Package ‘RFmstate’

September 9, 2026

Type Package

Title Random Forest-Based Multistate Survival Analysis

Version 0.1.9

Description Fits transition-specific cause-specific random survival forests on a clock-reset duration scale for acyclic, non-recurrent multistate processes. Entry-conditioned state-occupation probabilities are assembled from predicted cumulative hazards by semi-Markov entry-mass and sojourn convolution on a validated regular grid. The one-row-per-subject interface supports one common initial state, one recorded entry per state, baseline time-fixed covariates, competing exits, and independent right censoring. Left truncation, recurrent visits, directed cycles, time-dependent covariates, and ongoing-sojourn dynamic prediction are not supported. The package also provides calendar-time Aalen-Johansen point estimates as a covariate-free descriptive baseline, transition-specific permutation importance, genuine ranger edge OOB concordance, and patient-level cross-validated IPCW state-probability scoring. Methods are described in Ishwaran et al. (2008) <doi:10.1214/08-AOAS169> for random survival forests, Putter et al. (2007) <doi:10.1002/sim.2712> for multistate competing risks decomposition, and Aalen and Johansen (1978) <https://www.jstor.org/stable/4615704> for the nonparametric estimator.

License MIT + file LICENSE

URL https://github.com/EvalunaC/RFmstate

BugReports https://github.com/EvalunaC/RFmstate/issues

Encoding UTF-8

Depends R (>= 4.1.0)

Imports survival, ranger, stats, graphics, grDevices, utils

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

NeedsCompilation no
Author Yiqing Chen [aut, cre]
Maintainer Yiqing Chen <y.chen@tamu.edu>
Repository CRAN
Date/Publication 2026-09-09 18:30:02 UTC

Contents

RFmstate-package . . . 2
aalen_johansen . . . 3
clinical_states . . . 5
compute_trans_prob . . . 5
define_multistate . . . 7
diagnose . . . 9
head.msdata . . . 10
importance . . . 11
plot.aj_estimate . . . 13
plot.rfmstate_diag . . . 14
plot.rfmstate_importance . . . 15
plot.rfmstate_pred . . . 16
plot_transition_diagram . . . 17
predict.rfmstate . . . 18
prepare_data . . . 20
print.aj_estimate . . . 22
print.msdata . . . 24
print.mstate_structure . . . 24
rfmstate . . . 25
sim_clinical_data . . . 27
summary.rfmstate . . . 29

Index **31**

`RFmstate-package` *RFmstate: Random Forest-Based Multistate Survival Analysis*

Description

Fits transition-specific cause-specific random survival forests on a clock-reset duration scale for acyclic, non-recurrent multistate processes. Patient/profile state probabilities are assembled by semi-Markov convolution and are conditional on fresh state entry. The package supports a common initial state, one recorded entry per state, baseline time-fixed covariates, competing exits, and independent right censoring. It does not support left truncation, cycles/recurrent visits, time-dependent covariates, or ongoing-sojourn dynamic prediction. A strict predictor contract prevents IDs, event/censoring times, response fields, and arbitrary long-format columns from entering a forest. Every full-data or cross-validation fit learns its predictor schema only from its own fitting rows, and every reported OOB statistic has verified OOB coverage. A calendar-time Aalen-Johansen point estimator is provided as a covariate-free descriptive baseline. The package provides:

- State space and transition structure definition
- Wide-to-long data conversion for multistate counting processes
- Cause-specific random forest fitting per origin state
- Entry-conditioned state probabilities via semi-Markov convolution
- Aalen-Johansen point estimation (covariate-free baseline)
- Per-transition feature importance
- Genuine edge OOB concordance and patient-level cross-validated IPCW Brier scores
- Comprehensive visualizations

Author(s)

Maintainer: Yiqing Chen <y.chen@tamu.edu>

Authors:

- Yiqing Chen <y.chen@tamu.edu>

See Also

Useful links:

- https://github.com/EvalunaC/RFmstate
- Report bugs at https://github.com/EvalunaC/RFmstate/issues

`aalen_johansen` *Aalen-Johansen Point Estimator*

Description

Computes nonparametric estimates of transition probabilities using the Aalen-Johansen estimator via Nelson-Aalen cumulative hazard increments and product-integral construction.

Usage

```
aalen_johansen(msdata, s = 0)
```

Arguments

`msdata`	An `msdata` object from `prepare_data`.
`s`	Legacy numeric starting-time argument. Only `0` is accepted.

Details

The Aalen-Johansen estimator generalizes the Kaplan-Meier estimator to multistate models. RFmstate exposes it as a descriptive, covariate-free calendar-time population benchmark from the recorded common baseline.

The estimator uses calendar-time risk sets and is separate from the package's clock-reset semi-Markov forest solver. It is a covariate-free descriptive benchmark from the common baseline. Observed competing exits are transitions, not external censoring. No clipping or row normalization is applied.

Value

An object of class `"aj_estimate"` containing:

time Numeric vector of unique event times.

trans_prob List of calendar-time product-integral point-estimate matrices from the recorded common baseline at each event time.

state_occ Matrix of state occupation probabilities over time. Rows are time points, columns are states.

cum_hazard List of Nelson-Aalen cumulative hazard matrices.

hazard_inc List of hazard increment matrices at each event time.

n_risk Matrix of at-risk counts over time.

n_events Data frame of event counts per transition.

structure The multistate structure used.

s The starting time.

time_scale Calendar time from the common study origin.

uncertainty Statement that no validated variance or confidence interval is returned.

