

concrete R Paper: Title TBD

by David Chen, Thomas Gerds, Helene Rytgaard, Maya L. Petersen, Mark van der Laan, ...

Abstract Competing risks are a common but under-addressed feature in biomedical survival studies. This article outlines a causal inference workflow that illuminates whether or not the researchers' question of interest involves competing risks and introduces the R package **concrete** which implements a targeted maximum likelihood estimator for the cause-specific absolute risks for time-to-event outcomes measured in continuous or near-continuous time. The package can be used for survival analyses with or without competing risks and allows users to estimate causally-interpretable estimands such as risk ratios and risk differences using possibly misspecified cause-specific Cox models. Point estimates can be produced using G-formula plug-in or TMLE, and influence curve-based asymptotic inference will be provided for TMLE. Influence curve-based simultaneous inference is also possible if multiple times or events are targeted. This paper will demonstrate the use of **concrete** using the PBC dataset.

Recently targeted maximum likelihood-based estimation (TMLE) has been used to develop estimators for cause-specific absolute risks of time-to-event outcomes measured in continuous time. **concrete** implements in R a continuous-time TMLE for cause-specific absolute risks under static and dynamic interventions on a point treatment. **concrete** can also return estimates and asymptotic inference for contrastive parameters such as risk differences and risk ratios. Risk estimands can be jointly targeted at multiple times, and in the case of competing risks, multiple cause-specific risks can be jointly targeted. In this paper we review the TMLE methodology implemented in **concrete** and demonstrate its use and its features.

1 Introduction

In many biomedical survival applications, study subjects are susceptible to competing risks. Recognizing this, authors have cautioned against the use of the standard survival estimands (Andersen et al. (2012), Koller et al. (2012), Austin and Fine (2017)) and several methods have been developed for analyzing competing risks, including Fine-Gray regression (Fine and Gray (1999)), pseudovalue regression (Klein and Andersen (2005)), and direct binomial regression (Scheike et al. (2008), Gerds et al. (2012)). Nevertheless, reviews of clinical literature by Koller et al. (2012) and Austin and Fine (2017) found that most published trials failed to adequately address the effect of potential competing risks in their studies. We suggest that this pattern results not from a lack of awareness, but rather from understandable confusion about when and how to choose an appropriate competing risks method. The remedy we propose is a causal roadmap (Petersen and van der Laan (2014)), guiding the process of 1) transforming a clinical question into a rigorously defined causal estimand, 2) translating that causal quantity into a statistical estimand while clearly addressing the necessary assumptions for doing so, and 3) estimating the target statistical quantity via a fully prespecified analysis that provides reliable inference and achieves desirable statistical properties such as efficiency and consistency. In Section 2 we demonstrate a survival competing risks analysis following this roadmap and show how careful causal reasoning can clarify the process of making desirable statistical choices.

concrete is an R package written to help researchers answer causal questions about the effect of interventions on time-to-event outcomes measured in continuous time. Therefore, the package references concepts from causal inference even though **concrete** can be used purely for statistical inference without causal ramifications. When we analyze data to guide future actions, we are in essence asking causal questions - questions that can be made rigorous with the formal frameworks developed in recent decades (Pearl et al. (2016), Holland (1986)). Unfortunately the transition from causal thinking to statistical estimation is not always straightforward, and so a cohesive causal inference workflow ideally follows a structured roadmap (Petersen and van der Laan (2014)) that begins with 1) the causal model and causal estimands, 2) moves through identification to a statistical model and estimands, and then finally 3) performs estimation and provides inference. In subsequent sections we will briefly discuss these first two stages as they pertain to analyses of time-to-event data, but the majority of this manuscript will focus on the statistical estimation methodology implemented in **concrete**. Rigorous and detailed discussions of these first two stages can be found elsewhere (Laan and Rose (2011), Laan and Rose (2018), Rytgaard et al. (2021)).

1.1 What is in this manuscript

We write for readers looking for a hands-on introduction to the one-step targeted maximum likelihood estimation (TMLE) method for continuous time survival analysis described in Rytgaard and van der Laan (2021), as well as for readers wishing to use **concrete** for their own analyses. In Section 2 we briefly overview the one-step TMLE method as it is implemented in **concrete** and in Section 3 we demonstrate a competing risks analysis using **concrete**. For a full and rigorous treatment of the one-step TMLE for continuous-time survival and competing risks, see Rytgaard and van der Laan (2021) and Rytgaard et al. (2021).

1.2 What can concrete be used for

concrete can be used for estimands derived from cause-specific absolute risks (e.g. risk ratios and risk differences) under static and dynamic interventions on binary treatments given at baseline. The currently implemented estimators are the one-step TMLE and a g-formula plug-in, and both can be applied to data involving baseline covariate confounding, right-censoring and competing risks.

concrete cannot yet investigate stochastic interventions or interventions on multinomial and continuous treatments. Currently only Cox models can be used for initial estimation of conditional hazards though the incorporation of penalized Cox (coxnet) and hazard estimators based on highly adaptive lasso (HAL) are planned in a future package version. Methods have not yet been implemented to account for time-dependent treatments (e.g. drop-in) or time-dependent confounding.

concrete is not meant to be used for left truncation (i.e. delayed entry), interval censored data, or clustered data; and it cannot assess the assumptions needed for causal identification.

1.3 How concrete relates to other peoples work

concrete is the first general use R package implementing TMLE for continuous-time survival, with **continuousTMLE** being the methods development package accompanying (Rytgaard and van der Laan (2021), Rytgaard et al. (2021)). The **ltmle** (Schwab et al. (2020)), **stremr** (Sofrygin et al. (2017)), and **survtmle** (Benkeser and Hejazi (2019)) packages either natively or can be adapted to perform discrete-time TMLE for absolute risks of right-censored survival outcomes; **ltmle** and **stremr** use the method of iterated expectations while **survtmle** can target the hazard-based survival formulation. Notably these packages all implement discrete-time TMLEs and would necessitate discretizing continuous-time data which can negatively impact estimator performance.

The *Causal Inference* CRAN Task View shows just **riskregression** (Gerds et al. (2022)) as expressly estimating treatment effect estimands in survival settings. **riskregression** implements the g-formula plug-in, IPTW, and double-robust AIPTW estimators. The *Survival* CRAN Task View shows no packages to be implementing efficient semi-parametric estimators for survival estimands, though the non-CRAN packages **adjustedCurves** (Denz et al. (2022)) and **CFsurvival** (Westling et al. (2021)) implement the AIPTW and a cross-fitted doubly-robust estimator respectively.

1.4 What concrete does

concrete implements the one-step TMLE developed in Rytgaard and van der Laan (2021) to estimate cause-specific absolute risks for interventions on point treatments and covariate measured at baseline. Asymptotic inference for TMLE estimates are derived from the variance of the efficient influence curve (EIC) (Laan and Robins (2003)). **concrete** can also output estimates of risk differences and risk ratios, computed from absolute risk estimates with inference derived using the delta method.

Broadly speaking, this one-step TMLE procedure consists of two stages: 1) an initial estimation of nuisance parameters and 2) a targeted update of the initial estimators to solve the EIC of the target statistical estimand (Laan and Robins (2003), Kennedy (2016)).

