

Package: clinstats (via r-universe)

October 16, 2026

Title Clinical Research Statistics Toolkit

Version 0.1.2

Description Statistical utilities that recur in clinical and translational research workflows: receiver operating characteristic (ROC) summaries, odds ratios, univariable and multivariable Cox and logistic regression tables, time-dependent survival cut points, resampling-based Cox screening, TCGA clinical extraction, and expression-matrix cleaning. Includes curated colorectal cancer clinical datasets assembled from public repositories, a Quarto report template, and tidy data-frame output that is convenient for both interactive and programmatic use.

License Apache License (>= 2)

URL <https://github.com/gflab/clinstats>

BugReports <https://github.com/gflab/clinstats/issues>

Depends R (>= 4.1)

Imports cli, future.apply, pROC, survival, tibble, withr

Suggests AnnotationDbi, knitr, org.Hs.eg.db, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

LazyData true

VignetteBuilder knitr

Roxygen list(markdown = TRUE)

Config/roxygen2/version 8.1.0

Repository <https://gflab.r-universe.dev>

Date/Publication 2026-09-16 05:07:48 UTC

RemoteUrl <https://github.com/gflab/clinstats>

RemoteRef main

RemoteSha ac4cc66ddb925bb3eeb67db404d9c3239765e814

Contents

clean_expression	2
clin_crc	3
copy_report_template	4
cox_table	5
evaluate_model	6
odds_ratio	7
oncotype_crc	8
resample_cox	9
roc_summary	10
survival_cutoff	11
survival_response	12
tcga_clinical	12
Index	14

clean_expression	<i>Clean an expression matrix and collapse duplicated gene columns</i>
------------------	--

Description

Prepares a gene expression matrix for downstream analysis: drops unnamed and empty columns, expands columns with " /// " multi-mappings, maps gene identifiers when requested, and keeps the column with the largest median absolute deviation when several columns map to the same gene.

Usage

```
clean_expression(expression, id_from = NULL, id_to = "SYMBOL")
```

Arguments

expression	Numeric matrix or data frame with samples in rows and gene identifiers in columns.
id_from	Optional identifier type of the column names, for example "ENTREZID". When supplied, identifiers are translated to id_to with org.Hs.eg.db::org.Hs.eg.db , which must be installed.
id_to	Target identifier type used when id_from is supplied.

Value

A numeric matrix with samples in rows and unique gene identifiers in columns.

Examples

```
expr <- matrix(
  c(1, 2, 3, 4, 5, 6, NA, 8),
  nrow = 2,
  dimnames = list(NULL, c("A", "A", "B", ""))
)
clean_expression(expr)
```

clin_crc

Curated colorectal cancer clinical data

Description

Clinical information for colorectal cancer samples assembled from public repositories: the GEO datasets GSE14333, GSE17538, GSE18088, GSE26906, GSE31595, GSE33113, GSE37892, GSE39084, and GSE39582 (the PETACC-3 cohort), and TCGA colon and rectal cancer clinical exports. The data contain research-level variables only and no patient identifiers. These objects are carried over from the archived curatedClinicalData repository without changing the stored values.

Usage

clin_crc

clin_crc_gf

clin_crc_cell

Format

A data frame with 25 columns:

sample Sample or patient identifier.

sex Sex ("M"/"F").

age Age at diagnosis in years.

tnm.stage TNM stage as an ordered number.

tnm.t, **tnm.n**, **tnm.m** TNM T, N, and M categories.

lymphnodes Lymph node involvement.

tumor.location Tumour location ("L"/"R").

grade Tumour grade.

chemotherapy.adjuvant Adjuvant chemotherapy indicator.

rfs.event, **rfs.delay** Relapse-free survival event and time (years).

os.event, **os.delay** Overall survival event and time (years).

dfs.event, **dfs.delay** Disease-free survival event and time (years).

mmr.status, **cimp.status**, **cin.status** Molecular subtype indicators.

tp53.mutation, kras.mutation, braf.mutation Mutation status.

dataset Source dataset.

cms Consensus molecular subtype label.

A data frame with 2,868 rows and 25 columns, in the same layout as [clin_crc](#). This version adds the TCGA update of 2018.

A data frame with 2,865 rows and 25 columns, in the same layout as [clin_crc](#). This version carries the 2019 TCGA cell-line update.

Source

GEO series GSE14333, GSE17538, GSE18088, GSE26906, GSE31595, GSE33113, GSE37892, GSE39084, and GSE39582; TCGA clinical exports.

