Package 'tm.hyslaw'

November 23, 2026

Title 'teal' Modules for Hy's Law Monitoring and Evaluation of Drug-Induced Serious Hepatotoxicity (eDISH) Plots

Version 0.2.0

Description Provides open-source 'teal' modules and reusable builders for exploratory monitoring of Hy's Law (Hyman Zimmerman's observation that hepatocellular injury with jaundice predicts severe drug-induced liver injury) from Clinical Data Interchange Standards Consortium (CDISC) Analysis Data Model (ADaM) subject-level analysis dataset (ADSL) and laboratory analysis dataset (ADLB) records. Peak evaluation of Drug-Induced Serious Hepatotoxicity (eDISH), modified evaluation of Drug-Induced Serious Hepatotoxicity (mDISH; on-treatment peaks divided by baseline), and cholestatic screening plots are produced, and concurrent alanine aminotransferase (ALT) or aspartate aminotransferase (AST) plus bilirubin elevations, with optional alkaline phosphatase (ALP) exclusion, are classified without a study-specific drug-induced liver injury analysis dataset (ADDILI). Table and plot builders can be called without launching 'shiny'. Hy's Law and eDISH criteria are discussed by Temple (2006) <doi:10.1002/pds.1211>, the United States Food and Drug Administration (2009) ``Drug-Induced Liver Injury: Premarketing Clinical Evaluation'' <https://www.fda.gov/media/116737/download>, and Watkins et al. (2011) <doi:10.2165/11586600-000000000-00000>. Baseline-scaled mDISH plots follow the fold-baseline display in Lin et al. (2012) <doi:10.1007/BF03261982>.

License MIT + file LICENSE

URL https://github.com/ygma-bitc/tm.hyslaw, https://ygma-bitc.github.io/tm.hyslaw/

BugReports https://github.com/ygma-bitc/tm.hyslaw/issues

Depends R (>= 4.1)

Imports checkmate (>= 2.1.0), dplyr (>= 1.0.5), ggplot2 (>= 3.4.0), lifecycle (>= 1.0.0), rlang (>= 1.0.0), rtables (>= 0.6.0), shiny (>= 1.7.0), shinyjs (>= 2.0.0), shinyvalidate (>= 0.1.0), teal (>= 1.2.0), teal.picks (>= 0.1.0), teal.widgets (>= 0.4.0)

Suggests haven, knitr, lintr, pharmaverseadam, pkgload, pkgdown, random.cdisc.data, rmarkdown, teal.data (>= 0.5.0), testthat (>= 3.2.0), withr

Config/testthat/edition 3

Encoding UTF-8

Language en-US

RoxygenNote 8.0.0

Config/roxygen2/version 8.0.0

NeedsCompilation no

Author Yaoguang Ma [aut, cre]

Maintainer Yaoguang Ma <ygma0929@gmail.com>

Repository CRAN

Date/Publication 2026-11-23 15:10:08 UTC

Contents

build_hyslaw_table 2
build_liver_elevation_table 4
classify_hyslaw 5
default_cdish_quadrants 6
default_edish_quadrants 6
default_hyslaw_footnotes 7
hyslaw_listing 7
make_synthetic_hyslaw_data 8
plot_edish 8
prep_hyslaw_data 10
standardize_adlb 11
standardize_adsl 11
tm_g_edish 12
tm_t_hyslaw 14

Index **16**

`build_hyslaw_table` *Build Hy's Law / eDISH review tables*

Description

Default output is two stacked sections: maximal treatment-emergent (peak eDISH) quadrant counts, then biochemical concurrent Hy's Law in the selected window. Use `build_liver_elevation_table()` for the nested ALT/AST/BILI/ALP elevation shell.

Usage

```
build_hyslaw_table(
  adsl = NULL,
  adlb = NULL,
  arm_var = "TRT01A",
  show_total = TRUE,
  title = "Hy's Law monitoring (exploratory)",
  footnotes = NULL,
  classified = NULL,
  which = c("quadrant", "concurrent"),
  ...
)
```

Arguments

adsl	ADSL-like data. Ignored when classified is supplied.
adlb	ADLB-like data.
arm_var	Treatment / grouping variable.
show_total	Whether to add a Total column.
title	Table title.
footnotes	Optional footnotes.
classified	Optional list from classify_hyslaw().
which	Character vector of sections: "quadrant", "concurrent".
...	Passed to classify_hyslaw() when classified is NULL.

