Package: mectx (via r-universe)

May 11, 2026

Type Package

Title MEchanistic Clustering - Treatment eXposure Framework

Version 1.1.1

Description Implements the MEC-TX (MEchanistic Clustering - Treatment eXposure) framework for encoding, clustering, and survival analysis of real-world oncology treatment timelines. Provides functions for normalising medication records, computing treatment intervals, performing k-means clustering in PCA space, assigning line-of-therapy labels, and comparing survival outcomes across treatment groups. Designed for use with registry-based cohorts such as the ORIEN AVATAR dataset. Methods follow the digital-twin framework described in Dhrubo and Spakowicz (2026) <https://github.com/spakowiczlab/mec-tx>. treatment timelines using the MEC-TX digital-twin framework.

License MIT + file LICENSE

Encoding UTF-8

Language en-US

RoxygenNote 7.3.3

Depends R (>= 4.4.0)

Imports magrittr, broom, dplyr, forcats, ggnewscale, ggplot2, patchwork, purrr, stats, stringr, survival, tibble, tidyr, umap

Suggests testthat (>= 3.0.0), devtools, roxygen2

Config/testthat/edition 3

Date 2026-04-27

NeedsCompilation no

Author Dipankor Dhrubo [aut, cre], Daniel Spakowicz [aut] (ORCID: <https://orcid.org/0000-0003-2314-6435>)

Maintainer Dipankor Dhrubo <dhrubo.2@osu.edu>

Repository https://cran.r-universe.dev

Date/Publication 2026-05-11 20:09:26 UTC

RemoteUrl https://github.com/cran/mectx

RemoteRef HEAD

RemoteSha 71351649ec6489fabb1bb7bf4afe8b868e6ea2f3

Contents

`cox_forest_plot_from_df` *Cox Proportional Hazards Forest Plot*

Description

Fits an adjusted Cox model on a pre-filtered survival dataset and renders a publication-ready forest plot. Handles reference-level encoding, case-insensitive column resolution, numeric covariate rescaling, and events-per-variable (EPV) driven covariate selection automatically. Degrades gracefully — returns an annotated blank plot rather than erroring when data are insufficient.

Usage

```
cox_forest_plot_from_df(
  df,
  covars = c("CAlevel", "stage_group", "sex", "age", "smokingstatus"),
 ref_levels = list(CAlevel = "Low", stage_group = "Local", smokingstatus = "Never"),
  title = "Adjusted Cox",
  min_epv = 5,
  priority = NULL,
```

```
  numeric_scale = list(age = 10),
  numeric_units = list(age = "years"),
  numeric_pretty = NULL,
  base_size = 16,
  title_size = 22,
  axis_title_size = 14,
  axis_text_size = 12,
  bold = TRUE
)
```

Arguments

df	A data frame containing at minimum `diagsurvtime` (numeric, years), `status` (integer, 0/1), and all columns named in `covars`. Typically the `$data` slot from `tx_pooled_analysis` or `tx_compare_groups`. Column name matching is case-insensitive.
covars	Character vector of covariate column names to include in the model. The first element is always retained regardless of EPV. Default: `c("CAlevel", "stage_group", "sex", "age", "smokingstatus")`.
ref_levels	Named list specifying the reference level for each categorical covariate. Keys are matched case-insensitively to column names. Covariates not listed retain their existing factor ordering. Default: `list(CAlevel = "Low", stage_group = "Local", smokingstatus = "Never")`.
title	Character string. Plot title. Default `"Adjusted Cox"`.
min_epv	Integer. Minimum events-per-variable threshold used to cap the number of model parameters at `floor(events / min_epv)`. Covariates are dropped in reverse `priority` order until the cap is satisfied. The first covariate is always retained. Default 5.
priority	Character vector or `NULL`. Covariates listed first are retained preferentially when EPV forces dropping. `NULL` uses the `covars` order as-is.
numeric_scale	Named list mapping numeric covariate names to a divisor applied before modelling. The reported HR reflects one unit of the rescaled variable. Default `list(age = 10)` gives HR per 10-year increment.
numeric_units	Named list mapping numeric covariate names to a unit label used in axis annotations. Default `list(age = "years")`.
numeric_pretty	Named list or `NULL`. Display name overrides for numeric covariates on the forest plot axis. `NULL` auto-generates labels from `numeric_scale` and `numeric_units`.
base_size	Base font size (pt) passed to `ggplot2::theme_minimal()`. Default 16.
title_size	Font size (pt) for the plot title. Default 22.
axis_title_size	Font size (pt) for axis titles. Default 14.
axis_text_size	Font size (pt) for axis tick and covariate labels. Default 12.
bold	Logical. If `TRUE` all text elements use bold weight. Default `TRUE`.

Details

Column requirements: `df` must contain `diagsurvtime` (time from diagnosis in years) and `status` (0 = censored, 1 = event). These names are hardcoded; rename columns upstream if necessary.

EPV selection: Parameters are budgeted as `floor(events / min_epv)`. Categorical covariates consume `(n_levels - 1)` parameters each. The first element of `covars` (typically `CAlevel`) is always included regardless of budget.

Convergence fallback: If the full selected model fails, covariates are dropped one at a time from the end until the model converges.

Value

A `ggplot` object. The subtitle reports complete-case n, event count, and the covariates actually fitted. Returns an annotated blank plot (still a `ggplot`) if columns are missing, fewer than 10 complete cases exist, fewer than 3 events are observed, or the Cox model fails to converge.

See Also

`tx_pooled_analysis`, `tx_compare_groups`, `km_panel_from_df`

Examples

```
df <- data.frame(
  label    = c('High vs Low'),
  estimate = c(1.8),
  conf.low = c(1.1),
  conf.high = c(2.9),
  p.value  = c(0.02),
  stringsAsFactors = FALSE
)
p <- cox_forest_plot_from_df(df = df, title = 'Example Forest Plot')
class(p)
```

`dominant_exclusive` *Assign Mutually Exclusive Dominant Treatment Regimen Per Patient*

Description

Computes per-patient treatment duration shares (excluding ancillary types), identifies which treatment types exceed the share threshold, and assigns a single regimen label using a specificity-first hierarchy. Solves the overlap problem where a patient simultaneously qualifies for `Chemo[DOMINANT]`, `Chemo+IO[DOMINANT]`, and `all_types[DOMINANT]`.

Usage

```
dominant_exclusive(
  timeline,
  ancillary_types = c("Ancillary", "Others"),
  min_share = 0.2
)
```

Arguments

`timeline`	A long-format treatment intervals data frame — typically the `$timeline_long_intv` slot from `tx_intervals`. Must contain columns `sample`, `type`, `start_year`, and `end_year`.
`ancillary_types`	Character vector of treatment types to exclude from share calculation. These types contribute no duration to the denominator. Default `c("Ancillary", "Others")`.
`min_share`	Numeric in `(0, 1)`. Minimum proportion of total non-ancillary treatment duration for a type to qualify as dominant. Default `0.20` (20%). Increase to tighten, decrease to allow more multi-type combinations.

Details

Hierarchy logic: Multi-type combinations are matched before single-agent labels (longest combination first). A patient whose qualifying types are `"Chemo+IO"` is assigned `"Chemo+IO"`, not `"Chemo only"`, even though Chemo alone also exceeds the threshold. This ensures mutual exclusivity without post-hoc filtering.

Share calculation: Duration is computed as `end_year - start_year` per interval. Overlapping intervals are not merged before summing — call `tx_intervals` upstream to ensure non-overlapping intervals are passed in.

Unassigned patients: Patients present in the input but absent from the output had no qualifying type at the given `min_share` threshold (e.g. highly fragmented treatment with no dominant type). These are distinct from `"Other"` — they have no dominant signal, not an unrecognised one.

Value

A tibble with one row per patient and two columns:

sample Patient identifier, matching the input `timeline`.

regimen Character. Mutually exclusive dominant regimen label. One of: `"Chemo+Radiation+IO"`, `"Chemo+IO"`, `"Chemo+Radiation"`, `"Chemo+Targeted"`, `"Chemo only"`, `"Radiation only"`, `"IO only"`, `"Small Molecule only"`, `"Hormone only"`, `"Other"`.

Patients whose entire treatment record consists of `ancillary_types` are dropped from the output. A message reports the count of dropped patients. Use a left join against your full cohort to identify them — do not treat them as equivalent to the `"Other"` regimen stratum. A warning is raised if any patient is assigned more than one regimen, which indicates a gap in the hierarchy.