Examples

```
dim(clin_crc)
table(clin_crc$dataset)
```

copy_report_template *Copy the bundled Quarto analysis report template*

Description

Copies analysis-report, a Quarto document with lab-style defaults for HTML and PDF output, into a directory of the current project.

Usage

```
copy_report_template(path = ".", overwrite = FALSE)
```

Arguments

path	Directory into which the template files are copied.
overwrite	Replace existing files.

Value

The paths of the copied files, invisibly.

cox_table

*Regression tables for clinical variables***Description**

cox_table() and logistic_table() fit each variable on its own, fit one multivariable model, and return a single tidy table with the univariable and multivariable estimates side by side. Numeric variables contribute one row with a per-unit estimate; factors contribute one row per level against the reference level.

Usage

```
cox_table(
  data,
  time,
  event,
  factors = NULL,
  max_time = NULL,
  multivariable = c("all", "significant", "none"),
  significance = 0.05,
  ci_level = 0.95
)

logistic_table(
  data,
  outcome,
  factors = NULL,
  positive = NULL,
  multivariable = c("all", "significant", "none"),
  significance = 0.05,
  ci_level = 0.95
)
```

Arguments

data	Data frame holding the variables.
time, event	Column names of the follow-up time and event indicator used by cox_table().
factors	Character vector of variable names to analyse. Defaults to every column of data except the time and event columns.
max_time	Optional follow-up limit for cox_table(). Times beyond the limit are censored at the limit.
multivariable	Which variables enter the multivariable model: "all" (default), "significant", or "none".
significance	Significance level used by multivariable = "significant".
ci_level	Confidence level of the reported intervals.

outcome	Column name of the binary outcome used by <code>logistic_table()</code> .
positive	Value of outcome that represents the event.

Details

The multivariable model includes all requested variables by default. With `multivariable = "significant"` only variables whose univariable likelihood-ratio test is below significance are included, which reproduces the screening rule of the original helper functions.

Value

A tibble with one row per term and the columns `term`, `variable`, `level`, `n`, `n_event`, the univariable estimate (`hr_univariable` or `or_univariable`) with interval and p-value, and the corresponding multivariable columns.

Examples

```
set.seed(42)
dat <- data.frame(
  time = rexp(60, 0.1),
  event = rbinom(60, 1, 0.7),
  age = rnorm(60, 60, 8),
  stage = factor(sample(c("II", "III"), 60, replace = TRUE))
)
cox_table(dat, time = "time", event = "event")
set.seed(42)
dat <- data.frame(
  response = rbinom(60, 1, 0.4),
  age = rnorm(60, 60, 8),
  sex = factor(sample(c("F", "M"), 60, replace = TRUE))
)
logistic_table(dat, outcome = "response")
```

<code>evaluate_model</code>	<i>Evaluate a prediction model on training and test data</i>
-----------------------------	--

Description

Fits a logistic regression model on train, evaluates discrimination on both train and test, and compares the two areas under the curve with DeLong's test for unpaired samples.

Usage

```
evaluate_model(train, test, formula, positive = NULL, ci_level = 0.95)
```

Arguments

train, test	Data frames with the model variables.
formula	Model formula, for example outcome ~ marker1 + marker2.
positive	Value of the outcome that represents the event.
ci_level	Confidence level of the reported AUC intervals.

Value

A tibble with one row: sample sizes and event counts of both data sets, the training and test AUC with confidence intervals, and the DeLong p-value comparing them.

Examples

```
set.seed(1)
train <- data.frame(y = rbinom(80, 1, 0.5), x = rnorm(80))
test <- data.frame(y = rbinom(40, 1, 0.5), x = rnorm(40))
evaluate_model(train, test, y ~ x)
```

odds_ratio	<i>Odds ratio for a marker with a binary outcome</i>
------------	--

Description

Binaries a marker and reports the odds ratio for the event in the high group relative to the low group. Continuous markers are split at the Youden-optimal ROC threshold unless cutoff is supplied; two-level markers and factors are used as they are.

Usage

```
odds_ratio(x, outcome, positive = NULL, cutoff = NULL, ci_level = 0.95)
```

Arguments

x	Marker or factor to binarise. Continuous numeric values are split at the optimal threshold; numeric, logical, character, and factor inputs with two distinct values are used directly.
outcome	Binary outcome. Logical vectors are coded TRUE as the event; numeric 0/1 vectors use 1 as the event. Any other coding requires positive to name the event level.
positive	Value of outcome that represents the event.
cutoff	Optional threshold supplied by the user. Values greater than cutoff form the high group.
ci_level	Confidence level for the AUC interval.

