# Propensity score analysis for time-dependent exposure

**Zhongheng Zhang[1#], Xiuyang Li[2,3#], Xiao Wu[4], Huixian Qiu[5,6], Hongying Shi[7,8]; written on behalf of AME Big-Data Clinical Trial Collaborative Group**

[1]Department of Emergency Medicine, Sir Run-Run Shaw Hospital, Zhejiang University School of Medicine, Hangzhou 310016, China; [2]Department of Epidemiology & Biostatistics, Zhejiang University School of Medicine, Hangzhou 310058, China; [3]Department of Neurology of the Second Affiliated Hospital of Zhejiang University School of Medicine, Interdisciplinary Institute of Neuroscience and Technology of Qiushi Academy for Advanced Studies, Zhejiang University, Hangzhou 310029, China; [4]Department of Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA, USA; [5]Institute of Cardiovascular Development and Translation Medicine, Wenzhou Medical University, Wenzhou 325000, China; [6]Children's Heart Center, Second Affiliated Hospital & Yuying Children's Hospital of Wenzhou Medical University, Wenzhou 325027, China; [7]Department of Preventive Medicine, School of Public Health and Management, Wenzhou Medical University, Wenzhou 325035, China; [8]Department of Epidemiology and Biostatistics, Harvard T.H. Chan School of Public Health, Boston, MA, USA

[#]These authors contributed equally to this work as co-first authors.

*Correspondence to:* Zhongheng Zhang. Department of Emergency Medicine, Sir Run-Run Shaw Hospital, Zhejiang University School of Medicine, Hangzhou 310016, China. Email: zh_zhang1984@zju.edu.cn; Hongying Shi. Department of Preventive Medicine, School of Public Health and Management, Wenzhou Medical University, Wenzhou 325035, China. Email: shying918@163.com.

**Abstract:** Propensity score analysis (PSA) is widely used in medical literature to account for confounders. Conventionally, the propensity score (PS) is calculated by a binary logistic regression model using time-fixed covariates. In the presence of time-varying treatment or exposure, the conventional method may cause bias because subjects with early and late exposure are treated as the same. In effect, subjects who are treated latter can be different from those who are treated early. Thus, the conventional PSA must be modified to address this bias. In this paper, we illustrate how to perform analysis in the presence of time-dependent exposure. We conduct a simulation study with a known treatment effect. In the simulation study, we find the PSA method that directly adjust PS estimated by either a binary logistic regression model or a Cox regression model using time-fixed covariates still introduce significant bias. On the other hand, the time-dependent PS matching can help to achieve a result approaching the true effect. After time-dependent PS matching, the matched cohort can be analyzed with conventional Cox regression model or conditional logistic regression (CLR) model with time strata. The performance is comparable to the correctly specified Cox regression model with time-varying covariates (i.e., adjusting the exposure in a multivariable model as a time-varying covariate). We further develop a function called *TDPSM()* for time-dependent PS matching and it is applied to a real world dataset.

**Keywords:** Propensity score matching (PS matching); time-dependent; R

Submitted Dec 28, 2019. Accepted for publication Jan 29, 2020.
doi: 10.21037/atm.2020.01.33
**View this article at:** http://dx.doi.org/10.21037/atm.2020.01.33

## Introduction

Propensity score analysis (PSA) is widely used in medical literature. In observational studies, the causal inference cannot be easily made due to multiple measured and/or unmeasured confounding factors. The causation between exposure (*A*) and outcome (*Y*) is explored in counterfactual framework so that the allocation of treatment (i.e., the treatment a subject actually receives) is independent of potential outcome $Y^a$ conditional on confounders (*L*) such as the severity of illness and patients' preference: $Y^a \perp A | L$. The treatment effect can then be estimated in the strata with equal probability of receiving treatment. Here comes the idea of propensity score (PS) which is typically estimated by regressing the treatment on pre-treatment covariates using binary logistic regression models. After PS matching, the matched cohort is considered as that generated from randomized experiments in which the treatment allocation is independent of any pre-treatment covariates.

However, PS is generated for each individual at study entry without considering its time-dependent property. In effect, there are many interventions that are not given at the start of a study but may be given at any time during study period. For example, in a study exploring the association of tracheal intubation and survival in in-hospital cardiac arrest, the tracheal intubation (exposure) can happen at any time after cardiac arrest and intubation may not occur if return of spontaneous circulation (ROSC) or termination of efforts occurs first (1). If the duration of resuscitation is long enough, a patient is very likely to be intubated. Thus, the comparison between intubated and non-intubated subjects is essentially comparing the survival outcome for those with long versus short resuscitation time. Short resuscitation time can be the result of early ROSC or termination of efforts. To avoid such bias, time-dependent PS matching can be performed by iteratively matching the treated subjects to the "at-risk" controls across all time strata.

This article aims to provide a step-by-step tutorial on how to perform analysis for time-dependent treatment. We will show how the conventional PSA ignoring the time-to-exposure property of the treatment can bias the result, and then highlight the time-dependent PS matching for the analysis of such data. Specifically, we provide a function *TDPSM()* to perform time-dependent PS matching. Alternative methods such as Cox regression with time-varying covariate, adjustment with PS are also shown for comparison. Comparisons among these methods are performed by simulation. Finally, we illustrate how to use the *TDPSM()* in real world data.

