a different path through the ‘"quantal"‘ preset:

‘Graded‘ (Test_Number 101) A clean concentration-response with no control mortality, generated with an ED50 of 10 and a slope of 0.9. Pooled response runs 0, 0.10, 0.25, 0.50, 0.75, 0.90, so three concentrations sit inside the partial-effect band and a model fits normally. LC50 is 10 by construction.

‘Steep‘ (Test_Number 102) No partial responses anywhere: nothing affected at or below 10, everything affected at 32 and above. The slope is unidentifiable, so no dose-response model can be fitted and [interpolateECx()] applies instead. The expected estimate is the geometric mean of the bracketing concentrations, $\sqrt{10 \times 32} = 17.889$.

‘Background‘ (Test_Number 103) As ‘Graded‘, but with 10 percent control mortality, so the lower limit is not 0 and ‘LL.3u‘ has a reason to be preferred over ‘LL.2‘. The absolute 50 percent point is 7.804 rather than 10, because the ED50 parameter marks the midpoint between the curve’s limits, which is no longer the midpoint between 0 and 1 once the lower limit is raised.

Source

Simulated for this package. See ‘data-raw/acutetox.R‘.

See Also

[cellglow] and [toxresult] for continuous endpoints, [toxdrc_preset()] for the matching configuration.

Examples

```
# The proportion column with a group size, which is the default path.
runtoxdrc(
  dataset  = acutetox,
  Conc     = Conc,
  Response = Prop,
  N        = Total,
  IDcols   = c("Test_Number", "Substance"),
  preset   = "quantal",
  quiet    = TRUE
)

# The same data supplied as counts instead.
```

```
runtoxdrc(
  dataset  = acutetox,
  Conc     = Conc,
  Response = Affected,
  N        = Total,
  IDcols   = c("Test_Number", "Substance"),
  preset   = "quantal",
  endpoint = toxdrc_endpoint(type = "binomial", response.type = "count"),
  quiet    = TRUE
)
```

`averageresponse` *Average response variable*

Description

'averageresponse()' averages a given response variable by the experimental group, such as concentration or exposure length.

Usage

```
averageresponse(
  dataset,
  Conc,
  Response,
  N = NULL,
  IDcols = NULL,
  type = c("continuous", "binomial"),
  list_obj = NULL,
  quiet = FALSE
)
```

Arguments

`dataset`	A dataframe, containing the columns 'Conc' and 'Response'.
`Conc`	Bare (unquoted) column name in 'dataset' that groups the 'Response' variable.
`Response`	Bare (unquoted) column name in 'dataset' containing the response variable. For a quantal endpoint this is the affected fraction, between 0 and 1.
`N`	Bare (unquoted) column name in 'dataset' giving the number of organisms in each group. Required when 'type = "binomial"' and ignored otherwise.
`IDcols`	Optional. Character vector of columns used to identify the data. These columns are preserved in the collapsed 'dataset', taking the first non-blank value in each group, so their values should be constant within a 'Conc' group. Naming a column that does not exist is an error.

type	Character. ‘"continuous"‘ averages the response within each group. ‘"binomial"‘ pools it instead, weighting each replicate by ‘N‘ and returning the combined group size, so that a proportion from 30 of 60 counts for more than one from 1 of 4.
list_obj	Optional. List object used for integration with [runtoxdrc()].
quiet	Logical. Indicates if results should be hidden. Defaults to FALSE.

Value

A collapsed ‘dataset‘ with one row for each level of ‘Conc‘. If ‘list_obj‘ is provided, returns this within a list as ‘list_obj$dataset‘, along with an unmodified copy as ‘list_obj$pre_average_dataset‘.

Examples

```
averageresponse(
  dataset = toxresult,
  Conc = Conc,
  Response = RFU,
  IDcols = c("TestID", "Test_Number", "Dye", "Type", "Replicate")
)
```

---

blankcorrect *Blank correct response variable*

Description

‘blankcorrect()‘ subtracts a calculated correction value from all responses.

Usage

```
blankcorrect(
  dataset,
  Conc,
  blank_group = "Blank",
  Response,
  list_obj = NULL,
  quiet = FALSE
)
```

Arguments

dataset	A dataframe, containing the columns ‘Conc‘ and ‘Response‘.
Conc	Bare (unquoted) column name in ‘dataset‘ that groups the ‘Response‘ variable.
blank_group	Character. Name of the ‘Conc‘ level to calculate the correction value from.
Response	Bare (unquoted) column name in ‘dataset‘ containing the response variable.
list_obj	Optional. List object used for integration with [runtoxdrc()].
quiet	Logical. Indicates if results should be hidden. Defaults to FALSE.

Value

A modified ‘dataset‘ with an additional column, ‘c_response‘. If ‘list_obj‘ is provided, returns this within a list as ‘list_obj$dataset‘, along with statistics of the correction value as ‘list_obj$blank_stats‘.

Examples

```
blankcorrect(
     dataset = toxresult,
     Conc = Conc,
     blank_group = "Blank",
     Response = RFU
   )
```

`cellglow` *Example toxicity test data with multiple experimental groups.*

Description

A subset of data from a study using the RTgill-W1 assay (ISO 21115/OECD 249). Briefly, cells are exposed to a toxicant and the fluorescent signal is measured using 3 indicators.