See Also

`tx_intervals`, `tx_pooled_analysis`, `get_focus_cohort`

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                     = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact           = spec_ages[i] + 3,
    diagsurvtime               = 3,
    Status                     = i %% 2L,
    Medication                 = c('DrugA','DrugB','DrugC'),
    treatment_group            = tx_types[[i]],
    AgeAtMedStart              = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop               = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod    = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors           = FALSE
  )
}))
norm      <- tx_normalize(med_data)
intervals <- tx_intervals(norm)
res <- dominant_exclusive(intervals)
head(res)
```

`get_focus_cohort` *Extract Focus Cohort Sample IDs*

Description

Identifies patients matching a specific treatment focus and mode, returning a tibble of sample IDs with associated metadata. Pure data extraction — no plotting. Designed as a reusable upstream helper for cohort building before survival analysis (e.g. IO-only resistance analysis, concurrent chemoradiation cohort).

Usage

```
get_focus_cohort(
  metadata,
  timeline,
  focus_types,
```

```
  mode = c("only", "concurrent", "dominant"),
  min_share = 0.2,
  ancillary_types = c("Ancillary", "Others")
)
```

Arguments

`metadata`	A data frame containing survival and cluster columns for the analysis cohort. The timeline is scoped to patients present here before any filtering, ensuring the focus cohort cannot include samples excluded by upstream filters. Must contain a `sample` column.
`timeline`	A long-format treatment intervals data frame — typically the `$timeline_long_intv` slot from `tx_intervals`. Must contain columns `sample`, `type`, `start_year`, and `end_year`.
`focus_types`	Character vector of treatment types to focus on. Must match type labels in the `type` column of `timeline` (e.g. `c("Chemo", "Radiation")`). All specified types must be present for a patient to qualify.
`mode`	One of `"only"`, `"concurrent"`, or `"dominant"`. Controls the inclusion logic — see Details. Default `"only"`.
`min_share`	Numeric in `(0, 1)`. Minimum treatment duration share threshold. Used in `"only"` mode to exclude patients with substantial non-focus treatment, and in `"dominant"` mode to require that focus types collectively reach this share. Default `0.20`.
`ancillary_types`	Character vector of treatment types excluded from share calculations and mode filtering. Default `c("Ancillary", "Others")`.

Details

Cohort scoping: The first operation inside the function is filtering `timeline` to patients present in `metadata`. This ensures upstream exclusions (stage filters, duplicate removal, minimum follow-up requirements) are respected before any mode logic runs.

Mode definitions:

`"only"` Patient must have ALL `focus_types` present and must have NO non-focus type exceeding `min_share`. Use for pure-modality cohorts (e.g. radiation-only patients).

`"concurrent"` Patient must have ALL `focus_types` with at least one pairwise temporal overlap between each type combination. Requires `length(focus_types) >= 2`; returns an empty result with a warning otherwise. Use for regimens administered simultaneously (e.g. concurrent chemoradiation — the standard of care signal in locally advanced LUSC).

`"dominant"` Patient must have ALL `focus_types` and their combined share must meet or exceed `min_share`. Less restrictive than `"only"` — other treatment types may also be present. Use when focus types need not be exclusive but should dominate the treatment record.

Share calculation: Duration shares exclude `ancillary_types` from the denominator. Overlapping intervals are not merged — call `tx_intervals` upstream to ensure clean intervals.

Concurrent overlap check: Overlap is assessed pairwise across all combinations of `focus_types`. Two intervals overlap when `start1 < end2 & start2 < end1` (strict inequality — touching endpoints do not count).

Value

A tibble with one row per qualifying patient and five columns:

sample Patient identifier.

focus_share Numeric. Combined duration share of all `focus_types` for this patient (sum across types, excluding ancillary denominator).

mode Character. The `mode` argument used, recorded for traceability.

focus_types Character. `focus_types` collapsed with `"+"`, recorded for traceability.

n_patients Integer. Total number of qualifying patients (same value in every row).

Returns a zero-row tibble (with the same columns) if no patients qualify. A message is printed reporting the focus, mode, and n.

See Also

`tx_intervals`, `dominant_exclusive`, `tx_pooled_analysis`

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                    = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact          = spec_ages[i] + 3,
    diagsurvtime              = 3,
    Status                    = i %% 2L,
    Medication                = c('DrugA','DrugB','DrugC'),
    treatment_group           = tx_types[[i]],
    AgeAtMedStart             = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop              = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod   = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors          = FALSE
  )
}))
meta <- data.frame(
  sample       = paste0('P', seq_len(n)),
  diagsurvtime = rep(3, n),
  Status       = seq_len(n) %% 2L,
  CAlevel      = rep(c('High','Low'), n/2),
  stringsAsFactors = FALSE
```

```
)
norm      <- tx_normalize(med_data)
intervals <- tx_intervals(norm)
cohort <- get_focus_cohort(meta, intervals, focus_types = c('Chemo', 'IO'))
head(cohort)
```

`km_panel_from_df` *Kaplan-Meier Survival Panel with Optional Risk Table*

Description

Fits a Kaplan-Meier survival curve stratified by a grouping variable and renders a publication-ready plot with confidence ribbons, log-rank p-value, and an optional at-risk table below the main panel. Degrades gracefully — returns an annotated blank plot when fewer than two groups are present.

Usage

```
km_panel_from_df(
  df,
  group_col = "CAlevel",
  title = "",
  horizon_years = NULL,
  risk_table = TRUE,
  risk_times = NULL,
  risk_table_height = 0.28,
  risk_text_size = 4,
  base_size = 16,
  title_size = 22,
  axis_title_size = 14,
  axis_text_size = 12,
  legend_title_size = 14,
  legend_text_size = 12,
  bold = TRUE,
  group_colours = NULL
)
```

Arguments

`df`	A data frame containing at minimum `diagsurvtime` (numeric, years from diagnosis), `status` (integer, 0/1), and the column named in `group_col`. Column name matching for `group_col` is case-insensitive.
`group_col`	Character string. Name of the grouping variable column. Default `"CAlevel"`. When `group_col` is `"CAlevel"` and both `"High"` and `"Low"` levels are present, `ca_cols` from `constants.R` is used automatically for colouring.
`title`	Character string. Plot title. Default `""`.

horizon_years Numeric or NULL. If supplied, the x-axis is clipped to this value via coord_cartesian. Also used as the upper bound for risk_times when risk_times is NULL. Default NULL (full follow-up range).

risk_table Logical. If TRUE, an at-risk table is appended below the KM panel. Default TRUE.

risk_times Numeric vector or NULL. Time points at which at-risk counts are displayed. NULL auto-generates breakpoints from 0:horizon_years (if supplied) or pretty(range(diagsurvtime)). Default NULL.

risk_table_height Numeric. Relative height of the at-risk table panel as a fraction of the KM panel height, passed to patchwork::plot_layout(). Default 0.28.

risk_text_size Numeric. Font size for at-risk count labels. Default 4.

base_size Base font size (pt) for the KM panel. Default 16.

title_size Font size (pt) for the plot title. Default 22.

axis_title_size Font size (pt) for axis titles. Default 14.

axis_text_size Font size (pt) for axis tick labels. Default 12.

legend_title_size Font size (pt) for the legend title. Default 14.

legend_text_size Font size (pt) for legend item labels. Default 12.

bold Logical. If TRUE all text elements use bold weight. Default TRUE.

group_colours Named character vector mapping group levels to hex colours. Must be named with the levels of group_col. NULL auto-generates: uses ca_cols for CAlevel, otherwise cycles through a built-in 8-colour palette. Default NULL.

Details

Column requirements: diagsurvtime and status are hardcoded. status must be coded 0 (censored) / 1 (event). Rename columns upstream if necessary.

Time-zero rows: The function ensures a survival estimate of 1.0 at time 0 exists for every group before plotting, preventing stepped curves that start below 1.

Log-rank p-value: Computed via survival::survdiff() with 1 degree of freedom. Displayed in the subtitle as signif(p, 3). NA is shown if the chi-square statistic is non-finite.

Colour precedence: group_colours argument > ca_cols auto-detection > built-in palette.

Consistent return type: Always returns a patchwork object so downstream code assembling multi-panel figures does not need to branch on risk_table.

Value

A patchwork object in all cases. When risk_table = TRUE, the KM panel (top) and at-risk table (bottom) are combined via patchwork::plot_layout(). When risk_table = FALSE, the KM panel is wrapped in patchwork::wrap_plots() for a consistent return type. Returns an annotated blank ggplot if fewer than two groups are present in group_col.

See Also

cox_forest_plot_from_df, tx_pooled_analysis, tx_compare_groups

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                    = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact          = spec_ages[i] + 3,
    diagsurvtime              = 3,
    Status                    = i %% 2L,
    Medication                = c('DrugA','DrugB','DrugC'),
    treatment_group           = tx_types[[i]],
    AgeAtMedStart             = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop              = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod   = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors          = FALSE
  )
}))
meta <- data.frame(
  sample       = paste0('P', seq_len(n)),
  diagsurvtime = rep(3, n),
  Status       = seq_len(n) %% 2L,
  CAlevel      = rep(c('High','Low'), n/2),
  stringsAsFactors = FALSE
)
norm        <- tx_normalize(med_data)
cluster_res <- tx_cluster_surv(meta, norm, k_range = 2,
                               umap_neighbors = 5,
                               min_feature_variance = 0)
p <- km_panel_from_df(
  df        = cluster_res$Cluster_surv,
  group_col = 'CAlevel',
  title     = 'KM by CAlevel'
)
class(p)
```

plot_timeline_for_k *Treatment Timeline Facet Plot by Cluster Assignment*

Description

Renders a faceted swimlane plot of patient treatment timelines, with one facet per cluster. Each horizontal segment represents a treatment interval coloured by type. Patients within each facet are ordered by earliest treatment start or dominant treatment share. Optionally saves to disk.

Usage

