Package 'RFmstate'

March 11, 2026

Type Package

Title Random Forest-Based Multistate Survival Analysis

Version 0.1.2

Author Yiqing Chen [aut, cre]

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

Description Fits cause-specific random survival forests for flexible multistate survival analysis with covariate-adjusted transition probabilities computed via product-integral. State transitions are modeled by random forests. Subject-specific transition probability matrices are assembled from predicted cumulative hazards using the product-integral formula. Also provides a standalone Aalen-Johansen nonparametric estimator as a covariate-free baseline. Supports arbitrary state spaces with any number of states (three or more) and any set of allowed transitions, applicable to clinical trials, disease progression, reliability engineering, and other domains where subjects move among discrete states over time. Provides per-transition feature importance, bias-variance diagnostics, and comprehensive visualizations. Handles right censoring and competing transitions. 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

Encoding UTF-8

Depends R (>= 4.1.0)

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

Suggests knitr, rmarkdown, mstate, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

RoxygenNote 7.3.3

NeedsCompilation no

Repository CRAN
Date/Publication 2026-03-11 16:40:16 UTC

Contents

RFmstate-package . . . 2
aalen_johansen . . . 3
clinical_states . . . 4
compute_trans_prob . . . 5
define_multistate . . . 5
diagnose . . . 6
importance . . . 8
plot.aj_estimate . . . 9
plot.rfmstate_diag . . . 10
plot.rfmstate_importance . . . 11
plot.rfmstate_pred . . . 11
plot_transition_diagram . . . 12
predict.rfmstate . . . 13
prepare_data . . . 14
rfmstate . . . 16
sim_clinical_data . . . 18
summary.rfmstate . . . 19

Index **20**

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

Description

Fits cause-specific random survival forests for flexible multistate survival analysis with covariate-adjusted transition probabilities computed via product-integral. For each transient state, competing transitions are modeled by separate random forests, and patient-specific transition probability matrices are assembled from the predicted cumulative hazards using the product-integral formula. Also provides a standalone Aalen-Johansen nonparametric estimator as a covariate-free baseline. Supports arbitrary state spaces with any number of states (three or more) and any set of allowed transitions, applicable to clinical trials, disease progression, reliability engineering, and other domains where subjects move among discrete states over time. 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
- Transition probability matrices via product-integral of predicted cumulative hazards
- Aalen-Johansen nonparametric estimation (covariate-free baseline)
- Per-transition feature importance
- Bias-variance diagnostics with Brier score and C-index
- Comprehensive visualizations

Author(s)

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

`aalen_johansen` *Aalen-Johansen Nonparametric 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`	Numeric, the starting time for transition probabilities (default 0).

Details

The Aalen-Johansen estimator generalizes the Kaplan-Meier estimator to multistate models under the Markov assumption and independent right censoring. It provides population-level transition probability matrices without covariate adjustment.

Value

An object of class `"aj_estimate"` containing:

time Numeric vector of unique event times.

trans_prob List of transition probability matrices P(s,t) 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.

variance List of Greenwood-type variance estimates for state occupation probabilities.

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.

Examples

```
ms <- clinical_states()
set.seed(42)
dat <- sim_clinical_data(n = 200, structure = ms)
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.

Examples

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

compute_trans_prob *Compute Transition Probability Matrix via Product-Integral*

Description

Given cause-specific cumulative hazard functions for all transitions, computes the full transition probability matrix P(s,t) using the product-integral formula.

Usage

```
compute_trans_prob(cum_hazards, structure, s = 0, times = NULL)
```

Arguments

cum_hazards	A list of cumulative hazard data frames, one per transition. Each should have columns time and hazard.
structure	An mstate_structure object.
s	Numeric, starting time (default 0).
times	Numeric vector of times at which to evaluate P(s,t). If NULL, uses all unique event times from the cumulative hazards.

Value

An object of class "trans_prob" containing:

time Evaluation times.

P List of transition probability matrices at each time.

state_occ Matrix of state occupation probabilities.

structure The multistate structure.

s Starting time.

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.

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 for Random Forest Multistate Model*

Description

Computes diagnostic measures including OOB-based prediction error, Brier score, concordance index, and bias-variance decomposition for each transition-specific model.

Usage

```
diagnose(object, ...)