Usage

```
cellglow
```

Format

‘cellglow‘ A data frame with 1,080 rows and 7 columns:

TestID Combination of Test_Number, Dye, Type, and Replicate

Test_Number Identifying number of each effluent sample

Conc Concentration of reference toxicant (3,4 dichloranaline). 0 is solvent control, "control" is a lab control

RFU Fluoresence produced as determined by a plate reader

Dye Three cell viability indicators; aB = alamarBlue, CFDA = 5-CFDA-AM, NR = Neutral Red

Type Only spiked exists in this dataset; indicates a reference toxicant was added to the effluent.

Replicate The experimental replicate; replication occured at a well-plate level.

Details

Data collected as part of a study. Full dataset is available within a data repository: Salole, Jack; Wilson, Joanna; Taylor, Lisa, 2025, "RTgill-W1 Assay - Optimization and Effluent Testing", https://doi.org/10.5683/SP3/ES7GD Borealis, V2.

Source

https://doi.org/10.5683/SP3/ES7GDM

checktoxicity *Check for an effect*

Description

'checktoxicity()' flags if the response variable exceeds a limit in either direction as evidence of an effect.

Usage

```
checktoxicity(
  dataset,
  Conc,
  Response,
  effect,
  type = c("relative", "absolute"),
  direction = c("below", "above"),
  reference_group = "0",
  target_group = NULL,
  list_obj = NULL,
  quiet = FALSE
)
```

Arguments

dataset	A dataframe, containing the columns 'Conc' and 'Response'.
Conc	Bare (unquoted) column name in 'dataset' that groups the 'Response' variable.
Response	Bare (unquoted) column name in 'dataset' containing the response variable.
effect	Numeric. Dictates at the value beyond which observations are flagged as toxic. This value can be further customized; see see 'type' and 'direction'.
type	Character. Indicates if 'effect' is '"relative"' to 'reference group' or an '"absolute"' value. Defaults to relative.
direction	Character. Indicates if an effect occurs '"below"' or '"above"'. Defaults to below.
reference_group	Label used for reference group in 'Conc' column. Defaults to 0.
target_group	Optional. Limits the comparison to certain levels in 'Conc'.
list_obj	Optional. List object used for integration with [runtoxdrc()].
quiet	Logical. Indicates if results should be hidden. Defaults to FALSE.

Value

TRUE if the response variable exceeds a limit in either direction and FALSE otherwise. If 'list_obj' is provided, returns this within a list as 'list_obj$effect'.

Examples

```
checktoxicity(
 dataset = toxresult,
 Conc = Conc,
 Response = RFU,
 effect = 0.5
)
```

`config_runtoxdrc` *Configuration functions for the runtoxdrc pipeline*

Description

An overview of the modular configuration functions used by [runtoxdrc()]. These configuration functions provide default lists of parameters for customizing different stages of the pipeline to reduce the number of arguments required in [runtoxdrc()]. Each function returns a named list of configuration parameters suitable for passing directly to [runtoxdrc()].

Details

Available configuration functions:

- [toxdrc_endpoint()] - Continuous or quantal response, and how it is supplied
- [toxdrc_qc()] - Quality control and filtering options
- [toxdrc_normalization()] - Blank correction and normalization
- [toxdrc_toxicity()] - Toxicity threshold and response-level options
- [toxdrc_modelling()] - Model selection, fitting criteria, and EDx calculation
- [toxdrc_output()] - Output settings

See Also

[runtoxdrc()], [toxdrc_endpoint()], [toxdrc_qc()], [toxdrc_normalization()], [toxdrc_toxicity()], [toxdrc_modelling()], [toxdrc_output()]

flagCV *Check for groups with high CV*

Description

This function calculates the coefficient of variation (CV) of each of the exposure conditions, and flags them if they exceed a set value.

Usage

```
flagCV(dataset, Conc, Response, max_val = 30, list_obj = NULL, quiet = FALSE)
```

Arguments

dataset	A dataframe, containing the columns 'Conc' and 'Response'.
Conc	Bare (unquoted) column name in 'dataset' that groups the 'Response' variable.
Response	Bare (unquoted) column name in 'dataset' containing the response variable.
max_val	Numeric. The percent beyond which CV values are flagged. Defaults to 30.
list_obj	Optional. List object used for integration with [runtoxdrc()].
quiet	Logical. Indicates if results should be hidden. Defaults to FALSE.

Value

A modified 'dataset' with an additional column, 'CVflag'. If 'list_obj' is provided, returns this within a list as 'list_obj$dataset', along with a summary of the CV results as 'list_obj$CVresults'.

Examples

```
df <- data.frame(x = rep(1:2, each = 3), y = c(10, 11, 9, 20, 40, 60))
flagCV(dataset = df, Conc = x, Response = y, max_val = 30)
```

getECx *Get point estimates from model*

Description

Generate point estimates from a dose response curve.

