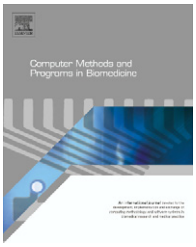

journal homepage: www.intl.elsevierhealth.com/journals/cmpb

SAS and R functions to compute pseudo-values for censored data regression

John P. Klein[a,*], *Mette Gerster*[b], *Per Kragh Andersen*[b], *Sergey Tarima*[a], *Maja Pohar Perme*[b,c]

[a] *Division of Biostatistics, Department of Population Health, Medical College of Wisconsin, 8701 Watertown Plank Road, Milwaukee, WI 53226, USA*

[b] *Department of Biostatistics, University of Copenhagen, Ø. Farimagsgade 5, PB 2099, DK 1014 Copenhagen K, Denmark*

[c] *Department of Biomedical Informatics, University of Ljubljana, Vrazov trg 2, SI-1000, Ljubljana, Slovenia*

ARTICLE INFO

Article history:
Received 13 July 2007
Received in revised form
27 November 2007
Accepted 27 November 2007

Keywords:
Cumulative incidence
GEE
Kaplan–Meier curves
Pseudo-values
Restricted mean survival

ABSTRACT

Recently, in a series of papers, a method based on pseudo-values has been proposed for direct regression modeling of the survival function, the restricted mean and cumulative incidence function with right censored data. The models, once the pseudo-values have been computed, can be fit using standard generalized estimating equation software. Here we present SAS macros and R functions to compute these pseudo-values. We illustrate the use of these routines and show how to obtain regression estimates for a study of bone marrow transplant patients.

1. Introduction

In many applications investigators are interested in regression modeling of covariates on a survival outcome. The outcome may be the time to some event or the time until a competing risk event has occurred. Most applications use a Cox regression [1] model for the data. This approach models the hazard rate of the time to an event or in the case of competing risk data the crude hazard rate of the event in the presence of all the other risks [2]. Statistical procedures for the Cox model are available in most statistical packages [3].

Recently [4–8], we have developed a flexible technique to directly model survival quantities based on right censored data. The technique allows direct regression modeling of the survival function [9], the restricted mean survival time [5] and the cumulative incidence function for competing risks data [4,6–8]. The approach uses the pseudo-values based on the difference between the complete sample and the leave-one-out estimators of relevant survival quantities. These pseudo-values are used in a generalized estimating equation (GEE) to model the effects of covariates on the outcome of interest.

To apply the methodology one needs to compute the pseudo-values for each observation. This needs to be performed only once. Once the pseudo-values are obtained they can be used in a standard GEE program to obtain regression estimates.

In this report we present three SAS macros and three R functions to compute the pseudo-values for right censored data. The SAS macro and R function "pseudosurv" compute pseudo-values for modeling the survival function based on

[*] *Corresponding author*. Tel.: +1 414 456 8379.
E-mail address: klein@mcw.edu (J.P. Klein).

0169-2607/$ – see front matter © 2007 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2007.11.017

the Kaplan–Meier estimator. The SAS macro and R function “pseudomean” provide pseudo-values for the restricted mean survival time. The SAS macro and R function “pseudoci” provide pseudo-values for the cumulative incidence function for competing risks data.

In Section 2 we present a summary of the statistical background for these regression models. In Section 3 we present our functions and macros. In Section 4 we present an example of the macros and functions. Section 5 concludes with some closing remarks.

2. Methods

2.1. *The general approach*

In this section we present a general approach to censored data regression based on pseudo-values [4]. This approach has been applied to regression models for the cumulative incidence functions in competing risks [4,6–8]; for state occupation probabilities in general multi-state models [4,8]; for the restricted mean [5] and to the survival function [9]. In its most general form let $X_1, \ldots, X_n$ be independent and identically distributed. The X_i’s may be random variables, vectors or processes. Let $\theta = E[f(X_i)]$ for some $f()$ which may be multivariate. Let $\hat{\theta}$ be an unbiased (or approximately unbiased) estimator of θ.

