STATISTICS IN MEDICINE
Statist. Med. 2008; **27**:2618–2634
Published online 8 October 2007 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sim.3092

Comparison of algorithms to generate event times conditional on time-dependent covariates

Marie-Pierre Sylvestre and Michal Abrahamowicz*, †

Department of Epidemiology and Biostatistics, McGill University, 1020 Pine Avenue West, Montreal, Que., Canada H3A 1A2

SUMMARY

The Cox proportional hazards model with time-dependent covariates (TDC) is now a part of the standard statistical analysis toolbox in medical research. As new methods involving more complex modeling of time-dependent variables are developed, simulations could often be used to systematically assess the performance of these models. Yet, generating event times conditional on TDC requires well-designed and efficient algorithms. We compare two classes of such algorithms: permutational algorithms (PAs) and algorithms based on a binomial model. We also propose a modification of the PA to incorporate a rejection sampler. We performed a simulation study to assess the accuracy, stability, and speed of these algorithms in several scenarios. Both classes of algorithms generated data sets that, once analyzed, provided virtually unbiased estimates with comparable variances. In terms of computational efficiency, the PA with the rejection sampler reduced the time necessary to generate data by more than 50 per cent relative to alternative methods. The PAs also allowed more flexibility in the specification of the marginal distributions of event times and required less calibration. Copyright © 2007 John Wiley & Sons, Ltd.

KEY WORDS: survival analysis; simulations; time-dependent covariates; proportional hazards model; algorithms; rejection sampling

1. INTRODUCTION

Models extending the Cox proportional hazards (PH) model [1] to include time-dependent covariates (TDC) are now a part of the conventional statistical toolbox in medical research [2–8]. In such models, the hazard at time t is conditioned on the updated covariate value vector $x(t)$:

$$h(t|X) = h_0(t)\exp[\beta' x(t)] \tag{1}$$

*Correspondence to: Michal Abrahamowicz, McGill University Health Centre, 687 Pine Avenue West, V Building, Montreal, Que., Canada H3A 1A1.
†E-mail: michal.abrahamowicz@mcgill.ca, michal@epimgh.mcgill.ca

Contract/grant sponsor: National Sciences and Engineering Research Council of Canada
Contract/grant sponsor: Canadian Institutes for Health Research

Received 6 March 2007
Accepted 30 August 2007

Some examples of time-dependent processes in longitudinal studies are the measurements of laboratory tests or clinical symptoms that are often repeated over the follow-up [9]. At the modeling stage, updating the covariates in the model using TDC may be preferable to using their baseline values only, as the predictive ability of the baseline values may decrease over time [9]. Functions of TDC have also been used to model complex situations such as the history of drug treatment or irregularly measured TDC-like white blood cell counts in chronic leukemia patients [10, 11]. In a study on the association between the history of Benzodiazepine use and fall-related injuries in the elderly, Abrahamowicz *et al*. observed that combinations of TDC such as current and cumulative dose or duration of use, weighted by recency, improved the fit over simpler models [10]. In their discussion on modeling longitudinal data that are measured irregularly, Bruijne *et al*. proposed to include three TDC to model one time-dependent process: the current value, the time elapsed since last observation and their interaction [11]. TDC can also be used to test the validity of PHs assumption in Cox's model [8, 12]. Clearly, most of the new and more specialized models being developed for survival analysis, including modeling of the time-dependent effects, are or should be extendable to incorporate TDC [13–16].

As such new models are developed, it may be useful to validate their performance using simulations [17]. Indeed, simulations are essential to validate complex methods, such as non-parametric extension of Cox's model, for which it is difficult or impossible to establish the properties of the estimates and inferential statistics analytically [13, 15, 18, 19]. Furthermore, simulations often provide the only means to systematically compare the alternative models.

In their guidelines for the design of simulation studies, Burton *et al*. refer to two algorithms to generate time-to-event data [20]. The first, discussed by Bender *et al*., relies on inverted survival functions to generate survival times for the Cox model [17, 21]. In most cases, this algorithm cannot be used to generate events conditional on TDC because this would require inverting the expression $-H_0(t)\exp(\beta' x(t))$, where $H_0(t)$ is the cumulative baseline hazard function. When changes over time in $x(t)$ cannot be described by a parametric function or are not defined over the whole range of t, then $-H_0(t)\exp(\beta' x(t))$ cannot be inverted.

The second algorithm is the permutational algorithm (PA), first introduced by Abrahamowicz *et al*. [18], and described in details by MacKenzie and Abrahamowicz [22]. The PA is a flexible tool to generate event and censoring times that follow user-specified distributions and that are conditional on user-specified covariates, possibly with time-dependent or non-linear effects [22]. The crux of the algorithm is to perform a one-to-one matching of the n observed times T_i with n independently generated vectors of covariate values x_i, $i = 1, \ldots, n$ [22]. The matching is performed based on a permutation probability law derived from the partial likelihood of the Cox PH model [1, 22, 23]. The PA is asymptotically exact [22]. Simulations also showed that the accuracy of the algorithm increases with decreasing rates of censoring [22]. Until now, the algorithm has been validated in univariate models with a fixed-in-time-independent variable and an exponential distribution of events and small sample sizes (50–400) [22]. However, because time-dependent effects can be represented by artificial TDC [1, 18], the PA can be adapted to the case of TDC with constant coefficients [24]. The accuracy and efficiency of the PA for generating events conditional on TDC have not been evaluated.

Although not mentioned by Burton *et al*. [20], another approach to generating survival data with event times depending on TDC is to consider each consecutive small interval of follow-up time separately and to generate events within these intervals using a binomial model. This method is simple but there is no direct control over the distribution of event times. Moreover, the accuracy and efficiency of the binomial method also have to be evaluated.

DOI: 10.1002/sim

The objective of this paper is to compare two classes of algorithms to generate event times conditional on TDC. The remainder of the paper is structured as follows. The PA and the algorithm based on the binomial model are described in Section 2, where we also propose a new version of the PA that incorporates a rejection sampler to increase efficiency. In Section 3, we describe the design of the simulation study performed to assess and compare the performance of the algorithms. The results of the simulations, in terms of accuracy, stability, and computational time, are presented in Section 4. Concluding remarks are included in Section 5.

2. ALGORITHMS

Here we describe alternative algorithms to generate a data set of N individuals, with event times conditional on TDC according to the PH model in equation (1) and with random censoring. We assume that TDC are updated, and events are observed at m subsequent, equal-length time intervals.