## S3 method for class 'rfmstate'
diagnose(object, eval_times = NULL, ...)
```

Arguments

`object`	A fitted `rfmstate` model.
`...`	Ignored.
`eval_times`	Numeric vector of times at which to evaluate diagnostics. If `NULL`, uses quantiles of event times.

Details

The bias-variance decomposition uses OOB predictions from the random forest ensemble. For each transition:

- **Bias**: systematic difference between predicted and observed survival.
- **Variance**: variability of predictions across trees (estimated from tree-level OOB predictions when available).
- **Brier score**: integrated prediction error combining bias and variance.
- **C-index**: concordance between predicted risk and observed event ordering.

Value

An object of class `"rfmstate_diag"` containing:

oob_error Data frame of OOB prediction errors per transition.

brier List of time-dependent Brier score components per transition.

concordance Data frame of concordance indices per transition.

bias_variance Data frame of bias-variance decomposition per transition.

eval_times Evaluation times used.

Examples

```
ms <- clinical_states()
set.seed(42)
dat <- sim_clinical_data(n = 200, structure = ms)
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, covariates = c("age", "sex", "BMI", "treatment"),
                num.trees = 100)
diag <- diagnose(fit)
print(diag)
```

`importance` *Feature Importance per Transition*

Description

Extracts and organizes variable importance scores from the fitted random forest models for each transition.

Usage

```
importance(object, ...)
```

Arguments

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

Value

An object of class `"rfmstate_importance"` containing:

importance Data frame with columns `variable`, `from`, `to`, `importance`.

importance_matrix Matrix with variables as rows and transitions as columns.

covariates Covariate names.

transitions Character vector of transition labels.

Examples

```
ms <- clinical_states()
set.seed(42)
dat <- sim_clinical_data(n = 200, structure = ms)
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, covariates = c("age", "sex", "BMI", "treatment"),
                num.trees = 100)
imp <- importance(fit)
print(imp)
```

`plot.aj_estimate` *Plot Aalen-Johansen Estimates*

Description

Visualizes state occupation probabilities and transition probabilities from the Aalen-Johansen estimator.

Usage

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

Arguments

`x`	An `aj_estimate` object.
`type`	Character, one of `"state_occupation"` (default), `"stacked_transition_prob"`, `"cumulative_hazard"`, `"transition_intensity"`.
`states`	Character vector of states to plot (default: all). For `"transition_intensity"`, filters by destination state.
`ci`	Logical, whether to show confidence intervals (default `TRUE`).
`col`	Colors for each state/transition. If `NULL`, default palette is used.
`main`	Title (default: auto-generated).
`...`	Additional arguments passed to `plot`.

Value

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

`plot.rfmstate_diag` *Plot Diagnostics*

Description

Visualizes diagnostic measures including Brier score curves, concordance indices, and bias-variance decomposition.

Usage

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

Arguments

x	An rfmstate_diag object.
type	Character, one of "brier" (default), "concordance", "bias_variance".
col	Colors.
main	Title.
...	Additional arguments.

Value

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

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, ...)
```

Arguments

x	An rfmstate_importance object.
type	Character, one of "barplot" (default), "heatmap".
col	Colors.
main	Title.
...	Additional arguments.

Value

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

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,
  ...
)
```

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	Colors. If NULL, default palette is used.
main	Title.
...	Additional arguments passed to plot.

Value

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

---

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",
  ...
)
```

Arguments

structure	An mstate_structure object.
msdata	Optional msdata object to annotate with counts.
col	Node colors. Default uses the standard palette.
main	Title.
...	Ignored.

Value

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

Examples

```
ms <- clinical_states()
plot_transition_diagram(ms)
```

`predict.rfmstate` *Predict Transition Probabilities for New Data*

Description

Predicts patient-specific transition probability matrices and state occupation probabilities using fitted random forest multistate models.

Usage

```
## S3 method for class 'rfmstate'
predict(object, newdata = NULL, times = NULL, s = 0, ...)
```

Arguments

`object`	A fitted `rfmstate` model.
`newdata`	A data frame with the same covariates used in fitting. If `NULL`, predictions are made for the training data.
`times`	Numeric vector of times at which to compute transition probabilities. If `NULL`, uses all unique event times.
`s`	Numeric, starting time (default 0).
`...`	Ignored.

Value

An object of class `"rfmstate_pred"` containing:

time Evaluation times.

P Array of transition probability matrices (n_subjects x n_states x n_states x n_times).

state_occ Array of state occupation probabilities (n_subjects x n_states x n_times).

cum_hazard List of per-subject cumulative hazard matrices.

structure The multistate structure.

newdata The prediction data.

Examples

```
ms <- clinical_states()
set.seed(42)
dat <- sim_clinical_data(n = 200, structure = ms)
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, covariates = c("age", "sex", "BMI", "treatment"),
                num.trees = 100)
newpat <- data.frame(age = c(50, 70), sex = c(0, 1),
                     BMI = c(25, 30), treatment = c(1, 0))
pred <- predict(fit, newdata = newpat, times = c(30, 90, 180, 365))
```

---

prepare_data *Prepare Data for Multistate Analysis*

Description

Converts wide-format clinical data into long counting-process format suitable for multistate survival analysis.

