

Package: multimcm (via r-universe)

October 11, 2026

Title Bayesian Relative Mixture Cure Modelling

Version 0.0.1

Description Jointly models two event time distributions (e.g. time to disease progression and time to death) within a Bayesian relative survival mixture cure model framework, using the ``Stan'' engine called from 'R'. It extends current methods to account for the dependence between event times to leverage additional information to make better inferences and decisions.

URL <https://github.com/StatisticsHealthEconomics/multimcm>,
<https://statisticshealtheconomics.github.io/multimcm/>

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Biarch true

Depends R (>= 4.2.0)

Imports dplyr, ggplot2, glue, gridExtra, rstan (>= 2.18.1),
reformulas, purrr, reshape2, rstanarm, survival, tidybayes,
readr

Suggests testthat, knitr, rmarkdown, tidyr, abind, here, brms,
shinytan, cmdstanr

Remotes stan-dev/cmdstanr

VignetteBuilder knitr

BugReports <https://github.com/StatisticsHealthEconomics/multimcm/issues>

Config/roxygen2/version 8.1.0

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libuv1-dev libxml2-dev libx11-dev zlib1g-dev

Repository <https://n8thangreen.r-universe.dev>

Date/Publication 2026-10-11 08:30:03 UTC

RemoteUrl <https://github.com/statisticshealtheconomics/multimcm>

RemoteRef HEAD

RemoteSha be9f760a571119bb60345cdf7ef7e9a4adc866f2

Contents

.default	2
bmcm_stan	3
cf_forest_cutpoint	5
cf_forest_plot	6
geom_kaplan_meier	7
grid_arrange_shared_legend	7
plot.bmcm	8
precompile_bmcm_model	9
prep_bg_data	10
prep_latent_data	11
prep_S_data	11
prep_S_joint_data	12
prep_S_joint_data.bmcm	12
prep_S_joint_data.default	13
prep_stan_params	14
Index	15

.default	<i>Set default values</i>
----------	---------------------------

Description

The `.default()` function is used internally to mark certain values as default, so that the user may be notified when default values are being used. For example, choosing a default reference treatment for a network, or using default prior distributions. The function `.is_default()` checks whether an argument/object is set to a default value. Neither of these functions are intended to be called by the user.

Usage

```
.default(x = list())  
  
.is_default(x)
```

Arguments

x An object

Value

For `.default()`, an identical object with additional attribute `.default`. For `.is_default()`, a logical value (TRUE or FALSE).

bmcm_stan

*bmcm_stan***Description**

Jointly models two or more event time distributions within a Bayesian relative survival mixture cure model framework. The function generates custom Stan code on-the-fly (or uses precompiled models) and estimates treatment effects simultaneously for both the cure fraction and the latent survival distributions. It allows for complex hierarchical structures in the cure fraction, background mortality adjustment, and multiple parametric families for the latent survival times (e.g., exponential, Weibull, Gompertz).

Usage

```
bmcm_stan(
  input_data,
  formula,
  cureformula = ~1,
  endpoint_var = NULL,
  family_latent = "exponential",
  prior_latent = .default(NA),
  prior_cure = .default(list()),
  centre_coefs = FALSE,
  bg_model = c("bg_distn", "bg_fixed"),
  bg_varname = "bg_rate",
  bg_hr = 1,
  t_max = 70,
  save_stan_code = FALSE,
  read_stan_code = FALSE,
  precompiled_model_path = NA,
  use_cmdstanr = FALSE,
  ...
)
```

Arguments

<code>input_data</code>	A long-format data frame containing the input data.
<code>formula</code>	An R formula object specifying the latent model component.
<code>cureformula</code>	An R formula object specifying the cure fraction model component. Default is <code>~ 1</code> .
<code>endpoint_var</code>	A string specifying the column in <code>input_data</code> that defines the endpoints. If provided, the package correctly identifies the endpoint grouping separately from the cure formula, allowing single shared cure fractions across multiple endpoints. Default is <code>NULL</code> .
<code>family_latent</code>	A string or vector specifying the distribution(s) for the latent model component. Supported values are "exponential", "weibull", "gompertz", "lognormal", "loglogistic", and "gengamma".

