

# randomForestSRC: Uno IPCW weighting, stabilization, and fast computation for improved concordance vignette

Hemant Ishwaran (<https://ishwaran.org/ishwaran.html>)

Min Lu (<https://www.luminwin.net/>)

Udaya B. Kogalur

2026-03-28



randomForestSRC  
Fast Unified Forests

## Overview

Harrell's concordance statistic  $C$  (the  $C$ -index) is widely used to quantify discriminative performance in right-censored survival analysis and is used throughout `randomForestSRC` to report out-of-bag (OOB) and test error rates. The package recently introduced two related enhancements for survival and competing risks forests. First, concordance is now computed using the inverse-probability-of-censoring weighted (IPCW) estimator of Uno et-al. (2011), which targets a censoring-free population concordance [1]; legacy unweighted Harrell concordance for survival forests can be recovered by setting `use.uno = FALSE`. Because IPCW weights may become inflated in the heavily censored tail, a weight-stabilization (gating) rule is applied that selects a truncation threshold  $\tau$  using an effective sample size criterion. Second, concordance is computed using the tree-based counting strategy described by [2], with the required rank-count updates and prefix queries implemented using a binary indexed tree [3], replacing the naive  $O(n^2)$  pairwise calculation with a significantly faster  $O(n \log n)$  time. This vignette describes these methodological and computational changes and presents simulation benchmarks illustrating the finite-sample behavior of unweighted and IPCW concordance under different levels of censoring in both survival and competing risk settings.

## Introduction

Let  $\{Y_i, \delta_i, \mathbf{X}_i\}_{i=1}^n$  denote observed survival data, where

$$Y_i = \min(T_i, C_i), \quad \delta_i = I(T_i \leq C_i),$$

with  $T_i > 0$  the event time,  $C_i > 0$  the censoring time, and  $\delta_i = 1$  indicating an observed event. Let  $\hat{\eta}_i$  be a model-based risk score such that larger values correspond to higher event risk; for example, in random survival forests (RSF),  $\hat{\eta}_i$  is the OOB mortality estimate returned by `rfsrc()`.

A central goal of performance assessment is to quantify how well  $\hat{\eta}$  ranks subjects by their latent failure times. For continuous event times, the concordance functional is

$$C(\hat{\eta}) = P(\hat{\eta}_i > \hat{\eta}_j \mid T_i < T_j),$$

the probability that, for a randomly selected pair  $(i, j)$ , the subject who fails earlier is assigned higher predicted risk. For a clear exposition, we present definitions under the simplifying assumption that there are no ties. In applications, ties in observed follow-up times and in predicted risk scores are common; `randomForestSRC` handles these using a deterministic time-ordering and partial-credit scheme described in Section on user interface.

Throughout, we focus on the concordance index  $C$  where larger is better. The exported `get.cindex()` function and the internal performance summaries in `randomForestSRC` return the  $1 - C$ ; this is a monotone reparameterization and follows the package's convention that smaller error is better.

## Harrell's C-statistic

Harrell's estimator replaces unobserved event times by observed follow-up times and restricts comparisons to pairs for which the earlier time corresponds to an observed event. The classical estimator can be written as

$$\hat{C}_H = \frac{\sum_{i \neq j} \delta_i I(Y_i < Y_j) I(\hat{\eta}_i > \hat{\eta}_j)}{\sum_{i \neq j} \delta_i I(Y_i < Y_j)}.$$

The denominator counts the set of pairs in which subject  $i$  experiences an observed event before subject  $j$ 's observed follow-up time; the numerator counts the subset of those pairs concordant with the predicted risk ordering.

When censoring is low, the set of comparable pairs approximates the set of all pairs with  $T_i < T_j$ , and  $\hat{C}_H$  is often a useful discrimination summary. Under heavier censoring, however, the comparable-pair restriction induces a selection that depends on the censoring mechanism. Even under independent censoring,  $\hat{C}_H$  converges to a censoring-dependent limit (a conditional concordance over comparable pairs) and can differ systematically from the target functional  $C(\hat{\eta})$  for a fixed risk score.