Limitations

Only the recorded common baseline `s = 0` is supported. The function does not support left truncation, general conditional $P(s, t)$ for `s > 0`, covariate adjustment, or validated variance/confidence bands. The point estimate should not be described as the covariate-free form of the clock-reset forest model.

Examples

```
ms <- clinical_states()
dat <- sim_clinical_data(n = 200, structure = ms, seed = 42)
msdata <- prepare_data(
  data = dat, id = "ID", structure = ms,
  time_map = list(
    Responded = "time_Responded",
    Unresponded = "time_Unresponded",
    Stabilized = "time_Stabilized",
    Progressed = "time_Progressed",
    Death = "time_Death"
```

```
  ),
  censor_col = "time_censored",
  covariates = c("age", "sex", "BMI", "treatment")
)
aj <- aalen_johansen(msdata)
print(aj)
```

`clinical_states` *Create Clinical Trial Multistate Structure*

Description

A convenience function that creates the standard clinical trial multistate structure with states: Baseline, Responded, Unresponded, Stabilized, Progressed, Death.

Usage

```
clinical_states()
```

Value

An `mstate_structure` object.

Limitations

This is a fixed demonstration DAG, not a claim that the package supports cyclic or recurrent clinical histories. Use `define_multistate()` for another supported single-root DAG.

Examples

```
ms <- clinical_states()
print(ms)
```

`compute_trans_prob` *Entry-Conditioned Semi-Markov State Probabilities*

Description

Combines one clock-reset cumulative cause-specific hazard curve per allowed edge using a regular-grid entry-mass/sojourn convolution. The result is not a general Markov $P(s,t)$ matrix: each row is conditional on fresh entry into its named starting state at elapsed duration zero.

Usage

```
compute_trans_prob(
  cum_hazards,
  structure,
  s = 0,
  times = NULL,
  start_state = NULL,
  grid_step = NULL,
  target_grid_points = 1024L,
  max_grid_points = 131073L,
  grid_tol = 5e-04,
  max_grid_refinements = 7L,
  check_grid = TRUE,
  prob_tol = 1e-08,
  extrapolate = c("error", "flat")
)
```

Arguments

`cum_hazards` Named list of data frames, one for every allowed edge. Names must be `"from->to"`; each frame contains sortable `time` and nondecreasing `hazard` (or `cum_hazard`). Duplicate times retain their last supplied value. An explicit duration zero must have cumulative hazard zero within tolerance; when the first supplied time is positive, `(0, 0)` is prepended.

`structure` An `mstate_structure` object.

`s` Legacy starting-time argument. Only zero is supported.

`times` Finite nonnegative elapsed durations. If `NULL`, the union of input curve times is used.

`start_state` State in which fresh entry occurs. The default is the common initial state stored in `structure`.

`grid_step` Optional initial internal grid step.

`target_grid_points` Initial number of regular-grid intervals when `grid_step` is not supplied.

`max_grid_points` Maximum number of internal grid points, including elapsed duration zero.

`grid_tol` Maximum probability change allowed on grid refinement. The default `5e-4` reflects discontinuous ranger hazard curves; analytic tests retain their separate `1e-3` truth-error gate.

`max_grid_refinements` Maximum number of step halvings.

`check_grid` Whether to require grid-refinement convergence. Keep `TRUE` for reported analyses. It must be one nonmissing logical; `FALSE` warns and records an unchecked result.

`prob_tol` Positive tolerance for probability bounds and unit mass.

`extrapolate` Either `"error"` (default) or explicit `"flat"` cumulative-hazard extension.

Details

Cumulative hazards are evaluated as right-continuous step functions. Within each regular interval, total exit probability is allocated to causes in proportion to their cumulative-hazard increments. Entry mass is assigned to the interval's right endpoint. Grid refinement controls these approximations. A positive cumulative hazard at duration zero is rejected because the data contract permits no instantaneous fresh-entry transition. Tiny monotonicity corrections within the declared tolerance are counted. Probabilities are never clipped or row-normalized.

Value

A `trans_prob` object. `entry_prob` and its temporary alias `P` are arrays ordered selected starting state by occupied state by elapsed time; the starting-state dimension has length one. `state_occ` is the selected starting-state slice. The object also records `time`, structure, fresh-entry conditioning, support and extrapolation policy, the final internal grid and step, convergence error/refinements, roundoff corrections, and `prob_tol`.

Limitations

Curves must cover every declared edge and represent cumulative cause-specific hazards on the clock-reset duration scale. The function does not estimate hazards, accept left truncation or `s > 0`, condition on an ongoing sojourn, support cyclic/recurrent graphs, or provide uncertainty. Prediction beyond conservative support fails unless the explicit flat-hazard sensitivity extension is selected.

Examples

```
ms <- define_multistate(c("A", "B"), "B", list(A = "B"))
tt <- seq(0, 2, length.out = 2001)
hazards <- list("A->B" = data.frame(time = tt, hazard = 0.4 * tt))
pr <- compute_trans_prob(hazards, ms, times = c(0, 1, 2),
                         target_grid_points = 512)
pr$state_occ
```

`define_multistate` *Define Multistate Structure*

Description

Defines the state space, absorbing states, and allowed transitions for a multistate model.

Usage

```
define_multistate(state_names, absorbing, transitions)
```

Arguments

`state_names`	Character vector of state names.
`absorbing`	Character vector of absorbing state names (must be a subset of `state_names`).
`transitions`	A named list where each element name is an origin state and the value is a character vector of destination states reachable from that origin. Absorbing states should not appear as list names.