## Working example

### *Mathematics underlying survival data simulation*

We need to spend a little time to review mathematics underlying the survival analysis. Assume that the survival time $T$ follows an exponential distribution, the baseline cumulative density function can be written as $F_0(t) = 1 - \frac{1}{e^{\lambda t}} = 1 - e^{-\lambda t}$, where $\lambda$ is a constant term. The function behaves reasonably that when $t$ tends to 0, $F_0(t)$ tends to 0, as it should be (e.g., the cumulative probability of the event is small). When $t$ tends to infinity, $F_0(t)$ tends to 1 (e.g., the event will eventually happen). Following this assumption, other important functions can be easily derived. The probability density function $f_0(t)$ is the first order derivative of the $F_0(t)$ w.r.t. $t$: $f_0(t) = F_0{}'(t) = \lambda e^{-\lambda t}$. The survivor function is $S_0(t) = 1 - F_0(t) = e^{-\lambda t}$; and the hazard function $h_0(t) = \frac{f_0(t)}{S_0(t)} = \frac{\lambda e^{-\lambda t}}{e^{-\lambda t}} = \lambda$, which is a constant hazard. The cumulative hazard function is $H_0(t) = \lambda t$. The above equations define as the baseline function that all covariates are zero. The hazard function in the presence of covariates can be expressed as $h(t) = \lambda e^{\beta x}$. Since $h(t) = \lambda e^{\beta x}$ is independent of $t$, $H(t) = \int_0^t h(u)\,du = h(t)t$. The survival time can be simulated by $T = \frac{H(t)}{h(t)} = \frac{-log(S(t))}{\lambda e^{\beta x}} = -\frac{log(U)}{\lambda e^{\beta x}}$, where $U$ follows a uniform distribution on the interval from 0 to 1, which is consistent with the survivor function $S(t)$ (2). Simulation of longitudinal data with time dependent exposure is well described in Xu's article (3). We adapted their approach as follows:

Next, we are going to simulate a dataset with time-varying exposure and survival outcome.

(I) Generate three confounders with standard normal distribution.

```
n = 2000  #The sample size
set.seed(223)
for (ii in 1:3) {
  assign(paste("X", ii, sep = "_"), rnorm(n))
}
X = cbind(X_1, X_2, X_3)
```

The sample size is 2000 and we also set a seed for reproducibility. In the *for* loop, three covariates *X_1*, *X_2* and *X_3* are generated.

(II) Generate the potential exposure time S given $X_1$, $X_2$ and $X_3$ from an exponential distribution with rate $e^{\alpha_0+\alpha_1 X_1+\alpha_2 X_2+\alpha_3 X_3}$, where $(a_0, a_1, a_2, a_3) = (1,1,1,1)$, and $S = -\frac{\log(U)}{\lambda e^{\alpha_0+\alpha_1 X_1+\alpha_2 X_2+\alpha_3 X_3}}$.

```
lambda = 1
alpha_0 = 1
alpha_1 = 1
alpha_2 = 1
alpha_3 = 1
ExpLin <- cbind(1, X) %*% c(alpha_0, alpha_1,
  alpha_2, alpha_3)
S = -log(runif(n))/(lambda * exp(ExpLin))
```

(III) Generate the event time T given X_1,X_2,X_3 and $Z_1(t) = I(t > S)$, where $I(\cdot)$ is the indicator function, according to a Cox model with hazard function

$$h(t)=h_0(t)e^{\beta_t Z_1(t)+\beta_1 X_1+\beta_2 X_2+\beta_3 X_3}, \qquad [1]$$

where $h_0(t)=1$, $\beta$ takes on the value of –0.5, and$(\beta_1,\beta_2,\beta_3)=(1,1,1)$. For the ease of annotation, we can write $\beta'X=\beta_1X_1+\beta_2X_2+\beta_3X_3$ as the linear predictor of time-fixed covariates The integration of h(t) w.r.t. t gives the cumulative hazard function

$$H(t)=\int_0^t h_0(u)e^{\beta_t Z_1(u)+\beta' X}\,du$$

$$du=\begin{cases}\lambda e^{\beta' X}t & if\ t<S\\ \lambda e^{\beta' X}\left[S+e^{\beta_t}t-e^{\beta_t}t_0\right] & if\ t\ge S\end{cases} \qquad [2]$$

This gives the survival function

$$S(t)=e^{-H(t)}=\begin{cases}exp\left(-\lambda e^{\beta' X}t\right) & if\ t<S\\ exp\left[-\lambda e^{\beta' X}\left(S+e^{\beta_t}t-e^{\beta_t}S\right)\right] & if\ t\ge S\end{cases} \qquad [3]$$

By sampling X and u~U(0,1), substituting u for S(t) and rearranging with simple algebra gives the following equation for T. (4)

$$T=\begin{cases}\dfrac{-log(u)}{\lambda e^{\beta' X}} & if\ -log(u)<\lambda e^{\beta' X}S\\ \dfrac{-log(u)-\lambda e^{\beta' X}S+\lambda e^{\beta' X+\beta_t}S}{\lambda e^{\beta' X+\beta_t}} & if\ -log(u)\ge\lambda e^{\beta' X}S\end{cases} \qquad [4]$$

where $\beta'X=\beta_1X_1+\beta_2X_2+\beta_3X_3$, and $u\sim U(0,1)$. This is effectively a piecewise exponential distribution, with different rates on the two intervals $(0,S)$ and $(S,\infty)$. The above mathematical equations can be coded as follows:

```
for (ii in 1:3) {
  assign(paste("beta", ii, sep = "_"),
    1)
}
beta_t = -0.5
U = runif(n)
LinFix <- X %*% c(beta_1, beta_2, beta_3)
EventT <- ifelse(-log(1 - U) < lambda * exp(LinFix) *
  S, -log(1 - U)/(lambda * exp(LinFix)),
  (-log(1 - U) - lambda * exp(LinFix) *
    S + lambda * exp(LinFix + beta_t) *
    S)/(lambda * exp(LinFix + beta_t)))
```