For TMLE to be consistent and efficient, the initial nuisance parameter estimators must achieve certain convergence requirements. Unfortunately it is often impossible to know in advance which estimator is best suited for a particular estimation problem, and so in **concrete** nuisance parameters are estimated using Super Learning, a cross-validated machine learning ensemble algorithm with oracle guarantees (Laan et al. (2007), Polley et al. (2021), Laan and Dudoit (2003), Vaart et al. (2006)). Super Learners with robust candidate libraries and valid loss functions give users the best chance of achieving the necessary conditions for TMLE's asymptotic properties. Guidance on how to best specify Superlearner estimators are discussed further in Phillips et al. (2022). Whenever possible, Super Learner libraries should include the highly adaptive lasso (HAL) which achieves the needed convergence rate (Laan (2017); Benkeser and Van Der Laan (2016); Rytgaard et al. (2021)) for TMLE's asymptotic properties.

The subsequent targeted update is based in semi-parametric efficiency theory (Laan and Rose (2011), Kennedy (2016)), specifically that a regular, asymptotically linear estimator of a statistical estimand is efficient if its influence function is equal to the target estimand's EIC. By updating initial estimators of nuisance parameters to solve the EIC, TMLE can use flexible machine-learning algorithms for initial estimation, and still recover asymptotically valid inference. Contrary to other semi-parametric efficient estimators based on the influence function, TMLE is a plug-in estimator

1.5 concrete in a nutshell - PBC example

```

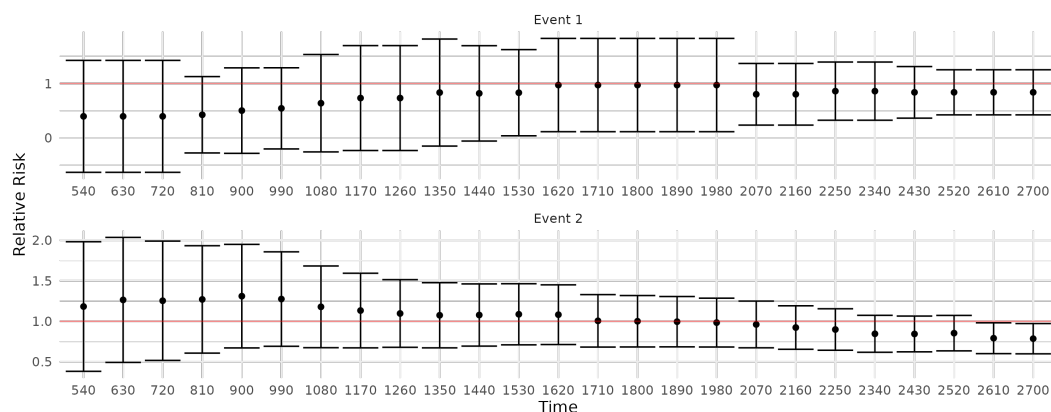
# Prepare Data
library(concrete)
library(data.table)
library(tidyverse)
set.seed(12345)
data <- as.data.table(survival::pbc)
data <- data[!is.na(trt), ][, trt := trt - 1]
data <- data[, c("time", "status", "trt", "age", "sex", "albumin")]

# Specify Analysis
ConcreteArgs <- formatArguments(DataTable = data,
                                EventTime = "time",
                                EventType = "status",
                                Treatment = "trt",
                                Intervention = makeITT(),
                                TargetTime = 90 * (6:30),
                                TargetEvent = 1:2,
                                MaxUpdateIter = 500)

# Compute
ConcreteEst <- doConcrete(ConcreteArgs)

# Return Output
ConcreteOut <- getOutput(ConcreteEst, Estimand = "RD")
plot(ConcreteOut, Estimand = "RD", GComp = TRUE, ask = FALSE)

```



2 Concepts

2.1 The Targeted Learning Roadmap

A basic targeted learning roadmap for analyzing continuous-time survival or competing risks consists of:

1. Defining the causal model and specifying a causal estimand (e.g. causal risk difference at time t). Considerations include defining time zero and a time horizon, identifying the intervention (i.e. treatment) variable, specifying the desired intervention(s), specifying the event(s) of interest.
2. Defining the statistical model and the statistical estimand, while stating and discussing the assumptions necessary for the statistical estimand to identify the causal estimand. Considerations include identifying confounding variables and sources of right-censoring, establishing positivity for censoring and desired interventions, and formalizing knowledge about the statistical model (e.g. dependency structures or functional structures such as proportional hazards).
3. Performing estimation and providing inference. Considerations include choosing an estimator with desirable properties (e.g. consistency and efficiency within a desired class), using outcome-blind simulations to practically assess the estimator's robustness and usability, and pre-specifying the estimation procedure to avoid misleading inference.

In the following sections we discuss the steps in this roadmap in more detail.

2.2 The Causal Model: Counterfactuals, Interventions, and Causal Estimands

With time-to-event data, the counterfactual outcomes are how long it takes for some event(s) occur if subjects hypothetically receive some intervention. Let A be this intervention variable and let d be the intervention rule, i.e. the function that assigns values to A . The simplest interventions are static rules to set A to some value a , $d = a$. More flexible dynamic treatment rules might assign treatments based on subjects' baseline covariates, while stochastic treatment rules incorporate randomness may even depend on the natural treatment assignment mechanism. Whatever the desired intervention, let d represent the intervention of interest and let W represent subjects' baseline covariates. Then a counterfactual survival dataset with J competing events, an intervention d delivered at baseline (i.e. time zero), and a time horizon of t_{max} takes the form:

$$X = (T^d, \Delta^d, W) \quad (1)$$

where $T^d \in (0, t_{max}]$ is earliest occurrence of any of the J events under intervention d , and $\Delta^d \in \{1, \dots, J\}$ shows which of the J events occurred first under intervention d . Notably, we do not include censoring in this counterfactual data and instead isolate just those events that experimenters would want to observe in their ideal hypothetical experiment. For ideal experiments tracking just one event, the causal setting is one of classic survival; if instead mutually exclusive events would be allowed to compete, then the causal setting is one with competing risks.

With the counterfactual data defined, causal estimands can then be specified as functions of this data. For instance, the causal event- j relative risks under some interventions d^* versus d^{**} can be written as Eq. (2)

$$\Psi_{F_{j,t}}^{\mathcal{F}} = P(T^{d^*} \leq t, \Delta^{d^*} = j) / P(T^{d^{**}} \leq t, \Delta^{d^{**}} = j) \quad (2)$$

and the causal difference in event-free survival at time t as Eq. (3).

$$\Psi_{S_t}^{\mathcal{F}} = P(T^{d^*} < t) - P(T^{d^{**}} < t) \quad (3)$$

Researchers may be interested in these estimands at a single endpoint or at many timepoints, potentially even full risk curves over a time interval (e.g. $\Psi_{F_{j,t}}^{\mathcal{F}} : t \in (0, t_{max}]$). In the case of competing risks, researchers may also be simultaneously interested in multiple cause-specific estimands (e.g. $\Psi_{F_{j,t}}^{\mathcal{F}} : j \in 1, \dots, J$). These estimands, and others, can all be rigorously defined as functions of the counterfactual data; efforts can then be made in the next step to identify these causal estimands with functions of observed data, i.e. statistical estimands.