2.1. Permutational algorithm

The adaptation of the PA [22] to generate event times conditional on TDC consists in the following five steps:

1. Generate N survival times T_i, $i=1,\ldots,N$ from a user-specified marginal distribution. PA allows the user to generate event times from an arbitrary pre-specified 'marginal' distribution, such as exponential, Weibull, or even a non-parametric distribution represented by a histogram. The marginal distribution is assumed to represent the distribution of events in the entire study population, regardless of the covariates.
2. Generate N censoring times C_i, $i=1,\ldots,N$ from a user-specified marginal distribution.
3. For each individual i, $i=1,\ldots,N$, let t_i^* be the last observed time, i.e. $t_i^*=\min\{T_i, C_i\}$, and δ_i an indicator of non-censoring, i.e. $\delta_i=I\{T_i\leqslant C_i\}$. Sort the N survival status tuples (t_i^*, δ_i^*) so that $t_i^*\leqslant t_{i+1}^*$.
4. Generate N individual matrices of covariate values X_s, $s=1,\ldots,N$. Each matrix has m rows, each of which represents an interval of follow-up time and p columns, corresponding to both fixed-in-time and TDC. In Section 3.2, we describe in detail how the covariate generation is performed. Define $X_s(t)$, the vector of covariate values at time t $(t=1,\ldots,m)$, for subject s $(s=1,\ldots,N)$, as the row of the matrix X_s corresponding to time t.
5. Starting from the earliest observed time t_1^*, randomly assign each consecutive survival status tuple (t_i^*, δ_i^*), $i=1,\ldots,N$ to a vector of current covariate values $X_s(t_i^*)$, $s=1,\ldots,N$.
   (a) If $\delta_i^*=1$, i.e. if t_i^* represents an event, then covariate vectors are sampled with probabilities based on the partial likelihood of the PH model [1]. Accordingly, for an individual s at time t_i^*, this probability is defined as

$$P_{s,t_i^*}=\frac{\exp[\beta' x_s(t_i^*)]}{\sum_{j\in R_i}\exp[\beta' x_j(t_i^*)]} \tag{2}$$

   where R_i is the risk set for time t_i^*. We define R_i as the set of vectors of covariate values at time t_i^*, from the individual covariate matrices that have not been selected yet.

DOI: 10.1002/sim

(b) If $\delta_i = 0$, assign a subject who is censored at time t_i^* by simple random sampling from the risk set R_i with equal probability $p_i = 1/\text{size}(R_i)$.

The subject s, associated with the vector $X_s(t_i^*)$, is then assigned the observed time t_i^*, and the corresponding matrix X_s represents a complete data history for this subject. Since t_i^* is the last observed time for that individual, all the rows of X_s for which $t > t_i^*$ are deleted from the individual data history. In addition, the selected covariate matrix X_s is then removed from all subsequent risk sets R_v $(v > t_i^*)$. Since in our context, each individual is initially fully described by his/her generated covariate vector, such matching is equivalent to sampling 'individuals' from the risk set R_i at time t_i^*.

The PA may become computationally intensive when large number of events need to be generated. The complexity of the PA is driven by step 5(a) in which the hazard ratios for *all* the subjects in the risk set must be calculated at every event time. In fact, when none of the subjects are censored, the computational complexity of PA grows in a quadratic fashion with sample size. In this case, PA proceeds as follows: at first, n hazard ratio calculations are required to assign the first event to a vector of covariate values. After the first event has been assigned, the corresponding vector is removed from the risk set and the same operations need to be performed on the remaining $(n-1)$ events, and so on until the last event is matched. Consequently, $M_{\text{PA}}(n)$, the total number of hazard ratio calculations required for PA to assign a total of n events to n covariate vectors equals

$$M_{\text{PA}}(n) = \sum_{i=1}^{n} i = \frac{n(n+1)}{2} \tag{3}$$

which is of order n^2. When censoring is allowed, the number of hazard ratio calculations is a quadratic function of the number of events generated. As a result, the time required to perform PA may become impractical for a large number of events.

2.2. *Permutational algorithm with rejection sampling (PARS)*

In order to improve the computational efficiency of the PA, we propose to replace the sampling method used in step 5(a) with a rejection sampler. The purpose is to avoid the systematic evaluation of the hazard ratio for all members of the risk set at each event time. Rejection sampling is a theoretically exact probabilistic algorithm that allows sampling from a distribution whose cumulative distribution function cannot be written in closed form and/or analytically inverted [25].

We briefly describe the basic principle of rejection sampling [26]. Suppose we want to generate a sample from a target probability distribution $P(x) = P^*(x)/Z_P$, where Z_P is a normalizing constant. To do so, we select a proposal distribution $Q(x) = Q^*(x)/Z_Q$, with Z_Q a normalizing constant, and $Q(x)$ a probability distribution that can readily be sampled from. The proposal distribution needs to satisfy $cQ^*(x) \geqslant P^*(x)$ for all x, with c a positive and finite constant. We can then draw a sample from $P(x)$ with the following steps [26, 27]:

1. Draw x from $Q(x)$.
2. Evaluate $cQ^*(x)$.
3. Draw U from U[0, 1].
4. If $U \leqslant P^*(x)/cQ^*(x)$, accept x as a value from $P(x)$, else reject x and go to step 1.

DOI: 10.1002/sim

The efficacy of the method depends on the choice of c and $Q^*(x)$. The distance between the envelope functions $cQ^*(x)$ and $P(x)$ should be small as to minimize the number of points that are rejected [25]. The number of trials required to sample one point from $P(x)$ has a geometric distribution with mean $c(Z_Q/Z_P)$ [26].

In the context of PA, the function $P(x)$ we wish to sample from is described in equation (2). Since, by definition, the values in (2) sum up to 1, it can be considered a probability distribution whose normalizing constant is $\sum_{j\in R_i}\exp\left[\beta' x_j(t_i^*)\right]$. An obvious candidate for the proposal distribution is $Q^*(x)=1$, with the normalizing constant size(R_t) so that $Q(x)$ corresponds to simple random sampling in the risk set R_i.

Over follow-up time, the target distribution in (2) changes because of (i) changes in TDC, and (ii) changes in the risk set, due to the elimination of subjects who were censored or had event in previous times. To maintain the efficiency of the algorithm, we update the constant c so as to keep the envelope function close to the target distribution. Setting $c_{t_i^*}$ to be the maximum hazard ratio among subjects in the current set at time t_i

$$c_{t_i}=\max(\exp[\beta' x_v(t_i)]),\quad v\in R_i \tag{4}$$

would represent an efficient choice. However, it would require calculating the hazard ratios for all the members of the risk set at each event time. Assuming that all regression coefficients (β's) are non-negative, the maximum hazard will correspond to the highest values of the covariates. For fixed-in-time covariates, we identify the empirical maximum among all N subjects and then updated this value each time a subject with the highest value was eliminated from the consecutive risk sets. For TDC, since identification of current empirical maxima would have required recalculating the TDC values for *all* subjects in the risk set, we used the theoretical maximum of the TDC at time t_i^* instead of the empirical maximum. For example, if the TDC was a binary indicator of current use, we used the maximum value of 1, while for a TDC representing the cumulative past duration of exposure at time t_i^*, we used t_i^*.

Therefore, the proposed PARS algorithm involves the following steps, for each observed event time t_i^*:

1. Sample a covariate vector $X_s(t_i^*)$ from the risk set R_i at time t_i^* of event, with equal probabilities $1/\text{size}(R_i)$.
2. Draw U from U[0, 1].
3. Compute the hazard ratio associated with the covariate vector X_s, $h(X_s(t_i^*))=\exp[\beta' x_s(t_i^*)]$.
4. If $U\leqslant h(X_s(t_i^*))/c_{t_i^*}$, match the covariate vector $X_s(t_i^*)$ with the event at time t_i^*. Otherwise, go back to step 1.