Now suppose we have covariates $\mathbf{Z}_1, \ldots, \mathbf{Z}_n$ which are an iid sample and define the conditional expectation of $f(X_i)$ given $\mathbf{Z}_i$ by

$$\theta_i = E[f(X_i)|\mathbf{Z}_i]$$

The ith pseudo-observation is defined as

$$\hat{\theta}_i = n \cdot \hat{\theta} - (n-1)\widehat{\theta^{-i}}$$

where $\widehat{\theta^{-i}}$ is the “leave-one-out” estimator for θ based on X_j, $j \neq i$.

The regression model for θ corresponds to a specification of the relationship between θ_i and $\mathbf{Z}_i$ which is provided by a generalized linear model:

$$g(\theta_i) = \beta^{\mathrm{T}}\mathbf{Z}_i \tag{1}$$

with $g(.)$ a link function. Typically we add a column $\mathbf{Z}_{i0}$ to $\mathbf{Z}_i$ to allow for an intercept β_0. When $\theta = (\theta(\tau_1), \ldots, \theta(\tau_M))$ we add to $\mathbf{Z}_i$ indicators of the time points, τ_j, $j = 1, \ldots, M$ to allow for different intercepts at each time. Deterministic time dependent covariates measured at each of the τ_j’s are also possible. Estimates of the β’s are based on the unbiased estimating equations:

$$\sum_i \left(\frac{\partial}{\partial\beta} g^{-1}(\beta^{\mathrm{T}}\mathbf{Z}_i)\right)^{\mathrm{T}} \mathbf{V}_i^{-1}(\hat{\theta}_i - g^{-1}(\beta^{\mathrm{T}}\mathbf{Z}_i)) = \sum_i U_i(\beta) = U(\beta) = 0 \tag{2}$$

Here $\mathbf{V}_i$ is a working covariance matrix. A sandwich estimator is used to estimate the variance of $\hat{\beta}$ let

$$\begin{aligned} I(\hat{\beta}) &= \sum_i \left(\frac{\partial g^{-1}(\hat{\beta}^{\mathrm{T}}\mathbf{Z}_i)}{\partial\beta}\right)^{\mathrm{T}} \mathbf{V}_i^{-1} \left(\frac{\partial g^{-1}(\hat{\beta}^{\mathrm{T}}\mathbf{Z}_i)}{\partial\beta}\right), \quad \text{and} \\ \widehat{\mathrm{Var}}\,(U(\hat{\beta})) &= \sum_i U_i(\hat{\beta})^{\mathrm{T}} U_i(\hat{\beta}), \quad \text{then} \\ \widehat{\mathrm{Var}}\,(\hat{\beta}) &\approx I(\hat{\beta})^{-1}\widehat{\mathrm{Var}}\,(U(\hat{\beta}))I(\hat{\beta})^{-1} \end{aligned} \tag{3}$$

The estimators of β can be shown to be asymptotically normal by results of Liang and Zeger [10]. One can show that the sandwich estimator converges in probability to the true variance.

Once the pseudo-values have been computed estimators of β can be obtained by using standard software for Generalized Estimating Equations (GEE) such as PROC GENMOD in SAS or the function “geese” in R. In the next sections we present R and SAS routines to compute the pseudo-values in the three situations.