Usage

```
getECx(
  dataset,
  model,
  EDx = 0.5,
  interval = c("tfls", "fls", "delta", "none"),
  level = 0.95,
  type = c("relative", "absolute"),
  quiet = FALSE,
  EDargs.supplement = list(),
  list_obj = NULL
)
```

Arguments

dataset	A dataframe used to generate ‘model‘.
model	A drm model, generated by ‘modelcomp()‘ or ‘drm()‘.
EDx	Numeric. The effect level(s) to estimate, given as a proportion between 0 and 1, so ‘0.5‘ requests the EC50. Defaults to 0.5. Note that ‘drc::ED()‘ expects a percentage under ‘type = "relative"‘; the conversion is handled here, so pass a proportion regardless of ‘type‘.
interval	Character. Method for calculating confidence intervals of EDx. Choices: ‘"tfls"‘, ‘"fls"‘, ‘"delta"‘, ‘"none"‘. Defaults to "tfls". See ‘drc::ED()‘ for more information.
level	Numeric. Confidence level for the interval calculation. Defaults to 0.95.
type	Character. ‘"relative"‘ expresses ‘EDx‘ as a fraction of the range between the fitted limits, so ‘EDx = 0.5‘ is the concentration halfway down the curve. ‘"absolute"‘ treats ‘EDx‘ as a value on the response scale itself, which is only meaningful when the response is on a bounded scale such as a proportion. Defaults to relative, matching [toxdrc_modelling()].
quiet	Logical. Indicates if results should be hidden. Defaults to FALSE.
EDargs.supplement	List. Optional user-supplied list of additional arguments compatible with ‘drc::ED()‘.
list_obj	Optional. List object used for integration with [runtoxdrc()].

Value

A dataframe of the point estimates. If ‘list_obj‘ is provided, returns this within a list as ‘list_obj$effectmeasure‘.

See Also

[modelcomp()] to fit the model, and [drc::ED()] for the underlying estimation.

getmetadata *Generate metadata from a dataframe*

Description

Collects identifying or important values from an experimental replicate.

Usage

```
getmetadata(dataset, IDcols = NULL, list_obj = NULL, quiet = FALSE)
```

Arguments

dataset	A dataframe.
IDcols	Optional. Character. Columns given as a vector used in the identification of data. These columns are preserved in the modified 'dataset' with the first non-blank value. These values should be identical within observations grouped by 'Conc'.
list_obj	Optional. List object used for integration with [runtoxdrc()].
quiet	Logical. Indicates if results should be hidden. Defaults to FALSE.

Value

A 1 row dataframe of the identifying parameters of an experimental replicate. If 'list_obj' is provided, returns this within a list as 'list_obj$metadata'.

interpolateECx *Estimate a point estimate by log-linear interpolation*

Description

'interpolateECx()' estimates an effect concentration by interpolating between the observed concentrations that bracket the target response, linearly on log concentration. It is intended for quantal data where no concentration produced a partial effect, so no dose-response model can be fitted.

Usage

```
interpolateECx(
  dataset,
  Conc,
  Response,
  EDx = 0.5,
  type = c("relative", "absolute"),
  partial.tol = 0.2,
  list_obj = NULL,
  quiet = FALSE
)
```

Arguments

`dataset`	A dataframe, containing the columns ‘Conc‘ and ‘Response‘.
`Conc`	Bare (unquoted) column name in ‘dataset‘ giving concentration. Non-numeric and non-positive levels are dropped, since the interpolation works on log concentration. Control rows are still used to establish the background response.
`Response`	Bare (unquoted) column name in ‘dataset‘ giving the affected fraction, between 0 and 1.
`EDx`	Numeric. The effect level(s) to estimate. Defaults to 0.5.
`type`	Character. ‘"relative"‘ expresses ‘EDx‘ relative to the background response, targeting ‘control + EDx * (1 - control)‘, which is the interpolation equivalent of an Abbott correction and matches [getECx()]. ‘"absolute"‘ treats ‘EDx‘ as the target proportion directly, so ‘EDx = 0.5‘ finds the concentration affecting half the organisms regardless of control response. Defaults to relative. With no control response the two coincide.
`partial.tol`	Numeric. How far a proportion must sit from both the control response and complete effect to count as a partial effect, used only to decide whether to warn. Defaults to 0.2.
`list_obj`	Optional. List object used for integration with [runtoxdrc()].
`quiet`	Logical. Indicates if results should be hidden. Defaults to FALSE.

Details

Interpolating linearly on log concentration is the standard treatment of a test with no partial responses. Between the highest concentration with no effect, ‘A‘, and the lowest with complete effect, ‘B‘, the estimate is

$$EC_p = A \times (B/A)^p$$

so the EC50 is $\sqrt{A \times B}$, the geometric mean of the two. That value is the geometric midpoint and does not depend on the shape of the underlying curve, which is what makes it defensible when the shape is unknown.

The same is not true of other effect levels. An EC10 estimated this way lands just above ‘A‘, and an EC90 just below ‘B‘, so both largely restate the bracketing concentrations rather than estimating anything from them. Where they fall between ‘A‘ and ‘B‘ is a consequence of the straight-line assumption, which is precisely what the data cannot support. Requesting a level other than 0.5 when no partial responses exist therefore raises a warning.

Interpolation never extrapolates. If the target response is not bracketed by two observed concentrations, the estimate is NA.

Because the estimate comes from two points rather than a fitted model, there is no standard error and no confidence interval. Those columns are returned as NA so the output matches [getECx()].

Value

A dataframe of point estimates with the same columns as [getECx()]: ‘Effect Measure‘, ‘Estimate‘, ‘Std. Error‘, ‘Lower‘ and ‘Upper‘, the last three being NA. If ‘list_obj‘ is provided, returns this within a list as ‘list_obj$effectmeasure‘.

See Also

[getECx()] for model-based estimation, and [runtoxdrc()] for using this automatically when a quantal test has no partial responses.