## Uno's IPCW concordance

Uno et-al. (2011) proposed an IPCW modification that reweights observed events to represent the full cohort at the corresponding follow-up time [1]. Let

$$G(t) = P(C > t)$$

denote the censoring survival function and let  $\widehat{G}$  be the Kaplan–Meier estimator fit to the reverse-censoring data  $\{Y_i, 1 - \delta_i\}$ . The IPCW weight for subject  $i$  is

$$W_i = \widehat{G}(Y_i^-)^{-2},$$

where  $\widehat{G}(Y_i^-)$  denotes the left limit of the Kaplan–Meier estimate at  $Y_i$ . The squared term arises because, for a pair to be comparable at an event time  $T_i$ , both subjects must be uncensored beyond  $T_i$ , giving probability  $G(T_i)^2$  under independent censoring. Uno’s estimator replaces  $\delta_i$  by the weighted event contribution  $\delta_i^* = \delta_i W_i$ :

$$\widehat{C}_U = \frac{\sum_{i \neq j} \delta_i^* I(Y_i < Y_j) I(\widehat{\eta}_i > \widehat{\eta}_j)}{\sum_{i \neq j} \delta_i^* I(Y_i < Y_j)}.$$

Compared to Harrell’s estimator, the IPCW formulation increases the contribution of events occurring at times where the probability of remaining uncensored is small, i.e., in the tail where  $\widehat{G}(t)$  is small.

Under independent censoring and consistent estimation of  $G$ , the IPCW estimator targets the population concordance functional  $C(\widehat{\eta})$  for a fixed risk score and is consistent for that target [1]. In finite samples, IPCW typically reduces the bias of unweighted concordance that arises when heavy censoring disproportionately removes later events from the comparable-pair set.

## Default weight stabilization (gating)

IPCW weights can be highly variable when  $\widehat{G}(t)$  approaches 0. Because  $W_i$  scales as  $\widehat{G}(Y_i^-)^{-2}$ , even a small number of events can generate extremely large weights if these occur deep in the tail and can lead to unstable finite-sample behavior for  $\widehat{C}_U$ . For this reason, `randomForestSRC` therefore applies a default stabilization (or ) rule.

## Gated weights and a cap on tail inflation

In finite samples, a small number of late events can yield extremely large Uno weights when  $\widehat{G}(Y_i^-)$  is small, which can destabilize  $\widehat{C}_U$ . To avoid this, `randomForestSRC` introduces a gate  $\tau > 0$  and uses the stabilized weight vector

$$\widetilde{W}_i(\tau) = \widehat{G}(Y_i^-)^{-2} I\{\widehat{G}(Y_i^-) \geq \tau\} + \epsilon_{\text{keep}} I\{\widehat{G}(Y_i^-) < \tau\},$$

where  $\epsilon_{\text{keep}}$  is a very small constant (defaulting to machine precision). The stabilized IPCW concordance replaces  $\delta_i^* = \delta_i W_i$  in Section on Uno’s IPCW concordance by  $\delta_i \widetilde{W}_i(\tau)$ :

$$\widehat{C}_U^{(\tau)} = \frac{\sum_{i \neq j} \delta_i \widetilde{W}_i(\tau) I(Y_i < Y_j) I(\widehat{\eta}_i > \widehat{\eta}_j)}{\sum_{i \neq j} \delta_i \widetilde{W}_i(\tau) I(Y_i < Y_j)}.$$

Because  $\widehat{G}(Y_i^-) \geq \tau$  implies  $\widehat{G}(Y_i^-)^{-2} \leq \tau^{-2}$ , the gate caps the magnitude of the non-negligible event weights by  $\tau^{-2}$  and prevents a small number of tail events from dominating the criterion. Events with  $\widehat{G}(Y_i^-) < \tau$  receive negligible weight, so their event contribution is effectively removed. The small positive value  $\epsilon_{\text{keep}}$  is used (instead of a literal zero) so that these observations remain available rather than excluding the observation entirely.