2.3 Observed Data, Identification, and Statistical Estimands

Observed time-to-event data with J competing events might be written as:

$$O = (\tilde{T}, \tilde{\Delta}, A, W) \quad (4)$$

where $\tilde{T} \in (0, t_{max}]$ is the earlier of the first event time T or the right-censoring time C , $\tilde{\Delta} \in \{0, \dots, J\}$ indicates which event occurs (with 0 indicating right-censoring), A is the observed treatment and W is the set of baseline covariates.

To link causal estimands such as Eq. (2) and (3) to statistical estimands, we then need several untestable identification assumptions to hold: consistency, positivity, no unmeasured confounding, and conditionally independent censoring (For more details, see Section 3 in [Rytgaard et al. \(2021\)](#)). Given these assumptions, the cause- j absolute risk at time t under intervention d can be identified by the g-computation formula

$$\begin{aligned} F_j^d(t) &= \mathbb{E}_{\mathcal{W}} \left[\mathbb{E}_{\pi^d} [F_j(t \mid a, w)] \right] \\ &= \mathbb{E}_{\mathcal{W}} \left[\int_{\mathcal{A}} \left[\int_0^t \lambda_j(s \mid a, w) S(s- \mid a, w) ds \right] \pi^d(a \mid w) da \right] \end{aligned} \quad (5)$$

where the cause- j conditional hazard at time t is

$$\lambda_j(t \mid a, w) = \lim_{h \rightarrow 0} \frac{1}{h} P(\tilde{T} \leq t+h, \tilde{\Delta} = j \mid \tilde{T} \geq t, a, w)$$

the conditional event-free survival at time t is

$$S(t \mid a, w) = \exp \left(- \int_0^t \sum_{j=1}^J \lambda_j(s \mid a, w) ds \right)$$

and $\pi^d(a | w)$ is the treatment propensity implied by the intervention d .

With this identification result in Eq (5), the causal cause- j relative risk (2) is simply identified by the statistical estimand (6)

$$\Psi_{F_{j,t}} = F_j^{d^*}(t) / F_j^{d^{**}}(t) \quad (6)$$

and the causal difference in survival (3) estimand can then be identified by the statistical estimand (7).

$$\Psi_{S_t} = \left[1 - \sum_{j=1}^J F_j^{d^*}(t) \right] - \left[1 - \sum_{j=1}^J F_j^{d^{**}}(t) \right] \quad (7)$$

It should be noted here that when the identification assumptions do not hold, these statistical estimands in Eq. (6) and (7) may still have valuable interpretations as standardized risks isolating the importance of the "intervention" variable (Laan (2006)).

2.4 Targeted Estimation

The TMLE procedure for estimands derived from cause-specific absolute risks begins with estimating the treatment propensity π , the conditional hazard of censoring λ_c and the conditional hazards of events $\lambda_j : j = 1, \dots, J$. In **concrete** these nuisance parameters are estimated using the Super Learner algorithm, which involves specifying a cross-validation scheme, compiling a library of candidate algorithms, and designating a cross-validation loss function and a Super Learner meta-learner.

Specifying Super Learners

For V -fold cross validation, let $Q_n = \{O_i\}_{i=1}^n$ be the observed n i.i.d observations of $O \sim P_0$ and let $B_n = \{1, \dots, V\}^n$ be a random vector that assigns the n observations into V validation folds. For each $v \in \{1, \dots, V\}$ we then define a training set $Q_v^T = \{O_i : B_n(i) = v\}$ with the corresponding validation set $Q_v^V = \{O_i : B_n(i) \neq v\}$.

When specifying Super Learner libraries, candidate algorithms should range in flexibility and while respecting pre-existing knowledge about the data-generating mechanism should be selected. For instance, candidate estimators for conditional hazards should incorporate domain knowledge about which covariates may be most predictive of event times. If the number of independent observations n is much greater than the number of covariates, then more highly flexible candidate algorithms such as Highly Adaptive Lasso (HAL) should be included in the Super Learner library. If on the other hand the number of covariates is not much less than n , then libraries should be comprised of fewer and less flexible candidate algorithms, potentially with native penalization as with coxnet or being paired with covariate screening algorithms.

Super Learner loss functions should be bounded and minimized by the true data-generating process and for maximal robustness, the discrete selector that simply selects the best performing candidate should be used as the Super Learner metalearner.

Currently the default cross-validation setup in **concrete** attempts to follow the guidelines laid out in Phillips et al. (2022), with the number of cross-validation folds increasing as sample size decreases. Default Super Learner libraries are provided, but should be amended to suit users' data as detailed above. Further details on the default **concrete** Super Learner specifications for each nuisance parameter are provided in the following sections.

Estimating Treatment Propensity

For estimating the treatment propensity, let $\pi_0(\cdot | W)$ be the true conditional distribution of A given W , $\mathcal{M}_\pi = \{\hat{\pi} : Q_n \rightarrow \hat{\pi}(Q_n)\}$ be the candidate library of propensity score estimators, and L_π be a loss function such that the risk $\mathbb{E}_0[L_\pi(\hat{\pi}, O)]$ is minimized when $\hat{\pi} = \pi_0$. The current **concrete** default for estimating treatment propensity uses the default **Superlearner** loss functions with glmnet and xgboost as candidate algorithms. The discrete Superlearner estimator is then the the candidate propensity estimator $\hat{\pi} \in \mathcal{M}_\pi$ that has minimal cross validated risk

$$\hat{\pi}^{SL} = \underset{\hat{\pi} \in \mathcal{M}_\pi}{\operatorname{argmin}} \sum_{v=1}^V P_{Q_v^V} L_\pi(\hat{\pi}(Q_v^T), Q_v^V) \quad (8)$$

Estimating Conditional Hazards

For estimating the conditional hazards, let $\lambda_{0,\delta} : \delta = 0, \dots, J$ be the true conditional hazards for censoring ($\delta = 0$) and events ($\delta \in \{1, \dots, J\}$). Let $\mathcal{M}_\delta = \{\hat{\lambda}_\delta : Q_n \rightarrow \mathbb{R}\}$ for $\delta = 0, \dots, J$ be the libraries of candidate estimators for the censoring and cause-specific hazards. The current **concrete** default is a library of two Cox models, treatment-only and main-terms. The corresponding loss function is then the negative log Cox partial-likelihood loss function $L_\delta(\beta) = -\sum_{i=1}^n [\beta W_i - \log [\sum_{h \in \mathcal{R}(\tilde{T}_h)} \exp(\beta W_h)]]$. The discrete SuperLearner selector for each δ chooses the candidate $\hat{\lambda}_\delta \in \mathcal{M}_\delta$ that has minimal cross validated risk

$$\hat{\lambda}_\delta^{SL} = \operatorname{argmin}_{\hat{\lambda}_\delta \in \mathcal{M}_\delta} \sum_{v=1}^V P_{Q_v^v} L_\pi(\hat{\lambda}_\delta(Q_v^T), Q_v^V) : \delta = 0, \dots, J \quad (9)$$

These conditional hazard estimators are then used to compute the cause-specific absolute risks and survivals that are needed to compute the efficient influence curve (EIC).