Examples

```
# No partial responses: nothing affected at 10, everything at 30.
quantal <- data.frame(
  Conc = c(0, 10, 30, 100),
  Prop = c(0, 0, 1, 1)
)
interpolateECx(quantal, Conc = Conc, Response = Prop, quiet = TRUE)
# EC50 is sqrt(10 * 30) = 17.32
```

---

modelcomp *Compare model fits and select best model*

Description

Data is fitted to provided models, typically from the drc package. Models fitted successfully are compared using multiple goodness-of-fit scores, and organized using the score given as the ‘metric‘ argument.

For a continuous response the default ‘model_list‘ holds a single model, so by default this fits ‘LL.4‘ rather than choosing between candidates. Supply more than one entry to make ‘metric‘ meaningful. For a quantal endpoint the default holds two, ‘LL.2‘ and ‘LL.3u‘, which differ in whether background response in the controls is estimated.

Usage

```
modelcomp(
  dataset,
  Conc,
  Response,
  N = NULL,
  model_list = NULL,
  metric = c("IC", "Res var", "Lack of fit"),
  type = c("continuous", "binomial"),
  list_obj = NULL,
  quiet = FALSE
)
```

Arguments

dataset	A dataframe, containing the columns ‘Conc‘ and ‘Response‘.
Conc	Bare (unquoted) column name in ‘dataset‘ that groups the ‘Response‘ variable.

`Response`	Bare (unquoted) column name in ‘dataset‘ containing the response variable. For a quantal endpoint this is the affected fraction, between 0 and 1.
`N`	Bare (unquoted) column name in ‘dataset‘ giving the number of organisms in each group. Required when ‘type = "binomial"‘ and ignored otherwise.
`model_list`	Named list, or NULL. Model functions to be tested. NULL, the default, selects a list appropriate to ‘type‘: ‘LL.4‘ for a continuous response, and ‘LL.2‘ together with ‘LL.3u‘ for a quantal one. Supply your own list to override, e.g. ‘list("LL.4" = LL.4(), "W1.4" = W1.4())‘. Most models from the drc package are compatible; use ‘drc::getMeanFunctions()‘ for more options. Each entry must be named for the model function it holds. See details for formatting.
`metric`	Character. Criterion used to select the best model. Choices are ‘"IC"‘, ‘"Res var"‘, ‘"Lack of fit"‘. Defaults to "IC". See details for how each is ordered. ‘"Res var"‘ is not available when ‘type = "binomial"‘.
`type`	Character. ‘"continuous"‘ for a measured response, or ‘"binomial"‘ for a quantal endpoint, which is fitted with ‘drc::drm(type = "binomial")‘ and weighted by ‘N‘.
`list_obj`	Optional. List object used for integration with [runtoxdrc()].
`quiet`	Logical. Indicates if results should be hidden. Defaults to FALSE.

Details

The ‘model_list‘ argument requires a specific style. The argument must be a named list, where each name is the shorthand of the model function held in that entry, and each entry is the *result* of calling that function. An example of this is ‘list("LL.4" = LL.4(), "W1.4" = W1.4())‘.

The names are not cosmetic: they are used to label the comparison table and to retrieve the winning model, so a name that disagrees with the function it holds is rejected rather than silently ignored.

Models that fail to converge are retained in the comparison table as ‘NA‘ rows and are never selected as the best model.

Ordering of ‘metric‘. ‘"IC"‘ and ‘"Res var"‘ are ordered ascending, since a smaller value indicates the better model. ‘"Lack of fit"‘ is ordered descending, because it is a goodness-of-fit p-value where a larger value means less evidence against the model. Note that this differs from [drc::mselect()], which sorts every criterion ascending.

Value

A fitted drm model. If ‘list_obj‘ is provided, returns this within the list as ‘list_obj$best_model‘, along with the model name (‘list_obj$best_model_name‘), and the model comparison dataframe (‘list_obj$model_df‘). If model fitting fails, returns NULL.

See Also

[drc::getMeanFunctions()] for compatible models and their shorthand for ‘model_list‘.

Examples

```
toxresult2 <- toxresult[!toxresult$Conc %in% c ("Control", "Blank"),]
toxresult2$Conc <- as.numeric(toxresult2$Conc)
modelcomp(toxresult2, Conc, RFU, metric = "IC")
```

`normalizeresponse` *Normalize response variable*

Description

Express a response variable relative to a reference group.

Usage

```
normalizeresponse(
  dataset,
  Conc,
  reference_group = "0",
  Response,
  list_obj = NULL,
  quiet = FALSE
)
```

Arguments

`dataset`	A dataframe, containing the columns 'Conc' and 'Response'.
`Conc`	Bare (unquoted) column name in 'dataset' that groups the 'Response' variable.
`reference_group`	Label used for the group values will be normalized to. Defaults to 0.
`Response`	Bare (unquoted) column name in 'dataset' containing the response variable.
`list_obj`	Optional. List object used for integration with [runtoxdrc()].
`quiet`	Logical. Indicates if results should be hidden. Defaults to FALSE.

Value

A modified 'dataset' with an additional column, 'normalized response'. If 'list_obj' is provided, returns this within a list as 'list_obj$dataset', along with summary statistics surrounding the reference group as 'list_obj$normalize_response_summary'.

Examples

```
normalizeresponse(
 dataset = toxresult,
 Conc = Conc,
 Response = RFU
)
```

pctl *Check for positive control effect*

Description

This function evaluates the difference between a two groups to determine if the difference between them exceeds a set amount. Commonly used to determine if a solvent introduces effects.