## Automatic choice of $\tau$ using an effective sample size constraint

The gate  $\tau$  is selected on the training data using only the distribution of event-time weights. Let  $\mathcal{E} = \{i : \delta_i = 1\}$  denote the set of observed events and  $d = |\mathcal{E}|$  its size. For a candidate threshold  $\tau$ , define the retained event-time weights

$$W_i(\tau) = \widehat{G}(Y_i^-)^{-2} I\{\widehat{G}(Y_i^-) \geq \tau\}, \quad i \in \mathcal{E}.$$

Define the effective sample size functional-[4]

$$\text{ESS}(W(\tau)) = \frac{(\sum_{i \in \mathcal{E}} W_i(\tau))^2}{\sum_{i \in \mathcal{E}} W_i(\tau)^2}.$$

The default ESS target is specified by fixed constants `ess_frac` and `ess_min`:

$$\begin{aligned} d &:= \sum_{i=1}^n I(\delta_i = 1), \\ \text{ESS}_{\text{target}} &:= \max\{\text{ess\_min}, \lceil \text{ess\_frac} \cdot d \rceil\}, \\ \text{ess\_frac} &= 0.20, \quad \text{ess\_min} = 20. \end{aligned}$$

The automatic gating rule chooses the smallest  $\tau$  such that

$$\text{ESS}(W(\tau)) \geq \text{ESS}_{\text{target}}.$$

Because  $W_i(\tau)$  is monotone in  $\widehat{G}(Y_i^-)$ , this procedure is equivalent to dropping the largest event weights (the smallest values of  $\widehat{G}(Y_i^-)$ ) until the ESS constraint is satisfied;  $\tau$  is the corresponding cutoff on  $\widehat{G}(Y_i^-)$ .

# User interface and implementation in randomForestSRC

For survival and competing risks forests, the package now adopts a default concordance-based performance calculation using Uno's IPCW estimator. This affects all concordance-derived performance summaries, including OOB and test error rates, as well as all downstream calculations based on those values. The legacy unweighted Harrell concordance can be recovered by setting `use.uno = FALSE` when fitting a survival forest:

```
fit <- rfsrc(Surv(time, status) ~ ., data = dat, use.uno = FALSE)
```

## Standalone calculations

For standalone computation of concordance for a given outcome and risk score, the exported `get.cindex()` provides a direct interface to the native concordance routines:

```
err <- get.cindex(time, censoring, predicted, weight = NULL, fast = TRUE)
C <- 1 - err
```

The arguments correspond to the observed outcomes, risk score, and (optionally) precomputed IPCW weights:

$$\begin{aligned} \text{time}_i &= Y_i \\ \text{censoring}_i &= \delta_i \\ \text{predicted}_i &= \hat{\eta}_i \\ \text{weight}_i &= W_i^{(\tau)} && \text{(optional)} \\ \text{fast} &\in \{\text{TRUE}, \text{FALSE}\} && \text{(optional)}. \end{aligned}$$

For large samples, `fast = TRUE` yields major speedups by using the Fenwick-tree implementation (if not specified, by default the fast version is used when  $n$  exceeds 500 or the number of events exceed 250). When `weight = NULL`, `get.cindex()` computes unweighted Harrell concordance; when `weight` is supplied, it computes Uno IPCW concordance using the provided weights (and therefore reflects any stabilization encoded in those weights).

## Handling ties

So far our definitions have assumed continuous time and strictly ordered risk scores. The `randomForestSRC` concordance implementation handles time ties and prediction ties using the following set of conventions.

1. Pairs with equal observed times are handled by imposing an ordering on  $(Y, \delta)$  at each unique time: events are treated as occurring before censorings recorded at the same time. Thus, if  $Y_i = Y_j$  with  $(\delta_i, \delta_j) = (1, 0)$ , the pair is treated as comparable with subject  $i$  the earlier event.
2. If  $\hat{\eta}_i = \hat{\eta}_j$  for a comparable pair, the pair contributes a half-credit to the numerator.