(IV) Generate censoring time CensorT as Uniform (0,b), where b =0.8 is chosen so that a pre-specified percentage of censoring is achieved. Then all simulated variables are merged into a data frame.

```
options(width = 50)
CensorT <- runif(n, min = 0, max = 0.8)
# merge into a data frame
dt <- data.frame(EventT = pmin(EventT, CensorT),
  EventFlg = EventT < CensorT, X_1, X_2,
  X_3, ExposeFlg = -log(1 - U) >= lambda *
    exp(LinFix) * S & CensorT > S)
dt$ExposeT <- pmin(dt$EventT, S)
head(dt, 5)
##      EventT EventFlg        X_1        X_2
## 1 0.07963784    FALSE  0.93208196 -1.3232478
## 2 0.04824934    FALSE -1.03154448 -0.5216332
## 3 0.13187827     TRUE -0.09910448  0.1234164
## 4 0.05206076     TRUE  0.29929004  1.7372739
## 5 0.23296485    FALSE  0.80511751  0.6064613
##          X_3 ExposeFlg    ExposeT
## 1 -0.2277021     FALSE 0.07963784
## 2 -0.8751707     FALSE 0.04824934
## 3  2.0982547      TRUE 0.03277726
## 4  0.2783973      TRUE 0.03840657
## 5 -0.3938989      TRUE 0.02207852
dt$id <- 1:n
table(dt$ExposeFlg)
##
## FALSE  TRUE
##  1116   884
table(dt$EventFlg)
##
## FALSE  TRUE
##  1291   709
```

The *dt* is the prototype of a data set for survival analysis with time-dependent exposure. *EventT* is the observed time until censoring or the event of interest (i.e., mortality, recurrence and development of complications) occurs. *EventFlg* is whether a subject is right censored *FALSE* or event occurs *TRUE*. *X_1* to *X_3* are time-fixed covariates. *ExposeFlg* is exposure status and *ExposeT* is the time on which a subject is exposed. Note that the time-dependent exposure is a stochastic process (counting process) that equals zero from $t_0$ until exposure, then it equals to one until the end of observation.

### *Alternative method to generate a data frame with counting process*

Another way to generate simulated dataset with time-dependent exposure is to loop through individual subjects. We adapt code used in the *genTDCM()* function in the *genSurv* package (5,6).

```
mat <- matrix(ncol = 8, nrow = 1)
for (k in 1:n) {
  status <- 1
  u <- U[k]
  z1 <- X[k, ]
  c <- CensorT[k]
  if (u < 1 - exp(-lambda * S[k] * exp(LinFix[k]))) {
    t <- -log(1 - u)/(lambda * exp(LinFix[k]))
    z2 <- 0
  } else {
    t <- -(log(1 - u) + lambda * S[k] *
      exp(LinFix[k]) * (1 - exp(beta_t)))/(lambda *
      exp(LinFix[k] + beta_t))
    x12 <- S[k]
    z2 <- 1
  }
  time <- min(t, c)
  ifelse(t > c, status <- 0, status <- 1)
  if (u < 1 - exp(-lambda * S[k] * exp(LinFix[k]))) {
    aux1 <- c(k, 0, time, status, z1,
      0)
    mat <- rbind(mat, aux1)
  } else {
    if (c > x12) {
      aux1 <- c(k, 0, x12, 0, z1, 0)
      mat <- rbind(mat, aux1)
      aux2 <- c(k, x12, time, status,
        z1, 1)
      mat <- rbind(mat, aux2)
    } else {
      aux1 <- c(k, 0, time, status,
        z1, 0)
      mat <- rbind(mat, aux1)
    }
  }
}
data <- data.frame(mat, row.names = NULL)
names(data) <- c("id", "start", "stop", "event",
  "X_1", "X_2", "X_3", "tdcov")
data <- data[-1, ]
row.names(data) <- as.integer(1:nrow(data))
head(data)
```

```
##   id      start       stop event         X_1
## 1  1 0.00000000 0.07963784     0  0.93208196
## 2  2 0.00000000 0.04824934     0 -1.03154448
## 3  3 0.00000000 0.03277726     0 -0.09910448
## 4  3 0.03277726 0.13187827     1 -0.09910448
## 5  4 0.00000000 0.03840657     0  0.29929004
## 6  4 0.03840657 0.05206076     1  0.29929004
##          X_2        X_3 tdcov
## 1 -1.3232478 -0.2277021     0
## 2 -0.5216332 -0.8751707     0
## 3  0.1234164  2.0982547     0
## 4  0.1234164  2.0982547     1
## 5  1.7372739  0.2783973     0
## 6  1.7372739  0.2783973     1
```

The generated *data* is the same as *dt*, except that *dt* is not expanded. The variable *tdcov* corresponds to the *ExposeFlg*. In the next chunk, we are going to expand the *dt* with the *tmerge()* function so that each subject can have multiple intervals and thus takes multiple rows.

## Effect estimates using Cox regression model with time-varying exposure

Because the exposure in the example is time-varying, i.e., subjects can receive treatment at different observation period, the treatment effect can be estimated by using the *coxph()* function (7). The following chunk first splits the variables into those that are time-varying (*dtTV*) or time-fixed (*dtBase*), and then generate the counting process table with *tmerge()* function.

```
dtBase <- dt[, c(1:5, 8)]
dtTV <- dt[, 6:8]
library(survival)
dtLong <- tmerge(dtBase, dtBase, id = id,
  endpt = event(time = EventT, as.numeric(EventFlg)))
dtLong <- tmerge(dtLong, dtTV, id = id, Expose.flg =
tdc(ExposeT,
  as.numeric(ExposeFlg)))
dtLong$Expose.flg <- as.numeric(!is.na(dtLong$Expose.flg))
head(dtLong)
##       EventT EventFlg         X_1        X_2
## 1 0.07963784    FALSE  0.93208196 -1.3232478
## 2 0.04824934    FALSE -1.03154448 -0.5216332
## 3 0.13187827     TRUE -0.09910448  0.1234164
## 4 0.13187827    TRUE -0.09910448  0.1234164
## 5 0.05206076    TRUE  0.29929004  1.7372739
## 6 0.05206076    TRUE  0.29929004  1.7372739
##          X_3 id     tstart      tstop endpt
## 1 -0.2277021  1 0.00000000 0.07963784     0
## 2 -0.8751707  2 0.00000000 0.04824934     0
## 3  2.0982547  3 0.00000000 0.03277726     0
## 4  2.0982547  3 0.03277726 0.13187827     1
## 5  0.2783973  4 0.00000000 0.03840657     0
## 6  0.2783973  4 0.03840657 0.05206076     1
##   Expose.flg
## 1          0
## 2          0
## 3          0
## 4          1
## 5          0
## 6          1
```

