Package: blendR (via r-universe)

May 14, 2026

Title Blended Survival Curves

Version 1.0.0

Description Create a blended curve from two survival curves, which is particularly useful for survival extrapolation in health technology assessment. The main idea is to mix a flexible model that fits the observed data well with a parametric model that encodes assumptions about long-term survival. The two curves are blended into a single survival curve that is identical to the first model over the range of observed times and gradually approaches the parametric model over the extrapolation period based on a given weight function. This approach allows for the inclusion of external information, such as data from registries or expert opinion, to guide long-term extrapolations, especially when dealing with immature trial data. See Che et al. (2022) <doi:10.1177/0272989X221134545>.

License GPL (>= 3)

Additional_repositories https://giabaio.r-universe.dev, https://inla.r-inla-download.org/R/stable

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.2.9000

Imports dplyr, flexsurv, ggplot2, manipulate, sn, survHE, tibble

Depends R (>= 4.4.0)

Suggests INLA, knitr, remotes, rlang, rmarkdown, survHEhmc, survival, testthat (>= 3.0.0)

VignetteBuilder knitr

URL https://github.com/StatisticsHealthEconomics/blendR/, https://StatisticsHealthEconomics.github.io/blendR/

BugReports https://github.com/StatisticsHealthEconomics/blendR/issues/

Config/testthat/edition 3

Config/pak/sysreqs cmake make default-jdk libicu-dev libuv1-dev

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

Date/Publication 2025-10-16 07:41:13 UTC

RemoteUrl https://github.com/statisticshealtheconomics/blendR

RemoteRef HEAD

RemoteSha 37756194473dd1e41ca4794cf4dbc8de3112fbb9

Contents

`blendsurv` *Blended survival object*

Description

This is the main function in the **blendR** package. Two survival curves are supplied and blended according to the blending distribution characterised by the blending interval and the beta distribution parameters.

Usage

```
blendsurv(
  obs_Surv,
  ext_Surv,
  blend_interv,
  beta_params = list(alpha = 3, beta = 3),
  times = NULL,
  nsim = 100
)
```

Arguments

`obs_Surv`, `ext_Surv`	Observed and external data survival curves. These can come from **survHE**, **INLA** or **flexsurv** fits.
`blend_interv`	Maximum and minimum values for the blending interval.
`beta_params`	Coefficients of a beta distribution.
`times`	A vector of times for which the survival curves are to be computed; optional.
`nsim`	The number of simulations from the distribution of the survival curves; default 100.

Value

List of survival probabilities for observed, external and blended curves, with other relevant data.

Examples

```
library(survHE)

## trial data
data("TA174_FCR", package = "blendR")

## externally estimated data
data_sim <- ext_surv_sim(t_info = 144,
                         S_info = 0.05,
                         T_max = 180)
# observed survival times
obs_Surv <- fit.models(formula = Surv(death_t, death) ~ 1,
                       data = dat_FCR,
                       distr = "exponential",
                       method = "hmc")

# external survival times
ext_Surv <- fit.models(formula = Surv(time, event) ~ 1,
                       data = data_sim,
                       distr = "exponential",
                       method = "hmc")

# blending parameter values
blend_interv <- list(min = 48, max = 150)
beta_params <- list(alpha = 3, beta = 3)

ble_Surv <- blendsurv(obs_Surv, ext_Surv, blend_interv, beta_params)

# all survival curves
plot(ble_Surv)
```

dat_FCR *Survival data*

Description

Survival data

ext_surv_sim *Create an external survival data based on expert opinion*

Description

Generates an individual patient-level survival dataset from aggregate survival probabilities that might be elicited from expert opinion.

The function creates a synthetic dataset of event times for `n` patients, consistent with a piecewise-constant hazard rate implied by the expert-provided survival probabilities.

Usage

```
ext_surv_sim(t_info, S_info, T_max, n = 100)
```

Arguments

`t_info`	A numeric vector of time points for which expert opinion is elicited.
`S_info`	A numeric vector of mean survival probabilities estimated by experts corresponding to time points in `t_info`.
`T_max`	The maximum survival time for the simulation, at which the survival probability is assumed to be 0.
`n`	The total number of patients to construct the artificial external data set; default 100

Details

The simulation uses a two-step sampling process based on partitioning the time horizon from `0` to `T_max`. First, the total number of patients `n` is allocated to different time intervals `[t_i, t_{i+1}]`. The probability of an event occurring in an interval is derived from the drop in the survival curve over that interval. This allocation is done via a multinomial distribution:

$$i \sim \text{Multinomial}(\pi)$$

where $\pi_i = S(t_i-1) - S(t_i)$ is the probability of an event in interval `i`.

Second, for the patients assigned to each interval, a specific event time is simulated from a uniform distribution covering that interval:

$$T|i \sim U(t_{i-1}, t_i)$$

Value

A data frame with n rows and two columns:

- **time**: The simulated event time for each patient.
- **event**: The event indicator, which is always 1 as this function does not simulate censoring.