2.2. *Pseudo-values for the survival function*

First, we present a routine for use when the event of interest is the survival function. Here we need pseudo-values for $S(\tau_j) = P[T > \tau_j]$ at a grid of time points $\tau_1 < \cdots < \tau_M$. That is, $\theta = (\theta_1, \ldots, \theta_M)$ where $\theta_j = S(\tau_j)$ is estimated using the Kaplan–Meier estimator $\hat{S}(\tau_j)$ as explained below. When $M = 1$ this allows for a regression model for the survival (or failure) probability at a single point in time. When $M > 1$ then inference is to an entire survival curve. Our experience [5,6] suggests that five to ten time points equally spaced on the event scale works well in most cases when fitting models for the entire curve. For this parameter the pseudo-values are based on the Kaplan–Meier estimator [11], $\hat{S}(\bullet)$, defined by

$$\hat{S}(t) = \prod_{t_j \le t} \frac{Y_j - d_j}{Y_j} \tag{4}$$

where $t_1 < \cdots < t_D$ are the distinct event times, Y_j the number at risk and d_j the number of events at time t_j. Note that when there is no censoring the pseudo-value at τ reduces to the indicator that the observation is greater than τ.

2.3. *Pseudo-values for the restricted mean survival time*

The second set of routines computes pseudo-values for the restricted mean lifetime [5]. Note that for survival data the mean time to event is the area under the survival curve:

$$\mu = \int_0^\infty S(u)\,du \tag{5}$$

For right censored data, and in particular when the largest on study time is censored, the estimated survival curve does not drop to zero and the estimator of μ obtained by plugging the Kaplan–Meier estimator into (5) does not work well. An alternative to μ is, for $\tau > 0$, the restricted mean defined as the area under the survival curve up to time τ [2]. This quantity is equal to $\theta = \mu_\tau = E[\min(T, \tau)]$ and is estimated by the area under the Kaplan–Meier curve up to time τ. That is

$$\hat{\theta} = \hat{\mu}_\tau = \int_0^\tau \hat{S}(u)\,du$$

2.4. *Competing risks pseudo-values using the cumulative incidence function*

The final set of functions deal with regression models for the cumulative incidence function [6–8], $C_k(t)$, $k = 1, 2$. For two competing risks with crude hazard rates, $h_1(t)$ and $h_2(t)$ the cumulative incidence function is given by

$$C_k(t) = \int_0^t h_k(u) \exp\left\{ -\int_0^u [h_1(v) + h_2(v)]\,dv \right\} du, \quad k = 1, 2$$

For a fixed set of time points $\tau_1, \ldots, \tau_M$ we have for cause k that the parameter is $\theta = (\theta_1, \ldots, \theta_M) = (C_k(\tau_1), \ldots, C_k(\tau_M))$.

If $t_1 < \cdots < t_D$ are the distinct times where one of the events occurs, Y_j the number at risk, d_{1j} (d_{2j}) the number of type 1 (type 2) events at time t_j then the estimate of the cumulative incidence is

$$\hat{C}_k(t) = \sum_{t_j \le t} \left[\frac{d_{jk}}{Y_j}\right] \prod_{t_i < t_j} \left[\frac{Y_i - (d_{1i} + d_{2i})}{Y_i}\right], \quad k = 1, 2$$

That is $\hat{\theta}_j = \hat{C}_k(\tau_j)$.

3. The functions

3.1. *SAS macros*

The SAS macro for the survival function is pseudosurv (indata, time, event, howmany, datatau, outdata). The arguments are

Indata. An input data set.
Time. The name of the variable which contains the on study times.
Event. The death or event indicator (1 event, 0 censored).
Howmany. The sample size (n).
Datatau. A SAS data set with the single variable tau which is the M time points at which the pseudo-values are to be computed.
Outdata. The name of the SAS data set that will contain the pseudo-values.

The macro uses PROC LIFETEST to compute the Kaplan–Meier estimators at the time points in the data set datatau. The output data set consists of M new lines for each observation each of which includes the original data and two new variables: pseudo which contains the pseudo-value for this observation and tpseudo which contains the time point at which the pseudo-value was computed.

To find pseudo-values for the restricted mean we have the macro pseudomean. The arguments of the macro are the same as above, with the exception that the data set datatau is replaced by the maximum cut-off point τ for the restricted mean. The value of τ needs to be an interior point of the data. The macro is again based on PROC LIFETEST in SAS.