Value

An object of class `"mstate_structure"` containing:

state_names Character vector of all state names.

n_states Integer, number of states.

absorbing Character vector of absorbing states.

transient Character vector of transient (non-absorbing) states.

transitions Named list of allowed transitions.

trans_matrix Integer matrix where entry `[i,j]` is the transition number for allowed transition i->j, or `NA` if not allowed.

n_transitions Total number of allowed transitions.

trans_list Data frame listing all transitions with columns `trans_id`, `from`, `to`.

initial_state The unique graph root used as the common initial state.

topological_order State order used by the semi-Markov solver.

Limitations

Only directed acyclic, non-recurrent structures with exactly one root and at least one absorbing state are supported. Every transient state must be reachable from the root and must have a directed path to absorption. Cycles, self-transitions, recurrent visits, disconnected states, and multiple subject-specific initial states are rejected.

Examples

```
ms <- define_multistate(
  state_names = c("Baseline", "Responded", "Progressed", "Death"),
  absorbing = "Death",
  transitions = list(
    Baseline = c("Responded", "Progressed", "Death"),
    Responded = c("Progressed", "Death"),
    Progressed = c("Death")
  )
)
print(ms)
```

diagnose *Diagnostics and Patient-Level Validation for RFmstate*

Description

Reports genuine ranger edge-specific OOB concordance. Optionally, complete patient-level cross-validation refits every edge forest and returns IPCW full-state Brier scores and an integrated Brier score.

Usage

```
diagnose(object, ...)

## S3 method for class 'rfmstate'
diagnose(
  object,
  eval_times = NULL,
  method = c("edge_oob", "cv"),
  folds = 5L,
  repeats = 1L,
  seed = 2026L,
  g_min = 0.05,
  ...
)
```

Arguments

`object`	A fitted `rfmstate` model.
`...`	Ignored; validation settings must use the documented arguments.
`eval_times`	Prespecified calendar-time validation horizons. For `method="cv"`, defaults to nine points inside 80 percent of the fitted conservative support.
`method`	`"edge_oob"` (default) or `"cv"`.
`folds`	Positive integer number of patient-level folds for cross-validation.
`repeats`	Positive integer number of repeated fold assignments.
`seed`	Nonnegative integer fold-assignment and refit seed.
`g_min`	Minimum allowed training-fold censoring survival, strictly between zero and one.

Details

Ranger OOB concordance is `1 - prediction.error` for each binary cause-specific edge endpoint. It does not validate the assembled multistate probability vector. Full-state scores use subject-level held-out predictions and training-fold Kaplan–Meier censoring estimates. The stronger marginal independent-censoring assumption applies to these KM weights. Fold assignment occurs before schema construction; unordered factor levels and numeric ranges are reconstructed from training subjects only. A held-out-only factor level is an explicit fold failure. No bias–variance decomposition is provided.

Value

An rfmstate_diag object. Edge OOB tables (edge_oob, oob_error, and concordance) are always present. With cross-validation, brier is a full-state IPCW score table, ibs is its trapezoidal integral over integration_interval, and fold assignments/support/seeds are stored in assignments and fold_summary; folds is a backward-compatible alias for fold_summary. The object also records evaluation times, whether their grid was prespecified or exploratory, censoring model, g_min, method, and an explicit validation label.

Limitations

Edge OOB concordance applies only to separate binary cause-specific endpoints and is not a full-pipeline validation score. Cross-validation currently uses a training-fold marginal Kaplan–Meier censoring model and therefore requires marginal independent censoring for the score. Every fold must fit every declared edge and cover all evaluation horizons; failures, sparse edges, or censoring survival below g_min stop validation. No calibration model, prediction interval, or bias–variance decomposition is returned. Automatically generated evaluation times are exploratory and are labeled as such; confirmatory work should supply prespecified times.

Examples

```
ms <- clinical_states()
dat <- sim_clinical_data(300, structure = ms, seed = 42)
long <- prepare_data(
  dat, "ID", ms,
  list(Responded="time_Responded", Unresponded="time_Unresponded",
       Stabilized="time_Stabilized", Progressed="time_Progressed",
       Death="time_Death"),
  "time_censored", c("age", "sex", "BMI", "treatment")
)
fit <- rfmstate(long, num.trees = 100, min_events = 3, seed = 42)
diagnose(fit)
```

---

head.msdata *Return ordinary data rows from an msdata object*

Description

Return ordinary data rows from an msdata object

Usage

```
## S3 method for class 'msdata'
head(x, n = 6L, ...)
```

Arguments

`x`	An `msdata` object.
`n`	Number of rows to return.
`...`	Ignored.

Value

An ordinary `data.frame` containing the first `n` rows.

Limitations

Class-specific metadata are intentionally removed from the returned preview, so it should not be passed to `rfmstate()`. The source `msdata` object is unchanged.

Examples

```
ms <- define_multistate(c("A", "B"), "B", list(A = "B"))
dat <- data.frame(id = 1:3, x = 1:3, time_B = 1:3,
                  censor = NA_real_)
long <- prepare_data(dat, "id", ms, list(B = "time_B"), "censor", "x")
head(long, 2)
```

`importance` *Feature Importance per Transition*

Description

Extracts and organizes variable importance scores from the fitted random forest models for each transition. These are ranger permutation-importance scores based on edge-specific OOB predictive loss, not causal effects.

Usage

```
importance(object, ...)

## S3 method for class 'rfmstate'
importance(object, ...)
```

Arguments

`object`	A fitted `rfmstate` model (must have been fit with `importance != "none"`).
`...`	Ignored.

Details

Values are transition-specific and may have different effective sample sizes. Negative values can arise from Monte Carlo variation, sparse events, correlated predictors, or noise; they do not indicate protective or causal effects. Repeated seeds/resamples are needed to assess stability.