```
plot_timeline_for_k(
  kc,
  metadata,
  segs,
  out_file = NULL,
  ncols = 3,
  horizon_years = 6,
  base_size = 14,
  title_size = 20,
  axis_title_size = 13,
  axis_text_size = 11,
  legend_title_size = 12,
  legend_text_size = 11,
  bold = TRUE,
  order_by = c("first_start", "dominant_share")
)
```

Arguments

kc	Character string. Name of the cluster assignment column in metadata (e.g. "Cluster_k3"). Used both to facet the plot and to label each panel as c1 (n=X), c2 (n=X), etc.
metadata	A data frame containing at minimum sample and the column named in kc. Patients with NA in kc are silently dropped. Typically Cluster_surv (LUSC) or LUAD_metadata (LUAD).
segs	A segments data frame passed to prep_segs for recoding and horizon clipping. Must contain columns sample, type, t0, and t1.
out_file	Character string or NULL. If supplied, the plot is saved to this path via pdf() at 20 – 12 inches. Always use a .pdf extension — png() requires X11/Cairo which is unavailable on OSC. If NULL, the plot is returned invisibly without saving. Default NULL.
ncols	Integer. Number of columns in the facet_wrap layout. Default 3.
horizon_years	Numeric. Maximum x-axis extent in years, passed to prep_segs. Segments extending beyond this value are clipped. Default 6.

base_size	Base font size (pt). Default 14.
title_size	Font size (pt) for the plot title. Default 20.
axis_title_size	Font size (pt) for axis titles. Default 13.
axis_text_size	Font size (pt) for axis tick labels. Default 11.
legend_title_size	Font size (pt) for the legend title. Default 12.
legend_text_size	Font size (pt) for legend item labels. Default 11.
bold	Logical. If TRUE all text elements use bold weight. Default TRUE.
order_by	One of "first_start" or "dominant_share". Controls row ordering within each cluster facet. "first_start" orders by earliest treatment start time (ascending). "dominant_share" orders by the proportion of total treated time accounted for by the single most common treatment type (descending), with earliest start as a tiebreaker. Default "first_start".

Details

Colour palette: Treatment type colours are defined locally inside the function (matching tx_cols in constants.R) and do not depend on any global object. Only types present in the data are included in the legend.

Facet labels: Each panel is labelled cN (n=X) where N is the numeric cluster index extracted from the kc column value and X is the patient count. Non-numeric cluster values are used as-is.

Y-axis: Patient identity labels are suppressed on the y-axis for readability. Row position within each facet reflects order_by only.

Value

The ggplot object, returned invisible(). If out_file is supplied the plot is also written to disk via pdf() and a message is printed. Returns invisible(NULL) with a warning if no segments remain after joining to metadata.

See Also

prep_segs, timeline_panel, tx_cluster_surv

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
```

```
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                     = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact           = spec_ages[i] + 3,
    diagsurvtime               = 3,
    Status                     = i %% 2L,
    Medication                 = c('DrugA','DrugB','DrugC'),
    treatment_group            = tx_types[[i]],
    AgeAtMedStart              = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop               = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod    = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors           = FALSE
  )
}))
meta <- data.frame(
  sample       = paste0('P', seq_len(n)),
  diagsurvtime = rep(3, n),
  Status       = seq_len(n) %% 2L,
  CAlevel      = rep(c('High','Low'), n/2),
  stringsAsFactors = FALSE
)
norm        <- tx_normalize(med_data)
intervals   <- tx_intervals(norm)
segs        <- prep_segs(intervals)
cluster_res <- tx_cluster_surv(meta, norm, k_range = 2,
                               umap_neighbors = 5,
                               min_feature_variance = 0)
p <- plot_timeline_for_k(
  kc       = 'Cluster_k2',
  metadata = cluster_res$Cluster_surv,
  segs     = segs
)
class(p)
```

`prep_segs` *Recode and clip treatment segments to canonical types Validates a raw segments data frame, recodes free-text treatment type labels to the eight canonical MEC-TX types via regex, and clips segment start/end times to* `[0, horizon_years]`. *Zero-duration segments (after clipping) and rows with* `NA` *in* `sample`, `t0`, *or* `t1` *are dropped. Called upstream by* `plot_timeline_for_k()` *and* `timeline_panel()` *— not intended for direct use.*

Description

Recode and clip treatment segments to canonical types Validates a raw segments data frame, recodes free-text treatment type labels to the eight canonical MEC-TX types via regex, and clips segment start/end times to `[0, horizon_years]`. Zero-duration segments (after clipping) and rows

with NA in sample, t0, or t1 are dropped. Called upstream by plot_timeline_for_k() and timeline_panel() — not intended for direct use.

Usage

```
prep_segs(segs, horizon_years = 5)
```

Arguments

segs	A data frame with at minimum columns sample, type (free-text treatment label), start_year, and end_year (numeric, years since first treatment).
horizon_years	Numeric. Maximum time horizon in years. Segment endpoints are clipped to [0, horizon_years] via pmin/ pmax. Default 5.

Details

Regex priority: Rules are applied in case_when() order — first match wins. Order: Radiation — IO — Chemo — Targeted — Hormone — Small_Molecule — Ancillary — Others. All patterns are case-insensitive. Types already matching a canonical label pass through correctly since the regexes cover the canonical names themselves. **Local valid types:** The canonical type vector is defined internally rather than referencing valid_types from constants.R, avoiding a global scope dependency (Bug 4.2 fix).

Value

A data frame with the same rows as segs (minus dropped rows) and two new columns: type (recoded canonical label, one of "Radiation", "IO", "Chemo", "Targeted", "Hormone", "Small_Molecule", "Ancillary", "Others") and t0 / t1 (clipped start and end times in years). The original start_year / end_year columns are retained alongside t0 / t1.

---

standardise_status *Standardise Survival Status Column to 0/1 Integer*

Description

Detects the coding scheme of a survival status column and converts it to the MEC-TX convention: 0 = alive/censored, 1 = dead/event (integer). Adds a status_label factor column for display use in tables and plots. Issues an informative message reporting what was detected and how it was mapped.

Usage

```
standardise_status(df, status_col = "status")
```

Arguments

df	A data frame containing a survival status column.
status_col	Character string. Name of the status column to standardise. Case-sensitive. Default "status".

Details

Detection logic: Two coding paths are handled:

Numeric 0/1 If the column is numeric and all non-NA values are in {0, 1}, no recoding is performed — the column is coerced to integer, renamed to "status", and status_label is added.

Character / factor Values are lowercased and matched against known dead patterns ("dead", "deceased", "died", "death", "1") and alive patterns ("alive", "living", "censored", "censor", "0"). Any value matching a dead pattern maps to 1; all others map to 0.

If no known patterns are detected the function stops with an informative error listing the values found, prompting manual recoding upstream.

Column rename: The output column is always named "status" (lowercase) regardless of status_col. This ensures compatibility with all downstream MEC-TX functions which hardcode "status" in survival::Surv() calls.

AVATAR-specific note: The LUSC Cluster_surv object uses lowercase "status" while LUAD LUAD_metadata uses capitalised "Status". Pass status_col = "Status" for LUAD — the output column will be renamed to lowercase "status" automatically.

status_label warning: status_label is a convenience column for display only. Never pass it to survival::Surv() — this will produce incorrect results.

Value

The input data frame with two modifications:

status The status column standardised to integer 0/1 and renamed to "status" regardless of the input status_col name. Safe to pass directly to survival::Surv(). Convention: 0 = alive/censored, 1 = dead/event.

status_label A new factor column with levels c("Alive", "Dead") added for use in plots, tables, and summaries. Do not use this column in survival::Surv() — use the integer status column instead.

See Also

tx_normalize, resolve_col

Examples

```
df <- data.frame(sample = 1:3, diagsurvtime = c(1.2, 3.4, 2.1),
                 status = c(0, 1, 1))
standardise_status(df)
```

`timeline_panel` *Treatment Timeline Panel for Selected Patients*

Description

Renders a horizontal swimlane plot showing treatment type over time for a specified set of patients (e.g. cluster representatives or "top twins"). Each patient occupies one row; treatment types are drawn as coloured horizontal segments with a small vertical offset per type to reduce overplotting. Row order is driven by a focus-type share score.

Usage

```
timeline_panel(
  segs_prepped,
  share_df,
  twin_ids,
  title = "Top twins",
  horizon_years = 5,
  focus_types = NULL,
  base_size = 16,
  title_size = 22,
  axis_title_size = 14,
  axis_text_size = 12,
  legend_title_size = 14,
  legend_text_size = 12,
  bold = TRUE
)
```

Arguments

`segs_prepped`	A pre-processed segments data frame produced by `prep_segs`. Must contain columns `sample`, `type`, `t0`, and `t1` (treatment start and end in years since first treatment).
`share_df`	A data frame with one row per patient containing `sample` and any number of `share_*_tx` columns (treatment type duration shares) plus `dur_treated`. Typically the `$shares` slot from `treatment_shares`.
`twin_ids`	Character vector of patient `sample` identifiers to display. Rows in `segs_prepped` not in `twin_ids` are silently dropped.
`title`	Character string. Plot title. Default `"Top twins"`.
`horizon_years`	Numeric. Maximum x-axis extent in years. Default `5`.
`focus_types`	Character vector or `NULL`. Treatment type(s) used to compute the row ordering score. Patients with higher combined share of these types appear at the top. Must match type labels used in `share_*_tx` column names (e.g. `"Radiation"` maps to `share_Radiation_tx`). `NULL` falls back to the first `share_*_tx` column found.

base_size	Base font size (pt). Default 16.
title_size	Font size (pt) for the plot title. Default 22.
axis_title_size	Font size (pt) for axis titles. Default 14.
axis_text_size	Font size (pt) for axis tick labels. Default 12.
legend_title_size	Font size (pt) for the legend title. Default 14.
legend_text_size	Font size (pt) for legend item labels. Default 12.
bold	Logical. If TRUE all text elements use bold weight. Default TRUE.

Details

Colour palette: Treatment type colours are defined locally inside the function (matching tx_cols in constants.R) and do not depend on any global object. Only types present in the data are included in the legend.

Row ordering: The ordering score is the row-wise sum of all focus_share_cols shares per patient. Ties are broken by the factor level order of sample.

Vertical offset: Each treatment type within a patient row is offset by 0.10 * (type_idx - mean(type_idx)) on the y-axis to visually separate overlapping segments without displacing the row label.

Value

A ggplot object. Print to display or save with pdf() / ggsave().