Usage

```
pctl(
  dataset,
  Conc,
  reference_group = "Control",
  positive_group = 0,
  Response,
  max_diff = 10,
  list_obj = NULL,
  quiet = FALSE
)
```

Arguments

dataset	A dataframe, containing the columns 'Conc' and 'Response'.
Conc	Bare (unquoted) column name in 'dataset' that groups the 'Response' variable.
reference_group	Label used for the true control level. Defaults to "Control".
positive_group	Label used for the positive control level. Defaults to 0.
Response	Bare (unquoted) column name in 'dataset' containing the response variable.
max_diff	Numeric. Percent difference of the response in the 'ref.label' and 'pctl.label' groups beyond which tests are flagged. Defaults to 10.
list_obj	Optional. List object used for integration with [runtoxdrc()].
quiet	Logical. Indicates if results should be hidden. Defaults to FALSE.

Value

A modified 'dataset' with an additional column, 'Validity'. If 'list_obj' is provided, returns this within a list as 'list_obj$dataset', along with statistics of the positive and reference group as 'list_obj$pctlresults'.

Examples

```
pctl(
 dataset = toxresult,
 Conc = Conc,
 Response = RFU,
 reference_group = "Control",
```

```
 positive_group = "0"
)
```

`print.toxdrc_preset` *Print a toxdrc preset*

Description

Print a toxdrc preset

Usage

```
## S3 method for class 'toxdrc_preset'
print(x, ...)
```

Arguments

`x`	A ‘toxdrc_preset‘.
`...`	Ignored.

Value

‘x‘, invisibly.

`removeoutliers` *Remove outliers iteratively using Grubbs' test*

Description

Removes statistical outliers from each group of ‘Conc‘ by applying Grubbs' test repeatedly, dropping the most extreme value each time the test is significant at $p < 0.05$. Groups of two or fewer observations are left untouched, since Grubbs' test needs at least three.

Usage

```
removeoutliers(dataset, Conc, Response, list_obj = NULL, quiet = FALSE)
```

Arguments

`dataset`	A dataframe, containing the columns ‘Conc‘ and ‘Response‘.
`Conc`	Bare (unquoted) column name in ‘dataset‘ that groups the ‘Response‘ variable.
`Response`	Bare (unquoted) column name in ‘dataset‘ containing the response variable.
`list_obj`	Optional. List object used for integration with [runtoxdrc()].
`quiet`	Logical. Indicates if results should be hidden. Defaults to FALSE.

Value

A modified ‘dataset‘ with outliers removed. If ‘list_obj‘ is provided, returns this within a list as ‘list_obj$dataset‘, along with dataframe of removed outliers as ‘list_obj$removed_outliers‘.

Examples

```
df <- data.frame(x = rep(1:2, each = 3),y = c(3, 5, 7, 3, 4, 30))
removeoutliers(dataset = df, Conc = x, Response = y)
```

---

runtoxdrc *Point estimation pipeline*

Description

‘runtoxdrc()‘ is the pipeline for function in the toxdrc package. This function allows the automated analysis of large datasets, while maintaining a consistent process for each subset of data.

Usage

```
runtoxdrc(
  dataset,
  Conc,
  Response,
  N = NULL,
  IDcols = NULL,
  quiet = FALSE,
  preset = NULL,
  endpoint = NULL,
  qc = NULL,
  normalization = NULL,
  toxicity = NULL,
  modelling = NULL,
  output = NULL
)
```

Arguments

dataset	A dataframe, containing the columns ‘Conc‘ and ‘Response‘.
Conc	Bare (unquoted) column name in ‘dataset‘ that groups the ‘Response‘ variable.
Response	Bare (unquoted) column name in ‘dataset‘ containing the response variable. For a quantal endpoint this is the affected fraction, or the number affected if ‘endpoint = toxdrc_endpoint(response.type = "count")‘.
N	Bare (unquoted) column name in ‘dataset‘ giving the number of organisms in each group. Required for a quantal endpoint and ignored otherwise.

IDcols	Optional. Character. Columns given as a vector used in the identification of data. These columns are preserved in the modified ‘dataset‘ with the first non-blank value. These values should be identical within observations grouped by ‘Conc‘.
quiet	Logical. Indicates if results should be hidden. Defaults to FALSE.
preset	A configuration set from [toxdrc_preset()], or the name of one such as ‘"quantal"‘. Supplies defaults for every configuration block left unset below. Defaults to ‘"continuous"‘, which is where this function’s own defaults come from.
endpoint	Endpoint options, declaring whether the response is continuous or quantal, overriding the preset. See [toxdrc_endpoint()] for more detail and defaults.
qc	Quality control and filtering options, overriding the preset. See [toxdrc_qc()] for more detail and defaults.
normalization	Normalization options, overriding the preset. See [toxdrc_normalization()] for more detail and defaults.
toxicity	Toxicity threshold and response-level options, overriding the preset. See [toxdrc_toxicity()] for more detail and defaults.
modelling	Model selection, fitting criteria, and EDx calculation options, overriding the preset. See [toxdrc_modelling()] for more detail and defaults.
output	Settings for output, overriding the preset. See [toxdrc_output()] for more detail and defaults.

Value

By default, returns a list of lists with each subset of data having its own entry. Each subset contains dataframes, models, and other objects that track the pipeline process. If ‘output = list(condense = TRUE)‘, the results are summarized into a single dataframe containing the ‘IDcols‘ and model information of each data subset.