The reformatted data frame is the same as the *data* generated above, with *tstart* and *tstop* corresponding to the *start* and *stop* variables in the *data*, respectively. Next, we will fit a Cox model with time-varying exposure.

```
modCoxTV <- coxph(Surv(tstart, tstop, endpt) ~
  X_1 + X_2 + X_3 + Expose.flg, data = dtLong)
library(tableone)
## Warning: package 'tableone' was built under R
## version 3.5.2
ShowRegTable(modCoxTV, exp = F)
##            coef [confint]       p
## X_1        0.92 [0.83, 1.01]  <0.001
## X_2        0.89 [0.80, 0.98]  <0.001
## X_3        0.94 [0.85, 1.04]  <0.001
## Expose.flg -0.42 [-0.62, -0.23] <0.001
```

The estimated treatment effect is −0.42 (P<0.001), which has little biased, and importantly the corresponding confidence interval includes the true effect of −0.5.

### PSA by considering exposure as a binary variable

Conventionally, patients are simply divided into those exposed and non-exposed groups without considering the time-to-exposure property. In this scenario, PS can be generated by regressing the exposure status $Z_1(t)$ on baseline covariates $X$. Here we define the $PS_1 = \widehat{\psi_1}X_1 + \widehat{\psi_2}X_2 + \widehat{\psi_3}X_3 + \widehat{\psi_0}$ as a linear combination of covariates, which is monotone function of the probability of treatment exposure. Finally, the treatment effect can be estimated by adjusting for this $PS_1$.

```
PSmodLogit <- glm(ExposeFlg ~ X_1 + X_2 +
  X_3, dt, family = "binomial")
# propensity score for each subject
psLogit <- predict.glm(PSmodLogit)
psLogit[1:10]
##          1          2          3          4
## -0.7100094 -1.7237368  1.0244820  1.1132018
##          5          6          7          8
##  0.2938331  0.8558305  0.1828180 -0.5405188
##          9         10
## -0.4656803  0.5068986
EffectMod <- coxph(Surv(EventT, EventFlg) ~
  ExposeFlg + psLogit, data = dt)
ShowRegTable(EffectMod, exp = F)
##               coef [confint]        p
## ExposeFlgTRUE -1.83 [-2.02, -1.65] <0.001
## psLogit        1.97 [1.85, 2.08]   <0.001
```

The estimated coefficient for the exposure is −1.83 (95% CI: −2.02 to −1.65), which is significantly downward biased comparing to the true effect of −0.5. PS can be used to match subjects with similar probability of receiving treatment exposure (8,9). Let's see how the estimated treatment effect differs from the true one when the time-to-exposure is ignored. Here we use the *MatchIt* package to perform PS matching (10).

```
library(MatchIt)
m.out <- matchit(ExposeFlg ~ X_1 + X_2 +
  X_3, data = dt, method = "nearest", distance = "logit")
m.data <- match.data(m.out)
ShowRegTable(coxph(Surv(EventT, EventFlg) ~
  ExposeFlg, data = m.data), exp = F)
##               coef [confint]        p
## ExposeFlgTRUE -0.32 [-0.47, -0.17] <0.001
```

The results show that the estimated treatment effect is slightly upward biased comparing to the true effect.

### PS generated by Cox regression model

In this instance, the PS is estimated by regressing the time to exposure on $X$ with Cox regression model. The Cox regression model gives estimated coefficients $\widehat{\phi}_0$, $\widehat{\phi}_1$, $\widehat{\phi}_2$ and $\widehat{\phi}_3$. Here we define $PS_2 = \widehat{\phi}_1 X_1 + \widehat{\phi}_2 X_2 + \widehat{\phi}_3 X_3 + \widehat{\phi}_0$ as a linear combination of covariates with estimated coefficients obtained from the Cox regression model. Strictly speaking, the $PS_2$ cannot be called a PS because PS should be the probability of receiving treatment given covariates. $PS_2$ is a surrogate for PS. The Cox regression model concerns the time-to-exposure feature in the estimation of $PS_2$. Then the treatment effect is estimated in multivariate model by adjusting for the $PS_2$.

```
PSmodCox <- coxph(Surv(time = ExposeT, ExposeFlg) ~
  X_1 + X_2 + X_3, data = dt)
psCox <- predict(PSmodCox, type = "lp")
EffectMod <- coxph(Surv(EventT, EventFlg) ~
  ExposeFlg + psCox, data = dt)
ShowRegTable(EffectMod, exp = F)
##               coef [confint]       p
## ExposeFlgTRUE -1.83 [-2.01, -1.65] <0.001
## psCox          1.21 [1.14, 1.28]   <0.001
```

The results show that the coefficient for the exposure is −1.83, which is an underestimation of the true treatment effect. The reason is probably that adjusting for $PS_2$ directly in the outcome model is subject to model misspecification and thus fail to recover true effects. This lead us to consider PS matching, a non-parametric technique to reduce model dependence (8,10).

