

Package: ambs (via r-universe)

October 7, 2026

Type Package

Title Bayesian Alpha-Mixture Survival Models

Version 0.1.0

Description Implements Bayesian estimation for alpha-mixture survival models with right-censored data. Weibull-Weibull, Gamma-Weibull, and Lognormal-Lognormal component specifications are supported, with all component parameters treated as unknown. The package provides identifiability handling, adaptive Markov chain Monte Carlo sampling, convergence diagnostics, model comparison criteria, and posterior survival, hazard, and density estimation. The methodology extends the framework described by Luan et al. (2026) [doi:10.3390/math14101772](https://doi.org/10.3390/math14101772). Danish Ezwan, David Goldberg, and Ting Huang contributed equally to the package.

License GPL (>= 3)

Depends R (>= 4.0.0)

Imports graphics, stats, utils

Encoding UTF-8

LazyData true

LazyDataCompression xz

Suggests survival, testthat (>= 3.0.0)

Config/testthat/edition 3

NeedsCompilation no

Author Zhexuan Yang [aut, cre], Danish Ezwan [ctb], David Goldberg [ctb], Ting Huang [ctb]

Maintainer Zhexuan Yang <zky5198@psu.edu>

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

Date/Publication 2026-10-06 14:00:02 UTC

RemoteUrl <https://github.com/cran/ambs>

RemoteRef HEAD

RemoteSha 0a5ab095deb283fbfc170e9bb72e7b71828de1e5

Contents

alpmixsurv	2
getcurve	4
mcmcdiag	5
plotmcmc	6
veteran_alpmix	6
Index	8

alpmixsurv	<i>Fit a Bayesian Alpha-Mixture Survival Model</i>
------------	--

Description

Fits a right-censored Bayesian alpha-mixture survival model by Metropolis-Hastings within Gibbs sampling.

Usage

```
alpmixsurv(  
  time,  
  status,  
  model = c("ww", "gw", "ll"),  
  mcmc = list(nburn = 5000, nsamp = 5000, thin = 1),  
  prior = NULL,  
  ident = c("auto", "relabel", "none"),  
  init = NULL  
)
```

Arguments

time	Positive observed survival times.
status	Event indicators: 1 for an event and 0 for right censoring.
model	Component families: Weibull-Weibull ("ww"), Gamma-Weibull ("gw"), or Lognormal-Lognormal ("ll").
mcmc	A list containing non-negative integer nburn and positive integers nsamp and thin. Exactly nsamp draws are retained from one chain. Proposal scales start at the reference simulation values, adapt during burn-in, and then remain fixed.
prior	NULL for defaults, or a named list with entries alpha, p, component1, and component2. alpha uses mean and sd; p uses beta shape1 and shape2. Weibull-Weibull scale parameters use gamma shape and scale. Positive shape parameters use lognormal meanlog and sdlog. Lognormal locations use normal mean and sd; their squared standard deviations use inverse-gamma shape and scale. A legacy list with k, theta, alpha = c(mean, variance), and p is accepted for compatibility.

<code>ident</code>	Identifiability handling. The default "auto" relabels exchangeable Weibull-Weibull and Lognormal-Lognormal components by increasing median survival and runs identifiability diagnostics. "relabel" explicitly requests this relabeling, and "none" disables relabeling and diagnostics.
<code>init</code>	Optional named initial-value list, or a list of four named initial-value lists. If NULL, four dispersed data-based initial values are generated automatically.

Details

Alpha is always estimated as a continuous parameter. At alpha equal to zero and numerically near zero, the likelihood is evaluated stably using the geometric-mixture limit or its continuous expansion. Posterior draws are not rounded or forced to zero. The fitted object reports $\Pr(|\alpha| < 0.01 \mid \text{data})$ as a near-zero interval probability; this is not a posterior point mass at alpha equal to zero.

