

Package ‘pcslr’

July 30, 2026

Title Progressive Censoring Schemes with Competitive Latent-Risk

Version 0.1.1

Maintainer Okechukwu J. Obulezi <oj.obulezi@unizik.edu.ng>

Description Implements simulation, numerical maximum likelihood estimation via fourth-order Runge-Kutta path optimization, and high-speed Bayesian Markov Chain Monte Carlo (MCMC) samplers for Weibull lifetimes under progressive censoring setups with competitive latent risks. Both point estimation and interval estimation are provided for the model parameters.

License MIT + file LICENSE

Encoding UTF-8

Imports graphics, Rcpp (>= 1.0.0), stats

LinkingTo Rcpp, RcppArmadillo

Config/roxygen2/version 8.0.0

NeedsCompilation yes

Author Okechukwu J. Obulezi [aut, cre]

Repository CRAN

Date/Publication 2026-07-30 17:10:31 UTC

Contents

fit_pcsclr	2
sim_pcsclr	3
Index	5

fit_pcsclr

*Fit PCS-CLR Parameter Estimation Models***Description**

Fit PCS-CLR Parameter Estimation Models

Usage

```
fit_pcsclr(
  time,
  delta,
  removal,
  method = c("RK4", "Bayes_Kernel"),
  control = list(),
  is_hybrid = FALSE,
  t_max = 3,
  r_star = 0,
  ...
)
```

Arguments

time	Vector of observed lifetime intervals.
delta	Attrition risk indicator matrix.
removal	Active dynamic progressive censorship counts.
method	Selection between "RK4" paths or "Bayes_Kernel" sampler loops.
control	Optional configuration list parameters.
is_hybrid	Boolean indicator for hybrid censoring regimes.
t_max	Truncation time milestone boundary constraint.
r_star	Escaped survival components unfailed beyond cutoff point.
...	Additional structural arguments passed to internal optimization engines.

Value

An object of S3 class "pcsclr_fit" consisting of a list with the following elements:

estimates	A numeric vector containing the estimated Weibull shape (α) and scale (β) parameters.
method	A character string indicating the algorithm used ("RK4" or "Bayes_Kernel").
iterations	Number of iterations to reach convergence (returned only when method = "RK4").
chains	A list of posterior MCMC samples for alpha and beta parameters (returned only when method = "Bayes_Kernel").

Examples

```
# 1. Generate small sample mock variables
set.seed(123)
n <- 30
mock_time <- rweibull(n, shape = 1.5, scale = 2.0)
mock_delta <- sample(c(0, 1), n, replace = TRUE, prob = c(0.2, 0.8))
mock_removal <- rbinom(n, 5, 0.1)

# 2. Fit using 4th-order Runge-Kutta Optimization
res_rk4 <- fit_pcsclr(time = mock_time, delta = mock_delta,
                     removal = mock_removal, method = "RK4")
print(res_rk4$estimates)

# 3. Fit using the C++ Bayesian MCMC Sampler
res_bayes <- fit_pcsclr(time = mock_time, delta = mock_delta,
                       removal = mock_removal, method = "Bayes_Kernel",
                       control = list(M = 1000, burn_in = 200))
print(res_bayes$estimates)
```

sim_pcsclr

*Simulate Datasets Under PCS-CLR Configurations***Description**

Simulate Datasets Under PCS-CLR Configurations

Usage

```
sim_pcsclr(
  n,
  target,
  alpha = 1.8,
  beta = 2.5,
  p = 0.15,
  scheme = c("baseline", "hybrid"),
  cs_layout = c("CS_I", "CS_II"),
  T_max = 3
)
```

Arguments

n	Total initial sample population count.
target	Scheduled tracking truncation parameter target threshold.
alpha	Shape vector coefficient.
beta	Scale distribution tracker.
p	Attrition or competitive latent risk probability.
scheme	Setting options: "baseline" or "hybrid".

cs_layout	Distribution setups: "CS_I" or "CS_II".
T_max	Explicit global temporal limitation window boundary.

Value

An object of S3 class "pcsclr_data" consisting of a list with the following elements:

T_obs	A numeric vector of observed failure times up to the target or truncation cutoff.
delta	A numeric vector of risk indicators corresponding to each observed time.
R_vector	A numeric vector of progressive censoring removal counts at each failure point.
scheme	A character string indicating the active censoring scheme ("baseline" or "hybrid").
is_truncated	A logical flag indicating whether time truncation (T_max) occurred during simulation.
T_max	The temporal limit applied (returned only when is_truncated = TRUE).
R_star	The count of remaining survival components unfailed beyond the truncation cutoff (returned only when is_truncated = TRUE).

Index

`fit_pcsclr`, [2](#)

`sim_pcsclr`, [3](#)