Solving the Efficient Influence Curve

The treatment propensity estimator Eq. (8) and conditional hazard estimators Eq. (9) are used to compute the cause-specific absolute risk EICs corresponding to estimands like Eq. (6) and (7). Parameters that contrast multiple cause-specific absolute risks at multiple time points involve vector EICs comprised of the absolute risk EICs for each targeted event, each target time, and each intervention. For example, the vector EIC element corresponding to the event j , time t , and intervention propensity π^* cause-specific absolute risk ($\Psi_{\pi^*,j,t}(P_0) = F_j^{\pi^*}(t)$) is:

$$D_{\pi^*,j,t}^*(\lambda, \pi, S_c)(O) = \sum_{l=1}^J \int h_{\pi^*,j,l,t,s}(\lambda, \pi, S_c)(O) \left(N_l(ds) - \mathbf{1}(\tilde{T} \geq s) \lambda_l(s | A, W) \right) + \sum_{a \in \mathcal{A}} F_j(t | A = a, W) \pi^*(a | W) - \Psi_{\pi^*,j,t}(P_0) \quad (10)$$

where $N_l(s) = \mathbf{1}\{\tilde{T} \leq s, \tilde{\Delta} = l\}$ are the cause-specific counting processes and $h_{\pi^*,j,l,t,s}(\lambda, \pi, S_c)(O)$ is the TMLE "clever covariate" with the form

$$h_{\pi^*,j,l,t,s}(\lambda, \pi, S_c)(O) = \frac{\pi^*(A | W) \mathbf{1}(s \leq t)}{\pi(A | W) S_c(s | A, W)} \left(\mathbf{1}(l = j) - \frac{F_j(t | A, W) - F_j(s | A, W)}{S(s | A, W)} \right) \quad (11)$$

where $F_j(t | a, w)$ is the conditional cause- j absolute risk, $S_c(t | A, W)$ is the conditional censoring survival, $S(t | A, W)$ is the conditional event-free survival, and $N_j(t) = \mathbf{1}\{\tilde{T} \leq t, \tilde{\Delta} = j\}$ is the event- j counting process. The treatment propensity π and the conditional event and censoring hazard functions ($\lambda_c, \lambda_j : j = 1, \dots, J$) are directly estimated with Eq. (8) and (9) while the conditional absolute risks and survivals are computed from the hazard estimates as described in Section 2.3. The clever covariate is a function of the **intervention propensity**, **observed conditional distributions** which are not changed by TMLE targeting, and lastly the **outcome-related conditional distributions** which are updated by targeting.

The one-step continuous-time survival TMLE updates the cause-specific hazards along the universally least favorable submodel in the following manner:

$$\lambda_{j,\epsilon}(t) = \lambda_j(t) \exp \left(\int_0^\epsilon \frac{\langle \mathbb{P}_n \tilde{D}^*(\lambda_x, \pi, S_c)(O), h_{j,s}(\lambda_x, \pi, S_c)(O) \rangle_\Sigma}{\|\tilde{D}^*(\lambda_x, \pi, S_c)(O)\|_\Sigma} dx \right) \quad (12)$$

where

$$\langle x, y \rangle_\Sigma = x^\top \Sigma^{-1} y, \quad \|x\|_\Sigma = \sqrt{x^\top \Sigma^{-1} x}$$

\tilde{D}^* is the vector of efficient influence functions

$$\tilde{D}^*(\lambda, \pi, S_c)(O) = \left(D_{\pi^*,j,t_k}^*(\lambda, \pi, S_c)(O) : \pi^* \in \mathcal{A}, j \in \mathcal{J}, t_k \in \mathcal{T} \right)$$

and $h_{j,s}$ is the vector of clever covariates

$$h_{j,s}(\lambda, \pi, S_c)(O) = \left(h_{\pi^*,j,l,t_k,s}(\lambda, \pi, S_c)(O) : \pi^* \in \mathcal{A}, j \in \mathcal{J}, t_k \in \mathcal{T} \right)$$

In practice this integral is approximated by recursively summing along a series of locally least favorable models and ends when

$$\mathbb{P}_n D^*(\lambda_\epsilon, \pi, S_c)(O) \leq \frac{\sqrt{\mathbb{P}_n D^*(\lambda_\epsilon, \pi, S_c)(O)^2}}{\sqrt{n} \log(n)} \quad (13)$$

Estimating Variance

In **concrete**, the variance of TMLE estimates of targeted risks is computed by dividing the corresponding EIC's variance by the sample size, $\frac{\mathbb{P}_n D^*(\lambda_\epsilon, \pi, S_c)(O)^2}{n}$, which is a consistent estimator of the asymptotic variance of asymptotically linear estimators. In the presence of significant positivity violations (which may be seen as propensity scores close to 0), this EIC-derived variance estimator will be anti-conservative and variance estimation by bootstrap may be more reliable. However, bias resulting from positivity violations cannot be remedied in this way, and so other methods of addressing positivity violations (Petersen et al. (2012)) are recommended instead. For multidimensional estimands, simultaneous confidence intervals can be computed by simulating the $1 - \alpha$ quantile of a multivariate normal distribution with the correlation structure of the estimand EICs.

3 Usage

There are 3 main user-facing functions in **concrete**: `formatArguments()`, `doConcrete`, and `getOutput`. Specification of the estimation problem is done through input into `formatArguments()`, which checks the estimation specification and outputs a "ConcreteArgs" object that encapsulates the estimation specification. When the estimation procedure is adequately specified, the "ConcreteArgs" object then passed into `doConcrete()` to perform the specified continuous-time one-step survival TMLE. The output of `doConcrete()` is a "ConcreteEst" object which can be passed into `getOutput` to output tables and plots for cause-specific absolute risk derived estimands such as risk differences and relative risks.

3.1 formatArguments()

3.2 ConcreteArgs

The arguments of `formatArguments()` are primarily involved in specifying 1) the observed data structure, 2) the target estimand, and 3) the TMLE estimator.

```
data <- as.data.table(survival::pbc)
data <- data[, c("time", "status", "trt", "age", "sex", "albumin")]
data <- data[!is.na(trt), ][, trt := trt - 1]

ConcreteArgs <- formatArguments(DataTable = data,
  EventType = "time",
  EventType = "status",
  Treatment = "trt",
  ID = NULL,
  Intervention = 0:1,
  TargetEvent = 1:2,
  TargetTime = 90 * (6:30),
  MaxUpdateIter = 500)
```

Data

Observed data is passed into the `DataTable` argument as either a `data.frame` or `data.table` object. This data must include columns corresponding to the observed times \tilde{T} , indicator of which event occurred (Δ), and treatment variable A , as well as any number of columns containing baseline covariates W . Note that the treatment values in A must be numeric, with binary treatments encoded as 0 or 1. The data must not have missingness; imputation of missing covariates should be done prior to passing data into **concrete** while data with missing treatment or outcome values is not supported by **concrete**. If the data includes a column with uniquely identifying subject IDs, its name should be passed into the `ID=` argument; this is for compatibility with planned future functionality for analyzing clustered or longitudinal data.