Relabeling resolves label switching but does not remove structural non-identifiability. The default assessment also checks component overlap, near-boundary mixture weights, and known model-specific degeneracies. Detected problems are recorded in the fitted object, while `summary()` reports the overall identifiability status.

One chain is run by default for a fast initial fit. Single-chain effective sample size is reported, but convergence is marked as not assessed because R-hat requires multiple chains. Use `mcmcdiag()` to run three additional chains and obtain four-chain split R-hat and effective sample-size diagnostics. The default proposals follow the two-component unknown-shape simulations: alpha starts at zero with a normal random-walk standard deviation of 0.05, the mixture weight starts at 0.5 and is updated on the logit scale, positive shape parameters start at 2 and are updated on the log scale, and Weibull scale parameters use relative-variance gamma random walks, whose conditional variance is proportional to the square of the current parameter value. Positive shape parameters are updated by symmetric normal random walks on the log-parameter scale, using posterior targets expressed on that transformed scale. Every proposal scale is adapted in 50-iteration batches during burn-in toward an acceptance rate of 0.44 and is then frozen before retained sampling. Sampling acceptance rates are reported separately from burn-in acceptance rates. Internal sampling diagnostics identify extreme acceptance, low one-chain ESS, non-finite draws, or extreme parameter tails; multi-chain assessment with `mcmcdiag()` remains recommended. For burn-in of at least 100 iterations, the first half of burn-in is also used to select the two transformed parameters most correlated with alpha. A three-dimensional normal random walk for alpha and those parameters is then adapted during the remainder of burn-in and frozen for retained sampling. The original scalar updates remain in place.

WW and GW estimation is performed internally using `time / median(time)`. Their default scale-parameter priors and automatic initial values are defined on this standardized scale. Theta draws, automatic initial values, likelihoods, and model criteria are transformed back to the original time unit before being returned. User-specified theta priors and initial values retain their original-scale interpretation through exact internal transformation. This makes the default computation invariant to whether time is recorded in days, months, or years.

For Lognormal-Lognormal, the corresponding fixed first-chain values are $\mu_1 = 0$, $\mu_2 = -\log(2)$, and $\text{sig}_1 = \text{sig}_2 = 1$. The location parameters use normal random walks with standard deviation 0.10, and positive standard deviations use log-scale random walks with standard deviation 0.10. LL estimation is performed internally on $(\log(\text{time}) - \text{mean}(\log(\text{time}))) / \text{sd}(\log(\text{time}))$. The default working-scale priors are exchangeable, with normal location priors and $\sigma^2 \sim \text{inverse-gamma}(0.5, 0.01)$. This proper heavy-tailed prior allows both narrow and diffuse components. Posterior draws,

initial values, likelihoods, and model criteria are returned on the original time scale. User-specified LL priors and initial values are transformed internally and retain their original-scale interpretation.

Value

An object of class `alpmixsurv` containing posterior draws, acceptance rates, model criteria, the normalized prior, and the supplied data. The object also contains `alpha_near_zero_band` and `alpha_near_zero_probability` for assessing proximity to the geometric mixture. The `posterior_raw` and `identifiability` elements retain the unmodified draws and the identification assessment, respectively. Chain-level draws, initial values, convergence status, calibration and warm-up acceptance rates, and final proposal scales are also retained. The `block_proposals` element records the selected transformed parameters, frozen covariance and scale, and warm-up and retained-sample acceptance rates.

Examples

```
set.seed(1)
data(veteran_alpmix)
fit <- alpmixsurv(
  time = veteran_alpmix$time,
  status = veteran_alpmix$status,
  model = "ll", mcmc = list(nburn = 50, nsamp = 100, thin = 1)
)
summary(fit)
```

getcurve

Plot a Fitted Alpha-Mixture Curve

Description

Evaluates and plots a survival, hazard, or density curve from a fitted alpha-mixture model.