## Time-dependent PS matching

The PS is estimated by Cox proportional hazard regression model, regressing time-to-exposure on time-fixed and time-varying covariates (e.g., only time-fixed covariates *Xs* are available in our example). Then the matching can be performed by sequential matching or simultaneous matching (11). Here we implement the sequential matching algorithm. The sequential matching takes place within risk set $R_t$ at time $t$; $R_t$ consists of all patients at risk of exposure at $t$. The matching procedure proceeds chronologically for each of the risk sets. There can be several subjects being treated at each risk set, and these patients compete with each other for matching controls. Note that the at-risk subjects being matched include those who are not exposed before or within the time interval $t$, and thus they also include those who are exposed latter. Remember that the matching should not depend on future data. The optimal matching can be easily performed by using the *optmatch* package (12). Then, matched individual subjects are removed from later risk sets $R_{t+1,t+2,...}$, and the process continues with the next risk set $R_{t+1}$ (11).

```
PSmodCox <- coxph(Surv(time = ExposeT, ExposeFlg) ~
  X_1 + X_2 + X_3, data = dt)
surObj <- survfit(PSmodCox, newdata = dt)
PScore <- data.frame(Time = surObj$time,
  surObj$cumhaz)
```

The above code chunk regresses the time to exposure on covariates *Xs* and estimates cumulative hazard for each individual over all time points that are experienced by sample patients. The *survfit()* function generates values for creating survival curves from previously fitted Cox model *PSmodCox* (13). Thus, the resulting *PScore* has the dimension of 2,000×2,001 with rows and columns corresponding to the time points and individual subjects, respectively. The values in the matrix are cumulative hazards. Next, let's reshape the matrix for further matching.

```
PScoreTolong <- reshape(PScore, direction = "long",
  varying = paste("X", 1:2000, sep = ""),
  sep = "", v.names = "cumhaz", timevar = "PtID")
dtScore <- merge(PScoreTolong[, -4], dt,
  by.x = "PtID", by.y = "id")
dim(dtScore)
## [1] 4000000      10
head(dtScore)
##   PtID         Time       cumhaz    EventT
## 1    1 0.0000549661 0.000000e+00 0.07963784
## 2    1 0.0001406952 6.982241e-05 0.07963784
## 3    1 0.0001826458 6.982241e-05 0.07963784
## 4    1 0.0002024824 6.982241e-05 0.07963784
## 5    1 0.0002539862 6.982241e-05 0.07963784
## 6    1 0.0003042782 1.407800e-04 0.07963784
##   EventFlg      X_1       X_2        X_3
## 1    FALSE 0.932082 -1.323248 -0.2277021
## 2    FALSE 0.932082 -1.323248 -0.2277021
## 3    FALSE 0.932082 -1.323248 -0.2277021
## 4    FALSE 0.932082 -1.323248 -0.2277021
## 5    FALSE 0.932082 -1.323248 -0.2277021
## 6    FALSE 0.932082 -1.323248 -0.2277021
##   ExposeFlg   ExposeT
## 1    FALSE 0.07963784
## 2    FALSE 0.07963784
## 3    FALSE 0.07963784
## 4    FALSE 0.07963784
## 5    FALSE 0.07963784
## 6    FALSE 0.07963784
```

The *reshape()* function is employed to convert wide to long format (14). Then the long format data frame is merged with the original data frame *dt*. The resulting *dtScore* has 4,000,000×7 dimension. The matching procedure have to proceed at a specified time interval; thus the entire follow up time is divided into 10 equally spaced intervals (e.g., each interval contains equal number of subjects). We create a new variable *TimeStrata* to store the information.

```
# Create time invertal for matching
strataNo = 10
breaks <- quantile(PScore$Time, seq(0, 1,
  length.out = strataNo + 1))
dtScore$TimeStrata <- cut(dtScore$Time, breaks = breaks,
  labels = 1:10, include.lowest = T)
dtScore$StrataCut <- NA
for (ii in 1:10) {
  dtScore$StrataCut <- ifelse(dtScore$TimeStrata ==
    ii, breaks[ii + 1], dtScore$StrataCut)
}
dtScore$ExposeFlgStrata <- ifelse(dtScore$ExposeFlg,
  dtScore$ExposeT <= dtScore$StrataCut,
  dtScore$ExposeFlg)
```

The *dtScore* must be formatted for matching procedure. Each individual subject has 2,000 rows with each representing a unique follow up time point. We only need to keep one row with the maximum cumulative hazard for each combination of *PtID* and *TimeStrata*, which can be easily performed by using the *ddply()* function.

```
library(plyr)
dtScoreStrata <- ddply(dtScore, .(PtID, TimeStrata),
  function(xx) {
    xx[xx$cumhaz == max(xx$cumhaz), ][1,
      ]
  })
library(lattice)
densityplot(~X_1 | TimeStrata, group = ExposeFlgStrata,
  dtScoreStrata, xlab = "X_1", auto.key = T)
```

The density plot shows that the distribution of *X_1* is different between the treated and the controls (*Figure 1*). Let's see how to match them (i.e., select control subjects with similar cumulative hazard to the treated in each risk set) with sequential matching.

```
# match
library(optmatch)
## Warning: package 'optmatch' was built under R
## version 3.5.2
## The optmatch package has an academic license. Enter re-
laxinfo() for more information.
dtFull <- dtScoreStrata
DtMatched <- NULL
for (ii in 1:strataNo) {
  dtStrata1 <- dtFull[dtFull$TimeStrata ==
    ii, ]
  if (sum(dtStrata1$ExposeFlg) != 0) {
    mahal.match <- pairmatch(match_on(ExposeFlgStrata ~
      cumhaz, data = dtStrata1), data = dtStrata1,
      controls = 1)
    DTwithGrp <- cbind(dtStrata1, matches = mahal.match)
    dtMatched <- DTwithGrp[!is.na(DTwithGrp$matches),
      ]
    dtFull <- dtFull[!(dtFull$PtID %in%
      dtMatched$PtID), ]
    DtMatched <- rbind(DtMatched, dtMatched)
  } else {
    next
  }
}
```