See Also

prep_segs, treatment_shares, plot_timeline_for_k

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                    = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact          = spec_ages[i] + 3,
    diagsurvtime              = 3,
    Status                    = i %% 2L,
```

```
    Medication                = c('DrugA','DrugB','DrugC'),
    treatment_group           = tx_types[[i]],
    AgeAtMedStart             = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop              = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod   = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors          = FALSE
  )
}))
norm      <- tx_normalize(med_data)
intervals <- tx_intervals(norm)
segs      <- prep_segs(intervals)
sh        <- treatment_shares(segs)
p <- timeline_panel(segs, sh, twin_ids = unique(intervals$sample))
class(p)
```

`treatment_shares` *Compute per-patient treatment type durations and shares Takes a clipped and recoded segments data frame (output of* `prep_segs`*) and computes per-patient treatment duration and share for each of the eight canonical treatment types. Returns a wide-format tibble with one row per patient. Called upstream by* `timeline_panel` *and* `plot_timeline_for_k` *— not intended for direct use.*

Description

Compute per-patient treatment type durations and shares Takes a clipped and recoded segments data frame (output of `prep_segs`) and computes per-patient treatment duration and share for each of the eight canonical treatment types. Returns a wide-format tibble with one row per patient. Called upstream by `timeline_panel` and `plot_timeline_for_k` — not intended for direct use.

Usage

```
treatment_shares(segs_prepped)
```

Arguments

`segs_prepped`	A data frame produced by `prep_segs`. Must contain columns `sample`, `type` (canonical label), `t0`, and `t1` (clipped segment start/end in years).

Details

Input requirement: `segs_prepped` must be the output of `prep_segs` — specifically it must contain `t0` and `t1` columns (clipped start/end times). Passing raw `timeline_long_intv` directly will error because `start_year` / `end_year` are not the same as `t0` / `t1`. **Local valid types:** The canonical type vector is defined internally rather than referencing `valid_types` from `constants.R`, avoiding a global scope dependency (Bug 4.2 fix). **Share calculation:** Shares are computed as a proportion of `dur_treated` (non-ancillary, non-None duration), not `dur_all`. A patient with only `"None"`

segments gets share_*_tx = 0 for all types. **Dominant type:** dom_type_tx uses max.col() with ties.method = "first" on the share matrix. Column order follows local_valid_types (Bug 4.3 fix — previously used unreliable column name inference).

Value

A wide-format tibble with one row per patient and the following columns:

sample Patient identifier.

Ancillary, Chemo, Hormone, IO, Small_Molecule, Targeted, Radiation, Others Total duration (years) for each treatment type within the clipped horizon. Zero-filled for types not present.

dur_all Total duration across all types including "None".

dur_treated Total duration across the eight canonical treatment types (excludes "None").

share_*_tx Duration share for each canonical type as a proportion of dur_treated. Zero when dur_treated == 0. Column names follow the pattern share_<Type>_tx (e.g. share_Chemo_tx).

dom_type_tx Character. The canonical type with the highest share for this patient. Ties broken by first match in column order.

---

tx_cluster_surv *Cluster Treatment Timelines and Attach Survival Data*

Description

Takes normalised treatment segments and patient metadata, builds a binary treatment feature matrix, performs PCA dimensionality reduction, clusters patients using k-means (k = 3 to 20 by default), and attaches survival and covariate data to the cluster assignments. UMAP is computed for visualisation only — clustering runs in PCA space.

Usage

```
tx_cluster_surv(
  metadata,
  timeline_long_norm,
  sample_col = "sample",
  time_col = "TimeSinceTreatmentStart",
  tx_col = "treatment_group",
  surv_time_col = "diagsurvtime",
  status_col = "status",
  meta_keep = NULL,
  horizon_years = 5,
  grid_weeks = 4,
  include_none = TRUE,
  drop_none_cols = TRUE,
  min_feature_variance = 0.01,
  seed = 42,
  n_pcs = 50,
```

```
  umap_neighbors = 30,
  umap_min_dist = 0.3,
  k_range = 3:20,
  kmeans_nstart = 50
)
```

Arguments

metadata	A data frame containing patient-level metadata. Must contain columns named in sample_col, surv_time_col, and status_col (case-insensitive). Typically Cluster_surv (LUSC) or LUAD_metadata (LUAD).
timeline_long_norm	A long-format normalised treatment timeline — the direct output of tx_normalize. Must contain columns named in sample_col, time_col, and tx_col (case-insensitive).
sample_col	Character string. Name of the patient identifier column in both metadata and timeline_long_norm. Case-insensitive. Default "sample".
time_col	Character string. Name of the treatment time column in timeline_long_norm. Case-insensitive. Default "TimeSinceTreatmentStart".
tx_col	Character string. Name of the treatment group column in timeline_long_norm. Case-insensitive. Default "treatment_group".
surv_time_col	Character string. Name of the survival time column (years from diagnosis) in metadata. Case-insensitive. Default "diagsurvtime".
status_col	Character string. Name of the survival status column in metadata. Auto-detected and standardised to integer 0/1 via standardise_status. Case-insensitive. Default "status".
meta_keep	Character vector or NULL. Additional metadata columns to carry through to the output Cluster_surv data frame. NULL retains all columns. Default NULL.
horizon_years	Numeric. Maximum follow-up horizon in years. Time bins beyond this value are excluded. Must match the horizon_years used in tx_normalize. Default 5.
grid_weeks	Numeric. Time bin width in weeks. Must match the grid_weeks used in tx_normalize. Common values: 1 (weekly), 2 (biweekly), 4 (monthly, default). Default 4.
include_none	Logical. If TRUE, time bins with no recorded treatment are filled with "None" before encoding. Default TRUE.
drop_none_cols	Logical. If TRUE, binary feature columns corresponding to "None" treatments are removed before PCA, reducing noise from untreated time bins. Default TRUE.
min_feature_variance	Numeric. Near-zero-variance threshold. Feature columns with variance at or below this value are dropped before PCA. Default 0.01.
seed	Integer. Random seed for reproducibility of PCA, UMAP, and k-means. Default 42.

n_pcs	Integer. Maximum number of principal components to retain for clustering and UMAP. Capped at the actual number of PCs available. Default 50.
umap_neighbors	Integer. Number of neighbours for UMAP. Must be strictly less than the number of patients — use min(30, n_patients - 1) as a rule of thumb. Default 30.
umap_min_dist	Numeric. Minimum distance parameter for UMAP layout. Smaller values produce tighter clusters visually. Default 0.3.
k_range	Integer vector. Range of k values for k-means clustering. All values must be >= 2 and < n_patients. Default 3:20.
kmeans_nstart	Integer. Number of random starts for k-means. Higher values improve stability at the cost of runtime. Default 50.

Details

Pipeline:

1. Resolve column names case-insensitively via resolve_col.
2. Standardise status column via standardise_status.
3. Build time-bin matrix: bin TimeSinceTreatmentStart into grid_weeks-wide bins up to horizon_years.
4. Multi-hot encode treatment types per bin (handles "+"-separated combination regimens).
5. Remove near-zero-variance features (var <= min_feature_variance).
6. PCA on the NZV-filtered binary matrix (scale. = FALSE).
7. UMAP on PCA scores — visualisation only.
8. K-means on PCA scores for each k in k_range.
9. Merge cluster assignments with survival metadata.

Clustering vs UMAP: K-means runs in PCA space, not UMAP space. UMAP is computed solely for visualisation. Do not use UMAP coordinates as clustering input.

Parameter alignment: grid_weeks and horizon_years must match the values used in the upstream tx_normalize call. Mismatches produce incorrect time bin alignment.

Sample audit: A message block is printed reporting sample counts at each pipeline stage — metadata input, timeline encoding, clustering, and survival merge. Samples lost at each stage are identified by ID.

Duplicate handling: Duplicate sample IDs in metadata produce a warning; the first occurrence is retained.

Value

A named list with eight elements:

Cluster_surv Tibble. One row per patient with cluster assignments for every k in k_range (Cluster_k3, Cluster_k4, ...), plus diagsurvtime, status (integer 0/1), status_label (factor "Alive"/"Dead"), and any columns in meta_keep. This is the primary output for downstream survival analyses.

cluster_results Tibble. Cluster assignments only — one row per patient, all k columns, no survival data.

umap_df Data frame with columns `sample`, `UMAP1`, `UMAP2`. Visualisation only — not used in clustering.

pca_matrix Numeric matrix. PCA scores for the first `n_pcs` components. Rownames are sample IDs. This is the space in which clustering is performed.

pca_var_explained Named numeric vector. Proportion of variance explained by each retained PC.

X Numeric matrix. NZV-filtered binary feature matrix used as PCA input. Rownames are sample IDs, column names are `<TimeBin>_<TreatmentType>`.

treatment_encoded Tibble. Wide-format multi-hot encoded treatment matrix before NZV filtering.

treatment_matrix_ordered Tibble. Wide-format treatment matrix with time bins as columns, ordered chronologically.

See Also

`tx_normalize`, `standardise_status`, `tx_pooled_analysis`, `km_panel_from_df`, `plot_timeline_for_k`

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                    = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact          = spec_ages[i] + 3,
    diagsurvtime              = 3,
    Status                    = i %% 2L,
    Medication                = c('DrugA','DrugB','DrugC'),
    treatment_group           = tx_types[[i]],
    AgeAtMedStart             = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop              = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod   = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors          = FALSE
  )
}))
meta <- data.frame(
  sample       = paste0('P', seq_len(n)),
  diagsurvtime = rep(3, n),
  Status       = seq_len(n) %% 2L,
  CAlevel      = rep(c('High','Low'), n/2),
  stringsAsFactors = FALSE
```

```
)
norm <- tx_normalize(med_data)
res  <- tx_cluster_surv(meta, norm, k_range = 2,
                        umap_neighbors = 5, min_feature_variance = 0)
head(res$Cluster_surv)
```

`tx_compare_groups` *Flexible Survival Comparison by Grouping Variable or Custom Groups*

Description

Produces a paired Kaplan-Meier panel and adjusted Cox forest plot for any categorical grouping variable or user-defined sample ID lists. Designed to work with any column in the `Cluster_surv` output — not limited to `CAlevel`. Returns plots, tidy Cox results, and a group summary table.

