Package: ROCsurvcomp (via r-universe)

October 14, 2026

Title ROC-Based Methods for Comparing Survival Distributions with Right, Left, and Doubly Censored Data

Version 0.1.4

Date 2026-10-13

Maintainer Mohammod Mahmudur Rahman <mahmudur.asif13@gmail.com>

Description Implements nonparametric and semiparametric methods for comparing two survival distributions under non-proportional hazards (non-PH) with right, left, and doubly censored data. The methods are based on the Receiver Operating Characteristic (ROC) curve length and its kernel-based estimate (Bantis et al. (2021) <doi:10.1002/sim.8869>) and the overlap coefficient (OVL) and its kernel-based estimate (Schmid and Schmidt (2006) <doi:10.1016/j.csda.2005.01.014>), as well as a new joint inference approach involving both the kernel-based ROC length and the kernel-based OVL. These methods do not require prior knowledge of the underlying non-PH pattern. The methods accommodate the ``last is censored'' phenomenon for right-censored data by using flexible parametric approaches only for the portion of the tail of the distribution that corresponds to time values beyond the last observed event (see Bantis et al. (2017) <doi:10.1002/sim.7394>). The analogous ``first is censored'' phenomenon is also accommodated for left-censored data.

License GPL-3

Encoding UTF-8

Roxygen list(markdown = TRUE)

Imports survival, interval, stats, utils, ggplot2

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

Depends R (>= 3.5)

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

Repository https://mmrahman13.r-universe.dev

Date/Publication 2026-10-14 01:57:05 UTC

RemoteUrl https://github.com/mmrahman13/rocsurvcomp

RemoteRef HEAD

RemoteSha e1da0041688ecd7bdb4093cfc30ea204a0a16244

Contents

surv.comp 2

Index **5**

`surv.comp` *ROC-Based Methods for Comparing Survival Distributions with Right, Left, and Doubly Censored Data*

Description

Performs nonparametric and semiparametric comparisons of two survival distributions under right, left, or double censoring using ROC-based metrics, including ROC curve length, overlap coefficient (OVL), and a joint ROC length–OVL test.

Usage

```
surv.comp(
  time,
  status,
  group,
  censor_type,
  method,
  n_perm,
  progress = TRUE,
  plot = FALSE
)
```

Arguments

`time`	Numeric vector of observed follow-up times (event or censoring times) for all observations.
`status`	Numeric vector indicating censoring status for each observation: • For `censor_type = "right"`: 0 = event, 1 = right-censored • For `censor_type = "left"`: 0 = event, -1 = left-censored • For `censor_type = "double"`: 0 = event, 1 = right-censored, -1 = left-censored.

`group` Numeric vector indicating the group label for each observation. Must contain exactly two groups coded as 1 and 2.

`censor_type` Character string specifying the type of censoring. Must be either `"right"`, `"left"`, or `"double"`.

`method` Character string specifying the test to perform. Must be one of:

- `"roc_length"`: ROC curve length-based test
- `"ovl"`: overlap coefficient-based test
- `"joint_method"`: joint ROC length-OVL-based test

`n_perm` Integer specifying the number of permutation samples used to compute p-values. A small value is used in the example for computational efficiency; larger values are recommended in practice (typically 50,000 or more) to obtain more stable and reliable inference.

`progress` Logical value indicating whether to display a progress bar during the permutation test. Default is `TRUE`. If `FALSE`, the computation runs silently without showing progress updates.

`plot` Logical value indicating whether to generate a convex hull plot based on the joint ROC length–OVL test. Default is `FALSE`. Plotting is only available when `method = "joint_method"`; otherwise, an error is returned if `plot = TRUE`.

Details

This function implements permutation-based two-sided hypothesis tests for comparing two survival distributions without relying on proportional hazards assumptions. The ROC length measures global separation between survival curves, while the overlap coefficient (OVL) quantifies distributional similarity. Density estimation is performed using nonparametric kernel methods, where the cumulative distribution function (CDF) is estimated using the Kaplan–Meier estimator under right censoring and the Turnbull estimator under left and double censoring.

Value

A list containing:

- `message`: A character string describing the testing procedure.
- `result`: A data frame with rows corresponding to the selected methods (ROC Length, OVL, and/or Joint ROC Length-OVL) and columns:
  - `estimate`: Estimated values of ROC length and/or OVL
  - `p_value`: Permutation-based two-sided p-values.
- `plot`: A `ggplot2` object showing the convex hull visualization of the permutation distribution for the joint ROC length-OVL test. This is returned only when `method = "joint_method"` and `plot = TRUE`; otherwise, it is `NULL`.

References

Pepe MS (2003). *The Statistical Evaluation of Medical Tests for Classification and Prediction*. Oxford University Press.

Bantis LE, Tsimikas JV, Chambers G, Capello M, Hanash S, Feng Z (2021). The length of the ROC curve and the two cutoff Youden index within a robust framework for discovery, evaluation, and cutoff estimation in biomarker studies involving improper ROC curves. *Statistics in Medicine*, 40(7), 1767–1789.

Franco-Pereira AM, Nakas CT, Reiser B, Pardo MC (2021). Inference on the overlap coefficient: The binormal approach and alternatives. *Statistical Methods in Medical Research*, 30(12), 2672–2684.

Examples

```
# Example usage
library(ROCsurvcomp)
library(PWEXP)

# Generating right-censored data with crossing survivals
set.seed(126)
n_trt <- 50
break_trt <- c(2, 4)
rate_trt <- c(log(2)/3, log(2)/7, log(2)/20)
rate.censor_trt <- c(log(2)/55, log(2)/62, log(2)/68)
event_trt <- PWEXP::rpwexp(n_trt, rate = rate_trt, breakpoint = break_trt)
censor_trt <- PWEXP::rpwexp(n_trt, rate = rate.censor_trt, breakpoint = break_trt)
n_ctrl <- 50
rate_ctrl <- log(2)/10
rate.censor_ctrl <- log(2)/58
event_ctrl <- rexp(n_ctrl, rate = rate_ctrl)
censor_ctrl <- rexp(n_ctrl, rate = rate.censor_ctrl)

# Observed time and censoring status (0 = event, 1 = right-censored)
time_trt <- pmin(event_trt, censor_trt)
status_trt <- ifelse(event_trt <= censor_trt, 0, 1)
time_ctrl <- pmin(event_ctrl, censor_ctrl)
status_ctrl <- ifelse(event_ctrl <= censor_ctrl, 0, 1)
time <- c(time_trt, time_ctrl)
status <- c(status_trt, status_ctrl)
group <- c(rep(1, n_trt), rep(2, n_ctrl))

# Run surv.comp()
# Note: n_perm = 10 is used here only for illustration purposes.
# Highly recommended n_perm to be set at 50,000 or more.
surv.comp(time = time, status = status, group = group, censor_type = "right",
          method = "roc_length", n_perm = 10, progress = TRUE, plot = FALSE)
```

Index

surv.comp, 2