The sequential matching proceeds iteratively from *TimeStrata = 1 to 10*, thus we used a *for* loop to perform the task. Each *TimeStrata* is considered as a risk set, within which the exposed subjects is matched to the controls. Recall that we have estimated cumulative hazard for the controls at each time strata. Here the control subjects are those that are not exposed to the treatment before or within that *TimeStrata*, which can also include subjects who are exposed at latter time. In our example, equal number of controls are selected to match the treated subjects; but the treat/control ratio can be $1:n$. Then the matched samples are removed from the latter risk set. In above chunk, the *dtFull* begins with the full dataset and the number of rows

**Figure 1** Density plot showing distribution of X_1 in treated and control groups before matching across time strata.

is reducing with each iteration. In contrast, the *DtMatched* begins with nothing but the matched pairs are appended to it with each loop.

```
# the distribution of cumulative hazard
# between the two groups
densityplot(~X_1 | TimeStrata, group = ExposeFlgStrata,
  DtMatched, xlab = "X_1", auto.key = T)
ModPSM <- coxph(Surv(EventT, EventFlg) ~
  ExposeFlgStrata, data = DtMatched)
ShowRegTable(ModPSM, exp = F)
##                     coef [confint]      p
## ExposeFlgStrataTRUE -0.48 [-0.65, -0.31] <0.001
```

The result shows that the matched pairs have similar density distribution for *X_1* (*Figure 2*) and the estimated coefficient (–0.48) is unbiased to the true effect size, and importantly the corresponding confidence interval includes the true effect. This method outperforms all previous methods in term of absolute bias.

## Conditional logistic regression (CLR) model after sequential matching

Alternatively, the CLR model can be used to estimate the association between exposure and survival outcome after sequential matching (15). Recall that CLR is a specialized type of logistic regression usually employed when exposed/treated subjects with particular features are each matched with *n* control subjects with similar features. In our example, the exposed subjects were matched to control subjects with similar cumulative hazard within each *TimeStrata*. The matched dataset is stored in the data frame *DtMatched*.

```
library(survival)
ModCLR <- clogit(EventFlg ~ ExposeFlgStrata +
  strata(TimeStrata), data = DtMatched)
ShowRegTable(ModCLR, exp = F)
##                     coef [confint]      p
## ExposeFlgStrataTRUE -0.42 [-0.66, -0.18]  0.001
```

The output shows that the estimated treatment effect is comparable to that estimated using Cox regression model after sequential matching.

## Simulation study to compare PSs generated by treating treatment as binary and time-to-exposure variable

In the simulation study we compare two methods (i.e., those

**Figure 2** Density plot showing distribution of X_1 in treated and control groups after matching across time strata.

with and without considering the time-to-exposure property of the treatment) to calculate the PS, as the time-to-exposure property is always ignored in the literature, introducing the immortal time bias (16). After the comparison, we show the equivalence of time-dependent PS matching method and Cox regression model with time-varying covariates. The former is recommended as the first choice for such study design, because it also allows time-varying covariates to be included for the estimation of PS. Firstly, we define a function for the time-dependent PS matching.

```
TDPSM <- function(dt, ExposeT, ExposeFlg,
  id, Cov, strataNo = 10, StrataFlg = T) {
  n = nrow(dt)
  names(dt)[names(dt) %in% id] <- "id"
  names(dt)[names(dt) %in% ExposeT] <- "ExposeT"
  names(dt)[names(dt) %in% ExposeFlg] <- "ExposeFlg"
   PSmodCox <- coxph(formula(paste("Surv(time =
ExposeT,ExposeFlg) ~",
    paste(Cov, collapse = "+"))), data = dt)
  surObj <- survfit(PSmodCox, newdata = dt)
  PScore <- data.frame(Time = surObj$time,
    surObj$cumhaz)
  PScoreTolong <- reshape(PScore, direction = "long",
    varying = paste("X", 1:n, sep = ""),
    sep = "", v.names = "cumhaz", timevar = "PtID")
  dtScore <- merge(PScoreTolong[, -4],
    dt, by.x = "PtID", by.y = "id")
  if (StrataFlg) {
    breaks <- quantile(PScore$Time, seq(0,
      1, length.out = strataNo + 1))
    dtScore$TimeStrata <- cut(dtScore$Time,
      breaks = breaks, labels = 1:strataNo,
      include.lowest = T)
  } else {
    breaks <- c(0, sort(unique(dtScore$ExposeT)))
    strataNo = length(breaks) - 1
    dtScore$TimeStrata <- cut(dtScore$Time,
      breaks = breaks, labels = 1:strataNo,
      include.lowest = T)
  }

  dtScore$StrataCut <- NA
  for (ii in 1:strataNo) {
    dtScore$StrataCut <- ifelse(dtScore$TimeStrata ==
      ii, breaks[ii + 1], dtScore$StrataCut)
  }
  dtScore$ExposeFlgStrata <- ifelse(dtScore$ExposeFlg,
    dtScore$ExposeT <= dtScore$StrataCut,
    dtScore$ExposeFlg)
  dtScoreStrata <- ddply(dtScore, .(PtID,
    TimeStrata), function(xx) {
    xx[xx$cumhaz == max(xx$cumhaz), ][1,
      ]
  })

  dtFull <- dtScoreStrata
  DtMatched <- NULL
  for (ii in 1:strataNo) {
    dtStrata1 <- dtFull[dtFull$TimeStrata ==
      ii, ]
    if (sum(dtStrata1$ExposeFlg) != 0) {
      mahal.match <- pairmatch(match_on(ExposeFlgStrata ~
        cumhaz, data = dtStrata1),
        data = dtStrata1, controls = 1)
      DTwithGrp <- cbind(dtStrata1,
        matches = mahal.match)
      dtMatched <- DTwithGrp[!is.na(DTwithGrp$matches),
        ]
      dtFull <- dtFull[!(dtFull$PtID %in%
        dtMatched$PtID), ]
      DtMatched <- rbind(DtMatched,
        dtMatched)
    } else {
      next
    }
  }
  return(DtMatched)
}
```