Usage

```
tx_compare_groups(
  df,
  group_var = NULL,
  custom_groups = NULL,
  ref_level = NULL,
  horizon_years = 5,
  cox_covars = c("stage_group", "sex", "age", "smokingstatus"),
  ref_levels = NULL,
  numeric_scale = list(age = 5),
  numeric_units = list(age = "years"),
  min_epv = 5,
  title = NULL,
  group_colours = NULL,
  risk_table = TRUE,
  show_forest = TRUE,
  base_size = 14,
  title_size = 16,
  widths = c(1, 1)
)
```

Arguments

`df`	A data frame with survival data. Must contain `diagsurvtime` (numeric, years) and `status` (integer 0/1) columns (case-insensitive). Typically the `$Cluster_surv` slot from `tx_cluster_surv` or the `$df` slot from `tx_pooled_analysis`.
`group_var`	Character string or `NULL`. Name of the grouping column in `df`. Matched case-insensitively. Ignored if `custom_groups` is also supplied (with a message). `NULL` requires `custom_groups` to be provided. Default `NULL`.

custom_groups	Named list or NULL. Each element is a character vector of sample IDs defining a group. Must have at least two named elements. Sample IDs not found in df are dropped with a warning. If both group_var and custom_groups are supplied, custom_groups takes precedence. Default NULL.
ref_level	Character string or NULL. Reference level for group_var in the Cox model and KM plot. NULL uses the first factor level. Default NULL.
horizon_years	Numeric. X-axis clip for KM plot and risk table, in years. Default 5.
cox_covars	Character vector. Adjustment covariates for the Cox model, in addition to group_var. Columns not present in df are silently dropped. Default c("stage_group", "sex", "age", "smokingstatus").
ref_levels	Named list. Reference levels for all Cox covariates. NULL auto-sets group_var reference to its first factor level. Default NULL.
numeric_scale	Named list. Divisors for numeric covariates in the Cox model. Default list(age = 5) (HR per 5-year increment) — intentionally smaller than the package-wide default of 10, since group comparison cohorts may be subsets with reduced power.
numeric_units	Named list. Unit labels for numeric covariates in the forest plot. Default list(age = "years").
min_epv	Integer. Minimum events-per-variable for Cox covariate selection. Passed to cox_forest_plot_from_df. Default 5.
title	Character string or NULL. Plot title. NULL auto-generates "Survival by <group_var> (n=<N>)". Default NULL.
group_colours	Named character vector or NULL. Hex colours for each group level. NULL auto-generates: uses ca_cols for CAlevel, cycles through a 10-colour palette for up to 10 levels, falls back to grDevices::hcl.colors() for larger groupings. Default NULL.
risk_table	Logical. If TRUE, an at-risk table is appended below the KM panel. Passed to km_panel_from_df. Default TRUE.
show_forest	Logical. If TRUE, the Cox forest plot is computed and included in combined. Default TRUE.
base_size	Base font size (pt). Default 14.
title_size	Font size (pt) for plot titles. Default 16.
widths	Numeric vector of length 2. Relative widths of the KM and forest panels in the combined layout, passed to patchwork::plot_layout(). Default c(1, 1).

Details

Group resolution: When group_var is supplied, it is matched case-insensitively to column names in df. When custom_groups is supplied, a temporary "custom_group" column is built via an inner join on sample, restricting df to only the listed patients.

Cox model: group_var is always included as the first covariate regardless of EPV. Additional covariates from cox_covars are selected by EPV budget. The Cox model is refitted cleanly for the cox_results slot — independently of the forest plot.

Colour fallback: For group_var = "CAlevel" with High/Low levels, ca_cols from constants.R is used automatically. For more than 10 group levels, grDevices::hcl.colors() is used with a message.

Value

A named list with nine elements:

km A patchwork object — KM panel with optional risk table.

forest A ggplot object — Cox forest plot. NULL if show_forest = FALSE.

combined A patchwork object — KM and forest side by side. Equal to km alone if show_forest = FALSE.

cox_results Tibble from broom::tidy() with columns term, estimate (HR), conf.low, conf.high, p.value. NULL if Cox model fails.

group_summary Tibble with one row per group level: n, n_events, median_surv_years.

df The analysis data frame after group column resolution, NA filtering, and factor relevel — useful for downstream inspection.

group_var Character. The resolved (actual) group column name used.

grp_levels Character vector of group factor levels in plot order.

n Integer. Number of patients in the analysis.

Returns a list of NULLs (with a warning) if group_var has fewer than two non-NA levels after filtering.

See Also

tx_pooled_analysis, km_panel_from_df, cox_forest_plot_from_df, tx_cluster_surv

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                    = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact          = spec_ages[i] + 3,
    diagsurvtime              = 3,
    Status                    = i %% 2L,
    Medication                = c('DrugA','DrugB','DrugC'),
    treatment_group           = tx_types[[i]],
    AgeAtMedStart             = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop              = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod   = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors          = FALSE
```

```
  )
}))
meta <- data.frame(
  sample       = paste0('P', seq_len(n)),
  diagsurvtime = rep(3, n),
  Status       = seq_len(n) %% 2L,
  CAlevel      = rep(c('High','Low'), n/2),
  stringsAsFactors = FALSE
)
norm        <- tx_normalize(med_data)
cluster_res <- tx_cluster_surv(meta, norm, k_range = 2,
                               umap_neighbors = 5,
                               min_feature_variance = 0)
res <- tx_compare_groups(
  df        = cluster_res$Cluster_surv,
  group_var = 'CAlevel',
  title     = 'Overall Survival by CAlevel'
)
names(res)
```

`tx_duration` *Treatment Duration Analysis by Grouping Variable*

Description

Computes per-patient treatment duration (calendar time) broken down by treatment type and as a merged total. Compares groups using Wilcoxon rank-sum tests and produces box/violin plots.

Usage

```
tx_duration(
  timeline,
  meta,
  group_var,
  sample_col = "sample",
  type_col = "type",
  start_col = "start_year",
  end_col = "end_year",
  duration_unit = "months",
  exclude_types = NULL,
  min_n = 3L,
  plot = TRUE,
  plot_type = "box",
  title = NULL,
  palette = NULL
)
```

Arguments

timeline	data.frame — output of tx_intervals(). Must contain columns for sample, type, start_year, end_year.
meta	data.frame — patient-level metadata (e.g., Cluster_surv). Must contain sample_col and group_var.
group_var	Character — column name in meta for the grouping variable (e.g., "CAlevel").
sample_col	Character — patient ID column (default "sample").
type_col	Character — treatment type column (default "type").
start_col	Character — interval start column (default "start_year").
end_col	Character — interval end column (default "end_year").
duration_unit	Character — "months" (default) or "years". Controls the unit in output tables and plot axis labels.
exclude_types	Character vector — treatment types to exclude from analysis (e.g., c("Others")). Default NULL.
min_n	Integer — minimum patients per group – type to run a test (default 3). Types with fewer are flagged but kept.
plot	Logical — produce a plot? (default TRUE)
plot_type	Character — "box" (default) or "violin".
title	Character — plot title. NULL for auto-generated.
palette	Named character vector — colours keyed by group levels. NULL for automatic palette.

Value

A named list:

duration_per_type data.frame: sample, type, group_var, duration (in duration_unit)

duration_total data.frame: sample, group_var, duration_total

summary_table data.frame: type, group, n, mean, median, q25, q75, p_value (Wilcoxon)

plot ggplot object (or NULL if plot = FALSE)

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
```

```
  data.frame(
    sample                    = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact          = spec_ages[i] + 3,
    diagsurvtime              = 3,
    Status                    = i %% 2L,
    Medication                = c('DrugA','DrugB','DrugC'),
    treatment_group           = tx_types[[i]],
    AgeAtMedStart             = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop              = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod   = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors          = FALSE
  )
}))
meta <- data.frame(
  sample       = paste0('P', seq_len(n)),
  diagsurvtime = rep(3, n),
  Status       = seq_len(n) %% 2L,
  CAlevel      = rep(c('High','Low'), n/2),
  stringsAsFactors = FALSE
)
norm      <- tx_normalize(med_data)
intervals <- tx_intervals(norm)
res <- tx_duration(timeline = intervals, meta = meta,
                   group_var = 'CAlevel', plot = FALSE)
res$summary_table
```

`tx_focus_dt` *Digital Twin Focus Analysis for a Specific Treatment Type*

Description

Identifies patients ("twins") whose treatment history is dominated by one or more specified treatment types, then produces a three-panel composite figure: a treatment timeline swimlane, a Kaplan-Meier survival panel, and an adjusted Cox forest plot. Designed for deep inspection of a treatment-type-specific subgroup, optionally restricted to a single k-means cluster.