Value

An object of class "rfmstate_importance" containing:

importance Data frame with columns variable, from, to, transition, n_events, and importance.

importance_matrix Matrix with variables as rows and transitions as columns.

covariates Covariate names.

transitions Character vector of transition labels.

Limitations

Importance describes transition-specific predictive contribution under the fitted ranger endpoint. Values are not causal effects, are not directly comparable across edges with different risk sets/event counts, and do not validate the assembled full-state probability model. Models fitted with importance = "none" return unavailable values.

Examples

```
ms <- clinical_states()
dat <- sim_clinical_data(n = 200, structure = ms, seed = 42)
msdata <- prepare_data(
  data = dat, id = "ID", structure = ms,
  time_map = list(
    Responded = "time_Responded",
    Unresponded = "time_Unresponded",
    Stabilized = "time_Stabilized",
    Progressed = "time_Progressed",
    Death = "time_Death"
  ),
  censor_col = "time_censored",
  covariates = c("age", "sex", "BMI", "treatment")
)
fit <- rfmstate(msdata, num.trees = 100, seed = 42)
imp <- importance(fit)
print(imp)
```

plot.aj_estimate *Plot Aalen-Johansen Estimates*

Description

Visualizes calendar-time state occupation, product-integral point estimates, Nelson–Aalen cumulative cause-specific hazards, or hazard increments from the Aalen–Johansen estimator.

Usage

```
## S3 method for class 'aj_estimate'
plot(
  x,
  type = c("state_occupation", "stacked_transition_prob", "cumulative_hazard",
    "hazard_increment", "transition_intensity"),
  states = NULL,
  ci = FALSE,
  col = NULL,
  main = NULL,
  xlab = "Time",
  ylab = NULL,
  ...
)
```

Arguments

`x`	An `aj_estimate` object.
`type`	Character, one of `"state_occupation"` (default), `"stacked_transition_prob"`, `"cumulative_hazard"`, `"hazard_increment"`. The legacy `"transition_intensity"` alias warns because increments are not intensities.
`states`	Character vector of states to plot (default: all). For cumulative-hazard and hazard-increment plots, transitions are filtered by destination state.
`ci`	One nonmissing logical value. This is a deprecated compatibility argument: `TRUE` warns because no validated AJ covariance or confidence band is returned.
`col`	One or more colors for selected states/transitions. Values are recycled when necessary; `NULL` uses the package palette.
`main`	Title (default: auto-generated).
`xlab`, `ylab`	Axis labels.
`...`	Additional arguments passed to `plot`.

Value

The input `x` object, returned invisibly. Called for its side effect of producing a plot.

Limitations

AJ plots are covariate-free calendar-time point estimates from the common baseline. Confidence bands are unavailable. Hazard increments are discrete Nelson–Aalen increments, not smoothed transition intensities. The legacy `"transition_intensity"` type is deprecated and draws the same hazard-increment plot with a warning. The historical `"stacked_transition_prob"` label displays the selected state-occupation components from the recorded initial state.

Examples

```
ms <- define_multistate(c("A", "B"), "B", list(A = "B"))
dat <- data.frame(id = 1:6, x = 1:6, time_B = 1:6,
                  censor = NA_real_)
long <- prepare_data(dat, "id", ms, list(B = "time_B"), "censor", "x")
aj <- aalen_johansen(long)
plot(aj, states = "B", xlab = "Study day", ylab = "Probability")
plot(aj, type = "hazard_increment", states = "B")
```

`plot.rfmstate_diag` *Plot Diagnostics*

Description

Visualizes genuine edge OOB concordance or patient-level cross-validated full-state Brier scores.

Usage

```
## S3 method for class 'rfmstate_diag'
plot(
  x,
  type = c("brier", "concordance"),
  col = NULL,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  ...
)
```

Arguments

`x`	An `rfmstate_diag` object.
`type`	Character, one of `"brier"` (default), `"concordance"`.
`col`	One or more plot colors; recycled when necessary.
`main`	Title.
`xlab, ylab`	Axis labels.
`...`	Additional arguments.

Value

The input `x` object, returned invisibly. Called for its side effect of producing a plot.

Limitations

Concordance plots contain separate edge-level ranger OOB statistics, not assembled full-state validation. Brier plots require an object produced by `diagnose(..., method = "cv")` and inherit its censoring/support limitations. No bias–variance plot is available.

Examples

```
d <- structure(list(
  concordance = data.frame(transition = "A->B", c_index = 0.7),
  brier = NULL
), class = "rfmstate_diag")
plot(d, type = "concordance", xlab = "Edge", ylab = "OOB C-index")
```

---

plot.rfmstate_importance

Plot Feature Importance

Description

Visualizes per-transition feature importance as a grouped barplot or heatmap.

Usage

```
## S3 method for class 'rfmstate_importance'
plot(
  x,
  type = c("barplot", "heatmap"),
  col = NULL,
  main = NULL,
  xlab = NULL,
  ylab = NULL,
  ...
)
```

Arguments

`x`	An `rfmstate_importance` object.
`type`	Character, one of `"barplot"` (default), `"heatmap"`.
`col`	Bar colors or heatmap gradient-anchor colors. `NULL` uses the package defaults.
`main`	Title.
`xlab, ylab`	Axis labels.
`...`	Additional graphical arguments. For a heatmap they are passed to `image`; for a barplot they are passed to `barplot`.

Value

The input x object, returned invisibly. Called for its side effect of producing a plot.