Consequently, the expected number of trials to sample one vector of covariate values from $P(x)$ at time t_i^* is

$$\frac{c_{t_i^*}\times\text{size}(R_i)}{\sum_{j\in R_i}\exp[\beta' x_j(t_i^*)]} \tag{5}$$

Note that equation (5) represents the ratio of the maximum hazard ratio at time t_i^* to the mean, across all subjects in risk set R_i, of the hazard ratios at the same time. This ratio will depend on both the strength of the covariate effects (β's) and the distribution of the covariates, with stronger effects and larger range of covariate values corresponding to the higher ratios. However, in most simulations, this ratio should not change drastically with an increasing size of the risk set. Assume,

DOI: 10.1002/sim

for simplicity, that in a given simulated scenario, the ratio is close to a constant r, independent of the size of the risk set. Then the total number of hazard ratio calculations required for PARS to assign a total of n uncensored events to n covariate vectors can be approximated by

$$M_{\mathrm{PARS}}(n)=\sum_{i=1}^{n} r=nr \tag{6}$$

This suggests that PARS is an $O(n)$ algorithm, i.e. its computing time is an approximately linear function of n. Moreover, comparison of equation (3) with (6) shows that the number of calculations required by PARS is smaller than that required by PA as long as $r<(n+1)/2$.

Because the maximum hazard ratio in the numerator of r may be expended to be close to the empirical maximum of size (R_i) components of the sum in its denominator, this inequality will be easily satisfied, and gains in efficiency obtained with PARS may be very important.

2.3. Algorithm based on a binomial model

The second class of algorithms that we considered was based on a binomial model. The binomial algorithm requires that the continuous follow-up time is partitioned into a finite number of m time intervals, which we assumed were of equal length. The algorithm involves three steps, performed iteratively from time $t=1$ to either the end of follow-up ($t=m$), or the time the individual is assigned the event or censored, separately for each individual $i=1,\ldots,n$ still at risk:

1. Compute the individual conditional probability of event $p_{i,t}$ based on a binomial model with parameters β_j, $j=1,\ldots,w$ corresponding to those of equation (1)

$$p_{i,t}=\mathrm{logit}\left[\beta_0+\sum_{j=1}^{w}\beta_j X_{j,i}(t)\right] \tag{7}$$

2. Generate $U_{i,t}$ from U[0, 1].
3. If $U_{i,t}\leqslant p_{i,t}$, assign an event to subject i at time t, i.e. $T_i=t$, and stop the follow-up for this subject. Otherwise, increase t by 1 unit and return to step 1.

Random censoring was applied by generating N censoring times C_i, $i=1,\ldots,N$ from a pre-specified probability distribution and, in the case where $C_i<T_i$, assigning the subject a censored status ($\delta_i=0$) at time $t_i=C_i$.

In addition to the parameters specified in equation (1), the user must supply a value for β_0 in equation (7). The pre-specified value of $\mathrm{logit}(\beta_0)$ represents the baseline risk, i.e. the probability of an event for an individual with all covariate values set at 0. A baseline risk that is constant over time implies event times that are exponentially distributed. The higher the baseline risk, the bigger p, the probability of event at a given time, is, and the more likely that events will be generated early in the follow-up. However, it is difficult to control the exact form of the resulting distribution of the generated event times. We further discuss the calibration of the binomial algorithm in Section 3.5.

We also implemented an algorithm based on the binomial model in which intercept (7) was time-dependent $\beta_0(t)$ to allow for baseline risk to change over time. This allowed some control over the distributions of the event times. For example, using a $\beta_0(t)$ that increases over time would produce an increasing hazard similar to a Weibull distribution with a shape parameter greater than 1.

DOI: 10.1002/sim

3. SIMULATIONS

We assessed the performance of the algorithms using simulations. The design of the simulations was based on the general features of our previous study, which investigated the association between the effect of current use and/or cumulative exposure to a prescribed drug represented by TDC, and an event, adjusting for fixed-in-time baseline covariates: age, sex, and a comorbidity index [10]. We simulated a one-year prospective cohort study. This section presents the protocol of the simulations and analysis [20], which comprised the generation of covariate values, the generation of events conditional on the covariates values according to two models, and the description of the set of parameters used in different scenarios.

3.1. Covariates' generation

First, the values of three fixed-in-time covariates were generated independently of each other. Specifically, we generated: age (AGE$\sim$N[75, 10]), a binary indicator for male sex (MALE$\sim$ Binomial[0.4]), and a quantitative index of comorbidity (COM$\sim$log N[1.6, 0.8]).

3.2. Exposure generation

Current drug exposure was represented by a time-dependent binary indicator (BIN(t)) taking the value of 1 when the subject was under treatment and 0 otherwise. To generate individual histories of drug use, we first assumed that the probability of being ever exposed during the 1-year follow-up (Pr(user$_i$ = 1)) depended on the fixed-in-time covariates, according to the following equation:

$$\Pr(\text{user}_i = 1) = \text{logit}[\gamma_0 + \log(1.01)\,\text{AGE}_i + \log(1.1)\,\text{MALE}_i + \log(1.05)\,\text{COM}_i] \tag{8}$$

where γ_0 was calculated by taking the logit of the proportion of subjects who were assumed to be ever exposed (0.3), minus the expression $\log(1.01)\,\text{AGE} + \log(1.1)\,\text{MALE} + \log(1.05)\,\text{COM}$, with the values of the three fixed-in-time covariates corresponding to their mean values.

Next, we assumed that users were equally likely to initiate drug use at any point during follow-up. Thus, once a user was identified, using equation (8) the date that she started the drug use was generated from a Uniform distribution (U[0, 365]). We also allowed drug treatment to be interrupted and resumed repeatedly: the duration of periods of drug use and the duration of interruptions were generated conditional on the fixed-in-time covariates and calculated in terms of weeks. Each of these periods was assumed to last at least two weeks. The duration of drug use was generated using the expression 14 days + 7 $\times$ log N[α_D, 5], where

$$\alpha_D = -0.5 + 0.002\,\text{AGE} + 0.001\,\text{COM} \tag{9}$$

The duration of interruptions of drug use was generated similarly, with 14 days + 7 $\times$ log N[α_I, 3] and

$$\alpha_I = 1 - 0.003\,\text{AGE} \pm 0.001\,\text{COM} \tag{10}$$

In summary, each user's history of drug exposure started with the date of first use, and then included subsequent periods of use and interruptions, as generated above, until day 365. Subjects assigned 1 for the BIN(t) indicator were considered users while the remaining subjects were assigned no exposure for the entire one-year follow-up. In addition to BIN(t), the current exposure TDC, we created an additional TDC that represented the cumulative duration of past drug use, $\text{CUM}(t) = \sum_{k=1}^{t} \text{BIN}(t)$.