To find pseudo-values for the cumulative incidence functions the SAS macro 'pseudoci' is used. This macro makes use of a macro 'cuminc' to compute the cumulative incidence function. The arguments of the SAS macro cuminc are

datain. The input data set.
x. The event time variable.
re. The indicator of the first competing risk (1, occurred; 0, otherwise).
de. The indicator of the second competing risk (1, occurred; 0, otherwise).
dataout. The name of an output data set.
cir, *cid*. Variable names for the cumulative incidence function of the first and second competing risks, respectively.

The macro uses PROC PHREG to obtain the crude hazard rates, $h_k(t)$, by fitting two Cox models, one for each competing risk, with a single covariate defined to be zero for all cases. The output statement yields the cumulative crude hazard rate which is converted to the hazard rate at the event times. These are combined in a data step to yield the cumulative incidence functions.

The cumulative incidence macro is called in the macro pseudoci (datain, x, re, de, howmany, datatau, dataout) which computes the pseudo-values at the time points in the data set datatau. An expanded data set, dataout, includes all the data in the dataset datain and for each tau in datatau an entry for each observation with the variables rpseudo, dpseudo, the pseudo-values for risks one and two, respectively, and tpseudo, the time point at which each pseudo-value is computed.

All the SAS macros are available at our website at http://www.biostat.mcw.edu/software/SoftMenu.html.

3.2. *The R functions*

The R function for computing the pseudo-values for the survival function is the object pseudosurv which has arguments

time. Event time variable.
event. The event indicator (1 event, 0 censored).
tmax. A vector with the time points at which the pseudo-values are to be computed.

The function returns a new object with the original time and censoring variables and new variables containing the pseudo-values. Here for M time points in tmax an additional M columns are appended to the time and censoring matrix. Since no sorting of the data occurs in the function this can be appended to the original data to obtain an augmented file with the pseudo-values.

To find pseudo-values for the restricted mean we have the R function pseudomean. The arguments of this function are the same as above, the vector tmax now represents the maximum cut-off point τ for the restricted mean. Again, this value of τ needs to be an interior point of the data.

The R function "pseudoci" has three arguments: time (the event time variable), event (1 if occurrence of risk 1, 2 if occurrence of risk 2 and 0 otherwise), and tmax (a list of time points at which the pseudo-values are to be computed). The routine produces an object containing the pseudo-values for both competing risks. The output object consists of columns for the time and status variable and the pseudo-values, alternating between the two competing risks.

The R functions are available from the package “pseudo” which can be found on the CRAN site [13].

4. Example

To illustrate the macros and functions we use a data set on HLA matched sibling donor bone marrow transplants [12]. This data set, which consists of data on 137 transplant patients, can be found at http://www.biostat.mcw.edu/homepgs/klein/bmt.html and is available in the R package “KMsurv” with data(bmt).

An abbreviated data set constructed from these data consists of the time to death, relapse or lost to follow-up (tdfs), the indicators of relapse and death (relapse, trm), the indicator of treatment failure (dfs = relapse + trm), an id number from 1 to 137 (id) and three factors that may be related to outcome: disease (1-Acute Lymphocytic Leukemia (ALL), 2-Low risk Acute Myeloid Leukemia (AML) and 3-High risk AML), the French-American-British Disease grade for AML (fab = 1 if AML and Grade 4 or 5, 0 otherwise), and recipient age at transplant (age).