Limitations

Values are transition-specific ranger importance scores. Negative values may arise from Monte Carlo noise, sparse events, correlated predictors, or irrelevant variables and are not protective or causal effects. Scales may differ across transitions.

Examples

```
imp <- structure(list(
  importance_matrix = matrix(
    c(0.2, -0.1), nrow = 2,
    dimnames = list(c("x", "z"), "A->B")
  )
), class = "rfmstate_importance")
plot(imp, xlab = "OOB loss increase", ylab = "Predictor")
```

---

plot.rfmstate_pred *Plot RF Multistate Predictions*

Description

Visualizes predicted state occupation probabilities and transition probabilities for individual patients.

Usage

```
## S3 method for class 'rfmstate_pred'
plot(
  x,
  type = c("state_occupation", "transition_prob"),
  subject = 1L,
  col = NULL,
  main = NULL,
  states = NULL,
  xlab = "Elapsed duration",
  ylab = NULL,
  ...
)
```

Arguments

x	An rfmstate_pred object.
type	Character, one of "state_occupation" (default), "transition_prob".
subject	Integer, which subject to plot (default 1). Use 0 for mean across all subjects.
col	One or more state colors. Values are recycled when necessary; NULL uses the package palette.
main	Title.
states	Occupied states to display.
xlab, ylab	Axis labels.
...	Additional arguments passed to plot.

Value

The input x object, returned invisibly. Called for its side effect of producing a plot.

Limitations

Curves are entry-conditioned on fresh entry into the fitted start_state; they are not general Markov $P(s,t)$ curves and have no confidence bands. subject = 0 is an arithmetic mean of profile predictions, not a population-standardized estimator. The historical "transition_prob" plot type displays the occupied-state components conditional on the selected fresh-entry state.

Examples

```
ms <- define_multistate(c("A", "B"), "B", list(A = "B"))
dat <- data.frame(id = 1:60, x = seq(-1, 1, length.out = 60),
                  time_B = 1:60, censor = NA_real_)
long <- prepare_data(dat, "id", ms, list(B = "time_B"), "censor", "x")
fit <- rfmstate(long, num.trees = 20, min.node.size = 3,
                min_events = 3, sparse_warning = Inf, seed = 42)
pred <- predict(fit, data.frame(x = 0), times = c(0, 5, 10),
                target_grid_points = 64, max_grid_points = 1025)
plot(pred, states = c("A", "B"), xlab = "Elapsed day")
plot(pred, type = "transition_prob", states = "B")
```

---

plot_transition_diagram

Plot Transition Diagram

Description

Draws a state transition diagram with event counts annotated on edges. Uses a layered layout that adapts to any number of states and automatically routes arrows around intermediate state boxes using Bezier curves when needed.

Usage

```
plot_transition_diagram(
  structure,
  msdata = NULL,
  col = NULL,
  main = "Transition Diagram",
  xlab = "",
  ylab = "",
  ...
)
```

Arguments

`structure`	An `mstate_structure` object.
`msdata`	Optional `msdata` object to annotate with counts.
`col`	Node colors. Default uses the standard palette.
`main`	Title.
`xlab`, `ylab`	Optional axis labels; defaults are blank because graph coordinates have no statistical scale.
`...`	Ignored.

Value

No return value, called for its side effect of producing a plot.

Limitations

The layout is descriptive and preserves the validated DAG. It does not represent cycles/recurrent visits, transition intensity, uncertainty, or edge direction over calendar time. Counts are shown only when a compatible `msdata` object is supplied.

Examples

```
ms <- clinical_states()
plot_transition_diagram(ms, xlab = "Graph layer")
```

`predict.rfmstate` *Predict Entry-Conditioned State Probabilities*

Description

Predicts clock-reset semi-Markov state-occupation probabilities conditional on fresh entry into a selected state at elapsed duration zero.

Usage

```
## S3 method for class 'rfmstate'
predict(
  object,
  newdata = NULL,
  times = NULL,
  s = 0,
  start_state = NULL,
  grid_step = NULL,
  target_grid_points = 1024L,
  max_grid_points = 131073L,
  grid_tol = 5e-04,
  max_grid_refinements = 7L,
  check_grid = TRUE,
  extrapolate = c("error", "flat"),
  ...
)
```

Arguments

`object`	A fitted `rfmstate` model.
`newdata`	Prediction profiles. `NULL` uses training profiles and labels the result apparent/training.
`times`	Finite nonnegative elapsed durations. `NULL` uses fitted edge event times that fall within the conservative support horizon.
`s`	Legacy start-time argument; only zero is supported.
`start_state`	Fresh-entry starting state. Defaults to the common initial state.
`grid_step`	Optional positive initial internal grid step.
`target_grid_points`	Initial regular-grid interval count when `grid_step` is `NULL`.
`max_grid_points`	Maximum allowed internal grid-point count.
`grid_tol`	Maximum change permitted between successive refinements.
`max_grid_refinements`	Maximum number of grid-step halvings.
`check_grid`	Whether convergence is required. Keep `TRUE` for inferential or reported work.
`extrapolate`	`"error"` (default) or explicit `"flat"` cumulative-hazard sensitivity extension.
`...`	Ignored; no ranger arguments are accepted during prediction.

Details

These are not general Markov $P(s,t)$ matrices. Predictions do not condition on an already elapsed sojourn and carry no confidence bands. By default the horizon cannot exceed the minimum observed sojourn support among reachable transient states.