## Train/test concordance using `get.cindex()`

The exported function `get.cindex()` is a direct interface to the native concordance routines used by `randomForestSRC`. Advanced users may use this interface to reproduce (or customize) the concordance calculations performed internally for OOB and test performance when calling `rfsrc()` and `predict.rfsrc()`.

In the Uno IPCW formulation, concordance weights depend on the censoring survival function  $G(t) = P(C > t)$ . When evaluating on an external test set, `randomForestSRC` estimates  $G$  on the training data and then evaluates  $\hat{G}(\tau)$  at test follow-up times to construct test weights, using the same stabilization (gating) threshold selected on the training sample. This avoids reusing test outcomes to estimate the censoring model.

The code below illustrates a train/test workflow on the `pbcc` data using the `get.cindex()` function. The example makes use of helpers `get.uno.weights.train()` and `get.uno.weights.test()` which are currently internal (non-exported) functions that mirror the package's default construction of stabilized Uno IPCW weights. The training call returns both the weight vector and a fitted censoring model (including the automatically selected gate  $\tau$ ), which can then be reused to evaluate  $\hat{G}(\tau)$  and construct weights on an external test set using the test call.

For survival outcomes, `predicted` is a vector risk score. For competing risks outcomes (discussed next), `predicted` is a matrix with one column per event type, and `get.cindex()` returns a vector of cause-specific concordance errors. The same train/test weight construction applies, since the censoring survival model does not depend on event type.

```

library(randomForestSRC)

## PBC data: remove NAs, split train/test
data(pbc, package = "randomForestSRC")
pbc.clean <- na.omit(pbc)

## right-censoring indicator (1=event, 0=censor)
pbc.clean$status <- as.integer(pbc.clean$status > 0)

set.seed(1)
trn <- sample(1:nrow(pbc.clean), nrow(pbc.clean)/2, replace = FALSE)
pbc.trn <- pbc.clean[trn, ]
pbc.tst <- pbc.clean[-trn, ]

## Fit forest on training data (Uno weighting is default)
fit <- rfsrc(Surv(days, status) ~ ., data = pbc.trn)

## outcomes and risk scores
time.trn <- fit$yvar[,1]
status.trn <- fit$yvar[,2]
risk.trn <- fit$predicted.oob

## Uno weights estimated from the training censoring distribution
uno.trn <- randomForestSRC::get.uno.weights.train(time.trn, status.trn)
w.trn <- uno.trn$weight

C.harrell.trn <- 1 - get.cindex(time.trn, status.trn, risk.trn)
C.uno.trn <- 1 - get.cindex(time.trn, status.trn, risk.trn,
                           weight = w.trn)

## ----- test concordance (external risk) -----
pred <- predict(fit, newdata = pbc.tst)

time.tst <- pred$yvar[,1]
status.tst <- pred$yvar[,2]
risk.tst <- pred$predicted

## Construct test weights from the training censoring model and gate
w.tst <- randomForestSRC::get.uno.weights.test(time.tst, uno.trn$fit)

C.harrell.tst <- 1 - get.cindex(time.tst, status.tst, risk.tst)
C.uno.tst <- 1 - get.cindex(time.tst, status.tst, risk.tst,
                           weight = w.tst)

c(C.harrell.trn = C.harrell.trn,
  C.harrell.tst = C.harrell.tst,
  C.uno.trn = C.uno.trn,
  C.uno.tst = C.uno.tst)

C.harrell.trn C.harrell.tst C.uno.trn C.uno.tst
0.8038976 0.8277734 0.7953605 0.7817458

```

The above code shows train/test example using `get.cindex`. The censoring survival function  $\widehat{G}(t)$  is estimated on the training sample, yielding Uno inverse-probability-of-censoring weights  $W_i = \widehat{G}(Y_i^-)^{-2}$  together with an automatically chosen tail gate (truncation threshold)  $\tau$  for stability. The fitted censoring model and the same  $\tau$  are then reused to construct test-set weights, enabling computation of both Harrell's  $C$  (unweighted) and Uno's IPCW  $C$  (gated) on training (using out-of-bag risk scores) and on an independent test set (using predicted risk scores).