In the above PBC example, the observed data is contained in the data object, \tilde{T} is the column {"time"}, Δ is the column {"status"}, A is the column {"trt"}, and covariates L are the remaining columns: "age", "sex", and "albumin".

Target Estimand

concrete implements a continuous-time one-step TMLE targeting absolute risk derived estimands indexed by interventions, target events, and target times.

Intervention For a binary A and static interventions d setting all observations to $A = 0$ or $A = 1$, interventions can be specified with 0, 1, or `c(0, 1)` for both. More complex interventions can be specified with a list containing a pair of functions: an "intervention" function which outputs desired treatment **assignments** and a "g.star" function which outputs desired treatment **probabilities**. Dynamic interventions can be passed in as "intervention" functions without an accompanying "g.star" function, while stochastic interventions should be passed in as "g.star" functions. These functions can take treatment and covariates as arguments and must produce treatment assignments and probabilities respectively, each with the same dimensions as the observed treatment. The function `makeITT()` creates list of functions corresponding to binary static interventions, which can be used as a template for specifying more complex interventions.

Target Events The `TargetEvent` argument specifies the event types of interest. Event types must be coded as integers, with 0 reserved for censoring. If `TargetEvent=` is left NULL, then all non-zero event types present in the observed data will be targeted.

In the pbc dataset, there are 3 event values encoded by the status column: 0 for censored, 1 for transplant, and 2 for death. To analyze pbc with transplants treated as right-censoring, `TargetEvent` should be set to 2, whereas for a competing risks analysis one could either leave `TargetEvent = NULL` or set `TargetEvent = 1:2` as in the above example.

Target Time The `TargetTime=` argument specifies the time(s) at which estimates of the cause-specific absolute risks or event-free survival are desired. Target times should be restricted to the time range in which target events are observed and `formatArguments()` will return an error if target time is after the last observed failure event time. If no `TargetTime` is provided, then **concrete** will target the last observed event time, though this is likely to result in a highly variable estimate if prior censoring is substantial. The `TargetTime=` argument can either be a single number or a vector, as one-step TMLE can target cause-specific risks at multiple times simultaneously. For estimands involving full curves, `TargetTime=` should be set to a fine grid covering the desired interval.

Estimator Specification

The `formatArguments()` arguments involved in estimation are the cross-validation setup `CVArg`, the Superlearner candidate libraries `Model`, the software backends `PropScoreBackend` and `HazEstBackend`, and the practical TMLE implementation choices `MaxUpdateIter`, `OneStepEps`, and `MinNuisance`. Note that `Model` is used in this section in line with common usage in statistical software, rather than to refer to statistical or causal models as in preceding sections.

Cross-Validation **concrete** uses **origami** to specify cross-validation folds, specifically the function `origami::make_folds()`. If no input is provided to the `formatArguments(CVArg=)` argument, **concrete** will implement simple V-fold cross-validation following the recommendations in Phillips et al. (2022).

```
CVArgs <- list(n = nrow(data), V = 10)
```

Estimating Nuisance Parameters Super Learner libraries for estimating nuisance parameters are specified in the `Model` argument. The input should be a named list with an element for the treatment variable and one for each event type including censoring. The list element corresponding to treatment must be named with the column name and the list elements corresponding to each event type must be named for the numeric value of the event type (e.g. "0" for censoring). Any missing specifications will be filled in with defaults, and the resulting list of libraries can be accessed in the output `.[["Model"]]` which can be then edited by the user.

```
Model <- list("trt" = c("SL.glmnet", "SL.ranger", "SL.xgboost", "SL.glmn"),
             "0" = list("~ trt", "~."),
```



```
"1" = list("~ trt", "~ .", "~ trt:.")
"2" = list("~ trt", "~ .", "~ trt:.")
)
```

3.3 doConcrete()

Once `formatArguments()` runs satisfactorily, the resulting object of class "ConcreteArgs" is passed into the `doConcrete()` function which will then perform the specified TMLE algorithm. The output is an object of class "ConcreteEst" which contains TMLE point estimates and influence curves for the cause-specific absolute risks for each targeted event at each targeted time. If `formatArguments(GComp=TRUE)`, then the "ConcreteEst" object will also contain the result of using the Superlearner predictions as a plug-in g-formula estimate of the targeted risks.

```
ConcreteEst <- doConcrete(ConcreteArgs)
```

Detailed explanations of the one-step TMLE for continuous-time absolute risk derived estimands can be found in [Rytgaard and van der Laan \(2021\)](#) and [Rytgaard et al. \(2021\)](#). This manuscript briefly reviews this estimation procedure in Section 2.4 and details how a TMLE is specified in `concrete` in Section 3.2.3, subsections 3.2.3.1 through 3.2.3.5. Here we will call out the non-exported functions in `doConcrete()` which perform each of the steps of the one-step continuous-time survival TMLE procedure, in case users wish to explore the implementation.

The cross-validation specification (Section 3.2.3.1) is checked and evaluated in `formatArguments()`, returning fold assignments as `.["CVFolds"]` of the "ConcreteArgs" object.

The initial estimation of nuisance parameters (Section 3.2.3.2) is performed by the function `getInitialEstimate()`; `getPropScore()` estimates propensity scores (Section 3.2.3.3) and `getHazEstimate()` estimates the conditional hazards (Section 3.2.3.4).

The one-step TMLE update procedure (Sections 2.4 and 3.2.3.5, Equations (10), (11), (12), and (13)) is performed by `doTmleUpdate()` with `getEIC()` computing the efficient influence curves (10).

ConcreteEst objects

The print method for "ConcreteEst" objects displays information about whether TMLE algorithm adequately converged, truncation of intervention-related nuisance parameter estimates, and the initial estimation of nuisance parameters.

```
print(ConcreteEst)
```

```
Continuous-Time One-Step TMLE targeting the Cause-Specific Absolute Risks for:
Interventions: "A=1", "A=0" | Target Events: 1, 2 | Target Times: 540, 630, 720, ... ,2520, 2610, 2700
```

```
**TMLE did not converge!!**
```

	Intervention	Time	Event	Pt Est	se	PnEIC	abs(PnEIC / Stop Criteria)
1:	A=1	810	1	0.005102	0.0004727	-0.0020070	24.385
2:	A=1	540	1	0.001573	0.0001504	-0.0004373	16.702
3:	A=1	630	1	0.001573	0.0001504	-0.0004373	16.702
4:	A=1	720	1	0.001573	0.0001504	-0.0004373	16.702
5:	A=1	990	1	0.012111	0.0057221	-0.0016876	1.694

```
For Intervention "A=1", no G-related nuisance weights fell below to 0.0493
For Intervention "A=0", no G-related nuisance weights fell below to 0.0493
```

```
Initial Estimators:
```

```
Treatment:
              Risk      Coef
SL.xgboost_All 0.2948025 0.09408743
SL.glmnet_All  0.2494030 0.90591257
```

```
Cens. 0:
```

```
              Risk Coef
TrtOnly      286.0636  0
MainTerms    285.3721  1
```

```
Event 1:
```

```
              Risk Coef
TrtOnly      44.61230  0
MainTerms    38.61817  1
```