Details

Pass either raw adsl / adlb or a list returned by classify_hyslaw() via classified so the teal server can prepare data once.

Value

An rtables VTableTree.

Examples

```
data <- make_synthetic_hyslaw_data(n_extra = 0L, seed = 1L)
build_hyslaw_table(data$ADSL, data$ADLB)
```

`build_liver_elevation_table` *Nested on-treatment liver-test elevation table*

Description

Counts subjects whose peak on-treatment ULN ratio meets 3/5/10/20x (transaminases) or 1.5/2x (bilirubin, ALP), then the combination of concurrent biochemical Hy's Law split by ALP in the window.

Usage

```
build_liver_elevation_table(
  adsl = NULL,
  adlb = NULL,
  arm_var = "TRT01A",
  show_total = TRUE,
  title = "Maximal on-treatment liver test elevations (exploratory)",
  footnotes = NULL,
  classified = NULL,
  at_cuts = c(3, 5, 10, 20),
  bili_cuts = c(1.5, 2),
  alp_cuts = c(1.5, 2),
  ...
)
```

Arguments

`adsl`	ADSL-like data. Ignored when `classified` is supplied.
`adlb`	ADLB-like data.
`arm_var`	Treatment / grouping variable.
`show_total`	Whether to add a Total column.
`title`	Table title.
`footnotes`	Optional footnotes.
`classified`	Optional list from `classify_hyslaw()`.
`at_cuts`	Transaminase ULN multiples.
`bili_cuts`	Bilirubin ULN multiples.
`alp_cuts`	ALP ULN multiples.
`...`	Passed to `classify_hyslaw()` when `classified` is `NULL`.

Value

An `rtables` `VTableTree`.

`classify_hyslaw` *Classify eDISH peak quadrants and concurrent Hy's Law*

Description

Peak eDISH uses each subject's **on-treatment** maximum ALT or AST / ULN versus maximum bilirubin / ULN (not necessarily the same day). Concurrent (biochemical potential) Hy's Law requires bilirubin meeting `bili_cut` on or after an ALT/AST meeting `alt_cut`, within `window_days`. ALP meeting `alp_cut` in the same window can optionally exclude the subject from potential Hy's Law.

Usage

```
classify_hyslaw(
  adsl = NULL,
  adlb = NULL,
  prepped = NULL,
  window_days = 14,
  alt_cut = 3,
  bili_cut = 2,
  alp_cut = 2,
  exclude_alp = TRUE,
  compare = c(">=", ">"),
  ...
)
```

Arguments

`adsl`	ADSL-like data. Ignored when `prepped` is supplied.
`adlb`	ADLB-like data. Ignored when `prepped` is supplied.
`prepped`	Optional list from `prep_hyslaw_data()`.
`window_days`	Concurrent window in days after the ALT/AST elevation. Default `14` matches the pharmaverse admiral Hy's Law vignette.
`alt_cut`	ALT/AST ULN multiple.
`bili_cut`	Bilirubin ULN multiple.
`alp_cut`	ALP ULN multiple used for cholestasis exclusion and the cholestatic screening plot.
`exclude_alp`	If `TRUE`, concurrent cases with ALP meeting `alp_cut` in the window are not counted as potential Hy's Law.
`compare`	`">="` (default) or `">"`. FDA / many CSR shells use strict greater-than; this package defaults to greater-or-equal.
`...`	Passed to `prep_hyslaw_data()` when `prepped` is `NULL`.

Details

Baseline records are kept for mDISH (peak / baseline) ratios. They are not used as eDISH maxima.

Value

The `prepped` list with a subject-level `subjects` data frame.

`default_cdish_quadrants` *Default cholestatic screening (ALP vs bilirubin) quadrant labels*

Description

Default cholestatic screening (ALP vs bilirubin) quadrant labels

Usage

```
default_cdish_quadrants()
```

Value

A named character vector of quadrant codes and display labels.

`default_edish_quadrants` *Default eDISH quadrant labels*

Description

Corner / table labels follow the hepatocellular screening plot (peak maxima). The upper-left quadrant is named Cholestasis, matching FDA ST&F / CDISC ADDILI slide language.