The above chunk defines a function *TDPSM()* which receives a dataframe containing time-dependent exposure, covariates and survival outcome. The *strataNo* argument defines the number of strata for continuous recorded data. However, for follow up data at several fixed time points, the number of strata is not needed and we need to set *StrataFlg* = *F* to switch off the *strataNo* argument. The following chunk is to repeat the computation for a number of times (*ii* =100) to see whether time-dependent PS method is superior to the conventional PS method by treating treatment as a binary variable.

```
library(genSurv)
library(plyr)
library(optmatch)
library(MatchIt)
dtCov <- data.frame()
for (ii in 1:100) {
  n = 2000
  for (ii in 1:3) {
    assign(paste("X", ii, sep = "_"),
      rnorm(n))
  }
  X = cbind(X_1, X_2, X_3)
  lambda = 1
  alpha_0 = 1
  alpha_1 = 1
  alpha_2 = 1
  alpha_3 = 1
  ExpLin <- cbind(1, X) %*% c(alpha_0,
    alpha_1, alpha_2, alpha_3)
  S = -log(runif(n))/(lambda * exp(ExpLin))
  for (ii in 1:3) {
    assign(paste("beta", ii, sep = "_"),
      1)
  }
  beta_t = -0.5
  U = runif(n)
  LinFix <- X %*% c(beta_1, beta_2, beta_3)
  EventT <- ifelse(-log(1 - U) < lambda *
    exp(LinFix) * S, -log(1 - U)/(lambda *
    exp(LinFix)), (-log(1 - U) - lambda *
    exp(LinFix) * S + lambda * exp(LinFix +
    beta_t) * S)/(lambda * exp(LinFix +
    beta_t)))
  CensorT <- runif(n, min = 0, max = 0.8)
  # merge into a data frame
  dt <- data.frame(EventT = pmin(EventT,
    CensorT), EventFlg = EventT < CensorT,
    X_1, X_2, X_3, ExposeFlg = -log(1 -
      U) >= lambda * exp(LinFix) *
      S & CensorT > S)
  dt$ExposeT <- pmin(dt$EventT, S)
  dt$id <- 1:n
  # survival analysis with time-varying
  # covariate
  dtBase <- dt[, c(1:5, 8)]
  dtTV <- dt[, 6:8]
  dtLong <- tmerge(dtBase, dtBase, id = id,
    endpt = event(time = EventT, as.numeric(EventFlg)))
  dtLong <- tmerge(dtLong, dtTV, id = id,
    Expose.flg = tdc(ExposeT, as.numeric(ExposeFlg)))
  dtLong$Expose.flg <- as.numeric(!is.na(dtLong$Expose.flg))
  modCoxTV <- coxph(Surv(tstart, tstop,
    endpt) ~ X_1 + X_2 + X_3 + Expose.flg,
    data = dtLong)
  Covtrue <- coef(modCoxTV)["Expose.flg"]
  # PS binary adjustment
  PSmodLogit <- glm(ExposeFlg ~ X_1 + X_2 +
    X_3, dt, family = "binomial")
  psLogit <- predict.glm(PSmodLogit)
  CovbiAdj <- coef(coxph(Surv(EventT, EventFlg) ~
    ExposeFlg + psLogit, data = dt),
    exp = F)[1]
  # PS binary outcome matched
  m.out <- matchit(ExposeFlg ~ X_1 + X_2 +
    X_3, data = dt, method = "nearest",
    distance = "logit")
  m.data <- match.data(m.out)
  CovbiMat <- coef(coxph(Surv(EventT, EventFlg) ~
    ExposeFlg, data = m.data), exp = F)
  # time-dependent PS
  DtMatched <- TDPSM(dt = dt, ExposeT = "ExposeT",
    ExposeFlg = "ExposeFlg", id = "id",
    Cov = c("X_1", "X_2", "X_3"))
  CovPStd <- coef(coxph(Surv(EventT, EventFlg) ~
    ExposeFlgStrata, data = DtMatched))
  triCov <- c(Covtrue, CovbiAdj, CovbiMat,
    CovPStd)
  dtCov <- rbind(dtCov, triCov)
  cat(".")
}
## ..................................................................................
........
names(dtCov) <- c("coefT", "coefPSbiAdj",
  "coefPSbiMat", "coefPStd")
sapply(dtCov, summary)
##             coefT coefPSbiAdj coefPSbiMat
## Min.    -0.7657473   -2.097617 -0.41360403
## 1st Qu. -0.5932910   -1.954420 -0.27906475
## Median  -0.5087681   -1.890186 -0.19954484
## Mean    -0.5181903   -1.883439 -0.21359265
## 3rd Qu. -0.4457967   -1.822158 -0.15428498
## Max.    -0.2778042   -1.692134  0.02120892
##          coefPStd
## Min.    -0.6333721
## 1st Qu. -0.5441774
## Median  -0.4999176
## Mean    -0.4962544
## 3rd Qu. -0.4460087
## Max.    -0.3577452
```

The above output shows that the coefficients obtained by survival model with time-varying covariate and by time-dependent PS matching are very close to the true value of -0.5. However, the coefficient obtained by treating exposure as binary variable is much greater than the true value, suggesting bias with this method. In the absence of time-varying covariates as in our example, both methods can be considered equivalently. However, time-dependent PS matching is recommended in the presence of time-varying covariates determining the hazard of exposure.