DOI: 10.1002/sim

3.3. Events' generation

The algorithms described in Section 2 were used to generate event and random censoring times, with event depending on covariates, according to two models. *Model 1* assumed that the risk depended only on the three fixed covariates and on the current use of the drug. Thus, the hazard at time t was a function of the binary time-dependent indicator of current drug use (BIN(t)) and the three fixed-in-time covariates (AGE, MALE, and COM):

$$h(t|X, Z(t)) = h_0(t)\exp[\beta_{\mathrm{AGE}}\,\mathrm{AGE} + \beta_{\mathrm{MALE}}\,\mathrm{MALE} + \beta_{\mathrm{COM}}\,\mathrm{COM} + \beta_{\mathrm{BIN}}\,\mathrm{BIN}(t)] \tag{11}$$

Model 2 was similar to *Model 1*, except that it assumed that risk also depended on the cumulative past duration of drug use. Accordingly, *Model 2* included an additional time-dependent variable that represented the cumulative duration of past drug use, CUM(t). In *Model 2*, the hazard was defined as

$$\begin{aligned} h(t|X, Z(t)) = h_0(t)\exp[&\beta_{\mathrm{AGE}}\,\mathrm{AGE} + \beta_{\mathrm{MALE}}\,\mathrm{MALE} \\ &+ \beta_{\mathrm{COM}} * \mathrm{COM} + \beta_{\mathrm{BIN}}\,\mathrm{BIN}(t) + \beta_{\mathrm{CUM}}\,\mathrm{CUM}(t)] \end{aligned} \tag{12}$$

3.4. Scenarios for simulations

We considered five scenarios selected to cover a range of plausible clinical situations (Table I). Scenarios 1–4 investigated the effect of the magnitude and sign of the coefficients of BIN(t) and CUM(t) on the performance of the algorithms. Scenario 1 assumed that both the binary and cumulative drug TDC increased the risk of an event (HR = 1.5 and 1.006, respectively). Scenario 2 assumed a protective effect of current exposure (HR = 0.8). Scenario 3 investigated the effect of setting the coefficient of CUM(t) to zero, to verify whether the algorithms would create a spurious effect of the cumulative TDC. Therefore, only *Model 2* was estimated in Scenario 3. In these three scenarios, the percentage of events observed during the follow-up was set at 66 per cent, implying a 34 per cent censoring rate. The distribution of event times, unless stated otherwise, was Uniform for the PAs. For the binomial model with the time-dependent intercept, we used a spline function of time selected so as to approximate the Uniform distribution of event times. This was tested using a goodness-of-fit test.

Finally, we added two variations of Scenario 1, specifically: Scenarios 1a and 1b used the same parameters specification as in Scenario 1 but the percentage of events during follow-up was decreased to 50 per cent in Scenario 1a, while in Scenario 1b, the distribution of event times was shifted towards the end of the follow-up. This was achieved by using a Weibull(13, 310) for the PAs and a different spline function for the binomial algorithm with a time-dependent intercept (Table I). For simplicity, we held the coefficients of the fixed-in-time variables ($\beta_{\mathrm{AGE}} = \log(1.002)$, $\beta_{\mathrm{MALE}} = \log(0.95)$, $\beta_{\mathrm{COM}} = \log(1.02)$) constant across the five scenarios.

We performed these scenarios with a sample size of 750. For each scenario, 1000 independent random samples were generated, which ensured high precision of the estimation of the mean value of relevant parameters: the standard error of the mean did not exceed 3 per cent of the sample-to-sample empirical standard deviation of the 1000 corresponding estimates. For Scenario 1, we varied the sample size ($n = 100, 250, 500, 750$) to assess the impact of increased n on the time

DOI: 10.1002/sim

to generate the data. This allowed one to assess the effect on the precision and stability of the estimates when varying the sample size of the data set generated.

3.5. Calibration of the algorithms

We calibrated the parameters of the two algorithms so that, on average, two-thirds of the subjects would experience an event during follow-up. Such calibration was necessary to ensure valid comparison of the data generation times, which crucially depend on the number of generated events. For the PA, the only parameter to calibrate was B, the upper bound of the Uniform distribution of censoring. If the marginal distribution of the event times was Uniform over follow-up U$[1, m]$, and we wanted two-thirds of the observed times to be events then the corresponding censoring distribution is U$[1, 3m/2]$. In cases where the distribution of event times was not Uniform, B was selected by trial and error to obtain the desired number of events (see Table I for B values used in different scenarios).

Calibrating the binomial algorithm involved tuning two parameters, the intercept of the binomial model and the upper bound B of the censoring distribution. When the intercept was constant over time (β_0), we proceeded by first setting B equal to the length of follow-up, and then finding the intercept that would produce the desired percentage of events over 100 simulations. For the binomial algorithm with a time-dependent intercept $\beta_0(t)$, we first selected the function of time t that would produce a pre-censoring distribution of event times that was both reasonably close to the desired distribution and with events occurring before the end of follow-up. We then selected B by trial and error to obtain the desired percentage of events over 100 simulations. We tested if the pre-censoring distribution of event times did not diverge significantly from the desired distribution with a Kolmogorov–Smirnov goodness-of-fit test for not-entirely continuous distributions ($\alpha = 0.05$) [28]. The parameters selected through the calibration are shown in Table I.

Table I. Regression parameters (log(HR)) for the simulation scenarios.

Scenario	Distributions		Coefficients		Per cent censoring
	Event times	Censoring	$\beta_{\mathrm{BIN}(t)}$	$\beta_{\mathrm{CUM}(t)}$	
1	PA, PARS: U[0, 365]	U$[1, B = (3 \times 365)/2]$	log(1.5)	log(1.006)	33
	Binomial: $\beta_0 = -5.17$	U$[1, B = 365]$			
	Binomial: $\beta_0(t) =$ cubic spline*	U$[1, B = 438]$			
2	PA, PARS: U[0, 365]	U$[1, B = (3 \times 365)/2]$	log(0.8)	log(1.006)	33
	Binomial: $\beta_0 = -5.17$	U$[1, B = 365]$			
	Binomial: $\beta_0(t) =$ cubic spline*	U$[1, B = 438]$			
3	PA, PARS: U[0, 365]	U$[1, B = (3 \times 365)/2]$	log(1.5)	log(1)	33
	Binomial: $\beta_0 = -5.17$	U$[1, B = 365]$			
	Binomial: $\beta_0(t) =$ cubic spline*	U$[1, B = 438]$			
1a	PA, PARS: U[0, 365]	U$[1, B = 365]$	log(1.5)	log(1.006)	50
	Binomial: $\beta_0 = -5.72$	U$[1, B = 365]$			
	Binomial: $\beta_0(t) =$ cubic spline*	U$[1, B = 365]$			
1b	PA and PARS: Weibull(13,310)	U$[1, B = 890]$	log(1.5)	log(1.006)	33
	Binomial: $\beta_0(t) =$ cubic spline†	U$[1, B = 852]$			

*Interpolated cubic spline with seven knots at −6.1, −6.0, −5.8, −5.6, −5.3, −5, −3.1.
†Interpolated cubic spline with seven knots at −15, −15, −15, −9, −6, −3, −3.