Value

An rfmstate_pred object. entry_prob and the temporary alias P have dimensions profile by selected starting state by occupied state by elapsed time; the starting-state dimension has length one. state_occ is that selected starting-state slice. The object also contains requested time, per-profile edge cum_hazard, validated newdata, used columns, support and extrapolation metadata, grid-convergence records, conditioning statement, predictor-support extrapolation records, fit-specific schema identifier, state structure, and package versions.

Limitations

Prediction is conditional on fresh entry into start_state at elapsed duration zero. General $P(s, t)$ prediction for s > 0, left truncation, ongoing-sojourn/landmark prediction, recurrent histories, and confidence intervals are unavailable. Missing covariates, nonfinite numeric values, incompatible classes, and unseen factor levels are rejected. Numeric values outside the fit-specific range warn and are recorded as predictor-support extrapolation; they are never truncated. Times beyond support fail unless extrapolate = "flat"; that option assumes zero additional hazard and is a sensitivity analysis only.

Examples

```
ms <- clinical_states()
dat <- sim_clinical_data(300, structure = ms, seed = 42)
long <- prepare_data(
  dat, "ID", ms,
  list(Responded="time_Responded", Unresponded="time_Unresponded",
       Stabilized="time_Stabilized", Progressed="time_Progressed",
       Death="time_Death"),
  "time_censored", c("age", "sex", "BMI", "treatment")
)
fit <- rfmstate(long, num.trees = 100, min_events = 3, seed = 42)
horizon <- min(fit$max_duration_by_origin)
predict(fit, data.frame(age=60, sex=1, BMI=26, treatment=1),
        times = c(0, horizon / 2))
```

---

prepare_data *Prepare Data for Multistate Analysis*

Description

Converts one-row-per-subject wide data into validated, long-form sojourn records for an acyclic, non-recurrent multistate process. Entry times are calendar times from a common origin; forest response times are the resulting clock-reset durations within states.

Usage

```
prepare_data(
  data,
  id,
  structure,
  time_map,
  censor_col,
  covariates,
  initial_state = NULL
)
```

Arguments

data	A data frame with one row per subject.
id	Name of the unique subject-ID column.
structure	An mstate_structure from define_multistate.
time_map	Named list mapping every noninitial state exactly once to its first and only calendar-time entry column. The initial state is excluded.
censor_col	Name of the external right-censoring column.
covariates	Explicit character vector of baseline, time-fixed predictors. Use character(0) only for preparation/AJ workflows.
initial_state	Common initial state. It must equal the unique graph root; the default is stored in structure.

Details

Delayed entry, recurrent visits, tied entry times, time-dependent covariates, and missing covariates are unsupported. Invalid trajectories produce an error; no event or interval is silently discarded. Character covariates are converted to factors in first-observed level order, and the resulting preparation schema is descriptive only. Every model fit and cross-validation refit rebuilds its schema from its own fitting rows. Subject ID, mapped entry times, censoring time, reserved long-format names, and names beginning .rfm_ cannot be predictors.

Value

An msdata data frame containing id, from, to, Tstart, Tstop, status, trans_id, duration, and retained covariates. Metadata store the graph, source-role table, predictor contract, descriptive preparation schema, counts, and observed duration support.

Limitations

The input must contain one row per subject, one common initial state at time zero, and at most one exact entry time per noninitial state. Left truncation, cycles/recurrent visits, tied or interval-censored transitions, time-dependent covariates, missing fitting covariates, events after censoring, and events after absorption are rejected. A nonabsorbed subject requires a finite censoring time strictly after the last state entry.

Examples

```
ms <- clinical_states()
dat <- sim_clinical_data(n = 50, structure = ms, seed = 42)
msdata <- prepare_data(
  dat, id = "ID", structure = ms,
  time_map = list(
    Responded = "time_Responded",
    Unresponded = "time_Unresponded",
    Stabilized = "time_Stabilized",
    Progressed = "time_Progressed",
    Death = "time_Death"
  ),
  censor_col = "time_censored",
  covariates = c("age", "sex", "BMI", "treatment")
)
head(msdata)
```

---

print.aj_estimate *Print RFmstate Result Objects*

Description

Concise print methods for fitted models, predictions, direct semi-Markov probability results, Aalen–Johansen point estimates, importance results, diagnostics, and fitted-model summaries.

Usage

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

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

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

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

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

## S3 method for class 'summary.rfmstate'
print(x, ...)

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

Arguments

x	An object of the class required by the selected print method: `rfmstate`, `rfmstate_pred`, `trans_prob`, `aj_estimate`, `rfmstate_importance`, `rfmstate_diag`, or `summary.rfmstate`.
...	Ignored.

Details

Printed values are deliberately concise and rounded only for display. Stored numerical results are unchanged. The fitted-model and summary methods identify edge-level ranger OOB quantities; prediction and direct-probability methods state their fresh-entry conditioning; the AJ method states that it returns point estimates only; and the diagnostic method distinguishes edge OOB output from patient-level cross-validation. The example is in a `donttest` block because it fits survival forests to exercise every result class.

Value

`x`, invisibly.

Limitations

Print methods are descriptive and do not revalidate, refit, or add uncertainty. Rounded importance or probability values must not be used in downstream calculations. Edge OOB statistics are not full-state validation, negative permutation importance is not a protective/causal effect, forest confidence intervals are unavailable, and AJ confidence intervals are not returned.