prior_latent	An optional prior specification for the latent model component. Default is NA.
prior_cure	An optional list specifying priors for the cure fraction model component. Default is an empty list.
centre_coefs	Logical. If TRUE, coefficients will be centered. Default is FALSE.
bg_model	A string specifying the background model type. Supported values are "bg_distn" and "bg_fixed".
bg_varname	A string specifying the background variable name in input_data. Default is "bg_rate".
bg_hr	A numeric value for the background hazard ratio adjustment. Default is 1.
t_max	A numeric value specifying the maximum time horizon for the model. Default is 70.
save_stan_code	Logical. If TRUE, saves the Stan model code to a file. Default is FALSE.
read_stan_code	Logical. If TRUE, reads the Stan model code from a file instead of generating it dynamically. Default is FALSE.
precompiled_model_path	A string specifying the path to a precompiled model. Default is NA.
use_cmdstanr	Logical. If TRUE, the cmdstanr package is used to fit the model. Default is FALSE.
...	Additional parameters to pass to the Stan sampler.

Value

An object of class `bmcm`, which is a list containing the following components:

- `output`: The fitted Stan model object (either a `stanfit` object from **rstan** or a `CmdStanMCMC` object from **cmdstanr**).
- `mcmc_params`: A list of the MCMC sampling parameters used (e.g., iterations, warmup, chains).
- `call`: The matched call to the function.
- `distns`: A character vector of the latent survival distributions used.
- `inputs`: A list of the formatted data inputs passed directly to the Stan model.
- `input_data`: The original `input_data` data frame provided.
- `formula`: A list containing the parsed cure and latent model formulas.

Examples

```
## Not run:
data("surv_input_data", package = "multimcm")
out <- bmcm_stan(
  input_data = surv_input_data,
  formula = "Surv(time=os, event=os_event) ~ 1",
  cureformula = "~ TRTA + (1 | center_id)",
  endpoint_var = "center_id",
  family_latent = "exponential",
  bg_model = "bg_fixed",
```

```

    bg_varname = "rate",
    t_max = 400
  )

  ## End(Not run)

```

cf_forest_cutpoint	<i>Cure fraction forest plot using cut-point Stan output</i>
--------------------	--

Description

Generates a forest plot of the cure fractions for different treatments using Stan model outputs fitted with specific time cut-points. Useful for comparing cure probabilities across interventions.

Usage

```

cf_forest_cutpoint(
  distns = list(c("exp", "exp"), c("lognormal", "lognormal")),
  folder = NA,
  save_name = c("_12", "_30", "_100"),
  is_hier = TRUE
)

```

Arguments

distns	A list or character vector of distribution names used in the model fits.
folder	A string specifying the directory path containing the saved model output files.
save_name	A string or vector of text strings appended to the file names (e.g., specific cut-points).
is_hier	Logical. Is it a hierarchical model? If TRUE, it accounts for a global grouping parameter.

Value

A ggplot2 object showing the cure fraction forest plot.

Examples

```

## Not run:
# independent model
fp_sep <- cf_forest_cutpoint(folder = "data/dbl_cut/separate", save_name = c("_30", "_12", "_100"))
#ggsave(fp_sep, filename = "plots/forest_plot_cf_sep_cpt.png",
#       dpi = 640, width = 16, height = 14)

# hierarchical model
fp_hier <- cf_forest_cutpoint(folder = "data/dbl_cut/hier", save_name = c("_30", "_12", "_100"))
#ggsave(fp_hier, filename = "plots/forest_plot_cf_hier_cpt.png",

```

```
#      dpi = 640, width = 16, height = 14)

## End(Not run)
```

cf_forest_plot

Cure fraction forest plot using Stan output

Description

Generates a forest plot displaying the estimated cure fractions across all treatments derived from the joint relative survival mixture cure model output.

Usage

```
cf_forest_plot(folder = "data/")
```

Arguments

folder	A string specifying the directory path containing the saved Stan model output files.
--------	--

Value

A ggplot2 object showing the cure fraction forest plot.