DOI: 10.1002/sim

3.6. Evaluation of the performances of the algorithms

We assessed the performance of the algorithms by comparing (i) the estimates obtained by analyzing the data sets and (ii) the parameters of the model used to generate them. The data were always generated using the PH model and the covariates were assumed to be observed without errors, and represented adequately in the analysis. Thus, we know *a priori* that the estimates from the PH model from which the data were generated should be unbiased. Accordingly, as in Le Teuff *et al*. [29]; we assumed that any systematic bias in the estimates would be due to inaccuracy at the data-generating step.

3.6.1. Accuracy and stability of parameter estimates. We evaluated the potential bias in the parameter estimates to assess the accuracy of the algorithms. Bias was estimated as the difference between the mean of the estimated parameter values, averaged over simulations, and their corresponding true value. We also investigated the stability of the estimates obtained with different algorithms by comparing the empirical standard errors, defined as the standard deviations of the estimated parameters across simulations. We used the root mean square error (RMSE) of the estimates as an overall measure of accuracy, which reflected both bias and variance. Finally, we estimated the coverage rates, defined as the proportion of the simulated samples in which the respective 95 per cent confidence interval included the true parameter value.

3.6.2. Computational time. In Scenario 1, we compared the CPU time required by each of the algorithms to generate and analyze 1000 simulated data sets of size $n = 100$, 250, 500, and 750. To enhance the comparability of the algorithms, we used the same procedures to generate the covariate matrices, and to analyze the data. We expected the relationship between computational time and sample size to be quadratic ($O(n^2)$) for the PA and linear ($O(n)$) for the other algorithms. We investigated the association between the computational time and the sample generated size using linear regression. To capture potential non-linear associations between computational time and sample size, we fitted polynomial models of degree up to 3, as well as a model in which the effect of sample size was represented with an exponential term. Goodness-of-fit testing was then used to select the model most consistent with the observed times.

3.7. Software specification

The algorithms were coded in R 2.3.1 [30] and tested on a Pentium 4 3.00 GHz, with Linux Fedora Core 5 kernel 2.6.15. The random number generator used was the *Mersenne-Twister* [31]. To analyze the simulated data, we used the R procedure `coxph` [30] with robust standard errors and Breslow method for tie handling.

4. RESULTS

4.1. Calibration of the algorithms

The purpose of calibration was to ensure similar pre-specified number of events for each algorithm. Calibrating the PAs (PA and PARS) consisted in finding the upper bound of the Uniform distribution of censoring, which in scenarios with a Uniform distribution of event times over the follow-up could be determined in advance. In other scenarios, calibrating was a matter of a few trials.

DOI: 10.1002/sim

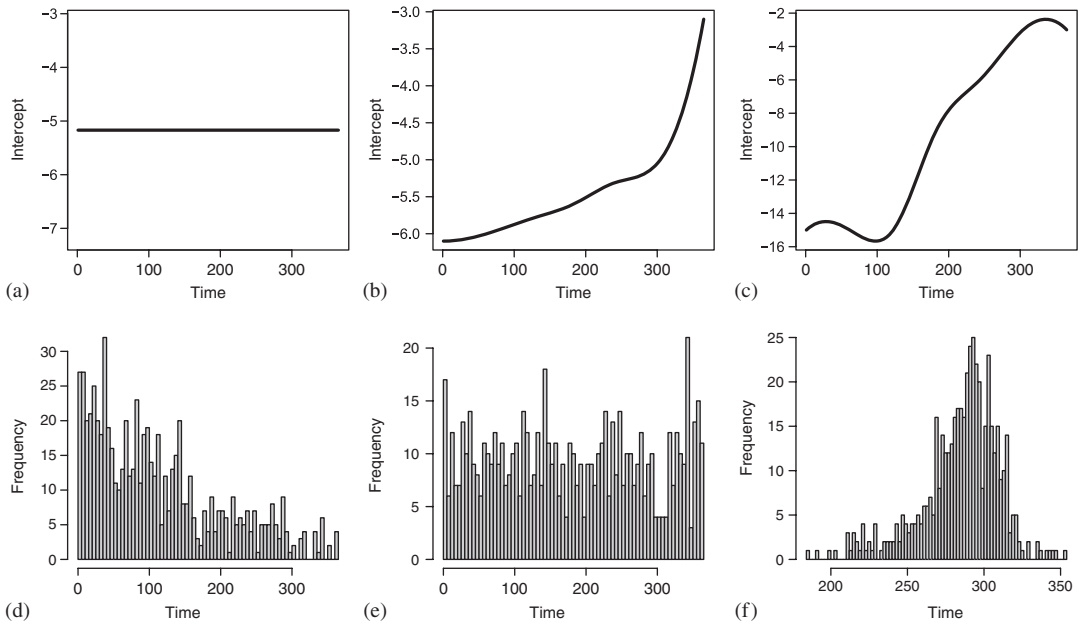

Figure 1. Binomial algorithm: time-dependent intercept and their corresponding distribution of event times before censoring: (a) constant intercept; (b) spline intercept; (c) spline intercept; (d) event time distribution corresponding to (a); (e) event time distribution corresponding to (b); and (f) event time distribution corresponding to (c).

Calibration for the class of binomial algorithms generated more work because both the intercept of the binomial model and the upper bound of the Uniform distribution of censoring had to be selected. In the case where the intercept was constant in time, we first set the upper bound of the censoring distribution to the end of follow-up and then selected the intercept by trial and error. In the case of the time-dependent intercept, we calibrated the spline function by trial and error and used the Kolmogorov–Smirnov test on 100 samples to assess whether the distribution obtained was similar to that desired. Finally, we applied censoring distributions with different upper bounds until the desired number of events were left uncensored. The bottom panels of Figure 1 show the distribution of event times generated with the binomial algorithm for a random sample, while the top panels show the corresponding intercepts.

Once the calibration had been achieved, the control over the number of events was very similar with all algorithms. For a sample size of 750 and a proportion of events of 66 per cent, all algorithms yielded on average 500 events with the sample-to-sample standard deviation of 12.4–13.4 events.

4.2. Accuracy and stability of parameter estimates

As expected, all the algorithms produced unbiased estimates for the coefficients of the fixed-in-time variables (data not shown). Therefore, in what follows, we focus on TDC.