Usage

```
default_edish_quadrants()
```

Value

A named character vector of quadrant codes and display labels.

default_hyslaw_footnotes *Default method footnotes for Hy's Law tables*

Description

Default method footnotes for Hy's Law tables

Usage

```
default_hyslaw_footnotes(classified = NULL)
```

Arguments

classified	Optional list from classify_hyslaw() used to fill window, comparison operator, and ALP settings.

Value

Character vector of footer notes.

hyslaw_listing *Subject listing for notable Hy's Law / eDISH cases*

Description

Subject listing for notable Hy's Law / eDISH cases

Usage

```
hyslaw_listing(
  adsl = NULL,
  adlb = NULL,
  classified = NULL,
  which = c("notable", "all"),
  ...
)
```

Arguments

adsl, adlb	Passed to classify_hyslaw() when classified is NULL.
classified	Optional list from classify_hyslaw().
which	"notable" keeps peak Hy's Law, concurrent, ALP-excluded, or concurrent cholestatic subjects; "all" returns every subject.
...	Passed to classify_hyslaw().

Value

A data frame. Symptoms, alternative causes, and confirmed Hy's Law are not derived from ADLB; see footnotes on `build_hyslaw_table()`.

`make_synthetic_hyslaw_data` *Build synthetic ADSL / ADLB fixtures for Hy's Law demos and tests*

Description

Seeds subjects into the four eDISH peak quadrants and two extra patterns that distinguish peak-only Hy's Law from concurrent-window monitoring: a peak pair outside the bilirubin window, and a concurrent pair with elevated ALP (cholestasis exclusion).

Usage

```
make_synthetic_hyslaw_data(n_extra = 6L, seed = 20260829L)
```

Arguments

`n_extra`	`integer(1)` Extra normal-range subjects per arm.
`seed`	`integer(1)` Random seed for extra subjects only.

Value

A list with `ADSL` and `ADLB`.

Examples

```
data <- make_synthetic_hyslaw_data(seed = 1L)
names(data)
unique(data$ADLB$PARAMCD)
```

`plot_edish` *Liver-injury screening scatter (eDISH / mDISH / cholestatic)*

Description

Default hepatocellular eDISH is one log-log panel colored by treatment so arm imbalance is visible. Quadrant names are annotated in the corners. Open red circles mark biochemical potential Hy's Law (bilirubin in the concurrent window after ALT/AST elevation, with optional ALP exclusion). Axis maxima are not required to occur on the same day.

Usage

```
plot_edish(
  adsl = NULL,
  adlb = NULL,
  classified = NULL,
  which = c("edish", "mdish", "cholestatic"),
  color_by = c("arm", "quadrant"),
  log_axes = TRUE,
  facet = FALSE,
  annotate_quadrants = TRUE,
  mark_potential = TRUE,
  title = NULL,
  ...
)
```

Arguments

adsl, adlb	Passed to classify_hyslaw() when classified is NULL.
classified	Optional list from classify_hyslaw().
which	"edish" is peak ALT or AST / ULN vs peak BILI / ULN; "mdish" uses peak / baseline; "cholestatic" is peak ALP / ULN vs peak BILI / ULN.
color_by	"arm" colors points by the grouping variable (default, for treatment imbalance). "quadrant" colors by screening quadrant.
log_axes	If TRUE, both axes use log10 scales.
facet	If TRUE, facet by the grouping variable when it has at least two levels. Default FALSE keeps arms on one panel.
annotate_quadrants	If TRUE, print quadrant names in the corners.
mark_potential	If TRUE, overlay open red circles on biochemical potential Hy's Law (eDISH / mDISH) or concurrent cholestatic cases.
title	Plot title. Generated from which when NULL.
...	Passed to classify_hyslaw().

Value

A ggplot object.

Examples

```
data <- make_synthetic_hyslaw_data(n_extra = 0L, seed = 1L)
plot_edish(data$ADSL, data$ADLB)
plot_edish(data$ADSL, data$ADLB, which = "cholestatic")
```

`prep_hyslaw_data` *Prepare ADSL / ADLB records for Hy's Law review*

Description

Filters liver-related laboratory parameters, computes ULN multiples, and joins the analysis population. An empty result is returned as a structured list (`empty = TRUE`) so teal modules can show a no-data message.