Examples

```
dat <- ext_surv_sim(t_info = c(10,20,50),
                    S_info = c(0.9, 0.8, 0.2),
                    T_max = 100, n = 100)
if (require(survival)) {
    # Kaplan-Meier curve
    km_fit <- survfit(Surv(time, event) ~ 1, data = dat)
    plot(km_fit)
}
```

---

fit_inla_pw *Fit a Piecewise Exponential Survival Model using INLA*

Description

A convenience wrapper to fit a Bayesian piecewise exponential model (PEM) for survival data using the INLA package. This function simplifies the process by automatically handling the necessary data transformation from a standard survival format to the Poisson regression format that INLA requires.

Usage

```
fit_inla_pw(
  inla.formula = inla.surv(death_t, death) ~ -1,
  data,
  cutpoints,
  nsim = 100,
  ...
)
```

Arguments

inla.formula	A formula specifying the survival model. The response must be an INLA::inla.surv object (e.g., inla.surv(time, event)). The right-hand side specifies the linear predictor. The default fits a baseline-hazard-only model.
data	A data frame containing the time-to-event data, including the variables named in the inla.formula.
cutpoints	A numeric vector of cut points used to partition the time axis into intervals for the piecewise constant hazard.
...	Additional arguments to be passed directly to the INLA::inla function (e.g., control.predictor, control.family).

Details

This function models the baseline hazard rate as a piecewise constant function. The `cutpoints` define the time intervals within which the hazard is assumed to be constant.

Internally, the function first uses `INLA::inla.coxph` to convert the survival data into the long format suitable for a Poisson generalized linear model (GLM). It then fits this model using `INLA::inla`, applying a random walk of order 1 (`rw1`) prior to the log-hazards for each interval. This prior provides a flexible and smoothed estimate of the baseline hazard over time.

Value

An object of class `inla`, which contains the full results of the fitted Bayesian model.

Examples

```
# INLA may require configuration on your system.
# See: https://www.r-inla.org/download-install
 if (requireNamespace("INLA", quietly = TRUE)) {
  data("TA174_FCR", package = "blendR")
  head(dat_FCR)

  # Fit a simple piecewise model with intervals every 5 time units
  obs_Surv <- fit_inla_pw(data = dat_FCR, cutpoints = seq(0, 180, by = 5))

  # summary(obs_Surv)
 }
```

`make_surv_methods` *Create survival probabilities*

Description

A generic S3 function and methods to generate a standardized matrix of survival probabilities from various fitted survival model objects.

This function standardizes the output from different survival modelling packages into a consistent format: a matrix where rows represent discrete time points and columns represent simulations from the model's posterior distribution. This standardized format is essential for use in downstream evidence blending functions. The methods are inspired by the `survHE::make.surv()` function.

Usage

```
make_surv(Surv, ...)

## S3 method for class 'survHE'
make_surv(Surv, t, nsim = 100, ...)
```

```
## S3 method for class 'flexsurvreg'
make_surv(Surv, t = NULL, nsim = 100, ...)

## S3 method for class 'inla'
make_surv(Surv, t = NULL, nsim = 100, ...)

## Default S3 method:
make_surv(Surv, t = NULL, nsim = 100, ...)
```

Arguments

Surv	A fitted survival model object or a matrix/vector of survival probabilities. Supported classes include survHE, flexsurvreg, inla, matrix, and numeric.
...	Additional arguments passed to specific methods (e.g., t, nsim).
t	A numeric vector of time points at which to calculate survival probabilities. The behaviour for NULL varies by method.
nsim	The number of simulations to generate from the model's posterior distribution. Defaults to 100.

Details

For flexsurvreg objects, parameters are sampled from the asymptotic normal distribution of the maximum likelihood estimates using flexsurv::normboot.flexsurvreg(). If t is NULL, the unique event/censoring times from the model's source data are used.

INLA Method:

The inla method requires the **INLA** package. As it is not available on CRAN, you must install it from its own repository: install.packages("INLA", repos = c(getOption("repos"), INLA = "https://inla.r-inla-download.org/R/stable"), dep = TRUE)

This method is designed for inla objects fitted with a poisson likelihood for piecewise exponential models. It samples from the joint posterior of the baseline hazard to calculate survival probabilities. If t is NULL, the interval cut-points for the baseline hazard from the model are used.

Default Method:

The default method handles pre-computed survival probabilities.

- If Surv is a **vector**, it is treated as a single survival curve (e.g., from a Kaplan-Meier estimate). The function replicates this curve nsim times to form the output matrix.
- If Surv is a **matrix**, it is assumed to already be in the desired (time x simulations) format. The function will simply subset rows based on t.

If t is NULL, a sequence 0, 1, 2, ... is generated based on the length or number of rows of Surv. Note that time points are used as 1-based indices, so t = 0 corresponds to the first row/element.

Value

A matrix with length(t) rows and nsim columns. Each element [i, j] is the survival probability at time t[i] for simulation j.