Examples

```
## Not run:
# independent model
fp_sep <- cf_forest_plot("data/separate/")
#ggsave(fp_sep, filename = "plots/forest_plot_sep_multimcm.png", dpi = 640, width = 12, height = 8)

# hierarchical model
fp_hier <- cf_forest_plot("data/")
#ggsave(fp_hier, filename = "plots/forest_plot_multimcm.png", dpi = 640, width = 12, height = 8)

## End(Not run)
```

geom_kaplan_meier	<i>Geom for Kaplan-Meier ggplot</i>
-------------------	-------------------------------------

Description

Geom for Kaplan-Meier ggplot

Usage

```
geom_kaplan_meier(out_dat, col = "black", add_marks = TRUE)
```

Arguments

out_dat	bmcm class output list
col	colour string
add_marks	logical

grid_arrange_shared_legend	<i>Use single legend for grid of plots</i>
----------------------------	--

Description

from <https://cran.r-project.org/web/packages/egg/vignettes/Ecosystem.html>

Usage

```
grid_arrange_shared_legend(  
  ...,  
  ncol = length(list(...)),  
  nrow = 1,  
  position = c("bottom", "right", "top")  
)
```

Arguments

...	ggplot objects
ncol	number of columns
nrow	number of rows
position	position of the legend

plot.bmcm

Plot survival curves for joint model and all treatments

Description

Generates a plot of the survival curves for the joint relative survival mixture cure model, overlaying the predicted survival probabilities and (optionally) the original Kaplan-Meier curves for all treatments.

Usage

```
## S3 method for class 'bmcm'
plot(x, ...)

plot_S_joint(
  bmcm_out,
  facet = TRUE,
  annot_cf = FALSE,
  add_km = FALSE,
  add_marks = TRUE,
  ...
)
```

Arguments

x	An object of class <code>bmcm</code>
...	Additional parameters passed to the plotting function.
bmcm_out	bmcm class output list, as returned by <code>bmcm_stan()</code>
facet	Logical. Should the plots for each endpoint be separated into facets? Default is TRUE.
annot_cf	Logical. Annotate the plot with the cure fractions? Default is FALSE.
add_km	Logical. Include an overlaid Kaplan-Meier curve of the raw data? Default is FALSE.
add_marks	Logical. Include Kaplan-Meier censoring marks? Default is TRUE.

Details

Use results of running Stan with `bmcm_stan()` relative survival joint mixture cure model.

Value

A `ggplot2` object showing the survival curves.

Examples

```
## Not run:
data("surv_input_data", package = "multimcm")

out <- bmcm_stan(
  input_data = surv_input_data,
  formula = "Surv(time=os, event=os_event) ~ 1",
  cureformula = "~ TRTA + (1 | center_id)",
  family_latent = "exponential",
  bg_model = "bg_fixed",
  bg_varname = "rate"
)

# Generate the survival plot, facetting by endpoint
surv_plot <- plot_S_joint(out, facet = TRUE, add_km = TRUE)
print(surv_plot)

## End(Not run)
```

precompile_bmcm_model *Precompile bmcm model*

Description

Precompiles the Stan model code generated by `create_stancode` for a given model specification. This is useful when you want to avoid recompiling the model multiple times during repeated analyses or simulations.

Usage

```
precompile_bmcm_model(
  input_data,
  family_latent = "exponential",
  cureformula = ~1,
  model_name = NA,
  use_cmdstanr = FALSE,
  file_path = NA
)
```

Arguments

<code>input_data</code>	Data frame
<code>family_latent</code>	String specifying latent distribution family
<code>cureformula</code>	Formula for cure fraction
<code>model_name</code>	Optional string for model name
<code>use_cmdstanr</code>	Logical. If TRUE, use cmdstanr to compile the model. Default is FALSE.
<code>file_path</code>	Optional string for file path to save compiled model

Value

A compiled stan model object

Examples

```
## Not run:
data("surv_input_data", package = "multimcm")

# Precompile the model
compiled_model <- precompile_bmcm_model(
  input_data = surv_input_data,
  family_latent = "exponential",
  cureformula = "~ TRTA + (1 | center_id)",
  model_name = "my_precompiled_model"
)

# Pass the compiled model file path to bmcm_stan to avoid recompilation
out <- bmcm_stan(
  input_data = surv_input_data,
  formula = "Surv(time=os, event=os_event) ~ 1",
  cureformula = "~ TRTA + (1 | center_id)",
  family_latent = "exponential",
  precompiled_model_path = "./my_precompiled_model.RDS"
)

## End(Not run)
```

```
prep_bg_data
```

```
Prepare background data
```

Description

Background hazard point values.