Usage

```
tx_focus_dt(
  Cluster_surv,
  segs,
  kc = "Cluster_k14",
  cl = NULL,
  focus_types = c("Radiation"),
  group_col = "CAlevel",
  horizon_years = 5,
  n_twins = 20,
```

```
  min_share_tx = 0.33,
  enforce_sequence = FALSE,
  seq_pattern = NULL,
  sequence_strict = FALSE,
  start_filter = c("all", "single_only", "combo_only"),
  pure_focus_only = FALSE,
  add_forest = TRUE,
  cox_covars = c("CAlevel", "stage_group", "sex", "age", "smokingstatus"),
 cox_ref_levels = list(CAlevel = "Low", stage_group = "Local", smokingstatus = "Never"),
  forest_min_epv = 1,
  forest_priority = c("CAlevel", "stage_group", "sex", "age", "smokingstatus"),
  forest_numeric_scale = list(age = 5),
  forest_numeric_units = list(age = "years"),
  forest_drop_stage_unknown = FALSE,
 forest_stage_unknown_levels = c("Unknown", "Unknown/Not Applicable", "Not Applicable"),
  km_risk_table = TRUE,
  km_risk_table_height = 0.26,
  show_km_legend = FALSE,
  base_size = 14,
  title_size = 16,
  widths = c(1.4, 1, 1)
)
```

Arguments

`Cluster_surv`	A data frame — the `$Cluster_surv` slot from `tx_cluster_surv`. Must contain `sample`, the column named in `kc`, and all columns in `cox_covars`.
`segs`	A data frame — the interval output of `tx_intervals`. Must contain `sample`, `type`, `start_year`, and `end_year`.
`kc`	Character string. Name of the cluster assignment column in `Cluster_surv` to use for subsetting (e.g. `"Cluster_k14"`). Default `"Cluster_k14"`.
`cl`	Integer or `NULL`. Cluster value to restrict twin selection to. `NULL` auto-selects the cluster with the highest median focus-type share across all patients. Default `NULL`.
`focus_types`	Character vector. Treatment type(s) defining the focus cohort. Must be canonical MEC-TX type labels. Patients are ranked by combined share of these types. Default `"Radiation"`.
`group_col`	Character string. Grouping variable for the KM and Cox panels. Matched case-insensitively. Default `"CAlevel"`.
`horizon_years`	Numeric. Timeline and KM x-axis extent in years. Default `5`.
`n_twins`	Integer. Maximum number of twins to display in the timeline and include in KM/Cox panels. Set to `999` to keep all. Default `20`.
`min_share_tx`	Numeric in `[0, 1]`. Minimum combined `focus_types` share threshold for twin selection. Automatically relaxed in 0.05 decrements if fewer than `n_twins` qualify at the initial threshold. Default `0.33`.

enforce_sequence
: Logical. If TRUE, only patients whose treatment sequence contains seq_pattern as a contiguous subsequence are included. Default FALSE.

seq_pattern
: Character vector or NULL. Subsequence to enforce when enforce_sequence = TRUE. NULL defaults to focus_types. Default NULL.

sequence_strict
: Logical. If TRUE, sequence matching requires strict temporal ordering of first occurrences (no re-ordering). If FALSE, uses has_subseq for contiguous subsequence matching. Default FALSE.

start_filter
: One of "all", "single_only", or "combo_only". Filters patients by what treatment type(s) they started with. "single_only" excludes patients who started two or more types simultaneously. "combo_only" requires at least two focus types to start together. Default "all".

pure_focus_only
: Logical. If TRUE, only patients whose entire treatment sequence consists exclusively of focus_types are included. Default FALSE.

add_forest
: Logical. If TRUE, an adjusted Cox forest plot is added as the third panel. Default TRUE.

cox_covars
: Character vector. Covariates for the Cox model. Default c("CAlevel", "stage_group", "sex", "age", "smokingstatus").

cox_ref_levels
: Named list. Reference levels for Cox covariates. Default list(CAlevel = "Low", stage_group = "Local", smokingstatus = "Never").

forest_min_epv
: Integer. Minimum EPV for Cox covariate selection in the forest panel. Default 1 (relaxed — twin cohorts are small).

forest_priority
: Character vector. Covariate priority order for EPV-based selection. Default matches cox_covars.

forest_numeric_scale
: Named list. Divisors for numeric covariates. Default list(age = 5).

forest_numeric_units
: Named list. Unit labels for numeric covariates. Default list(age = "years").

forest_drop_stage_unknown
: Logical. If TRUE, patients with stage values in forest_stage_unknown_levels are excluded from the Cox forest only (not from KM). Default FALSE.

forest_stage_unknown_levels
: Character vector. Stage values treated as unknown when forest_drop_stage_unknown = TRUE. Default c("Unknown", "Unknown/Not Applicable", "Not Applicable").

km_risk_table
: Logical. Show at-risk table below KM panel. Default TRUE.

km_risk_table_height
: Numeric. Relative height of risk table panel. Default 0.26.

show_km_legend
: Logical. If FALSE, the KM legend is suppressed to save space in the composite layout. Default FALSE.

base_size
: Base font size (pt) applied to all three panels. Default 14.

title_size
: Font size (pt) for panel titles. Default 16.

widths
: Numeric vector of length 3. Relative widths of the timeline, KM, and forest panels in the composite layout. Default c(1.4, 1, 1).

Details

Twin selection pipeline:

1. Compute treatment type shares via `treatment_shares`.
2. Restrict to the specified cluster (`cl`) or auto-select the cluster with highest median focus share.
3. Apply optional filters: `start_filter`, `pure_focus_only`, `enforce_sequence`.
4. Rank candidates by combined focus share, then by total treated time.
5. Apply `min_share_tx` dominance threshold, relaxing in 0.05 steps if fewer than `n_twins` qualify.
6. Take the top `n_twins` by focus share.

Accessing twin IDs: `attr(result, "twin_ids")` returns the sample IDs used in the figure without re-running the full pipeline.

Forest EPV: Twin cohorts are small by design. `forest_min_epv = 1` is the default to prevent all covariates from being dropped. Increase to 5 for larger cohorts.

Parameter naming note: `forest_numeric_scale` defaults to `list(age = 5)` here (per 5-year increment) rather than the package-wide default of `list(age = 10)`, reflecting that twin cohorts are smaller and the 5-year increment gives more estimable HRs.

Value

A `patchwork` object combining the three panels side by side. The selected twin sample IDs are attached as `attr(result, "twin_ids")` for downstream access without re-running the function.

See Also

`tx_cluster_surv`, `tx_intervals`, `timeline_panel`, `km_panel_from_df`, `cox_forest_plot_from_df`, `has_subseq`

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                    = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact          = spec_ages[i] + 3,
    diagsurvtime              = 3,
    Status                    = i %% 2L,
```

```
    Medication                = c('DrugA','DrugB','DrugC'),
    treatment_group           = tx_types[[i]],
    AgeAtMedStart             = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop              = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod   = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors          = FALSE
  )
}))
meta <- data.frame(
  sample       = paste0('P', seq_len(n)),
  diagsurvtime = rep(3, n),
  Status       = seq_len(n) %% 2L,
  CAlevel      = rep(c('High','Low'), n/2),
  stringsAsFactors = FALSE
)
norm        <- tx_normalize(med_data)
intervals   <- tx_intervals(norm)
segs        <- prep_segs(intervals)
cluster_res <- tx_cluster_surv(meta, norm, k_range = 2,
                               umap_neighbors = 5,
                               min_feature_variance = 0)
res <- tx_focus_dt(
  Cluster_surv = cluster_res$Cluster_surv,
  segs         = segs,
  kc           = 'Cluster_k2',
  focus_types  = c('Chemo'),
  n_twins      = 3
)
names(res)
```

`tx_intervals` *Convert Normalised Treatment Timeline to Treatment Intervals*

Description

Takes the long-format normalised treatment timeline produced by `tx_normalize` and converts it to a compact interval representation — one row per contiguous treatment run per patient per type. Intervals are defined on a regular time grid whose resolution must match the `grid_weeks` used in `tx_normalize`.

Usage

```
tx_intervals(
  df,
  id_col = "sample",
  time_col = "TimeSinceTreatmentStart",
  type_col = "treatment_group",
  drop_types = c("None"),
```

```
  horizon_years = 5,
  grid_weeks = 4
)
```

Arguments

df	A long-format data frame — the direct output of tx_normalize. Must contain columns named in id_col, time_col, and type_col.
id_col	Character string. Name of the patient identifier column. Default "sample".
time_col	Character string. Name of the time column (years since treatment start). Must be numeric. Default "TimeSinceTreatmentStart".
type_col	Character string. Name of the treatment type column. Default "treatment_group".
drop_types	Character vector. Treatment type labels to exclude before building intervals. Default c("None") drops untreated time bins. Set to character(0) to retain all types.
horizon_years	Numeric. Maximum follow-up horizon in years. Time points beyond this value are clamped to the horizon before interval construction. Must match the horizon_years used in tx_normalize. Default 5.
grid_weeks	Numeric. Time bin width in weeks. Must match the grid_weeks used in tx_normalize — mismatches produce incorrect interval boundaries. Common values: 1 (weekly), 2 (biweekly), 4 (monthly, default). Default 4.

Details

Grid alignment: Time values are snapped to the nearest grid point using round(t_year * grid_res) where grid_res = 52 / grid_weeks. This ensures interval boundaries align exactly with the bins created by tx_normalize.

Block vs run: A *block* groups all treatment types active within a contiguous set of grid points (no gap). A *run* is a contiguous sequence of the same treatment type within a block. The block structure captures treatment pauses; the run structure separates types within a block.

End-exclusive intervals: end_year is set to (max_grid_index + 1) / grid_res, making intervals end-exclusive. This convention is consistent with standard interval arithmetic and avoids zero-duration segments.

grid_weeks alignment: If grid_weeks does not match the value used in tx_normalize, interval boundaries will be misaligned and durations will be incorrect. Always pass the same value to both functions.

Grid alignment: Time values are snapped to the nearest grid point using round(t_year * grid_res) where grid_res = 52 / grid_weeks. This ensures interval boundaries align exactly with the bins created by tx_normalize.

Block vs run: A *block* groups all treatment types active within a contiguous set of grid points (no gap). A *run* is a contiguous sequence of the same treatment type within a block. The block structure captures treatment pauses; the run structure separates types within a block.

End-exclusive intervals: end_year is set to (max_grid_index + 1) / grid_res, making intervals end-exclusive. This convention is consistent with standard interval arithmetic and avoids zero-duration segments.

grid_weeks alignment: If `grid_weeks` does not match the value used in `tx_normalize`, interval boundaries will be misaligned and durations will be incorrect. Always pass the same value to both functions.

Value

A tibble with one row per contiguous treatment run and six columns:

sample Patient identifier.

type Treatment type label.

block Integer. Active-therapy block index per patient. A new block starts whenever there is a gap of one or more grid points with no recorded treatment of any type.

run Integer. Contiguous run index within each `sample -- block -- type` combination.

start_year Numeric. Interval start time in years (grid-aligned, inclusive).

end_year Numeric. Interval end time in years (grid-aligned, end-exclusive — i.e. the start of the next bin).

Returns a zero-row tibble with the same columns if no rows remain after filtering.

See Also

`tx_normalize`, `tx_duration`, `tx_lines`, `tx_pooled_analysis`, `dominant_exclusive`

Examples