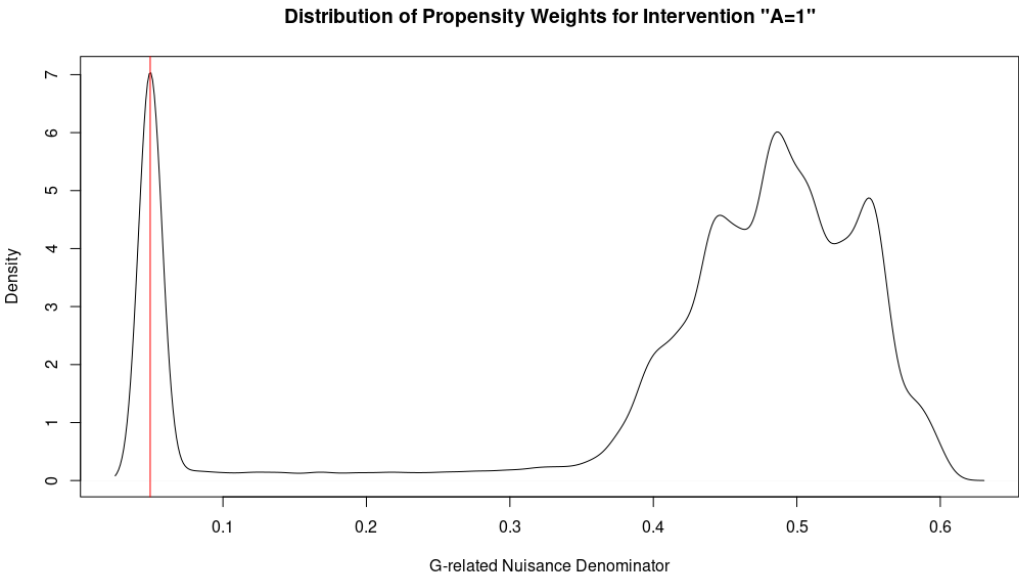
```
Event 2:
```

```
              Risk Coef
TrtOnly      277.4118  0
MainTerms    242.9892  1
```

If TMLE has not converged, the mean EICs that have not attained the desired cutoff will be displayed in a table. Convergence can be attained by increasing the maximum number of iterations, though as seen above, even very small PnEIC values may not meet the convergence criteria at target times when very few events have yet occurred.

The amount of g-related nuisance parameter truncation for each intervention is also printed, and if users suspect positivity issues, the plot method for "ConcreteEst" objects can be used to visualize the distribution of estimated propensity scores for each intervention, with the red vertical line marking the cutoff for truncation.

```
plot(ConcreteEst, propscores = TRUE)
```



Propensity scores close to 0 indicate the possibility of positivity violations and may warrant re-examining the target time(s), interventions, and covariate adjustment sets. In typical survival applications, positivity issues may arise when targeting times at which some subjects are highly likely to have been censored, or if certain subjects are unlikely to have received a desired treatment intervention.

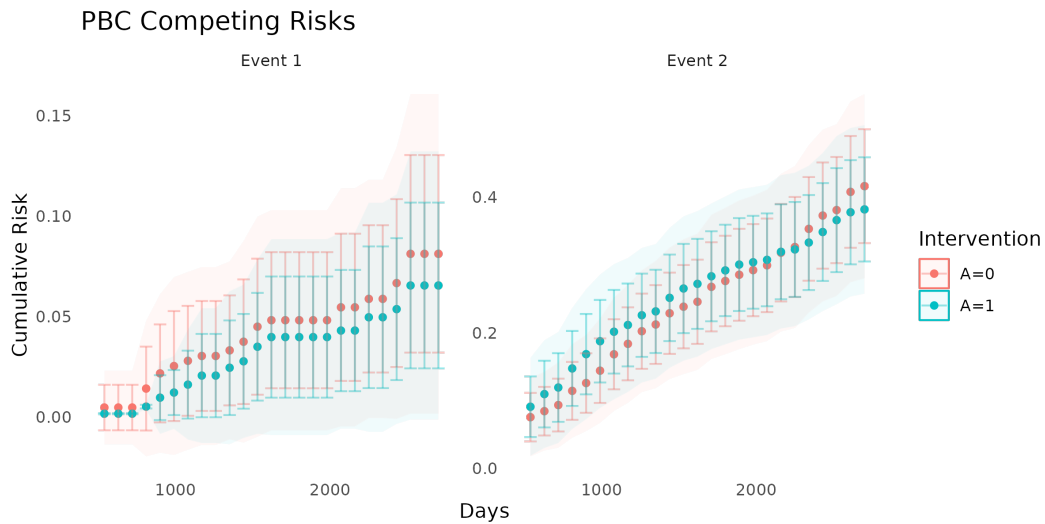
3.4 `getOutput()`

`getOutput()` takes as an argument the "ConcreteEst" object returned by `doConcrete()` and returns tables and produces plots of the cause-specific risks, risk differences, and/or relative risks. By default `getOutput()` returns a `data.table` with point estimates and pointwise standard errors for cause-specific absolute risks, risk differences, and risk ratios. Risk difference and ratios are by default computed by subtracting or dividing the second listed intervention from the first. Below we show a subset of the relative risk estimates produced by the "nutshell" estimation specification for the `pbc` dataset.

```
getOutput(ConcreteEst, "RR", GComp = FALSE)
```

	Intervention	Estimand	Estimator	Event	Time	Pt Est	se
1:	[A=1] / [A=0]	Rel Risk	tmle	1	540	0.3967609	0.52377394
2:	[A=1] / [A=0]	Rel Risk	tmle	1	630	0.3967609	0.52377394
3:	[A=1] / [A=0]	Rel Risk	tmle	1	720	0.3967609	0.52377394
4:	[A=1] / [A=0]	Rel Risk	tmle	1	2520	0.8378020	0.21099376
5:	[A=1] / [A=0]	Rel Risk	tmle	1	2610	0.8378020	0.21099376
6:	[A=1] / [A=0]	Rel Risk	tmle	1	2700	0.8378020	0.21099376
7:	[A=1] / [A=0]	Rel Risk	tmle	2	540	1.1821447	0.40796706
8:	[A=1] / [A=0]	Rel Risk	tmle	2	630	1.2638824	0.39355372
9:	[A=1] / [A=0]	Rel Risk	tmle	2	720	1.2535307	0.37561184
10:	[A=1] / [A=0]	Rel Risk	tmle	2	2520	0.8538055	0.11158560
11:	[A=1] / [A=0]	Rel Risk	tmle	2	2610	0.7910752	0.09701693
12:	[A=1] / [A=0]	Rel Risk	tmle	2	2700	0.7853989	0.09529657

From this small sample risk ratios we can see that the treatment decreases the incidence of transplant (event 1) and leads to an increased incidence of death (event 2) earlier on followed by a decreased risk of death. We also see that these treatment effect estimates do not reach a 95% level of significance. The full table is not shown here because it would take too much space, as can often be the case when estimands involve many time points. Instead results may be more easily interpreted when plotted, as we do below with the treated and control cause-specific risks for transplant and death.