Usage

```
prep_bg_data(input_data, bg_varname, formula_cure, event_type, suffix = TRUE)
```

Arguments

input_data	Data frame
bg_varname	String for background variable name
formula_cure	Parsed formula list
event_type	String specifying event type
suffix	Logical indicating whether to append suffix

Value

A list with background data

prep_latent_data	<i>Prepare Stan data in latent model</i>
------------------	--

Description

Data specific to end type for Stan input.

Usage

```
prep_latent_data(
  input_data,
  formula_cure,
  formula_latent,
  event_type,
  centre_coefs = FALSE,
  suffix = TRUE
)
```

Arguments

input_data	Data frame
formula_cure	parsed formula
formula_latent	parsed formula
event_type	cluster/group
centre_coefs	Logical
suffix	Logical

Value

List consisting of: sample size, times, censoring indicator, number of covariates, covariates

prep_S_data	<i>Prepare posterior survival data for plotting</i>
-------------	---

Description

Prepare posterior survival data for plotting

Usage

```
prep_S_data(
  stan_extract,
  event_type = NA,
  CI_probs = c(0.025, 0.5, 0.975),
  tx_idx = NA
)
```

Arguments

stan_extract	List of stan output
event_type	String specifying event type
CI_probs	Numeric vector of credible interval probabilities
tx_idx	Numeric vector of treatment indices

Value

dataframe

```
prep_S_joint_data      Prepare data for survival plot
```

Description

Prepare data for survival plot

Usage

```
prep_S_joint_data(bmcm_out, ...)
```

Arguments

bmcm_out	Output of Stan model
...	Additional arguments

```
prep_S_joint_data.bmcm
      Prepare data for survival plot
```

Description

An internal/helper function used to extract and format survival predictions from a fitted bmcm model. It generates a long-format data frame containing the mean survival estimates and their credible intervals across time points.

Usage

```
## S3 method for class 'bmcm'
prep_S_joint_data(bmcm_out, ...)
```

Arguments

bmcm_out	Output list from the bmcm_stan() function.
...	Additional arguments passed to methods.

Value

A data frame containing the summarized survival predictions.

Examples

```
## Not run:
data("surv_input_data", package = "multimcm")
out <- bmcm_stan(
  input_data = surv_input_data,
  formula = "Surv(time=os, event=os_event) ~ 1",
  cureformula = "~ TRTA",
  family_latent = "exponential",
  bg_model = "bg_fixed",
  bg_varname = "rate"
)
plot_data <- prep_S_joint_data(out)
head(plot_data)

## End(Not run)
```

prep_S_joint_data.default
already extracted matrix

Description

already extracted matrix

Usage

```
## Default S3 method:
prep_S_joint_data(bmcm_out, ...)
```

Arguments

bmcm_out	Output of Stan model
...	Additional arguments

prep_stan_params	<i>Default prior latent variable</i>
------------------	--------------------------------------

Description

Default parameters for survival model priors if not supplied.

Usage

```
default_prior_latent(formula_latent, formula_cure)
```

Arguments

- formula_latent Formula for latent
- formula_cure Formula for cure fraction

Value

hyper-parameters

Index

`.default`, [2](#)
`.is_default(.default)`, [2](#)

`bmcm_stan`, [3](#)

`cf_forest_cutpoint`, [5](#)
`cf_forest_plot`, [6](#)

`default_prior_latent`
 (`prep_stan_params`), [14](#)

`geom_kaplan_meier`, [7](#)
`grid_arrange_shared_legend`, [7](#)

`plot.bmcm`, [8](#)
`plot_S_joint(plot.bmcm)`, [8](#)
`precompile_bmcm_model`, [9](#)
`prep_bg_data`, [10](#)
`prep_latent_data`, [11](#)
`prep_S_data`, [11](#)
`prep_S_joint_data`, [12](#)
`prep_S_joint_data.bmcm`, [12](#)
`prep_S_joint_data.default`, [13](#)
`prep_stan_params`, [14](#)