```
med_data <- data.frame(
  sample                     = rep(c('P01','P02'), each = 3),
  Age.At.Specimen.Collection = rep(c(60, 65), each = 3),
  AgeAtLastContact           = rep(c(62, 67), each = 3),
  diagsurvtime               = rep(c(2, 2), each = 3),
  Status                     = rep(c(1L, 0L), each = 3),
  Medication                 = rep(c('DrugA','DrugB','DrugC'), 2),
  treatment_group            = c('Chemo','IO','Radiation',
                                 'Chemo','Targeted','Others'),
  AgeAtMedStart              = c(60.1, 60.5, 61.0, 65.1, 65.4, 66.0),
  AgeAtMedStop               = c(60.4, 60.9, 61.3, 65.3, 65.8, 66.2),
  AgeAtTreatmentStart.mod    = c(60.1, 60.5, 61.0, 65.1, 65.4, 66.0),
  stringsAsFactors           = FALSE
)
norm      <- tx_normalize(med_data)
intervals <- tx_intervals(norm)
head(intervals)
```

tx_lines *Detect Lines of Therapy from Treatment Timelines*

Description

Identifies lines of therapy for each patient by combining clinician-annotated MedLineRegimen labels (where available and non-Unknown) with a gap-based algorithm for unannotated records. Applies specimen-anchored record filtering to remove prior-cancer drug contamination from the AVATAR registry before any line detection.

Usage

```
tx_lines(
  timeline,
  annotations = NULL,
  meta = NULL,
  group_var = NULL,
  ann_id_col = "AvatarKey",
  ann_line_col = "MedLineRegimen",
  ann_start_col = "AgeAtMedStart",
  gap_threshold = 3/52,
  continuation_types = c("IO"),
  continuation_stages = c("I", "II", "III"),
  stage_col = NULL,
  exclude_types = NULL,
  specimen_age_col = "Age.At.Specimen.Collection",
  specimen_buffer = 0.25,
  filter_timeline = TRUE
)
```

Arguments

timeline	A data frame — the direct output of tx_intervals. Must contain columns sample, start_year, end_year, and type.
annotations	A data frame or NULL. The AVATAR Medication annotation file containing MedLineRegimen labels. Must contain columns named in ann_id_col, ann_line_col, and ann_start_col. NULL runs algorithm-only mode with no annotation coalescing. Default NULL.
meta	A data frame or NULL. Patient-level metadata (e.g. Cluster_surv or LUAD_metadata). Must contain sample. Used for: grouping variable join (group_var), stage extraction (stage_col), AvatarKey crosswalk for annotation joining, and specimen age for record-level filtering. Default NULL.
group_var	Character string or NULL. Column in meta for group comparison (e.g. "CAlevel"). If supplied, a Wilcoxon / Kruskal-Wallis comparison of n_lines and first_line_duration_months is computed between groups. Default NULL.
ann_id_col	Character string. Patient identifier column in annotations. Used for the AvatarKey — sample crosswalk via meta. Default "AvatarKey".

`ann_line_col` Character string. Line annotation column in `annotations`. Default `"MedLineRegimen"`.

`ann_start_col` Character string. Treatment start age column in `annotations` (years). Used for specimen-anchored filtering of annotation records. Default `"AgeAtMedStart"`.

`gap_threshold` Numeric. Gap in years above which consecutive treatment blocks are assigned to a new line. Default `3/52` (~3 weeks). PI-confirmed threshold for AVATAR data.

`continuation_types`
Character vector. Treatment types that may represent consolidation / adjuvant therapy rather than a true new line. IO-only lines containing only these types are flagged `"possible_consolidation"` in stage I—III patients. Default `"IO"`.

`continuation_stages`
Character vector. Stage values where consolidation flagging is applied. Default `c("I","II","III")`.

`stage_col` Character string or `NULL`. Column in `meta` containing cancer stage, used for consolidation flagging. `NULL` disables flagging. Default `NULL`.

`exclude_types` Character vector or `NULL`. Treatment types to remove from `timeline` before line detection. Recommended: `c("Ancillary", "Others")` to avoid short ancillary records triggering spurious line breaks. Default `NULL`.

`specimen_age_col`
Character string. Column in `meta` containing specimen collection age in years. Used for specimen- anchored record filtering in both `timeline` and `annotations`. Default `"Age.At.Specimen.Collection"`.

`specimen_buffer`
Numeric. Years before specimen collection date to allow medication records. Records with start age earlier than `specimen_age - specimen_buffer` are dropped. Default `0.25` (3 months) — covers the window where lung cancer treatment begins just before biopsy processing. Increase cautiously; large values risk re-introducing prior-cancer contamination.

`filter_timeline`
Logical. If `TRUE`, apply specimen-anchored filtering to `timeline` intervals. Set to `FALSE` when `timeline` uses relative time from `TimeSinceTreatmentStart` (cannot be compared to absolute specimen age). Annotation filtering always runs regardless of this flag. Default `TRUE`.

Details

Specimen-anchored record filtering: AVATAR captures all medications across a patient's lifetime, including drugs for prior cancers. Records are filtered at the individual record level — any record with start age earlier than `specimen_age - specimen_buffer` is dropped. This removes prior-cancer contamination (e.g. Letrozole for breast cancer) without excluding any patients from the cohort.

Coalesce logic: For each patient, if a non-Unknown `MedLineRegimen` annotation exists after filtering, it overrides the algorithm-computed label for line 1. Subsequent lines rely on the gap algorithm. Line 1 is the only line that can be annotation-coalesced — higher lines are always algorithm-derived.

Consolidation flagging: IO-only lines after line 1 in stage I—III patients are flagged `"possible_consolidation"` and must be reviewed by the PI before counting as true new therapy lines. These likely represent durvalumab consolidation or adjuvant pembrolizumab.

filter_timeline = FALSE: Use this when `timeline` was produced from `tx_intervals()` with relative `TimeSinceTreatmentStart` values. These cannot be compared to absolute `AgeAtMedStart` values and the filter would drop all records. Annotation filtering still runs.

Value

A named list with four elements:

lines Data frame with one row per patient – line. Columns: `sample`, `line_number`, `line_label`, `line_types`, `line_start`, `line_end`, `line_duration_months`, `line_source` (`"annotated"` or `"computed"`), `line_flag` (`"confirmed"` or `"possible_consolidation"`).

patient_summary Data frame with one row per patient. Columns: `sample`, `n_lines`, `max_line`, `first_line_label`, `first_line_types`, `first_line_duration_months`, `n_possible_consolidation`.

group_comparison Data frame of Wilcoxon / Kruskal-Wallis test results by `group_var` for `n_lines` and `first_line_duration_months`. `NULL` if `group_var` is not supplied.

params Named list recording all function settings for reproducibility.

Returns a list with an empty `lines` data frame and a warning if no lines are detected.

See Also

`tx_intervals`, `tx_duration`, `tx_pooled_analysis`, `dominant_exclusive`

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                     = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact           = spec_ages[i] + 3,
    diagsurvtime               = 3,
    Status                     = i %% 2L,
    Medication                 = c('DrugA','DrugB','DrugC'),
    treatment_group            = tx_types[[i]],
    AgeAtMedStart              = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop               = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod    = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors           = FALSE
  )
}))
```

```
meta <- data.frame(
  sample       = paste0('P', seq_len(n)),
  diagsurvtime = rep(3, n),
  Status       = seq_len(n) %% 2L,
  CAlevel      = rep(c('High','Low'), n/2),
  stringsAsFactors = FALSE
)
norm      <- tx_normalize(med_data)
intervals <- tx_intervals(norm)
res <- tx_lines(timeline = intervals, meta = meta, group_var = 'CAlevel')
head(res$patient_summary)
```

`tx_normalize` *Normalise Raw Medication Data into a Grid-Based Treatment Timeline*

Description

Converts raw per-medication-record data into a long-format treatment timeline on a regular time grid. Handles missing stop dates, clips exposure to specimen collection date, recodes non-canonical treatment group labels (PI-confirmed mappings), and optionally joins patient metadata and computes a per-patient dominant regimen assignment.

Usage

```
tx_normalize(
  med_data,
  metadata = NULL,
  grid_weeks = 4,
  dominant_regimen_share = 0.2
)
```

Arguments

`med_data`	A data frame of raw medication records. Must contain columns: `sample`, `AgeAtMedStart`, `AgeAtMedStop`, `Age.At.Specimen.Collection`, `AgeAtLastContact`, `Medication`, and `treatment_group`.
`metadata`	A data frame or `NULL`. Patient-level metadata joined to the timeline output. Must contain a `sample` column (case-insensitive). If supplied, `stage` and `status` columns are joined where present. `status` is automatically standardised to integer 0/1 via `standardise_status`. Default `NULL`.
`grid_weeks`	Numeric. Time bin width in weeks. Controls the resolution of the output timeline grid. Must match the value used in downstream calls to `tx_intervals` and `tx_cluster_surv`. Common values: 1 (weekly), 2 (biweekly), 4 (monthly, default). Default 4.

dominant_regimen_share
: Numeric in (0, 1]. Minimum treatment duration share threshold passed to dominant_exclusive for per-patient dominant regimen assignment. Default 0.20 (20%).

Details

Grid construction: Each medication record is expanded into a sequence of grid points from AgeAtTreatmentStart.mod to AgeAtMedStop at grid_weeks / 52 year intervals. TimeSinceTreatmentStart is then computed as the grid-snapped offset from each patient's earliest grid point.

Specimen clipping: Treatment exposure is clipped to begin no earlier than Age.At.Specimen.Collection. Records where AgeAtMedStart < Age.At.Specimen.Collection have their effective start shifted forward. This prevents pre-diagnosis drug exposure from contaminating the timeline.