Usage

```
prep_hyslaw_data(
  adsl,
  adlb,
  arm_var = "TRT01A",
  subject_var = "USUBJID",
  paramcd_var = "PARAMCD",
  aval_var = "AVAL",
  anrhi_var = "ANRHI",
  adt_var = "ADT",
  ady_var = "ADY",
  population_var = "SAFFL",
  population_value = "Y",
  ontrt_only = TRUE,
  liver_paramcds = c("ALT", "AST", "BILI", "ALP", "ALKPH")
)
```

Arguments

`adsl`	ADSL-like subject-level data.
`adlb`	ADLB-like BDS laboratory data.
`arm_var`	Treatment / grouping variable.
`subject_var`	Subject identifier.
`paramcd_var`	Parameter code variable.
`aval_var`	Analysis value variable.
`anrhi_var`	ULN variable.
`adt_var`	Analysis date variable.
`ady_var`	Analysis day variable.
`population_var`	Optional population flag variable.
`population_value`	Population flag value to keep.
`ontrt_only`	If `TRUE` and `ONTRTFL` exists, keep on-treatment records plus baseline (`ABLFL == "Y"`) when that flag is present.
`liver_paramcds`	Parameter codes to keep. `ALKPH` is treated as ALP.

Value

A list with long labs, adsl, alt_counts_df, metadata, and empty.

---

standardize_adlb *Standardize an ADLB / ADLBC-like data frame*

Description

Maps Pilot A1HI to ANRHI, ALKPH to ALP, and common value columns (AVAL, LBSTRESN) so prep_hyslaw_data() can compute ULN multiples.

Usage

```
standardize_adlb(adlb, adsl = NULL)
```

Arguments

adlb	A laboratory BDS data frame.
adsl	Optional standardized ADSL used to back-fill TRT01A.

Value

A standardized ADLB data frame.

---

standardize_adsl *Standardize an ADSL-like data frame for Hy's Law review*

Description

Harmonizes common CDISC Pilot / pharmaverseadam / random.cdisc.data column variants so prep_hyslaw_data() can join treatment and population flags.

Usage

```
standardize_adsl(adsl)
```

Arguments

adsl	A data frame with at least USUBJID.

Value

A standardized ADSL data frame.

tm_g_edish *teal Module: eDISH / mDISH / cholestatic screening plot*

Description

teal Module: eDISH / mDISH / cholestatic screening plot

Usage

```
tm_g_edish(
  label = "eDISH plot",
  adsl_name = "ADSL",
  adlb_name = "ADLB",
 arm_var = teal.picks::variables(choices = c("TRT01A", "TRT01P", "ARM", "SEX", "RACE",
    "COUNTRY"), selected = "TRT01A", multiple = FALSE),
 population_var = teal.picks::variables(choices = c("SAFFL", "ITTFL"), selected =
    "SAFFL", multiple = FALSE),
  population_value = "Y",
  window_days = 14,
  exclude_alp = TRUE,
  compare = c(">=", ">"),
  which = c("edish", "mdish", "cholestatic"),
  color_by = c("arm", "quadrant"),
  facet = FALSE,
  annotate_quadrants = TRUE,
  mark_potential = TRUE,
  log_axes = TRUE,
  group_vars = NULL,
  population_choices = NULL
)
```

Arguments

label	Module label.
adsl_name	ADSL dataset name in teal_data.
adlb_name	ADLB dataset name.
arm_var	(teal.picks::variables()) Treatment or grouping column on ADSL. multiple must be FALSE. The default selection is TRT01A. The control shows column labels from the data.
population_var	(teal.picks::variables()) Population flag column on ADSL. multiple must be FALSE. The default selection is SAFFL. A length-1 character string is deprecated and still accepted as the selected flag.
population_value	Population flag value to keep. Not shown in the UI.

window_days	Default concurrent window.
exclude_alp	Default ALP exclusion.
compare	Default threshold operator, ">=" or ">".
which	"edish", "mdish", or "cholestatic".
color_by	"arm" or "quadrant".
facet	Default faceting.
annotate_quadrants	Default corner labels.
mark_potential	Default red-circle overlay.
log_axes	Default log-axis setting.
group_vars	**[Deprecated]** Named character vector of treatment columns, such as c("Actual Treatment" = "TRT01A"). Still accepted. Converted with lifecycle::deprecate_warn() to teal.picks::variables(choices, selected, multiple = FALSE). Prefer arm_var.
population_choices	**[Deprecated]** Named character vector of population flag columns. Still accepted and converted the same way. Prefer population_var.