Usage

```
getcurve(
  object, type = c("s", "h", "d"), t = NULL, col = "black",
  add = FALSE, lwd = 1, lty = 1, xlab = "Time", ylab = NULL,
  main = NULL, level = 0.95, estimate = c("median", "mean"), ...
)
```

Arguments

<code>object</code>	A fitted <code>alpmixsurv</code> object.
<code>type</code>	Survival ("s"), hazard ("h"), or density ("d").
<code>t</code>	Optional positive time grid.
<code>col, add, lwd, lty, xlab, ylab, main</code>	Graphical controls.

level	Pointwise posterior credible level.
estimate	Pointwise posterior median or mean.
...	Additional arguments passed to plot.

Details

The curve is evaluated for every posterior draw and summarized pointwise with a credible band. At alpha equal to zero, the geometric-mixture limit is used consistently with the fitted likelihood.

Value

Invisibly, a data frame containing time, posterior estimate, lower limit, and upper limit.

mcmcdiag	<i>Run Multi-Chain Diagnostics for an Alpha-Mixture Fit</i>
----------	---

Description

Runs three additional chains from dispersed automatic initial values and combines them with the original chain for convergence assessment.

Usage

```
mcmcdiag(object, seed = NULL)
```

Arguments

object	A fitted <code>alpmixsurv</code> object containing one chain.
seed	Optional single integer seed for reproducible additional chains.

Details

The initial `alpmixsurv()` fit uses one chain for speed and therefore reports convergence as not assessed. This function adds three chains using the same data, model, priors, and MCMC settings. It then recomputes posterior summaries, identifiability diagnostics, model criteria, split R-hat, and effective sample size. The input object is not modified.

Value

An updated `alpmixsurv` object containing four chains and multi-chain convergence diagnostics.

Examples

```
set.seed(1)
fit <- alpmixsurv(
  c(0.5, 1, 1.5, 2), c(1, 1, 0, 1), model = "ww",
  mcmc = list(nburn = 10, nsamp = 20)
)
fit4 <- mcmcdiag(fit, seed = 2)
fit4$convergence$status
```

plotmcmc	<i>Plot Posterior Samples</i>
----------	-------------------------------

Description

Draws a trace plot or histogram for one retained posterior parameter. Trace plots display all available chains separately after mcmcdiag() has been run.

Usage

```
plotmcmc(object, param, type = c("line", "hist"), ...)
```

Arguments

object	A fitted alpmixsurv object.
param	Name of a posterior parameter.
type	A trace line plot or histogram.
...	Additional graphical arguments.

Value

Invisibly, the plotted posterior sample vector.

veteran_alpmix	<i>Veterans' Administration Lung Cancer Trial Data</i>
----------------	--

Description

A right-censored survival dataset from a randomized trial of two treatments for male patients with advanced, inoperable lung cancer. This bundled copy is provided as a directly runnable example for alpha-mixture survival models.

Usage

```
data(veteran_alpmix)
```

Format

A data frame with 137 observations and 8 variables:

trt Treatment code: 1 for standard and 2 for test treatment.

celltype Tumor cell type: squamous, small-cell, adenocarcinoma, or large-cell.

time Survival or censoring time in days.

status Event indicator: 1 for death and 0 for right censoring.

karno Karnofsky performance score.

diagtime Months from diagnosis to randomization.

age Age in years.

prior Prior therapy code: 0 for none and 10 for prior therapy.

Source

Veterans' Administration Lung Cancer Trial data distributed with the **survival** R package as `veteran`.

Examples

```
data(veteran_alpmix)

set.seed(123)
fit <- alpmixsurv(
  veteran_alpmix$time,
  veteran_alpmix$status,
  model = "l1",
  mcmc = list(nburn = 50, nsamp = 100, thin = 1)
)
summary(fit)
```

Index

* **datasets**

veteran_alpmix, [6](#)

alpmixsurv, [2](#)

getcurve, [4](#)

mcmcdiag, [5](#)

plotmcmc, [6](#)

veteran_alpmix, [6](#)