Usage

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

Arguments

data	A data frame in wide format with one row per patient.
id	Character string, name of the patient ID column.

structure	An mstate_structure object from define_multistate.
time_map	A named list mapping state names to column names in data containing the time of entry into that state (measured from baseline). The initial state should not be included. Use NA in the data for states not visited by a patient.
censor_col	Character string, name of the column containing the right censoring time (last follow-up time).
covariates	Character vector of covariate column names to carry into the long-format data.
initial_state	Character string, the starting state for all patients (default: first state in the structure).

Details

Each patient's trajectory is reconstructed from event times, validated against the allowed transitions, and expanded into start-stop intervals with covariates.

Value

An object of class "msdata" (a data frame) with columns:

id Patient identifier.

from Origin state for this interval.

to Destination state (or NA if censored).

Tstart Start time of the interval.

Tstop End time of the interval.

status 1 if a transition occurred, 0 if censored.

trans_id Integer transition ID (from structure) or NA.

duration Duration of the interval.

... Covariate columns.

The object also carries an attribute "structure" (the mstate_structure).

Examples

```
ms <- clinical_states()
set.seed(42)
dat <- sim_clinical_data(n = 50, structure = ms)
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")
```

```
)
head(msdata)
```

`rfmstate` *Fit Random Forest Multistate Model*

Description

Fits transition-specific cause-specific random survival forests for multistate survival analysis. For each transient origin state, a competing risks model is fit using random forests, where the competing events are the possible transitions to destination states.

Usage

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

Arguments

`msdata`	An `msdata` object from `prepare_data`.
`covariates`	Character vector of covariate column names to use as predictors. If `NULL`, all non-structural columns are used.
`num.trees`	Integer, number of trees per forest (default 1000).
`mtry`	Integer, number of variables to try at each split. Default `NULL` uses `floor(sqrt(p))` where p is number of covariates.
`min.node.size`	Integer, minimum node size (default 15).
`importance`	Character, variable importance mode. One of `"permutation"` (default), `"impurity"`, or `"none"`.
`seed`	Integer, random seed for reproducibility (default `NULL`).
`...`	Additional arguments passed to `ranger`.

Details

For each transient state h, the method:

1. Subsets all intervals where the patient is in state h.
2. Defines time as the duration in state h (Tstop - Tstart).
3. Codes competing events: 0 = censored, 1, 2, ... for each possible destination state.
4. Fits a cause-specific random survival forest using `ranger` with `survival` tree type.

Transition probabilities are then computed by combining per-origin-state predicted cumulative hazards via the product-integral formula.

Value

An object of class `"rfmstate"` containing:

models Named list of fitted `ranger` objects, one per origin state.

structure The multistate structure.

covariates Character vector of covariate names used.

origin_data Named list of per-origin-state data subsets.

event_times Named list of unique event times per origin state.

call The matched call.

params List of tuning parameters used.

Examples

```
ms <- clinical_states()
set.seed(42)
dat <- sim_clinical_data(n = 200, structure = ms)
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, covariates = c("age", "sex", "BMI", "treatment"))
print(fit)
```

`sim_clinical_data` *Simulate Clinical Trial Multistate Data*

Description

Generates realistic clinical trial data with covariates and multistate event times for testing and demonstration. Works with any multistate structure.

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

Transition intensities 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.

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.

Examples

```
set.seed(123)
dat <- sim_clinical_data(n = 100)
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.

Value

An object of class "summary.rfmstate", printed invisibly.

Examples

```
ms <- clinical_states()
set.seed(42)
dat <- sim_clinical_data(n = 200, structure = ms)
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, covariates = c("age", "sex", "BMI", "treatment"),
                num.trees = 100)
summary(fit)
```

Index

aalen_johansen, 3

clinical_states, 4, *18*
compute_trans_prob, 5

define_multistate, 5, *15*
diagnose, 6

importance, 8

plot, *9*, *12*
plot.aj_estimate, 9
plot.rfmstate_diag, 10
plot.rfmstate_importance, 11
plot.rfmstate_pred, 11
plot_transition_diagram, 12
predict.rfmstate, 13
prepare_data, *3*, 14, *16*

ranger, *16*, *17*
RFmstate *(*RFmstate-package*)*, 2
rfmstate, 16
RFmstate-package, 2

sim_clinical_data, 18
summary.rfmstate, 19