Quantal endpoints

Setting ‘endpoint = toxdrc_endpoint(type = "binomial")‘ fits models with ‘drc::drm(type = "binomial")‘, weighted by ‘N‘, and defaults to comparing ‘LL.2‘ against ‘LL.3u‘. Replicates are pooled by group size rather than averaged.

Three preprocessing steps are refused for quantal data, because they assume a continuous response on an unbounded scale: ‘blank.correction‘, ‘normalize.resp‘, and ‘outlier.test‘. Each raises an error naming the alternative. ‘"Res var"‘ is likewise unavailable as ‘model.metric‘, since a binomial fit has no residual variance.

Setting ‘modelling = toxdrc_modelling(interpolate = TRUE)‘ handles the degenerate case where no concentration produced a partial effect. There the slope is unidentifiable, so no model can be fitted at all, and the estimate comes from log-linear interpolation between the bracketing concentrations instead. Such entries are marked with ‘best_model_name = "interpolated"‘ and carry no confidence interval. See [interpolateECx()].

See Also

[toxdrc_preset()] for ready-made configurations, and [config_runtoxdrc()] for the individual settings they are built from.

Examples

```
 analyzed_data <- runtoxdrc(
dataset = cellglow,
Conc = Conc,
Response = RFU,
IDcols = c("Test_Number", "Dye", "Replicate", "Type"),
quiet = TRUE,
normalization = toxdrc_normalization(
  blank.correction = TRUE,
  normalize.resp = TRUE
),
modelling = toxdrc_modelling(EDx = c(0.2, 0.5, 0.7))
)
```

`toxdrc_endpoint` *Endpoint configuration for the runtoxdrc pipeline.*

Description

Declares whether the response is a continuous measurement or a quantal (binary) endpoint such as mortality, and how a quantal response is supplied.

Usage

```
toxdrc_endpoint(
  type = c("continuous", "binomial"),
  response.type = c("proportion", "count")
)
```

Arguments

`type`	Character. ‘"continuous"‘ for a measured response, or ‘"binomial"‘ for a quantal endpoint. Defaults to continuous.
`response.type`	Character. How a quantal response is given. ‘"proportion"‘ (the default) treats ‘Response‘ as the affected fraction, between 0 and 1. ‘"count"‘ treats ‘Response‘ as the number affected, and the proportion is derived as ‘Response / N‘. Ignored when ‘type = "continuous"‘.

Details

A quantal endpoint is the affected fraction of a group of organisms, and fitting it correctly requires knowing how many organisms were in each group. A response of 0.1 from 1 of 10 and from 10 of 100 are the same number but carry very different weight, so ‘runtoxdrc()‘ requires the ‘N‘ argument when ‘type = "binomial"‘. Models are then fitted with ‘drc::drm(type = "binomial")‘ and weighted by group size.

Several preprocessing steps assume a continuous response and are refused for quantal data: blank correction, normalization to a reference group, and Grubbs' outlier removal. See [runtoxdrc()] for what to use instead.

Value

A named list containing the endpoint configuration for use in [runtoxdrc()].

See Also

[config_runtoxdrc], [runtoxdrc()], [modelcomp()]

Examples

```
toxdrc_endpoint()
toxdrc_endpoint(type = "binomial")
toxdrc_endpoint(type = "binomial", response.type = "count")
```

---

toxdrc_modelling *Modelling configuration for the runtoxdrc pipeline.*

Description

Defines how dose-response models are fitted, selected, and how point estimates (EDx) are calculated.

Usage

```
toxdrc_modelling(
  model.list = NULL,
  model.metric = c("IC", "Res var", "Lack of fit"),
  EDx = 0.5,
  interval = c("tfls", "fls", "delta", "none"),
  level = 0.95,
  type = c("relative", "absolute"),
  quiet = FALSE,
  EDargs.supplement = list(),
  interpolate = FALSE,
  partial.tol = 0.2
)
```

Arguments

`model.list`	Named list, or NULL. Model functions to be tested. NULL, the default, selects a list appropriate to the endpoint: 'LL.4' for a continuous response, and 'LL.2' together with 'LL.3u' for a quantal one. Supply your own list to override, e.g. 'list("LL.4" = LL.4(), "W1.4" = W1.4())'; with a single entry no comparison

takes place and ‘model.metric‘ has no effect. Most models from the drc package are compatible; use ‘drc::getMeanFunctions()‘ for more options. Each entry must be named for the model function it holds. See [modelcomp()] for more information around formatting.

`model.metric` Character. Criterion used to select the best model. Choices are ‘"IC"‘, ‘"Res var"‘, ‘"Lack of fit"‘. Defaults to "IC". ‘"IC"‘ and ‘"Res var"‘ select the smallest value; ‘"Lack of fit"‘ selects the largest, since it is a goodness-of-fit p-value. See [modelcomp()] for details.

`EDx` Numeric. The effect level(s) to estimate, given as a proportion between 0 and 1, so ‘0.5‘ requests the EC50 and ‘c(0.1, 0.5)‘ requests the EC10 and EC50. Defaults to 0.5.

`interval` Character. Method for calculating confidence intervals of EDx. Choices: ‘"tfls"‘, ‘"fls"‘, ‘"delta"‘, ‘"none"‘. Defaults to "tfls". See ‘drc::ED()‘ for more information.