4.2.1. Model 1. Descriptive statistics for the coefficients of the TDC for Scenario 1 are shown in the upper part of Table II. The four algorithms yielded virtually unbiased estimates, with relative bias

DOI: 10.1002/sim

Table II. Accuracy and stability of parameter estimates of TDC, Scenario 1.*

Algorithms	Mean $\hat{\beta}$	SD $\hat{\beta}$	Bias	Per cent bias	RMSE	Coverage
Model 1						
Results for $\beta_{\mathrm{BIN}(t)}$, true = 0.405						
Permutational	0.400	0.123	−0.006	1.48	0.122	0.956
Permutational with rejection sampling	0.405	0.136	<1e−04	0.00	0.134	0.947
Binomial	0.398	0.126	−0.008	1.97	0.126	0.953
Binomial with $\beta(t)$	0.407	0.132	0.002	0.49	0.130	0.943
Model 2						
Results for $\beta_{\mathrm{BIN}(t)}$, true = 0.405						
Permutational	0.389	0.161	−0.016	−3.95	0.161	0.954
Permutational with rejection sampling	0.398	0.169	−0.008	−1.97	0.170	0.949
Binomial	0.389	0.153	−0.017	−4.19	0.155	0.951
Binomial with $\beta(t)$	0.399	0.162	−0.007	−1.72	0.161	0.944
Results for $\beta_{\mathrm{CUM}(t)}$, true = 0.006						
Permutational	0.006	0.002	<1e−04	0.00	<1e−04	0.939
Permutational with rejection sampling	0.006	0.002	<1e−04	0.00	<1e−04	0.961
Binomial	0.006	0.003	<1e−04	0.00	<1e−04	0.953
Binomial with $\beta(t)$	0.006	0.002	<1e−04	0.00	<1e−04	0.935

*Sample size of 750, 1000 iterations.

not exceeding 2 per cent of the true parameter. The variability of the estimates was similar across algorithms, but PA with the rejection sampler showed a slightly larger variance. All algorithms resulted in satisfying coverage rates, very close to the nominal value of 95 per cent. Similar results were obtained with Scenario 2, in which BIN(t) had a protective effect ($\beta_{\mathrm{BIN}(t)} = -0.223$), except that the distributions of the estimated parameters across simulations were slightly more variable when the true parameter was closer to the null, with the RMSE increased by at most 0.03 (data not shown).

4.2.2. Model 2. For all the algorithms, the estimates of coefficients for the cumulative TDC (CUM(t)) in *Model 2* were unbiased and had small sample-to-sample variance (bottom of Table II). However, unlike *Model 1*, in *Model 2* the four algorithms produced data sets that lead to slightly underestimated effects of BIN(t). Still, the relative biases in the estimates of BIN(t) were always smaller than 5 per cent. Estimates of BIN(t) also had a greater variance than their counterparts in *Model 1*, due to the correlation between BIN(t) and CUM(t). In Scenario 2, we simulated data sets from *Model 2* with the coefficient of CUM(t) set to zero. In addition, as in Scenario 1, there was a slight underestimation of the coefficients for BIN(t) (data not shown).

4.3. Sensitivity analyses

In Scenario 1a, we generated data sets according to the specification of Scenario 1, except the percentage of censored observed times was increased from 33 to 50 per cent. This did not affect the accuracy of the estimates, as expected, but increased their variance slightly (change in RMSE of the estimates were at most 0.03, other data not shown). We also generated data sets of smaller size ($n = 100$, 250, 750) with 66 per cent of events. As expected, the variance of the estimates of

DOI: 10.1002/sim

the coefficients across simulations decreased with increasing sample sizes but estimates remained virtually unbiased.

In Scenario 1b, we specified a Weibull(13, 310) for the distribution of event times in the PAs and a different spline function for the intercept in the binomial algorithm (shown in panel (b) of Figure 1, with the resulting distribution shown in panel (e)). This did not markedly affect the accuracy of the estimates, and only slightly increased their variance, resulting in an increase of RMSE of at most 0.02 relative to Scenario 1 (data not shown).

4.4. Computational time

For each of the sample sizes, we computed the time required to generate and analyze 1000 data sets generated from *Models 1* and *2* of Scenario 1. The results are shown in panels (a) and (b)

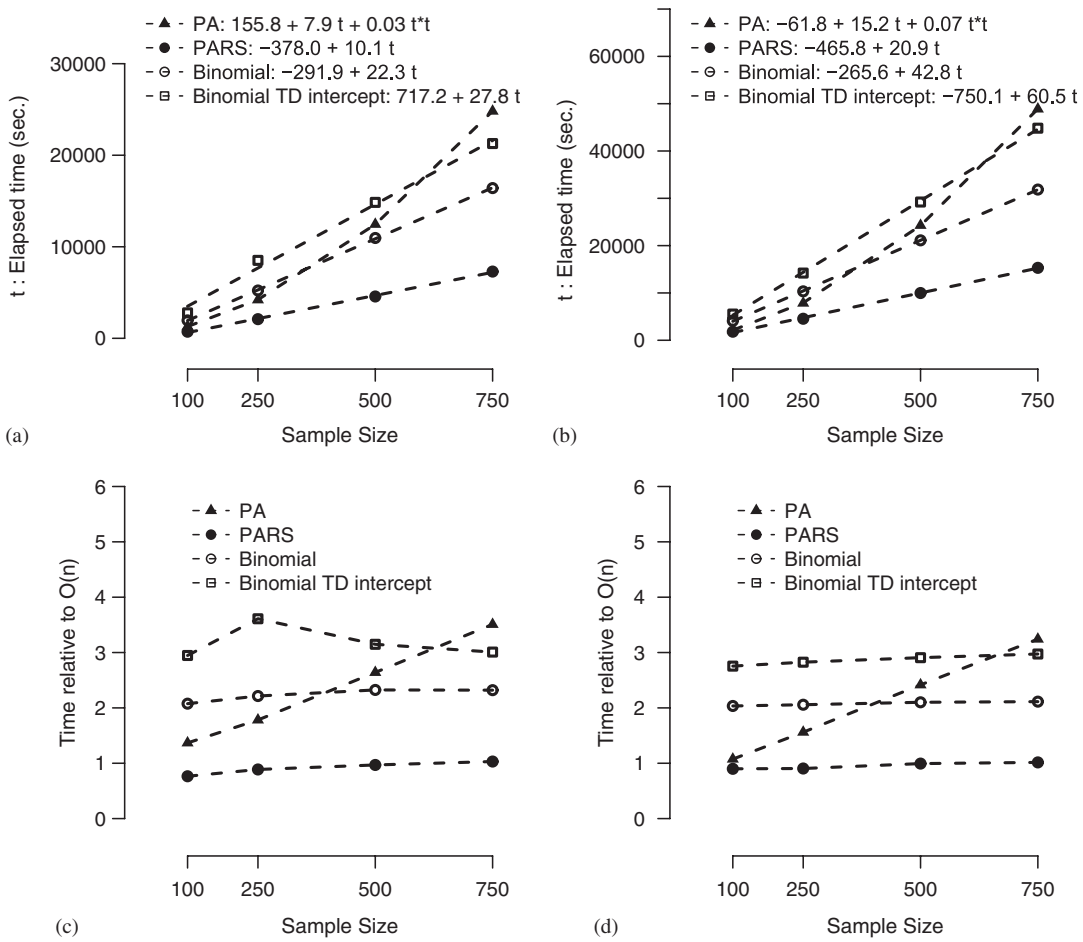

Figure 2. Mean computational times for *Model 1* and *2* for the four algorithms: (a) Model 1: absolute time; (b) Model 2: absolute time; (c) Model 1: time relative to $O(n)$; and (d) Model 2: time relative to $O(n)$.