Value

A tibble with one row: the threshold that was used, sample size, number of events, group sizes, the odds ratio with a Wald confidence interval, and the Wald p-value.

Examples

```
set.seed(1)
marker <- c(rnorm(40, 0), rnorm(40, 1))
outcome <- rep(c(0, 1), each = 40)
odds_ratio(marker, outcome)
```

oncotype_crc	<i>Oncotype DX colon cancer recurrence score</i>
--------------	--

Description

Computes the Oncotype DX colon cancer recurrence score from the published gene panel. The formula and the panel, including the reference genes, are unchanged from the original `calc_oncotypedx_crc()` helper.

Usage

```
oncotype_crc(expression, gene_axis = c("columns", "rows"))
```

Arguments

expression	Numeric matrix of gene expression values.
gene_axis	Axis that holds the gene symbols. "columns" (the default) expects samples in rows, which is the orientation returned by <code>clean_expression()</code> and the convention inherited from <code>gaofenglib</code> ; the two functions therefore compose directly. Use "rows" for matrices with genes in rows and samples in columns.

Value

A tibble with one row per sample: the stromal, cell-cycle, and individual gene scores, the reference score, the corrected scores, the recurrence score, and the recurrence score category ("Low", "Intermediate", or "High").

Examples

```
panel <- c(
  "BGN", "FAP", "INHBA", "MKI67", "MYC", "MYBL2", "GADD45B",
  "ATP5E", "GPX1", "PGK1", "VDAC2", "UBB"
)
samples_by_genes <- matrix(
  rnorm(3 * 12, 10),
  nrow = 3,
  dimnames = list(paste0("s", 1:3), panel)
```

```

)
oncotype_crc(samples_by_genes)

# Genes in rows also work when declared explicitly.
oncotype_crc(t(samples_by_genes), gene_axis = "rows")

```

resample_cox

*Resampling-based Cox screening of candidate markers***Description**

Repeatedly draws subsamples of the samples, fits a univariable Cox model for every candidate column, and reports how often each candidate reached `p_threshold`. This is the resampling screening previously provided by `calc_resamp_cox()`, reported as per-marker selection frequencies instead of a raw p-value matrix.

Usage

```

resample_cox(
  expression,
  time,
  event,
  ratio = 0.8,
  times = 1000,
  p_threshold = 0.05,
  seed = NULL
)

```

Arguments

<code>expression</code>	Numeric matrix or data frame with samples in rows and candidate markers in columns.
<code>time, event</code>	Follow-up time and event indicator, one value per sample.
<code>ratio</code>	Fraction of samples drawn in each resample.
<code>times</code>	Number of resamples.
<code>p_threshold</code>	Significance threshold that counts as a selection.
<code>seed</code>	Optional integer seed for reproducible resampling.

Details

The function uses the future framework. Set a parallel plan before calling it to use several workers, for example `future::plan(future::multisession)`.

Value

A tibble with one row per candidate marker: the number of valid resamples, the number of resamples in which the marker was significant, the selection fraction, and the median p-value.

Examples

```
set.seed(1)
expr <- matrix(rnorm(60 * 5), nrow = 60, dimnames = list(NULL, paste0("g", 1:5)))
time <- rexp(60)
event <- rbinom(60, 1, 0.7)
resample_cox(expr, time, event, times = 20, seed = 1)
```

roc_summary

Summarise the diagnostic performance of a numeric marker

Description

Computes the area under the receiver operating characteristic curve (AUC) with a confidence interval and the optimal operating point of a numeric marker against a binary outcome. A tibble with one row is returned, so the result can be combined across markers or subgroups.

Usage

```
roc_summary(
  marker,
  outcome,
  positive = NULL,
  direction = "auto",
  best_method = c("youden", "closest.topleft"),
  ci_level = 0.95,
  power = FALSE,
  alternative = c("one.sided", "two.sided")
)
```

Arguments

marker	Numeric vector of marker values.
outcome	Binary outcome. Logical vectors are coded TRUE as the event; numeric 0/1 vectors use 1 as the event. Any other coding requires positive to name the event level.
positive	Value of outcome that represents the event.
direction	Direction of the comparison passed on to <code>pROC::roc()</code> . "auto" lets pROC choose the direction from the data; "<" and ">" fix it explicitly.
best_method	Method used to choose the optimal operating point, "youden" (default) or "closest.topleft".
ci_level	Confidence level for the AUC interval.
power	If TRUE, add the power of the AUC test as reported by <code>pROC::power.roc.test()</code> . The value may be NA when power cannot be estimated for the sample.
alternative	Alternative hypothesis used for the power estimate.