Here, by comparing the treated and control cumulative incidence curves we can more clearly see the effect of treatment on incidence of both transplant and death, and especially how the treatment effect on risk of death varies over time.

4 Diagnostics and Debugging

4.1 Nuisance Weight Truncation Simulation

5 Appendix: Nice to have Concepts

5.1 Identification

In order to identify causal estimands such as absolute risk ratios and differences with functions of the observed data, some untestable structural assumptions must hold - namely the assumptions of consistency, positivity, randomization, and coarsening at random on the conditional density of the censoring mechanism.

1. The consistency assumption states that the observed outcome given a certain treatment decision is equal to the corresponding counterfactual outcome

$$T_j^d = T_j \text{ on the event that } A = d(L)$$

1. The positivity assumption states that the desired treatment regimes occur with non-zero probability in all observed covariate strata, and that remaining uncensored occurs with non-zero probability in all observed covariate strata at all times of interest t .

$$P_0(A = d(L) | W) > 0, a.e.$$

$$P(C \geq t | a, W), a.e.$$

1. The randomization assumption states that there is no unmeasured confounding between treatment and counterfactual outcomes

$$A \perp\!\!\!\perp (T_1^d, T_2^d) | W$$

1. Coarsening at random on censoring

$$C \perp\!\!\!\perp (T_1^d, T_2^d) | T > C, A, W$$

Given coarsening at random, the observed data distribution factorizes

$$p_0(O) = p_0(W) \pi_0(A | W) \lambda_{0,c}(\tilde{T} | A, W)^{\mathbf{1}(\Delta=0)} S_{0,c}(\tilde{T}^- | A, W) \prod_{j=1}^J S_{0,j}(\tilde{T}^- | A, W) \lambda_{0,j}(\tilde{T} | A, W)^{\mathbf{1}(\Delta=j)}$$

where $\lambda_{0,c}(t \mid A, W)$ is the true cause-specific hazard of the censoring process and $\lambda_{0,j}(t \mid A, W)$ is the true cause-specific hazard of the j^{th} event process. Additionally

$$S_{0,c}(t \mid a, w) = \exp \left(- \int_0^t \lambda_{0,c}(s \mid a, w) ds \right)$$

while in a pure competing risks setting

$$S_0(t \mid a, w) = \exp \left(- \int_0^t \sum_{j=1}^J \lambda_{0,j}(s \mid a, w) ds \right)$$

and

$$\begin{aligned} F_{0,j}(t \mid a, w) &= \int_0^t S(s \mid a, w) \lambda_{0,j}(s \mid a, w) ds \\ &= \int_0^t \exp \left(- \int_0^s \sum_{j=1}^J \lambda_{0,j}(u \mid a, w) du \right) \lambda_{0,j}(s \mid a, w) ds. \end{aligned}$$

Under the above identification assumptions, the post-intervention distribution of O under intervention $A = d(a, w)$ in the world of no-censoring, i.e the distribution of $(W, T_j^d, \Delta_j^d : j = 1, \dots, J)$, can be represented by the so-called G-computation formula. Let's denote this post-intervention probability distribution with P_d and the corresponding post-intervention random variable with O_d . The probability density of O_d follows from replacing $\pi_0(A \mid W)$ with the density that results from setting $A = d(a, w)$, $\pi_d(d(A, w) \mid W)$, and replacing the conditional probability of being censored at time t by no censoring with probability 1. In notation, $P(O_d = o)$ is given by

$$\begin{aligned} p_d(o) &= p_0(w) \pi_d(d(a, w) \mid w) \mathbf{1}(\delta \neq 0) \\ &\quad \prod_{j=1}^J \left[S_0(\tilde{t} \mid A = d(a, w), w) \lambda_{0,j}(\tilde{t} \mid A = d(a, w), w)^{\mathbf{1}(\delta=j)} \right] \end{aligned}$$

Recalling the censoring and cause-specific conditional hazards defined above in terms of observed data, we should note that given the identifiability assumptions they now identify their counterfactual counterparts, i.e.

$$\lambda_c(t \mid W, A) = \lim_{h \rightarrow 0} P(C < t + h \mid C \geq t, W, A)$$

$$\lambda_j(t \mid W, A) = \lim_{h \rightarrow 0} P(T < t + h, J = j \mid T \geq t, W, A)$$

Note that the cause-specific event hazards are not conditional on censoring once identifiability assumptions are met.

Since the density $P(O_d = o)$ implies any probability event about O_d , this g-computation formula for $P(O_d = o)$ also implies g-computation formulas for causal quantities such as event-free survival and cause- k absolute risk under intervention d .

Bibliography

- P. K. Andersen, R. B. Geskus, T. de Witte, and H. Putter. Competing risks in epidemiology: possibilities and pitfalls. *International Journal of Epidemiology*, 41(3):861–870, June 2012. ISSN 0300-5771. doi: 10.1093/ije/dyr213. URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3396320/>. [p1]
- P. C. Austin and J. P. Fine. Accounting for competing risks in randomized controlled trials: a review and recommendations for improvement. *Statistics in Medicine*, 36(8):1203–1209, 2017. ISSN 1097-0258. doi: 10.1002/sim.7215. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.7215>. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.7215>. [p1]
- D. Benkeser and N. Hejazi. survtmle: Compute Targeted Minimum Loss-Based Estimates in Right-Censored Survival Settings, Apr. 2019. URL <https://CRAN.R-project.org/package=survtmle>. [p2]
- D. Benkeser and M. Van Der Laan. The Highly Adaptive Lasso Estimator. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 689–696, Oct. 2016. doi: 10.1109/DSAA.2016.93. ISSN: null. [p2]
- R. Denz, R. Klaußen-Mielke, and N. Timmesfeld. A Comparison of Different Methods to Adjust Survival Curves for Confounders, Mar. 2022. URL <http://arxiv.org/abs/2203.10002>. Number: arXiv:2203.10002 arXiv:2203.10002 [stat]. [p2]
- J. P. Fine and R. J. Gray. A Proportional Hazards Model for the Subdistribution of a Competing Risk. *Journal of the American Statistical Association*, 94(446):496–509, June 1999. ISSN 0162-1459. doi: 10.1080/01621459.1999.10474144. URL <https://www.tandfonline.com/doi/abs/10.1080/01621459.1999.10474144>. Publisher: Taylor & Francis _eprint: <https://www.tandfonline.com/doi/pdf/10.1080/01621459.1999.10474144>. [p1]
- T. A. Gerds, T. H. Scheike, and P. K. Andersen. Absolute risk regression for competing risks: interpretation, link functions, and prediction. *Statistics in Medicine*, 31(29):3921–3930, 2012. ISSN 1097-0258. doi: 10.1002/sim.5459. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.5459>. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/sim.5459>. [p1]
- T. A. Gerds, J. S. Ohlendorff, P. Blanche, R. Mortensen, M. Wright, N. Tollenaar, J. Muschelli, U. B. Mogensen, and B. Ozenne. riskRegression: Risk Regression Models and Prediction Scores for Survival Analysis with Competing Risks, Sept. 2022. URL <https://CRAN.R-project.org/package=riskRegression>. [p2]
- P. W. Holland. Statistics and Causal Inference. *Journal of the American Statistical Association*, 81(396):945–960, 1986. ISSN 0162-1459. doi: 10.2307/2289064. URL <http://www.jstor.org/stable/2289064>. Publisher: [American Statistical Association, Taylor & Francis, Ltd.]. [p1]
- E. H. Kennedy. Semiparametric Theory and Empirical Processes in Causal Inference. In H. He, P. Wu, and D.-G. D. Chen, editors, *Statistical Causal Inferences and Their Applications in Public Health Research*, ICSA Book Series in Statistics, pages 141–167. Springer International Publishing, Cham, 2016. ISBN 978-3-319-41259-7. doi: 10.1007/978-3-319-41259-7_8. URL https://doi.org/10.1007/978-3-319-41259-7_8. [p2]
- J. P. Klein and P. K. Andersen. Regression Modeling of Competing Risks Data Based on Pseudovalues of the Cumulative Incidence Function. *Biometrics*, 61(1):223–229, 2005. ISSN 1541-0420. doi: 10.1111/j.0006-341X.2005.031209.x. URL <https://onlinelibrary.wiley.com/doi/abs/10.1111/j.0006-341X.2005.031209.x>. _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.0006-341X.2005.031209.x>. [p1]
- M. T. Koller, H. Raatz, E. W. Steyerberg, and M. Wolbers. Competing risks and the clinical community: irrelevance or ignorance? *Statistics in Medicine*, 31(11-12):1089–1097, May 2012. ISSN 1097-0258. doi: 10.1002/sim.4384. [p1]
- M. J. v. d. Laan. Statistical Inference for Variable Importance. *The International Journal of Biostatistics*, 2(1), Feb. 2006. ISSN 1557-4679. doi: 10.2202/1557-4679.1008. URL <https://www.degruyter.com/document/doi/10.2202/1557-4679.1008/html?lang=en>. Publisher: De Gruyter. [p5]
- M. J. v. d. Laan. A Generally Efficient Targeted Minimum Loss Based Estimator based on the Highly Adaptive Lasso. *The International Journal of Biostatistics*, 13(2), Oct. 2017. doi: 10.1515/ijb-2015-0097. [p2]
- M. J. v. d. Laan and S. Dudoit. Unified Cross-Validation Methodology For Selection Among Estimators and a General Cross-Validated Adaptive Epsilon-Net Estimator: Finite Sample Oracle Inequalities and Examples. *U.C. Berkeley Division of Biostatistics Working Paper Series*, Nov. 2003. [p2]