## Extensions to competing risks

We briefly describe how the above ideas extend to competing risks. Let  $K \geq 2$  denote the number of event types and write

$$Y_i = \min(T_i, C_i), \quad \delta_i = I(T_i \leq C_i), \quad D_i = \delta_i J_i \in \{0, 1, \dots, K\},$$

where  $J_i \in \{1, \dots, K\}$  is the latent event type and  $D_i = 0$  indicates censoring. For each event type  $k$ , let  $\widehat{\eta}_{ik}$  denote a cause-specific risk score, with larger values indicating higher predicted risk for event  $k$ . In `randomForestSRC`,  $\widehat{\eta}_{ik}$  is derived from an event-specific ensemble yielding cause- $k$  mortality risk scores [5], but the concordance definition below applies to any monotone risk score.

## Comparable pairs and the cause-specific concordance functional

For a fixed cause  $k$ , a pair of subjects can be informative in two distinct ways. Ignoring ties for notational clarity, define the following (directed) comparability indicators:

$$A_{ij}^{(1)}(k) = I(J_i = k, T_i < T_j), \quad A_{ij}^{(2)}(k) = I(J_i = k, J_j \neq k, T_j \leq T_i).$$

Type-(1) pairs compare a type- $k$  failure to a subject who is still event-free at  $T_i$ . Type-(2) pairs compare a type- $k$  failure to a subject who experiences a competing event before (or at)  $T_i$ ; such pairs are informative because the competing event precludes a type- $k$  failure occurring earlier than  $T_i$ .

Define  $\psi(a, b) = I(a > b) + \frac{1}{2}I(a = b)$  to be the concordance score for a pair of risk scores. The (population) cause-specific concordance for event  $k$  is then

$$C_k(\widehat{\eta}) = \frac{E\left[\sum_{j \neq i} \{A_{ij}^{(1)}(k) + A_{ij}^{(2)}(k)\} \psi(\widehat{\eta}_{ik}, \widehat{\eta}_{jk})\right]}{E\left[\sum_{j \neq i} \{A_{ij}^{(1)}(k) + A_{ij}^{(2)}(k)\}\right]}.$$

In applications, ties in the risk scores are handled directly by  $\psi$ . Ties in the observed follow-up times are handled by adopting a deterministic time-ordering convention (event before censoring at the same recorded time) together with partial-credit rules for event–event time ties similar to Section on user interface.

## Legacy conditional concordance

Previously, concordance for competing risks in `randomForestSRC` was computed by reducing the competing risks problem to a right-censored problem. For event  $k$ , only subjects with  $D_i \in \{0, k\}$  were retained (censoring or event  $k$ ), subjects experiencing competing events were discarded, and Harrell's estimator was applied to this subset. This produces a conditional discrimination measure that ignores comparisons involving subjects who experience competing events. However, for backwards compatibility, we continue to support this calculation, and revert to the legacy version if the user chooses the option `use.uno = FALSE` when fitting a competing risk forest.

## IPCW (Uno-style) concordance for competing risks

The current default is an IPCW estimator that retains both comparison types and corrects for censoring. Let  $G(t) = P(C > t)$  denote the censoring survival function and let  $\widehat{G}(t)$  be the reverse Kaplan–Meier estimate computed from the data, treating any observed event (of any type) as a failure in the censoring model. As in the survival setting, we evaluate left limits  $\widehat{G}(Y_i^-)$  and apply the same ESS-based gating rule of Section on default weight stabilization to select  $\tau$  and construct stabilized weights  $\widehat{W}_i(\tau)$ .