4.1. *The survival function*

We first will examine regression models for disease free survival based on the Kaplan–Meier estimator. We will use the SAS macro ‘pseudosurv’ to compute the pseudo-values. In this example we compute pseudo-values at 5 data points roughly equally spaced on the event scale: 50, 105, 170, 280 and 530 days. We assume that the macro is in a file ‘sasmac’ in the current directory. The SAS code to compute the pseudo-values and put them into a permanent SAS data set ‘pseudoval’ is as follows.

```
data one;
input tdfs trm relapse dfs id disease fab age;
lines;
2081 0 0 0 1 1 0 26
1602 0 0 0 2 1 0 21
        . . .
113 0 1 1 136 3 0 31
363 1 0 1 137 3 0 52
;
libname out ‘ ’;
%include ‘sasmac’;
data times;
input tau;
lines;
50
105
170
280
530
;
run;
%pseudosurv(one,tdfs,dfs,137,times,in.pseudoval)
proc print;
```

The data set in pseudoval contains the following.

```
Obs  tdfs  trm  rel  dfs   id  disease  fab  rage  pseudo   tpseudo
1       1    1    0    1   35     1       0    42    0          50
2       2    1    0    1  108     3       1    20    0          50
                                       . . .
465   276    1    0    1   17     1       0    18   -0.01036 280
466   288    1    0    1   75     2       0    45    1.00087 280
                                       . . .
684  2569    0    0    0   39     2       1    19    1.00325 530
685  2640    0    0    0   93     3       0    18    1.00325 530
```

To compute regression estimates we use PROC GENMOD. The code to fit a model using the complementary log–log link $\ln[-\ln\{S(t)\}] = \beta \mathbf{Z})$ is as follows:

```
proc genmod;
class id
      disease (param=ref ref=first)
      tpseudo (param=ref ref=first);
FWDLINK LINK=LOG(-LOG(_MEAN_));
INVLINK ILINK=EXP(-EXP(_XBETA_));
model pseudo= tpseudo disease fab age/dist=normal noscale;
repeated subject=id/corr=ind;
```

The output is

```
                     Analysis Of GEE Parameter Estimates
                     Empirical Standard Error Estimates
                                  Standard   95% Confidence
           Parameter        Estimate  Error        Limits        Pr > |Z|
           Intercept        -2.9816   0.5549  -4.0691   -1.8941  <.0001
           tpseudo   105     1.1143   0.3043   0.5178    1.7108  0.0003
           tpseudo   170     1.6262   0.3289   0.9815    2.2708  <.0001
           tpseudo   280     2.0043   0.3393   1.3392    2.6694  <.0001
           tpseudo   530     2.4953   0.3488   1.8117    3.1789  <.0001
           disease   2      -1.1955   0.4124  -2.0038   -0.3873  0.0037
           disease   3       0.0036   0.3766  -0.7345    0.7418  0.9923
           fab               0.7620   0.3360   0.1035    1.4204  0.0223
           age               0.0138   0.0138  -0.0139    0.0400  0.3419
```

Note, we used a user supplied link function. The built-in complementary link function fits the model $\ln[-\ln\{1 - S(t)\}]$.

For our model the estimated survival function for a patient at time t with a set of covariates $\mathbf{Z}$ is $S(t|\mathbf{Z}) = \exp(-\Lambda_0(t)e^{\beta\mathbf{Z}})$, where we have $\Lambda_0(50) = \exp(-2.9816) = 0.051$; $\Lambda_0(105) = \exp(-2.9816 + 1.1143) = 0.155$; $\Lambda_0(170) = \exp(-2.9816 + 1.6262) = 0.258$; $\Lambda_0(280) = \exp(-2.9816 + 2.0043) = 0.376$; $\Lambda_0(530) = \exp(-2.9816 + 2.4953) = 0.615$. The model shows that patients with AML low risk have better disease free survival than ALL patients (Relative Risk, RR = exp (−1.1955) = 0.30) and that AML patients with grade 4 or 5 FAB have a lower disease free survival (RR = exp (0.7620) = 2.14).