Examples

```
ms <- define_multistate(c("A", "B"), "B", list(A = "B"))
dat <- data.frame(id = 1:60, x = seq(-1, 1, length.out = 60),
                  time_B = 1:60, censor = NA_real_)
long <- prepare_data(dat, "id", ms, list(B = "time_B"), "censor", "x")
fit <- rfmstate(long, num.trees = 20, min.node.size = 3,
                min_events = 3, sparse_warning = Inf, seed = 42)
pred <- predict(fit, data.frame(x = 0), times = c(0, 10),
                target_grid_points = 64, max_grid_points = 1025)
tt <- seq(0, 10, length.out = 1001)
tp <- compute_trans_prob(
  list("A->B" = data.frame(time = tt, hazard = 0.1 * tt)),
  ms, times = c(0, 5, 10), target_grid_points = 128
)
aj <- aalen_johansen(long)
print(fit)
print(pred)
print(tp)
print(aj)
print(importance(fit))
print(diagnose(fit))
print(summary(fit))
```

print.msdata — *Print Prepared Multistate Data*

Description

Prints a concise validation summary for an `msdata` object, including patient and interval totals and an observed-transition table in the original state-definition order. Use `head()` to display actual data rows.

Usage

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

Arguments

`x`	An `msdata` object returned by `prepare_data()`.
`...`	Ignored.

Value

`x`, invisibly.

Limitations

The printout is descriptive and does not refit or revalidate a modified object. Re-run `prepare_data()` after changing rows or metadata.

Examples

```
ms <- define_multistate(c("A", "B"), "B", list(A = "B"))
dat <- data.frame(id = 1:3, x = 1:3, time_B = 1:3,
                  censor = NA_real_)
long <- prepare_data(dat, "id", ms, list(B = "time_B"), "censor", "x")
print(long)
```

print.mstate_structure — *Print a Multistate Structure*

Description

Prints the user-defined display order, absorbing states, common initial state, computational topological order, and numbered allowed transitions.

Usage

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

Arguments

x	An mstate_structure object.
...	Ignored.

Value

x, invisibly.

Limitations

The printout summarizes the validated graph; it does not imply support for cycles, recurrent visits, left truncation, or time-dependent covariates.

Examples

```
ms <- define_multistate(c("A", "B"), "B", list(A = "B"))
print(ms)
```

---

rfmstate *Fit Clock-Reset Random-Forest Multistate Models*

Description

Fits one ranger cause-specific survival forest for every declared edge of a validated acyclic, non-recurrent multistate structure. Forest response time is duration since fresh entry into the origin state.

Usage

```
rfmstate(
  msdata,
  covariates = NULL,
  num.trees = 1000L,
  mtry = NULL,
  min.node.size = 15L,
  min_events = 5L,
  sparse_warning = 50L,
  importance = "permutation",
  seed = NULL,
  ...
)
```

Arguments

msdata	An msdata object from prepare_data.
covariates	Predictor names. NULL uses the complete stored predictor contract; an explicit value must be a nonempty unique subset of that contract.
num.trees	Positive integer number of trees fitted for every edge.
mtry	Positive integer number of predictors considered at each split. NULL uses floor(sqrt(p)), bounded below by one.
min.node.size	Positive integer ranger minimum terminal-node size.
min_events	Explicit technical safeguard for target events per edge. It is not a universal adequacy threshold.
sparse_warning	Positive integer descriptive event-count threshold for an edge warning; set Inf to disable. It is not a fit threshold.
importance	One of "permutation", "impurity", "impurity_corrected", or "none". Permutation importance is recommended for interpretation under the limitations below.
seed	NULL or one nonnegative integer seed passed to every edge forest without changing the caller's RNG state.
...	Supported ranger controls: replace, sample.fraction, splitrule, num.random.splits, respect.unordered.factors, num.threads, save.memory, max.depth, always.split.variables, alpha, minprop, and verbose. Unnamed, conflicting, and unknown arguments are rejected. Configurations without genuine OOB observations are rejected.

Details

Every competing exit from an origin state remains an observed exit for that sojourn, but is coded as a non-target outcome in the binary cause-specific forest for a particular edge. An unestimable declared edge stops the whole fit; it is never omitted or represented by zero hazard. The ranger model frame is constructed anew as .rfm_time, .rfm_event, and the contract-approved predictors. Predictor factor levels and numeric ranges are learned from the actual fitting rows, not copied from a preparation-time or full-data schema.

Value

An rfmstate object containing models for every declared edge; structure; selected covariates and predictor_schema; origin-state fitting data; backend event-time grids; per-edge event, support, ranger-argument, and genuine OOB metadata; the validated msdata; fit params; package versions; verified OOB coverage; and the clock-reset semi-Markov time-scale contract.

Ranger argument contract

RFmstate controls and rejects duplicate specification of formula, data, num.trees, mtry, min.node.size, importance, seed, write.forest, oob.error, and keep.inbag. Unnamed and unknown arguments also fail. The only names accepted through ... are:

replace, sample.fraction Bootstrap/subsampling controls.

splitrule, num.random.splits, alpha, minprop Survival split controls supported by the installed ranger version.

respect.unordered.factors Unordered-factor handling.

num.threads, save.memory, verbose Computation controls.

max.depth Maximum tree depth.

always.split.variables Predictors always considered for splitting.

These arguments retain ranger's definitions and are checked against the installed ranger formal arguments before fitting. `case.weights`, `class.weights`, `split.select.weights`, response controls, and every other unlisted ranger argument are rejected in this release. Effective sampling defaults are resolved and stored. In particular, `replace = FALSE, sample.fraction = 1` is rejected because it leaves no OOB observations. RFmstate forces `oob.error = TRUE` and `keep.inbag = TRUE`, verifies finite ranger OOB error after every edge fit, and stores per-sojourn OOB-tree coverage.