See Also

survHE::make.surv()

Examples

```
# Define common time points and number of simulations for examples
time_points <- 1:100
n_sim <- 50

#--------------------------------------
## Method for a 'survHE' object
#--------------------------------------
if (rlang::is_installed("survHE") && rlang::is_installed("survival")) {
  library(survHE)
  library(survival)
  data(ovarian)

  # Fit a Weibull model using survHE (with MLE for speed)
  fit_she <- fit.models(
    formula = Surv(futime, fustat) ~ 1,
    data = ovarian,
    distr = "weibull",
    method = "mle"
  )

  # Generate survival probability matrix
  surv_matrix_she <- make_surv(fit_she, t = time_points, nsim = n_sim)
  cat("survHE method output dimensions:", dim(surv_matrix_she), "\n")
}

#--------------------------------------
## Method for a 'flexsurvreg' object
#--------------------------------------
if (rlang::is_installed("flexsurv") && rlang::is_installed("survival")) {
  library(flexsurv)
  library(survival)

  # Fit a log-logistic model using flexsurv
  fit_fsr <- flexsurvreg(
    formula = Surv(futime, fustat) ~ 1,
    data = ovarian,
    dist = "llogis"
  )

  # Generate survival probability matrix
  surv_matrix_fsr <- make_surv(fit_fsr, t = time_points, nsim = n_sim)
  cat("flexsurvreg method output dimensions:", dim(surv_matrix_fsr), "\n")
}

#--------------------------------------
## Default method for a numeric vector (e.g., from a Kaplan-Meier curve)
#--------------------------------------
```

```
if (rlang::is_installed("survival")) {
  library(survival)
  km_fit <- survfit(Surv(futime, fustat) ~ 1, data = ovarian)
  # Extract survival probabilities at our time points
  km_summary <- summary(km_fit, times = time_points)

  # Generate matrix by replicating the single survival curve
 surv_matrix_vec <- make_surv(km_summary$surv, t = 0:(length(km_summary$surv) - 1), nsim = n_sim)
  cat("Default (vector) method output dimensions:", dim(surv_matrix_vec), "\n")
}

#--------------------------------------
## Default method for a matrix (pre-simulated curves)
#--------------------------------------
# Create a sample matrix of survival probabilities (500 time points, 50 simulations)
pre_sim_matrix <- sapply(1:n_sim, function(i) 1 - pweibull(1:500, shape = 1.5, scale = 100 + i))

# Use make_surv to subset the matrix for our desired time points
surv_matrix_mat <- make_surv(pre_sim_matrix, t = time_points)
cat("Default (matrix) method output dimensions:", dim(surv_matrix_mat), "\n")

#--------------------------------------
## Method for 'inla' objects (conceptual example)
#--------------------------------------
## Not run:
if (rlang::is_installed("INLA")) {
  # This method requires a fitted 'inla' object, typically from a
  # piecewise exponential model (poisson likelihood).

  # Assuming 'fit_inla' is a valid model object from INLA:
  # surv_matrix_inla <- make_surv(fit_inla, t = time_points, nsim = n_sim)
  # print(dim(surv_matrix_inla))
}

## End(Not run)
```

`manip_plot` *Blended survival plot with manipulate*

Description

RStudio bug need to run base R first `manipulate(plot(1:x), x = slider(5, 10))`

Usage

```
manip_plot(obs_Surv, ext_Surv, blend_interv)
```

Arguments

obs_Surv	Observed survival
ext_Surv	External survival
blend_interv	Blending interval

Value

Blended survival plot

---

plot.blended *Plot Blended Survival Curves*

Description

S3 method for plotting objects of class blended. This function generates a ggplot2 visualization that displays the mean survival curves and 95% credible intervals for three components:

1. The curve fitted to the original trial data (Data fitting).
2. The curve representing external information (External info).
3. The final, combined blended curve (Blended curve).

Usage

```
## S3 method for class 'blended'
plot(x, alpha = c(0.1, 0.05), ...)
```

Arguments

x	An object of class blended, typically the output of blendsurv().
alpha	A numeric vector of length two specifying the opacity for the credible interval ribbons. The first value (alpha[1]) is for the blended curve, and the second (alpha[2]) is for the trial and external curves.
...	Additional graphical arguments passed to the plot, such as xlim, xlab, or ylab.

Value

A ggplot object representing the survival curves.

See Also

blendsurv(), weightplot()

Examples

```
library(survHE)

## trial data
data("TA174_FCR", package = "blendR")

## externally estimated data
data_sim <- ext_surv_sim(t_info = 144,
                         S_info = 0.05,
                         T_max = 180)

obs_Surv <- fit.models(formula = Surv(death_t, death) ~ 1,
                       data = dat_FCR,
                       distr = "exponential",
                       method = "hmc")

ext_Surv <- fit.models(formula = Surv(time, event) ~ 1,
                       data = data_sim,
                       distr = "exponential",
                       method = "hmc")

blend_interv <- list(min = 48, max = 150)
beta_params <- list(alpha = 3, beta = 3)

ble_Surv <- blendsurv(obs_Surv, ext_Surv, blend_interv, beta_params)

plot(ble_Surv)
```

---

weightplot *Plots the weights for the blending procedure*

Description

Plots the weights for the blending procedure

Usage

```
weightplot(x, ...)
```

Arguments

x	A blended survival curve object obtained from `blendsurv()`
...	Additional arguments

Value

ggplot2 object

See Also

blendsurv()

Index