Value

A tibble with one row: sample size, number of events, AUC and its confidence interval, the optimal threshold, and the sensitivity, specificity, positive and negative predictive values, and accuracy at that threshold.

See Also

[odds_ratio\(\)](#) for the odds ratio at the optimal threshold.

Examples

```
set.seed(1)
marker <- c(rnorm(40, 0), rnorm(40, 1))
outcome <- rep(c(0, 1), each = 40)
roc_summary(marker, outcome)
```

survival_cutoff

Survival cut point optimising the Youden index

Description

Estimates Kaplan-Meier survival above and below each candidate cut point of a marker at `predict_time` and returns the cut point that maximises the Youden index (sensitivity + specificity - 1), the same criterion that the long-unmaintained `survivalROC` package uses, computed here directly from `survival::survfit()`. Candidates are drawn from a quantile grid to keep the computation bounded.

Usage

```
survival_cutoff(time, event, marker, predict_time, grid_size = 200)
```

Arguments

<code>time</code>	Numeric follow-up times.
<code>event</code>	Event indicator (0/1, logical, or a two-level value).
<code>marker</code>	Numeric marker to split.
<code>predict_time</code>	Time point at which sensitivity and specificity are evaluated.
<code>grid_size</code>	Number of candidate cut points evaluated between the 1st and 99th percentile of the marker.

Value

A tibble with one row: the optimal cutoff, the Youden index (`youden`), the time-dependent true positive rate (`tp`) and false positive rate (`fp`) at that cut point, and the number of observations.

Examples

```
set.seed(1)
time <- rexp(100)
event <- rbinom(100, 1, 0.6)
marker <- rnorm(100)
survival_cutoff(time, event, marker, predict_time = 1)
```

survival_response	<i>Build a survival response from a clinical data frame</i>
-------------------	---

Description

Build a survival response from a clinical data frame

Usage

```
survival_response(data, type = c("rfs", "os", "dfs"))
```

Arguments

data	Data frame such as the included clin_crc dataset.
type	Endpoint prefix: "rfs", "os", or "dfs". The columns <type>.delay and <type>.event must exist.

Value

A [survival::Surv\(\)](#) object.

Examples

```
survival_response(clin_crc, type = "rfs")
```

tcga_clinical	<i>Extract clinical variables from a legacy TCGA clinical table</i>
---------------	---

Description

Converts the wide clinical export distributed with legacy TCGA data releases (column names of the form patient.age_at_initial_pathologic_diagnosis) into an analysis-ready tibble with follow-up times in days. Three column layouts are supported: the generic layout, the oesophageal carcinoma layout ("esca"), and the stomach adenocarcinoma layout ("stad").

Usage

```
tcga_clinical(clin, cancer = c("default", "esca", "stad"))
```

Arguments

clin	Data frame with one row per patient in the legacy TCGA layout.
cancer	Column layout: "default", "esca" (the historical ESAC = TRUE variant), or "stad".

Value

A tibble with the patient barcode, age, sex, stage, node status, T stage, disease-free survival, relapse and death status with their times, and, where available, chemotherapy and response. The "esca" layout additionally returns the histological subtype and tumour location.

Index

* datasets

clin_crc, [3](#)

clean_expression, [2](#)

clean_expression(), [8](#)

clin_crc, [3](#), [4](#), [12](#)

clin_crc_cell (clin_crc), [3](#)

clin_crc_gf (clin_crc), [3](#)

copy_report_template, [4](#)

cox_table, [5](#)

evaluate_model, [6](#)

logistic_table (cox_table), [5](#)

odds_ratio, [7](#)

odds_ratio(), [11](#)

oncotype_crc, [8](#)

org.Hs.eg.db::org.Hs.eg.db, [2](#)

pROC::power.roc.test(), [10](#)

pROC::roc(), [10](#)

resample_cox, [9](#)

roc_summary, [10](#)

survival::Surv(), [12](#)

survival::survfit(), [11](#)

survival_cutoff, [11](#)

survival_response, [12](#)

tcga_clinical, [12](#)