DOI: 10.1002/sim

of Figure 2, where the points represent the estimated mean computational time. The dashed lines represent the best fit models of the association between computational time and sample size, fitted to these points. The computational times for the four algorithms differed significantly for samples as small as 250. Computational time for PARS, as well as the two binomial algorithms, grew linearly in time, with PARS having the slowest rate of increase and the smallest computational time. As expected, the computational time for PA was a quadratic function of the sample size. In panels (c) and (d) of Figure 2, the times required by the four algorithms are shown as their ratios to the time of a (hypothetical) strictly $O(n)$ algorithm, in which the time increases as a perfectly linear function of the sample size n, with the intercept of zero and the slope estimated by least squares from the data on PARS. Panels (c) and (d) of Figure 2 indicate that the performance of PARS is reasonably consistent with the $O(n)$ hypothesis as the corresponding times follow an approximately horizontal line.

PARS reduced computational time by at least 50 per cent in comparison with other algorithms for a sample size of 750, reflecting the smaller number of computations it had to perform. In fact, the distribution of the number of hazard ratios computed by PARS across 1000 data sets of size $n=750$ had a median of 1935.5 for *Model 1*, and an interquantile range (IQR) of (1657.8; 2364.5), while it was 5099.0 and (4398.8; 6239.3) for *Model 2*. The inclusion of a cumulative TDC in *Model 2* resulted in a larger number of hazard ratio calculations because in some risk sets, particularly those at the end of follow-up, the theoretical maximum hazard that determined the envelope function in the rejection sampler was notably larger than the empirical maximum. The theoritical maximum at t_i^* was calculated by assuming that the cumulative TDC was t_i^*, while in most simulated data set, the value of the cumulative TDC was smaller then t_i^* because the individuals did not start to be exposed at t_0, or because they interrupted their drug exposure. In contrast, the number of hazard ratio calculations for PA was meaningfully larger than that required by PARS, with a median of 177 768.5 hazard ratio calculations, IQR = (173 794.2; 181 564.0) for *Model 1*, and 177 620.0 IQR = (174 019.2; 181 522.2) for *Model 2*.

Using a time-dependent intercept in the binomial algorithm significantly increased the time required by this algorithm. While PA was faster than the two binomial algorithms for samples smaller than 500, it was the slowest algorithm for sample sizes of 750.

Computational times for *Model 2* are shown in panel (b) of Figure 2. Adding another TDC to the model (CUM(t)) approximately doubled the computational time required for each algorithm. However, it did not affect the relative comparisons of computational times between different sample sizes and different algorithms. The PA with a rejection sampler remained the fastest algorithm. Generating 1000 data sets including two TDC using PA with the rejection sampler was in fact faster than generating 1000 data sets containing only one TDC with any of the other algorithms.

5. DISCUSSION

We compared and validated two general classes of algorithms to generate event times conditional on TDC. The first class involves the PA, which matches observed times with independently generated vectors of covariate values, based on a probability law derived from the partial likelihood of Cox's PH model. We adapted the original PA developed by MacKenzie and Abrahamowicz [22] to include a rejection sampler, which accelerated the data generation by 50 per cent without compromising the accuracy or stability of the results.

DOI: 10.1002/sim

The second class of algorithms was based on a binomial model for the probability of developing an event at a given time. This probability was updated for every subject throughout the follow-up, based on current values of TDC. We implemented two versions of this algorithm, one with a fixed intercept in the binomial model, and another with a time-dependent intercept (spline function of time).

We validated the algorithms by comparing the estimates obtained by analyzing the data sets with the true parameters of the model used to generate them. The four algorithms considered generated virtually unbiased estimates of the coefficients of the TDC variables and had coverage rates very close to the nominal coverage. This was irrespective of the distribution of event times considered (Uniform or Weibull), the strength of the effect of the TDC on hazard (HR = 1.5 or HR = 0.8) and the censoring level (33 or 50 per cent).

However, the algorithms differed in their speed and their easiness of implementation. The PA with rejection sampling reduced computation significantly to the point that generating data sets with two TDC with PARS required less time than generating data sets including one TDC with any of the other algorithms. The increased efficiency of PARS, relative to the original PA [22], is a consequence of the smaller number of hazard ratio calculations required by PARS. While, to assign an event to a vector of covariate values, PA requires a systematic evaluation of the hazard ratio for all the members of the risk set, PARS only requires a fraction of these calculations. The additional efficiency of PARS depends on the average number of rejections that occur until a randomly sampled covariate vector is successfully matched with an event. As discussed in Section 2.2, this depends, in turn, on the envelope function used in the rejection sampler. We improved the efficiency of PARS by using an envelope function that was updated for each risk set. By doing so in our simulations, the median number of trials necessary to match a single event was about 3.3 IQR(3.9; 4.7) and 10.2 IQR(8.8; 12.5), for the scenarios with one binary TDC, and a binary and a cumulative TDC, respectively. This corresponded to a reduction of the number of hazard ratio calculations by, respectively, about 90 and 35 times, relative to the original PA.

Both our analytical and empirical results suggested that while the PA is of order $O(n^2)$, PARS and the class of binomial algorithm are of order $O(n)$. Further research will be necessary for a more complete and accurate assessment of the asymptotic order of alternative algorithms. Still, it may be challenging to establish such orders with high accuracy. For example, it will be difficult to quantify the impact of the initial trial-and-error calibration of the intercept for the binomial algorithm on its asymptotic order, especially in simulations with complex TDC and restricted assumptions regarding the marginal distribution of the event times. On the other hand, the exact asymptotic performance of the PARS depends on the effectiveness of the rejection sampler, which—in turn—may vary substantially depending on the assumptions regarding TDC. From this perspective, the results of our comparison of the computational times of alternative algorithms should be considered as preliminary only and further studies will be necessary to assess their generalizability.

Still, the important differences in the computational time required by different algorithms, especially as sample size increases, have practical consequences, especially when simulations are used to validate a computer-intensive method of analysis. However, caution must be exerted in the interpretation of absolute computational times used by different algorithms because they depend on the code, processor, and operating system used.

In addition, while the class of binomial algorithms may appear more intuitive, the class of PAs were easier to implement because they only required the calibration of the censoring parameter, while the binomial algorithms also required the calibration of the intercept of the model. Although the use of a time-dependent intercept in the binomial algorithm offers some control over the

DOI: 10.1002/sim

distribution of the generated event times, the choice of the time-dependent function for the intercept may be cumbersome because there is no direct correspondence between that function and the distribution of event times. Furthermore, the intercept in model used in the binomial algorithm may become very difficult to calibrate as more complicated setting of time-varying covariates are used. In contrast, the class of PAs offers full control over the distribution of the event times. In addition, in some applications, the user may want to exactly replicate the distribution of both (i) uncensored events and (ii) censoring times observed in a particular empirical study. In such cases, the class of PAs allow the user to match the covariate vectors directly with the observed empirical times, instead of generating these times at steps 1 and 2 of the algorithm. Whereas Cox's model estimates remain valid regardless of the underlying event time distributions [1], in many simulation studies it may be important to ensure that the generated distribution is clinically plausible. Moreover, the power and precision for the analysis of the effect of a TDC will clearly depend on how observed events are distributed over time relative to changes over time in the covariates. For example, if the focus is on long-term effects of cumulative exposure, then generating exponentially distributed event times may be largely sub-optimal as most events will occur before sufficient variation in cumulative exposure can be observed.