Limitations

The fit supports baseline, time-fixed, complete covariates in single-root acyclic non-recurrent data. It does not support left truncation, time-dependent covariates, recurrent visits/cycles, missing fitting covariates, subject-specific frailty, clock-forward hazards, or confidence intervals. Structural, outcome-time, censoring, ID, and arbitrary long-format columns cannot be added as predictors. Every declared edge must meet `min_events`; this technical safeguard is not a universal adequacy threshold.

Examples

```
ms <- clinical_states()
dat <- sim_clinical_data(300, structure = ms, seed = 42)
long <- prepare_data(
  dat, "ID", ms,
  list(Responded="time_Responded", Unresponded="time_Unresponded",
       Stabilized="time_Stabilized", Progressed="time_Progressed",
       Death="time_Death"),
  "time_censored", c("age", "sex", "BMI", "treatment")
)
fit <- rfmstate(long, num.trees = 100, min_events = 3, seed = 42)
```

`sim_clinical_data` *Simulate Clinical Trial Multistate Data*

Description

Generates realistic clinical trial data with covariates and multistate event times for testing and demonstration. The structure must be an acyclic, non-recurrent graph with one common initial state.

Usage

```
sim_clinical_data(n = 500, structure = NULL, max_followup = 365, seed = NULL)
```

Arguments

n	Integer, number of patients to simulate.
structure	An mstate_structure object. Defaults to clinical_states().
max_followup	Numeric, maximum follow-up time (for generating censoring). Default 365.
seed	Optional integer for reproducibility.

Details

Cause-specific transition hazards follow Weibull distributions with covariate effects on the scale parameter. For the default clinical_states() structure, transition-specific parameters are calibrated to produce realistic clinical trial trajectories. For custom structures, sensible default parameters are used for all transitions.

Event waiting times remain at full numerical precision. External censoring is generated before each path and truncates it, so no event is retained after censoring or after entry into any absorbing state. Supplying the same seed, arguments, structure, and package version reproduces the returned data.

Value

A data frame in wide format with columns:

ID Patient identifier (1 to n).

age Continuous, simulated from Normal(60, 12).

sex Binary 0/1.

BMI Continuous, simulated from Normal(26, 5).

treatment Binary 0/1 (balanced arms).

time_*StateName* For each non-initial state in the structure, the time (days) of entry into that state, or NA if the state was not visited. Column names follow the pattern time_<StateName> (e.g., time_Death).

time_censored Days until last follow-up (right censoring time), or NA if an absorbing state was reached.

Limitations

This helper supplies demonstration and test data, not the final manuscript simulation design. It uses four fixed baseline covariates and simple transition-specific Weibull cause-specific hazards. It does not generate left truncation, recurrent/cyclic histories, time-dependent covariates, interval-censored transitions, or ongoing-sojourn predictions.

Examples

```
dat <- sim_clinical_data(n = 100, seed = 123)
head(dat)
summary(dat)
```

summary.rfmstate *Summary of Random Forest Multistate Model*

Description

Provides a comprehensive summary of the fitted model including per-origin state information, OOB prediction error, and transition event counts.

Usage

```
## S3 method for class 'rfmstate'
summary(object, ...)
```

Arguments

`object`	A fitted `rfmstate` model.
`...`	Ignored.

Details

The per-edge OOB concordance is `1 - prediction.error` from ranger for that edge's binary cause-specific endpoint. The printed summary also lists every forwarded ranger argument.

Value

An object of class `"summary.rfmstate"` containing the stored call, covariates, forest parameters, patient/event/interval totals, structure, and a per-edge table of risk-set size, target events, competing/non-target exits, external censoring, ranger OOB error, ranger OOB concordance, and verified OOB coverage.

Limitations

Edge OOB values do not validate assembled full-state probabilities and may be unstable for sparse transitions. Use `diagnose(..., method = "cv")` for patient-level held-out full-state Brier scoring.

Examples

```
ms <- clinical_states()
dat <- sim_clinical_data(n = 200, structure = ms, seed = 42)
msdata <- prepare_data(
  data = dat, id = "ID", structure = ms,
  time_map = list(
    Responded = "time_Responded",
    Unresponded = "time_Unresponded",
    Stabilized = "time_Stabilized",
    Progressed = "time_Progressed",
    Death = "time_Death"
  ),
```

```
  censor_col = "time_censored",
  covariates = c("age", "sex", "BMI", "treatment")
)
fit <- rfmstate(msdata, num.trees = 100, seed = 42)
summary(fit)
```

Index

aalen_johansen, 3

clinical_states, 5, *28*
compute_trans_prob, 5

define_multistate, 7, *21*
diagnose, 9

head.msdata, 10

importance, 11

plot, *13*, *17*
plot.aj_estimate, 13
plot.rfmstate_diag, 14
plot.rfmstate_importance, 15
plot.rfmstate_pred, 16
plot_transition_diagram, 17
predict.rfmstate, 18
prepare_data, *3*, 20, *26*
print.aj_estimate, 22
print.msdata, 24
print.mstate_structure, 24
print.rfmstate *(*print.aj_estimate*)*, 22
print.rfmstate_diag
*(*print.aj_estimate*)*, 22
print.rfmstate_importance
*(*print.aj_estimate*)*, 22
print.rfmstate_pred
*(*print.aj_estimate*)*, 22
print.summary.rfmstate
*(*print.aj_estimate*)*, 22
print.trans_prob *(*print.aj_estimate*)*, 22
print_rfmstate_objects
*(*print.aj_estimate*)*, 22

RFmstate *(*RFmstate-package*)*, 2
rfmstate, 25
RFmstate-package, 2

sim_clinical_data, 27
summary.rfmstate, 29