## Application of time-dependent PS matching to the Kawasaki dataset

The Kawasaki dataset is a cohort of children with Kawasaki disease, which has been described elsewhere (17). A random sample of 500 cases is used for the illustration purpose. In the study we intend to explore whether time-varying standard treatment as represented by *treat* and *treattime* can help to ameliorate the coronary artery lesion (CAL). CAL is recorded as a time-to-event outcome with 1 stands for lesion presence and 0 otherwise. *ytime* represents the observation time. Other time-fixed covariates included:

- *agemonth*: age in month at presentation;
- *gender*: 0 for female and 1 for male;
- *bithweight2*: birth weight in kilogram;
- *BMI*: body mass index;
- *feverday*: duration of fever in days at presentation;
- *keshi*: whether there is transferring between departments;
- *Kdgroup*: 1 for complete and 0 for incomplete Kawasaki;
- *preHB*, *prePLT*, and *preALB* are laboratory findings for hemoglobin, platelet and albumin, respectively.

```
dtKD <- read.csv("https://raw.githubusercontent.com/zh-
zhang1984/big-data-clinical-trial-column/master/dtKD.csv")
Cov <- names(dtKD)[c(3:7, 10:15)]
```

```
library(survival)
library(plyr)
library(optmatch)
DtMatched <- TDPSM(dt = dtKD, id = "ID",
  ExposeT = "treattime", ExposeFlg = "treat",
  StrataFlg = F, Cov = Cov)
CoxMod <- coxph(Surv(time = ytime, event = y) ~
  ExposeFlgStrata, data = DtMatched)
ShowRegTable(CoxMod, exp = F)
##                  coef [confint]      p
## ExposeFlgStrataTRUE -0.05 [-0.54, 0.45]  0.859
```

The result shows that there is no evidence to conclude that the outcome of patients between standard versus non-standard treatment is different. Readers can try to estimate the treatment effect with Cox regression model by treating the exposure as a time-varying covariate.

## Conclusions

The article provides a comprehensive tutorial about the statistical tools to analyze survival data with time-varying exposures. We firstly simulate a dataset with time-varying exposure as a working example. Several approaches are performed to estimate the association between the exposure and survival outcome. Through simulation study, we found that the conventional PSM without considering the time-to-exposure property significantly biased the true effect because it did not take the time component into account. Multivariate adjustment with linear predictors from Cox and logistic regression model are also unable to estimate the true effect unbiasedly. Including time-varying exposure in a Cox regression model or creating matched cohort by time-dependent PS matching are recommended to reduce potential confounding bias. However, we still recommend the time-dependent PS matching approach because it also allows the inclusion of time-varying covariates affecting the hazard of exposure and also does not rely on strong model assumptions. After sequential matching with time-dependent PS, the treatment effect can be consistently identified by Cox regression model or CLR model.

## Acknowledgments

*Funding:* This project is supported by the key soft science projects of Zhejiang Science and Technology Department (2019C25009).

## Footnote

*Provenance and Peer Review:* This article was commissioned by the Guest Editor (Zhongheng Zhang) for the series "Big-data Clinical Trial" published in Annals of Translational Medicine. The article was sent for external peer review organized by the Guest Editor and the editorial office.

*Conflicts of Interest:* The series "Big-data Clinical Trial" was commissioned by the editorial office without any funding or sponsorship. The authors have no other conflicts of interest to declare.

*Ethical Statement:* The authors are accountable for all aspects of the work in ensuring that questions related to the accuracy or integrity of any part of the work are appropriately investigated and resolved.

## References

1. Andersen LW, Granfeldt A, Callaway CW, et al. Association Between Tracheal Intubation During Adult In-Hospital Cardiac Arrest and Survival. JAMA 2017;317:494-506.
2. Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. Stat Med 2005;24:1713-23.
3. Xu R, Luo Y, Glynn R, et al. Time-dependent propensity score for assessing the effect of vaccine exposure on pregnancy outcomes through pregnancy exposure cohort studies. Int J Environ Res Public Health 2014;11:3074-85.
4. Austin PC. Generating survival times to simulate Cox proportional hazards models with time-varying covariates. Stat Med 2012;31:3946-58.
5. Meira-Machado L, Faria S. A simulation study comparing modeling approaches in an illness-death multi-state model. Commun Stat Simul Comput 2014;43:929-46.
6. Araújo A, Meira-Machado L, et al. GenSurv: Generating

multi-state survival data 2015. Available online: http://CRAN.R-project.org/package=genSurv
7. Terry M. Therneau, Patricia M. Grambsch. Modeling survival data: Extending the Cox model. New York: Springer, 2000.
8. Zhang Z. Propensity score method: a non-parametric technique to reduce model dependence. Ann Transl Med 2017;5:7-7.
9. Austin PC. An introduction to propensity score methods for reducing the effects of confounding in observational studies. Multivariate Behav Res 2011;46:399-424.
10. Ho DE, Imai K, King G, et al. MatchIt: Nonparametric preprocessing for parametric causal inference. J Statistical Software 2011;42:1-28.
11. Lu B. Propensity score matching with time-dependent covariates. Biometrics 2005;61:721-8.
12. Hansen BB, Klopfer SO. Optimal full matching and related designs via network flows. J Computational Graphical Statistics 2006;15:609-27.
13. Therneau TM. A package for survival analysis in 2015. Available online: https://CRAN.R-project.org/package=survival.
14. Wickham H. Reshaping data with the reshape package. Journal of Statistical Software 2007;21. http://www.jstatsoft.org/v21/i12/paper.
15. Andersen LW, Kurth T, Chase M, et al. Early administration of epinephrine (adrenaline) in patients with cardiac arrest with initial shockable rhythm in hospital: propensity score matched analysis. BMJ 2016;353:i1577.
16. Lévesque LE, Hanley JA, Kezouh A, et al. Problem of immortal time bias in cohort studies: example using statins for preventing progression of diabetes. BMJ 2010;340:b5087.
17. Shi H, Qiu H, Jin Z, et al. Coronary artery lesion risk and mediating mechanism in children with complete and incomplete Kawasaki disease. J Investig Med 2019;67:950-6.

**Cite this article as:** Zhang Z, Li X, Wu X, Qiu H, Shi H; written on behalf of AME Big-Data Clinical Trial Collaborative Group. Propensity score analysis for time-dependent exposure. Ann Transl Med 2020;8(5):246. doi: 10.21037/atm.2020.01.33