`level` Numeric. Confidence level for the interval calculation. Defaults to 0.95.

`type` Character. Indicates if EDx is ‘"absolute"‘ or ‘"relative"‘ to the curve. Defaults to absolute.

`quiet` Logical. Indicates if results should be hidden. Defaults to FALSE.

`EDargs.supplement` List. Optional user-supplied list of additional arguments compatible with ‘drc::ED()‘.

`interpolate` Logical. For a quantal endpoint, estimate the effect concentration by log-linear interpolation when no concentration produced a partial effect and therefore no model can be fitted. Defaults to FALSE. Has no effect on a continuous endpoint. See [interpolateECx()].

`partial.tol` Numeric. How far an affected proportion must sit from both the control response and complete effect to count as a partial effect. Defaults to 0.2, so with no control response a group must be between 20% and 80% affected. Only used when ‘interpolate = TRUE‘.

Value

A named list containing model fitting and selection settings for use in [runtoxdrc()].

See Also

[config_runtoxdrc], [runtoxdrc()], [drc::ED()], [drc::getMeanFunctions()], [modelcomp()]

Examples

```
toxdrc_modelling(EDargs.supplement = list(interval = "delta", level = 0.9))
```

`toxdrc_normalization` *Set normalization configuration for the runtoxdrc pipeline.*

Description

Control blank correction and normalization of the response variable.

Usage

```
toxdrc_normalization(
  blank.correction = FALSE,
  blank.label = "Blank",
  normalize.resp = FALSE,
  relative.label = 0
)
```

Arguments

`blank.correction`	Logical. Indicates if the response variable should be blank corrected. Defaults to FALSE.
`blank.label`	Character. Label used for the blank level. Defaults to "Blank".
`normalize.resp`	Logical. Indicates if response variable should be normalized to a given group. Defaults to FALSE.
`relative.label`	Label used for the group values will be normalized to. Defaults to 0.

Value

A named list containing normalization configuration for use in [runtoxdrc()].

See Also

[config_runtoxdrc], [runtoxdrc()], [blankcorrect()], [normalizeresponse()]

Examples

```
toxdrc_normalization(blank.correction = TRUE, relative.label = "Control")
```

toxdrc_output *Output configuration for the runtoxdrc pipeline.*

Description

Defines how [runtoxdrc()] output is returned.

Usage

```
toxdrc_output(
  condense = FALSE,
  sections = c("ID", "effectmeasure", "best_model_name", "effect")
)
```

Arguments

condense	Logical. Indicates if the results should be summarized into a single dataframe. Defaults to TRUE.
sections	Character. Columns given as a vector that should be present in the summary. Defaults to ‘c("ID", "effectmeasure", "best_model_name", "effect")‘.

Value

A named list containing output configuration for use in [runtoxdrc()].

See Also

[config_runtoxdrc], [runtoxdrc()]

Examples

```
toxdrc_output()
toxdrc_output(condense = TRUE)
```

toxdrc_preset *Ready-made configurations for common study designs*

Description

‘toxdrc_preset()‘ returns a complete set of configuration objects for a kind of study, suitable for passing to [runtoxdrc()] as ‘preset‘. The point is less to save typing than to record which combinations of settings belong together, which is not obvious from the individual configuration helpers.

Usage

```
toxdrc_preset(name = c("continuous", "quantal", "normalized"))
```

Arguments

name	Character. Which preset to return.

Details

A preset is an ordinary named list of the same objects [runtoxdrc()] takes individually, so it can be printed, inspected, and modified. Anything passed explicitly to [runtoxdrc()] overrides the corresponding entry, which makes a preset a starting point rather than a black box.

Available presets:

‘"continuous"‘ A measured response on an arbitrary scale, fitted with ‘LL.4‘. No preprocessing. This is the source of [runtoxdrc()]’s own defaults, so using it changes nothing; it exists so the baseline has a name and cannot drift from the defaults it defines.

‘"quantal"‘ A binary endpoint such as mortality, with few organisms per group. Fitted as binomial data weighted by ‘N‘, comparing ‘LL.2‘ against ‘LL.3u‘ so that background response in the controls is accounted for only if the data call for it. Log-linear interpolation is enabled, which covers the common case of a test with no partial responses, where no model can be fitted at all. Requires the ‘N‘ argument to [runtoxdrc()].

Effect levels are relative, which for a quantal fit is algebraically Abbott’s correction: the EC50 is the concentration affecting half of the organisms that would otherwise have survived, rather than half of all organisms. The two are identical unless there is control mortality, and so differ only when ‘LL.3u‘ is selected. Pass ‘modelling = toxdrc_modelling(type = "absolute")‘ to report raw response levels instead.

‘"normalized"‘ A continuous response in arbitrary units, blank corrected and expressed relative to a control group. Because normalization puts the control at 1 and complete effect at 0 by construction, the toxicity threshold is absolute rather than relative, and the models fix those bounds instead of estimating them: ‘LL.2‘ fixes both, and ‘LL.3‘ estimates the upper limit for data where responses run above 1 through noise or stimulation.

Presets set ‘blank.label‘ and ‘relative.label‘ to their defaults, since they cannot know how your groups are labelled. Override the whole ‘normalization‘ block if yours differ; see examples.

Value

A named list of configuration objects, with class ‘toxdrc_preset‘, containing ‘endpoint‘, ‘qc‘, ‘normalization‘, ‘toxicity‘, ‘modelling‘, and ‘output‘.