In conclusion, our finding that the inclusion of the rejection sampler increased the efficiency of the PA without affecting its accuracy will hopefully be useful in future simulation studies involving generation of event times conditional on complex TDC. Further refinements of the algorithm may include other sampling methods such as the so-called squeeze method [32].

ACKNOWLEDGEMENTS

We would like to thank Geneviève Lefebvre for her careful review of the article. M.-P. S. holds a Canadian Institutes for Health Research Doctoral Award. M. A. is a James McGill Professor at McGill University and holds grants from the National Sciences and Engineering Research Council of Canada and the Canadian Institutes for Health Research.

REFERENCES

1. Cox DR. Regression models and life tables (with discussion). *Journal of the Royal Statistical Society*, *Series B* 1972; **34**:187–220.
2. Sylvestre MP, Huszti E, Hanley JA. Do oscar winners live longer than less successful peers? A reanalysis of the evidence. *Annals of Internal Medicine* 2006; **145**(5):361–363.
3. Zhou Z, Rahme E, Abrahamowicz M, Pilote L. Survival bias associated with time-to-treatment initiation in drug effectiveness evaluation: a comparison of methods. *American Journal of Epidemiology* 2005; **162**(10):1016–1023.
4. Okin PM, Wachtell K, Devereux RB, Harris KE, Jern S *et al*. Regression of electrocardiographic left ventricular hypertrophy and decreased incidence of new-onset atrial fibrillation in patients with hypertension. *Journal of American Medical Association* 2006; **296**:1242–1248.
5. Finch M, Beiner M, Lubinski J, Lynch HT, Moller P, Rosen B, Murphy J *et al*. for the Hereditary Ovarian Cancer Clinical Study Group. Salpingo-oophorectomy and the risk of ovarian, fallopian tube, and peritoneal cancers in women with a BRCA1 or BRCA2 mutation. *Journal of American Medical Association* 2006; **296**:185–192.
6. Gogas H, Ioannovich J, Dafni U, Stavropoulou-Giokas C, Frangia K, Tsoutsos D, Panagiotou P, Polyzos A, Papadopoulos O, Stratigos A, Markopoulos C, Bafaloukos D, Pectasides D, Fountzilas G, Kirkwood JM. Prognostic significance of autoimmunity during treatment of melanoma with interferon. *The New England Journal of Medicine* 2006; **354**:709–718.
7. Houston TK, Person SD, Pletcher MJ, Liu K, Iribarren C, Kiefe CI. Active and passive smoking and development of glucose intolerance among young adults in a prospective cohort: CARDIA study. *British Medical Journal* 2006; **332**:1064–1069.

DOI: 10.1002/sim

8. Fisher LD, Lin DY. Time-dependent covariates in the Cox proportional hazards regression model. *Annual Review of Public Health* 1999; **20**:145–157.
9. Aydemir U, Aydemir S, Dirschedl P. Analysis of time-dependent covariates in failure time data. *Statistics in Medicine* 1999; **18**(16):2123–2134.
10. Abrahamowicz M, Bartlett G, Tamblyn R, du Berger R. Modeling cumulative dose and exposure duration provided insights regarding the associations between benzodiazepines and injuries. *Journal of Clinical Epidemiology* 2006; **59**(4):393–403.
11. de Bruijne MHJ, le Cessie S, Kluin-Nelemans HC, van Houwelingen HC. On the use of Cox regression in the presence of an irregularly observed time-dependent covariate. *Statistics in Medicine* 2001; **20**(24):3817–3829.
12. Andersen PK, Liestøl K. Attenuation caused by infrequently updated covariates in survival analysis. *Biostatistics* 2003; **4**(4):633–649.
13. Heinzl H, Kaider A. Gaining more flexibility in Cox proportional hazards regression models with cubic spline functions. *Computer Methods and Programs in Biomedicine* 1997; **54**(3):201–208.
14. Kooperberg C, Clarkson DB. Hazard regression with interval-censored data. *Biometrics* 1997; **53**(4):1485–1494.
15. Abrahamowicz M, MacKenzie TA. Joint estimation of time-dependent and non-linear effects of continuous covariates on survival. *Statistics in Medicine* 2007; **26**(2):392–408.
16. Giorgi R, Gouvernet J. Analysis of time-dependent covariates in a regressive relative survival model. *Statistics in Medicine* 2005; **24**(24):3863–3870.
17. Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models. *Statistics in Medicine* 2005; **24**(11):1713–1723.
18. Abrahamowicz M, MacKenzie T, Esdaile JM. Time-dependent hazard ratio: modelling and hypothesis testing with application in lupus nephritis. *Journal of the American Statistical Association* 1996; **91**:1432–1439.
19. Hess KR, Serachitopol DM, Brown BW. Hazard function estimators: a simulation study. *Statistics in Medicine* 1999; **18**(22):3075–3088.
20. Burton A, Altman DG, Royston P, Holder RL. The design of simulation studies in medical statistics. *Statistics in Medicine* 2006; **25**(24):4279–4292.
21. Bender R, Augustin T, Blettner M. Generating survival times to simulate Cox proportional hazards models (Letter to the Editor). *Statistics in Medicine* 2006; **25**(11):1778–1779.
22. MacKenzie T, Abrahamowicz M. Marginal and hazard ratio specific random data generation: applications to semi-parametric bootstrapping. *Statistics and Computing* 2002; **12**(3):245–252.
23. Cox DR. Partial likelihood. *Biometrika* 1975; **62**:269–276.
24. Leffondré K, Abrahamowicz M, Siemiatycki J. Evaluation of Cox's model and logistic regression for matched case–control data with time-dependent covariates: a simulation study. *Statistics in Medicine* 2003; **22**(24): 3781–3794.
25. Fishman GS. *A First Course in Monte Carlo* (1st edn). Thompson Learning: Belmont, CA, U.S.A., 2006.
26. Murray I. Note on rejection sampling and exact sampling with the metropolised independence sampler. Available from: http://www.gatsby.ucl.ac.uk/∼iam23/pub/04rejection_cftp/rejection_cftp.pdf. (Accessed on 19 January 2007).
27. von Neumann J. Various techniques in connection with random digits. In *Design of Computer, Theory of Automata and Numerical Analysis*, Taub AH (ed.), vol. V. Pergamon Press: New York, 1963; 768–770.
28. Abadie A. Bootstrap tests for distributional treatment effects in instrumental variable models. *Journal of the American Statistical Association* 2002; **97**(457):284–292.
29. Le Teuff G, Abrahamowicz M, Bolard P, Quantin C. Comparison of Cox's and relative survival models when estimating the effects of prognostic factors on disease-specific mortality: a simulation study under proportional excess hazards. *Statistics in Medicine* 2005; **24**(24):3887–3909.
30. R Development Core Team. *R*: *A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing: Vienna, 2006.
31. Matsumoto M, Nishimura T. Mersenne Twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation* 1998; **8**:3–30.
32. Marsaglia G. The squeeze method for generating gamma variates. *Computers and Mathematics with Applications* 1977; **3**:321–325.

DOI: 10.1002/sim