Without re-computing the pseudo-values we could examine the effect of FAB over time. We need to create in the data set a FAB indicator at each of the time points and rerun PROC GENMOD. The code is

```
data timedep;
set in.pseudoval;
if tpseudo=50  then fab1=fab;  else fab1=0;
if tpseudo=105 then fab2=fab;  else fab2=0;
if tpseudo=170 then fab3=fab;  else fab3=0;
if tpseudo=280 then fab4=fab;  else fab4=0;
if tpseudo=530 then fab5=fab;  else fab5=0;

proc genmod;
class id
      disease(param=ref ref=first)
      tpseudo (param=ref ref=first);
FWDLINK LINK=LOG(-LOG(_MEAN_));
INVLINK ILINK=EXP(-EXP(_XBETA_));
model pseudo= tpseudo disease fab1 fab2 fab3 fab4 fab5
repeated subject=id/corr=ind ;
contrast 'fab'
fab1 1 fab2 0 fab3 0 fab4 0 fab5 0,
fab1 0 fab2 1 fab3 0 fab4 0 fab5 0,
fab1 0 fab2 0 fab3 1 fab4 0 fab5 0,
fab1 0 fab2 0 fab3 0 fab4 1 fab5 0,
fab1 0 fab2 0 fab3 0 fab4 0 fab5 1/wald;
contrast 'fab by time'
fab1 1 fab2 -1 fab3 0 fab4 0 fab5 0,
fab1 0 fab2 1 fab3 -1 fab4 0 fab5 0,
fab1 0 fab2 0 fab3 1 fab4 -1 fab5 0,
fab1 0 fab2 0 fab3 0 fab4 1 fab5 -1/wald;
```

Here the two contrast statements test for an overall FAB effect and if the FAB effect changes with time, respectively. The relevant output is

```
                  Analysis Of GEE Parameter Estimates
                   Empirical Standard Error Estimates
                            Standard   95% Confidence
Parameter      Estimate  Error         Limits      Pr > |Z|
Intercept      -3.1184    0.5640 -4.2237  -2.0130   <.0001
tpseudo   105   1.3889    0.4937  0.4213   2.3566   <.0001
tpseudo   170   2.1149    0.5378  1.0609   3.1689   <.0001
tpseudo   280   2.4737    0.5483  1.3990   3.5484   <.0001
tpseudo   530   2.9565    0.5583  0.8622   4.0508   <.0001
disease   2    -1.0920    0.3858 -1.8482  -0.3359   0.0046
disease   3     0.1239    0.3558 -0.5734   0.8213   0.7276
fab1            1.4045    0.7509 -0.0673   2.8762   0.0614
fab2            0.9907    0.4588  0.0914   1.8901   0.0308
fab3            0.6297    0.3771 -0.1096   1.3691   0.0950
fab4            0.6428    0.3426 -0.0287   1.3143   0.0606
fab5            0.6319    0.3282 -0.0114   1.2752   0.0542
                        Contrast Results for GEE Analysis
                                 Chi-
    Contrast           DF     Square    Pr > ChiSq    Type
    fab                 5       6.26        0.2815    Wald
    fab by time         4       2.10        0.7179    Wald
```

This model shows that there is no difference in the FAB effect over time ($p = 0.7179$).

Now we implement the same operations with R. The data are available from the KMsurv package [13], we rename variables to match the SAS example.

```
library(KMsurv)
data(bmt)
names(bmt)[c(1,3:6,13,20)] <- c("disease", "tdfs", "relapse", "trm",
                                "dfs", "age","fab")
bmt$disease <- as.factor(bmt$disease)
```

We define the required time points and generate pseudo-values.

```
cutoffs <- c(50,105,170,280,530)
pseudo <- pseudosurv(bmt$tdfs, bmt$dfs, tmax=cutoffs)
```

The “pseudo” object is as follows.

```
> pseudo[order(pseudo$time),]
    time event tmax =50 tmax =105 tmax =170 tmax =280 tmax =530
35     1     1        0         0         0    0.0000    0.0000
108    2     1        0         0         0    0.0000    0.0000
                     .    .    .
17   276     1        1         1         1   -0.0104   -0.0080
75   288     1        1         1         1    1.0009   -0.0080
                     .    .    .

39  2569     0        1         1         1    1.0009    1.0032
93  2640     0        1         1         1    1.0009    1.0032
```