See Also

[runtoxdrc()], [config_runtoxdrc]

Examples

```
toxdrc_preset("quantal")

# A preset is a starting point. To keep its settings but change the group
# labels, pass a replacement for that block only.

runtoxdrc(
```

```
  dataset       = cellglow,
  Conc          = Conc,
  Response      = RFU,
  IDcols        = c("Test_Number", "Dye", "Replicate", "Type"),
  quiet         = TRUE,
  preset        = toxdrc_preset("normalized"),
  normalization = toxdrc_normalization(
    blank.correction = TRUE,
    normalize.resp   = TRUE,
    blank.label      = "Blank",
    relative.label   = 0
  )
)
```

`toxdrc_qc` *Set quality control options for the runtoxdrc pipeline.*

Description

Control outlier detection, CV calculation, averaging of response variable, and testing for positive control effects.

Usage

```
toxdrc_qc(
  outlier.test = FALSE,
  cv.flag = TRUE,
  cvflag.lvl = 30,
  pctl.test = FALSE,
  pctl.lvl = 10,
  ref.label = "Control",
  pctl.label = 0,
  avg.resp = TRUE
)
```

Arguments

`outlier.test`	Logical. Indicates if outliers should be tested for and removed. Defaults to FALSE.
`cv.flag`	Logical. Indicates if groups of the response variable should be flagged if the CV exceeds 'cvflag.lvl'. Defaults to TRUE.
`cvflag.lvl`	Numeric. The percent beyond which CV values are flagged. Defaults to 30.
`pctl.test`	Logical. Indicates if positive control/solvent effects should be tested for. Defaults to FALSE.
`pctl.lvl`	Numeric. Percent difference of the response in the 'ref.label' and 'pctl.label' groups beyond which tests are flagged. Defaults to 10.

ref.label	Label used for the true control level. Defaults to "Control".
pctl.label	Label used for the positive control level. Defaults to 0.
avg.resp	Logical. Indicates if responses should be averaged within each group. Defaults to TRUE.

Value

A named list containing the quality control configuration for use in [runtoxdrc()].

See Also

[config_runtoxdrc], [runtoxdrc()], [pctl()], [removeoutliers()], [flagCV()]

Examples

```
toxdrc_qc(outlier.test = TRUE, cvflag.lvl = 20)
```

---

toxdrc_toxicity *Toxicity configuration for the runtoxdrc pipeline.*

Description

Defines how toxicity is determined for model fitting.

Usage

```
toxdrc_toxicity(
  toxic.lvl = 0.7,
  toxic.type = c("relative", "absolute"),
  toxic.direction = c("below", "above"),
  comp.group = 0,
  target.group = NULL
)
```

Arguments

toxic.lvl	Numeric. Cutoff point to determine if modelling occurs. Defaults to 0.7.
toxic.type	Character. Indicates if ‘effect‘ is ‘"relative"‘ to ‘reference group‘ or an ‘"absolute"‘ value. Defaults to relative.
toxic.direction	Character. Indicates if an effect occurs ‘"below"‘ or ‘"above"‘. Defaults to below.
comp.group	Label used for reference group.
target.group	Optional. Limits the comparison to certain exposure conditions.

Value

A named list containing toxicity determination settings for use in [runtoxdrc()].

See Also

[config_runtoxdrc], [runtoxdrc()], [checktoxicity()]

Examples

```
toxdrc_toxicity(toxic.lvl = 0.5, toxic.direction = "above")
```

---

toxresult *Example toxicity test data from a single experimental subset.*

Description

A subset of data from a study using the RTgill-W1 assay (ISO 21115/OECD 249). Briefly, cells are exposed to a toxicant and the fluorescent signal is measured using 3 indicators.

Usage

```
toxresult
```

Format

'toxresult' A data frame with 1,080 rows and 7 columns:

TestID Combination of Test_Number, Dye, Type, and Replicate

Test_Number Identifying number of each effluent sample

Conc Concentration of reference toxicant (3,4 dichloranaline). 0 is solvent control, "control" is a lab control

RFU Fluoresence produced as determined by a plate reader

Dye Three cell viability indicators; aB = alamarBlue, CFDA = 5-CFDA-AM, NR = Neutral Red

Type Only spiked exists in this dataset; indicated a reference toxicant was added to the effluent.

Replicate The experimental replicate; replication occured at a well-plate level.

Details

Data collected as part of a study. Full dataset is available within a data repository: Salole, Jack; Wilson, Joanna; Taylor, Lisa, 2025, "RTgill-W1 Assay - Optimization and Effluent Testing", https://doi.org/10.5683/SP3/ES7GD Borealis, V2.

Source

https://doi.org/10.5683/SP3/ES7GDM

Index

∗ **datasets**
acutetox, 2
cellglow, 6
toxresult, 28

acutetox, 2
averageresponse, 4

blankcorrect, 5

cellglow, 6
checktoxicity, 7
config_runtoxdrc, 8

flagCV, 9

getECx, 9
getmetadata, 11

interpolateECx, 11

modelcomp, 13

normalizeresponse, 15

pctl, 16
print.toxdrc_preset, 17

removeoutliers, 17
runtoxdrc, 18

toxdrc_endpoint, 20
toxdrc_modelling, 21
toxdrc_normalization, 23
toxdrc_output, 24
toxdrc_preset, 24
toxdrc_qc, 26
toxdrc_toxicity, 27
toxresult, 28