Missing stop dates: AgeAtMedStop missing values are imputed as AgeAtLastContact, treating the drug as ongoing until last follow-up.

Treatment group recoding (PI-confirmed):

NA / "Other" — "Others"

"Onco_drug" + **Ado-Trastuzumab Emtansine** — "Targeted" (HER2-directed)

All other "Onco_drug" — "Others" (Ramucirumab, Rituximab, investigational agents)

Dominant regimen: A temporary interval table is built from the cleaned medication data and passed to dominant_exclusive to compute the per-patient dominant regimen. This column is used in Cox models stratified by regimen type.

Parameter alignment: grid_weeks must match the value used in tx_intervals and tx_cluster_surv. Mismatches produce incorrect time bin alignment downstream.

Value

A tibble in long format with one row per patient – grid time point – treatment type. Columns:

sample Patient identifier.

AgeGrid Numeric. Absolute age at each grid point (years).

treatment_group Character. Recoded canonical treatment type. One of the eight types in tx_cols from constants.R.

start_age Numeric. Age at first treatment grid point for this patient (used to compute TimeSinceTreatmentStart).

TimeSinceTreatmentStart Numeric. Time since first treatment in years, snapped to the grid. This is the primary time axis for all downstream analyses.

stage Character or NA. Joined from metadata if supplied.

status Integer 0/1 or NA. Joined and standardised from metadata if supplied.

end_followup Numeric. Age at last contact, joined from med_data for downstream truncation.

dominant_regimen Character. Per-patient mutually exclusive dominant regimen label from dominant_exclusive. Patients with no qualifying non-ancillary treatment are labelled "Ancillary/Supportive only".

See Also

tx_intervals, tx_cluster_surv, dominant_exclusive, standardise_status

Examples

```
med_data <- data.frame(
  sample                      = rep(c('P01','P02'), each = 3),
  Age.At.Specimen.Collection = rep(c(60, 65), each = 3),
  AgeAtLastContact            = rep(c(62, 67), each = 3),
  diagsurvtime                = rep(c(2, 2), each = 3),
  Status                      = rep(c(1L, 0L), each = 3),
  Medication                  = rep(c('DrugA','DrugB','DrugC'), 2),
  treatment_group             = c('Chemo','IO','Radiation',
                                  'Chemo','Targeted','Others'),
  AgeAtMedStart               = c(60.1, 60.5, 61.0, 65.1, 65.4, 66.0),
  AgeAtMedStop                = c(60.4, 60.9, 61.3, 65.3, 65.8, 66.2),
  AgeAtTreatmentStart.mod     = c(60.1, 60.5, 61.0, 65.1, 65.4, 66.0),
  stringsAsFactors            = FALSE
)
norm <- tx_normalize(med_data)
head(norm)
```

---

tx_pooled_analysis *Pooled Treatment Cohort Analysis*

Description

Builds a treatment-focused patient cohort using one of four selection modes, then produces a three-panel composite figure: a treatment timeline strip, a Kaplan-Meier survival panel, and an adjusted Cox forest plot. Returns all intermediate data objects for downstream inspection. Implements three-stage n_cohort transparency reporting.

Usage

```
tx_pooled_analysis(
  Cluster_surv,
  timeline,
  focus_types = c("Chemo", "IO"),
  mode = c("any", "only", "concurrent", "dominant"),
  group_var = "CAlevel",
  horizon_years = 5,
  min_share_tx = 0.33,
  concurrent_window = 4/52,
  n_twins = 999,
  enforce_sequence = FALSE,
  sequence_strict = FALSE,
  start_filter = "all",
```

```
  pure_focus_only = FALSE,
  cox_covars = c("stage_group", "sex", "age", "smokingstatus"),
  ref_levels = NULL,
  numeric_scale = list(age = 5),
  numeric_units = list(age = "years"),
  min_epv = 5,
  show_timeline = TRUE,
  group_colours = NULL,
  horizon_plot = NULL,
  base_size = 14,
  title_size = 16,
  widths = c(1.4, 1, 1)
)
```

Arguments

`Cluster_surv`	A data frame — the `$Cluster_surv` slot from `tx_cluster_surv`. Must contain `sample`, `diagsurvtime`, `status`, at least one `Cluster_kN` column, and the column named in `group_var`.
`timeline`	A data frame — the direct output of `tx_intervals`. Must contain `sample`, `type`, `start_year`, and `end_year`.
`focus_types`	Character vector. Treatment type(s) defining the cohort. Must be canonical MEC-TX type labels. Default `c("Chemo", "IO")`.
`mode`	One of `"any"`, `"only"`, `"concurrent"`, or `"dominant"`. Controls patient inclusion logic — see Details. Default `"any"`.
`group_var`	Character string. Grouping variable column in `Cluster_surv`. Matched case-insensitively. Default `"CAlevel"`.
`horizon_years`	Numeric. Analysis horizon in years used for timeline clipping and segment preparation. Default `5`.
`min_share_tx`	Numeric in `[0, 1]`. Minimum combined `focus_types` share for dominant mode. Passed to `tx_focus_dt`. Default `0.33`.
`concurrent_window`	Numeric. Maximum gap in years between two treatment intervals to still be classified as concurrent. Default `4/52` (4 weeks). A value of `0` means strict overlap only. Implements the PI-specified definition: a new treatment starting within 4 weeks of a prior treatment ending is classified as concurrent. Applied symmetrically — direction of the gap does not matter.
`n_twins`	Integer. Maximum twins per cluster for dominant mode. `999` retains all. Default `999`.
`enforce_sequence`	Logical. Passed to `tx_focus_dt` for dominant mode. Default `FALSE`.
`sequence_strict`	Logical. Passed to `tx_focus_dt` for dominant mode. Default `FALSE`.
`start_filter`	One of `"all"`, `"single_only"`, `"combo_only"`. Passed to `tx_focus_dt` for dominant mode. Default `"all"`.

pure_focus_only
: Logical. Passed to tx_focus_dt for dominant mode. Default FALSE.

cox_covars
: Character vector. Adjustment covariates for the Cox model. Columns absent from Cluster_surv are silently dropped. Default c("stage_group", "sex", "age", "smokingstatus").

ref_levels
: Named list or NULL. Reference levels for Cox covariates. NULL auto-sets group_var to its first factor level. Default NULL.

numeric_scale
: Named list. Divisors for numeric covariates in the Cox model. Default list(age = 5).

numeric_units
: Named list. Unit labels for numeric covariates in forest plot. Default list(age = "years").

min_epv
: Integer. Minimum events-per-variable for Cox covariate selection. Default 5.

show_timeline
: Logical. If TRUE, include the treatment timeline strip as the first panel. Default TRUE.

group_colours
: Named character vector or NULL. Default NULL.

horizon_plot
: Numeric or NULL. Default NULL.

base_size
: Base font size (pt). Default 14.

title_size
: Font size (pt) for panel titles. Default 16.

widths
: Numeric vector of length 3. Default c(1.4, 1, 1).

Details

Mode definitions:

"any" Patients who ever received ALL focus_types. No dominance filter — all n_raw patients used in KM and Cox.

"only" Patients who received ALL focus_types and no other treatment type (Ancillary allowed). No dominance filter — all n_raw patients used in KM and Cox.

"concurrent" Patients where all pairs of focus_types were administered within concurrent_window years of each other (default 4 weeks). No dominance filter — all n_raw patients used in KM and Cox.

"dominant" Patients whose dominant treatment type is one of focus_types. Dominance filter applied — only focus-type dominant patients used in KM and Cox.

Dominance filter: Applied ONLY for mode = "dominant". For "any", "only", and "concurrent", all patients passing the mode filter are included in KM and Cox, so n_cohort == n_raw for these three modes.

Value

A named list with fifteen elements: km, forest, timeline, ids, df, segs, shares, df_plot, mode, focus_types, group_var, n_cohort, n_raw, n_plot, group_table.

See Also

tx_cluster_surv, tx_intervals, km_panel_from_df, cox_forest_plot_from_df

Examples

```
set.seed(42)
n <- 6
spec_ages <- seq(55, 80, by = 5)
tx_types <- list(
  c('Chemo','IO','Radiation'),
  c('Chemo','Targeted','Others'),
  c('IO','Radiation','Chemo'),
  c('Targeted','Chemo','IO'),
  c('Radiation','Others','Chemo'),
  c('IO','Targeted','Chemo')
)
med_data <- do.call(rbind, lapply(seq_len(n), function(i) {
  data.frame(
    sample                     = paste0('P', i),
    Age.At.Specimen.Collection = spec_ages[i],
    AgeAtLastContact           = spec_ages[i] + 3,
    diagsurvtime               = 3,
    Status                     = i %% 2L,
    Medication                 = c('DrugA','DrugB','DrugC'),
    treatment_group            = tx_types[[i]],
    AgeAtMedStart              = spec_ages[i] + c(0.1, 0.5, 1.0),
    AgeAtMedStop               = spec_ages[i] + c(0.4, 0.9, 1.3),
    AgeAtTreatmentStart.mod    = spec_ages[i] + c(0.1, 0.5, 1.0),
    stringsAsFactors           = FALSE
  )
}))
meta <- data.frame(
  sample       = paste0('P', seq_len(n)),
  diagsurvtime = rep(3, n),
  Status       = seq_len(n) %% 2L,
  CAlevel      = rep(c('High','Low'), n/2),
  stringsAsFactors = FALSE
)
norm         <- tx_normalize(med_data)
intervals    <- tx_intervals(norm)
cluster_res  <- tx_cluster_surv(meta, norm, k_range = 2,
                                umap_neighbors = 5,
                                min_feature_variance = 0)
res <- tx_pooled_analysis(
  Cluster_surv = cluster_res$Cluster_surv,
  timeline     = intervals,
  focus_types  = c('Chemo', 'Radiation'),
  group_var    = 'CAlevel'
)
res$n_cohort
```

Index