For notational convenience, define

$$W_i^{(2)}(\tau) \equiv \widehat{W}_i(\tau), \quad W_i^{(1)}(\tau) \equiv \{\widehat{W}_i(\tau)\}^{1/2},$$

so that, in the ideal IPCW form,  $W^{(2)}(t) = 1/G(t)^2$  and  $W^{(1)}(t) = 1/G(t)$  (up to the gating convention in Section on default weight stabilization).

For a fixed event type  $k$ , introduce the index sets

$$\mathcal{R}_i \equiv \{j : Y_j > Y_i\}, \quad \mathcal{C}_{ik} \equiv \{j : D_j \notin \{0, k\}, Y_j \leq Y_i\},$$

(with minor tie adjustments as described earlier). The cause-specific IPCW concordance estimator can be written as

$$\widehat{C}_k = \frac{\widehat{N}_{k,1} + \widehat{N}_{k,2}}{\widehat{D}_{k,1} + \widehat{D}_{k,2}},$$

where

$$\begin{aligned} \widehat{N}_{k,1} &= \sum_{i:D_i=k} W_i^{(2)}(\tau) \sum_{j \in \mathcal{R}_i} \psi(\widehat{\eta}_{ik}, \widehat{\eta}_{jk}), \\ \widehat{D}_{k,1} &= \sum_{i:D_i=k} W_i^{(2)}(\tau) \sum_{j \in \mathcal{R}_i} 1, \\ \widehat{N}_{k,2} &= \sum_{i:D_i=k} W_i^{(1)}(\tau) \sum_{j \in \mathcal{C}_{ik}} W_j^{(1)}(\tau) \psi(\widehat{\eta}_{ik}, \widehat{\eta}_{jk}), \\ \widehat{D}_{k,2} &= \sum_{i:D_i=k} W_i^{(1)}(\tau) \sum_{j \in \mathcal{C}_{ik}} W_j^{(1)}(\tau). \end{aligned}$$

The type-(1) terms compare a type- $k$  event to subjects still event-free at  $Y_i$  and use the squared IPCW weights  $W^{(2)}$ , mirroring the right-censored IPCW concordance. The type-(2) terms compare a type- $k$  event to subjects with earlier competing events and use product weights  $W_i^{(1)}(\tau) W_j^{(1)}(\tau)$ , corresponding to  $1/\{\widehat{G}(Y_i^-) \widehat{G}(Y_j^-)\}$ .

### Cite this vignette as

H. Ishwaran, M. Lu, and U. B. Kogalur. 2026. "randomForestSRC: Uno IPCW weighting, stabilization, and fast computation for improved concordance vignette."  
<http://randomforestsrc.org/articles/uno.html> (<http://randomforestsrc.org/articles/uno.html>).

```
@misc{HemantUNO,
  author = "Hemant Ishwaran and Min Lu and Udaya B. Kogalur",
  title = "{randomForestSRC}: Uno IPCW weighting, stabilization, and fast computation",
  year = {2026},
  url = {http://randomforestsrc.org/articles/uno.html},
  howpublished = "\url{http://randomforestsrc.org/articles/uno.html}",
  note = "[accessed date]"}
}
```

## References

- Uno H, Cai T, Pencina MJ, D'Agostino RB, Wei L-J. On the c-statistics for evaluating overall adequacy of risk prediction procedures with censored survival data. *Statistics in medicine*. 2011;30:1105–17.
- Therneau T. Concordance. Vignette for the survival package. 2024.
- Fenwick PM. A new data structure for cumulative frequency tables. *Software: Practice and experience*. 1994;24:327–36.
- Kish L. Survey sampling. New York: John Wesley & sons. Am Polit Sci Rev. 1965;59:1025.
- Ishwaran H, Gerds TA, Kogalur UB, Moore RD, Gange SJ, Lau BM. Random survival forests for competing risks. *Biostatistics*. 2014;15:757–73.