The second step requires some data manipulation to prepare for the GEE step. The last line allows us to fit the complimentay log–log model to ipseudo since by default “geese” fits a model with link function $\log[-\log\{1-y\}]$.

```
b <- NULL
for(j in 3:ncol(pseudo))
b <- rbind(b,cbind(bmt,pseudo=pseudo[,j],
     tpseudo=cutoffs[j-2],id=1:nrow(bmt)))
b <- b[order(b$id),]
library(geepack)
b$tpseudo <- as.factor(b$tpseudo)
b$ipseudo <- 1-b$pseudo
```

The analysis is completed with GEE regression using the object “geese” in the package GEEPACK [13] on ipseudo. The “geese” function does not allow changing the definition of the link as we did in the SAS example, however, by modeling “ipseudo” instead of the “pseudo” variable we avoid this problem and can use the complementary log–log definition implemented in the “geese” function.

```
summary(fit <- geese(ipseudo ~ tpseudo + disease + fab + age, data = b,
        id=id, scale.fix=TRUE, family=gaussian, mean.link = "cloglog",
        corstr="independence"))
```

generating

```
              estimate  san.se     wald         p
(Intercept) -2.9816   0.5681   27.5437    <0.0001
tpseudo105   1.1143   0.3076   13.1225     0.0003
tpseudo170   1.6262   0.3325   23.9152    <0.0001
tpseudo280   2.0043   0.3431   34.1191    <0.0001
tpseudo530   2.4953   0.3522   50.1924    <0.0001
disease2    -1.1955   0.4125   8.3981      0.0038
disease3     0.0036   0.3801   0.0001      0.9924
fab          0.7620   0.3395   5.0368      0.0248
age          0.0131   0.0143   0.8398      0.3595
```

The parameter estimates are the same as those obtained using PROC GENMOD in SAS, however variance estimates are a bit different. In SAS the variance is estimated by a “sandwich” estimator $\widehat{\text{Var}}(\hat{\beta})$ presented in equation (3). By default, the “geese” function in R uses a different “sandwich” estimator of the variance proposed in Ref. [13]. An alternative to the “sandwich” estimator is the jackknife variance estimators [14]. The routine “geese” allows the user to decide between the fully iterated jackknife, the one-step jackknife, and approximate jackknife (AJ) variance estimates. We suggest using the AJ variance estimate. The code and results using that estimator are as follows:

```
fit <- geese(ipseudo~tpseudo+disease+ fab +age,
       data=b,scale.fix=TRUE,family=gaussian,jack=TRUE,
       mean.link="cloglog",corstr="independence")
round(cbind(mean = fit$beta,SD = sqrt(diag(fit$vbeta.ajs)),
      Z = fit$beta/sqrt(diag(fit$vbeta.ajs)),
      PVal = 2-2*pnorm(abs(fit$beta/sqrt(diag(fit$vbeta.ajs))))),4)

               mean     SD       Z   PVal
(Intercept) -2.9816 0.5561 -5.3615 0.0000
tpseudo105   1.1143 0.2988  3.7290 0.0002
tpseudo170   1.6262 0.3225  5.0427 0.0000
tpseudo280   2.0043 0.3325  6.0282 0.0000
tpseudo530   2.4953 0.3415  7.3079 0.0000
disease2    -1.1955 0.4105 -2.9121 0.0036
disease3     0.0036 0.3808  0.0095 0.9924
fab          0.7620 0.3391  2.2467 0.0247
age          0.0131 0.0141  0.9259 0.3545
```