- M. J. v. d. Laan and J. M. Robins. *Unified Methods for Censored Longitudinal Data and Causality*. Springer Series in Statistics. Springer-Verlag, New York, 2003. ISBN 978-0-387-95556-8. [p2]
- M. J. v. d. Laan and S. Rose. *Targeted Learning: Causal Inference for Observational and Experimental Data*. Springer Series in Statistics. Springer-Verlag, New York, 2011. ISBN 978-1-4419-9781-4. doi: 10.1007/978-1-4419-9782-1. [p1, 2]
- M. J. v. d. Laan and S. Rose. *Targeted Learning in Data Science: Causal Inference for Complex Longitudinal Studies*. Springer Series in Statistics. Springer International Publishing, 2018. ISBN 978-3-319-65303-7. doi: 10.1007/978-3-319-65304-4. [p1]
- M. J. v. d. Laan, E. C. Polley, and A. E. Hubbard. Super Learner. *Statistical Applications in Genetics and Molecular Biology*, 6(1), Sept. 2007. ISSN 1544-6115, 2194-6302. doi: 10.2202/1544-6115.1309. [p2]
- J. Pearl, M. Glymour, and N. P. Jewell. *Causal Inference in Statistics - A Primer*. Wiley, Chichester, West Sussex, 1st edition edition, Mar. 2016. ISBN 978-1-119-18684-7. [p1]
- M. L. Petersen and M. J. van der Laan. Causal models and learning from data: integrating causal modeling and statistical estimation. *Epidemiology (Cambridge, Mass.)*, 25(3):418–426, May 2014. ISSN 1531-5487. doi: 10.1097/EDE.0000000000000078. [p1]
- M. L. Petersen, K. E. Porter, S. Gruber, Y. Wang, and M. J. van der Laan. Diagnosing and responding to violations in the positivity assumption. *Statistical Methods in Medical Research*, 21(1):31–54, Feb. 2012. ISSN 0962-2802, 1477-0334. doi: 10.1177/0962280210386207. URL <http://journals.sagepub.com/doi/10.1177/0962280210386207>. [p7]
- R. V. Phillips, M. J. van der Laan, H. Lee, and S. Gruber. Practical considerations for specifying a super learner, Apr. 2022. URL <http://arxiv.org/abs/2204.06139>. arXiv:2204.06139 [stat]. [p2, 5, 8]
- E. Polley, E. LeDell, C. Kennedy, S. Lendle, and M. v. d. Laan. SuperLearner: Super Learner Prediction, May 2021. URL <https://CRAN.R-project.org/package=SuperLearner>. [p2]
- H. C. Rytgaard, T. A. Gerds, and M. J. van der Laan. Continuous-time targeted minimum loss-based estimation of intervention-specific mean outcomes, May 2021. URL <http://arxiv.org/abs/2105.02088>. arXiv:2105.02088 [math, stat]. [p1, 2, 4, 10]
- H. C. W. Rytgaard and M. J. van der Laan. One-step TMLE for targeting cause-specific absolute risks and survival curves. *arXiv:2107.01537 [stat]*, Sept. 2021. URL <http://arxiv.org/abs/2107.01537>. arXiv:2107.01537 version: 2. [p1, 2, 10]
- T. H. Scheike, M.-J. Zhang, and T. A. Gerds. Predicting cumulative incidence probability by direct binomial regression. *Biometrika*, 95(1):205–220, Mar. 2008. ISSN 0006-3444. doi: 10.1093/biomet/asm096. URL <https://doi.org/10.1093/biomet/asm096>. [p1]
- J. Schwab, S. Lendle, M. Petersen, M. v. d. Laan, and S. Gruber. ltmle: Longitudinal Targeted Maximum Likelihood Estimation, Mar. 2020. URL <https://CRAN.R-project.org/package=ltmle>. [p2]
- O. Sofrygin, M. J. v. d. Laan, and R. Neugebauer. streamr: Streamlined Estimation of Survival for Static, Dynamic and Stochastic Treatment and Monitoring Regimes, Jan. 2017. URL <https://CRAN.R-project.org/package=streamr>. [p2]
- A. W. v. d. Vaart, S. Dudoit, and M. J. v. d. Laan. Oracle inequalities for multi-fold cross validation. *Statistics & Decisions*, 24(3), Jan. 2006. ISSN 0721-2631. doi: 10.1524/std.2006.24.3.351. [p2]
- T. Westling, A. Luedtke, P. Gilbert, and M. Carone. Inference for treatment-specific survival curves using machine learning, June 2021. URL <http://arxiv.org/abs/2106.06602>. Number: arXiv:2106.06602 arXiv:2106.06602 [stat]. [p2]