Value

A teal module object.

Examples

```
synth <- make_synthetic_hyslaw_data(n_extra = 0L, seed = 1L)
data <- teal.data::teal_data(ADSL = synth$ADSL, ADLB = synth$ADLB)
app <- teal::init(
  data = data,
  modules = teal::modules(
    tm_g_edish(
      arm_var = teal.picks::variables(
        choices = c("TRT01A", "SEX", "RACE", "COUNTRY"),
        selected = "TRT01A",
        multiple = FALSE
      ),
      population_var = teal.picks::variables(
        choices = c("SAFFL", "ITTFL"),
        selected = "SAFFL",
        multiple = FALSE
      )
    )
  )
)
if (interactive()) {
  shiny::shinyApp(app$ui, app$server)
}
```

tm_t_hyslaw *teal Module: Hy's Law summary table and notable listing*

Description

teal Module: Hy's Law summary table and notable listing

Usage

```
tm_t_hyslaw(
  label = "Hy's Law table",
  adsl_name = "ADSL",
  adlb_name = "ADLB",
 arm_var = teal.picks::variables(choices = c("TRT01A", "TRT01P", "ARM", "SEX", "RACE",
    "COUNTRY"), selected = "TRT01A", multiple = FALSE),
 population_var = teal.picks::variables(choices = c("SAFFL", "ITTFL"), selected =
    "SAFFL", multiple = FALSE),
  population_value = "Y",
  window_days = 14,
  exclude_alp = TRUE,
  compare = c(">=", ">"),
  show_total = TRUE,
  table_which = c("both", "review", "elevation"),
  group_vars = NULL,
  population_choices = NULL
)
```

Arguments

label	Module label.
adsl_name	ADSL dataset name in teal_data.
adlb_name	ADLB dataset name.
arm_var	(teal.picks::variables()) Treatment or grouping column on ADSL. multiple must be FALSE. The default selection is TRT01A. The control shows column labels from the data.
population_var	(teal.picks::variables()) Population flag column on ADSL. multiple must be FALSE. The default selection is SAFFL. A length-1 character string is deprecated and still accepted as the selected flag.
population_value	Population flag value to keep. Not shown in the UI.
window_days	Default concurrent window.
exclude_alp	Default ALP exclusion.
compare	Default threshold operator, ">=" or ">".

show_total — Initial Total-column setting.

table_which — "review", "elevation", or "both".

group_vars — **[Deprecated]** Named character vector of treatment columns, such as c("Actual Treatment" = "TRT01A"). Still accepted. Converted with lifecycle::deprecate_warn() to teal.picks::variables(choices, selected, multiple = FALSE). Prefer arm_var.

population_choices — **[Deprecated]** Named character vector of population flag columns. Still accepted and converted the same way. Prefer population_var.

Value

A teal module object.

Examples

```
synth <- make_synthetic_hyslaw_data(n_extra = 0L, seed = 1L)
data <- teal.data::teal_data(ADSL = synth$ADSL, ADLB = synth$ADLB)
app <- teal::init(
  data = data,
  modules = teal::modules(
    tm_t_hyslaw(
      arm_var = teal.picks::variables(
        choices = c("TRT01A", "SEX", "RACE", "COUNTRY"),
        selected = "TRT01A",
        multiple = FALSE
      ),
      population_var = teal.picks::variables(
        choices = c("SAFFL", "ITTFL"),
        selected = "SAFFL",
        multiple = FALSE
      )
    )
  )
)
if (interactive()) {
  shiny::shinyApp(app$ui, app$server)
}
```

Index

build_hyslaw_table, 2
build_hyslaw_table(), *8*
build_liver_elevation_table, 4
build_liver_elevation_table(), *2*

classify_hyslaw, 5
classify_hyslaw(), *3, 4*, *7*, *9*

default_cdish_quadrants, 6
default_edish_quadrants, 6
default_hyslaw_footnotes, 7

hyslaw_listing, 7

lifecycle::deprecate_warn(), *13*, *15*

make_synthetic_hyslaw_data, 8

plot_edish, 8
prep_hyslaw_data, 10
prep_hyslaw_data(), *5*, *11*

standardize_adlb, 11
standardize_adsl, 11

tm_g_edish, 12
tm_t_hyslaw, 14