To examine the effect of FAB over time we create four new variables

```
b$fab50 <- 0;  b$fab50[b$tpseudo==50]   <- b$fab[b$tpseudo==50];
b$fab105 <- 0; b$fab105[b$tpseudo==105] <- b$fab[b$tpseudo==105];
b$fab170 <- 0; b$fab170[b$tpseudo==170] <- b$fab[b$tpseudo==170];
b$fab280 <- 0; b$fab280[b$tpseudo==280] <- b$fab[b$tpseudo==280];
b$fab530 <- 0; b$fab530[b$tpseudo==530] <- b$fab[b$tpseudo==530];
```

and use them in the GEE regression model

```
fit <- geese(ipseudo ~ tpseudo + disease + fab50 + fab105 + fab170+
      fab280 + fab530, data = b,id=id, jack=TRUE, scale.fix=TRUE,
      family=gaussian, mean.link = "cloglog", corstr="independence")
round(cbind(mean = fit$beta,SD = sqrt(diag(fit$vbeta.ajs)),
      Z = fit$beta/sqrt(diag(fit$vbeta.ajs)), PVal =
      2-2*pnorm(abs(fit$beta/sqrt(diag(fit$vbeta.ajs))))),4)
```

The results are

```
                 mean     SD       Z   PVal
(Intercept) -3.1184 0.5461 -5.7100 0.0000
tpseudo105   1.3889 0.4774  2.9093 0.0036
tpseudo170   2.1149 0.5201  4.0666 0.0000
tpseudo280   2.4736 0.5303  4.6649 0.0000
tpseudo530   2.9565 0.5399  5.4756 0.0000
disease2    -1.0920 0.3768 -2.8983 0.0038
disease3     0.1239 0.3515  0.3526 0.7244
fab50        1.4045 0.7335  1.9148 0.0555
fab105       0.9907 0.4509  2.1974 0.0280
fab170       0.6297 0.3722  1.6917 0.0907
fab280       0.6428 0.3382  1.9008 0.0573
fab530       0.6319 0.3229  1.9566 0.0504
```

To test the overall FAB effect we use the following R code.

```
C <- rbind( c(0,0,0,0,0,0,0,1,0,0,0,0),
            c(0,0,0,0,0,0,0,0,1,0,0,0),
            c(0,0,0,0,0,0,0,0,0,1,0,0),
            c(0,0,0,0,0,0,0,0,0,0,1,0),
            c(0,0,0,0,0,0,0,0,0,0,0,1))
SSH0 <- t(C %*% fit$beta)
            %*% solve(C %*% fit$vbeta.ajs %*% t(C))
            %*% (C %*% fit$beta)
1-pchisq(SSH0,nrow(C))
          [,1]
[1,] 0.2624323
```

To test if the FAB effect differs with time we use the following R code.

```
C <- rbind(c(0,0,0,0,0,0,0,1,-1, 0, 0, 0),
           c(0,0,0,0,0,0,0,0, 1,-1, 0, 0),
           c(0,0,0,0,0,0,0,0, 0, 1,-1, 0),
           c(0,0,0,0,0,0,0,0, 0, 0, 1,-1))
SSH0 <- t(C %*% fit$beta)
        %*% solve(C %*% fit$vbeta.ajs
        %*% t(C)) %*% (C %*% fit$beta)
1-pchisq(SSH0,nrow(C))

          [,1]
[1,] 0.6967408
```

4.2. *The restricted mean*

For the restricted mean time to treatment failure we use the SAS macro or the R function "pseudomean". To illustrate we look at a regression model for the mean time to treatment failure restricted to 2000 days. Here we use the identity link function. The SAS code, assuming the macro was in the file 'pseudomu' is

```
%include 'pseudomu';
%pseudomean(one, tdfs, dfs, 137,2000,outdata);
proc genmod;
class id disease (param=ref ref=first);
model psumean= disease fab rage/dist=normal link=id noscale;
repeated subject=id/corr=ind;
```

The relevant output is

```
                 Analysis Of GEE Parameter Estimates
                  Empirical Standard Error Estimates
                         Standard   95% Confidence

Parameter    Estimate     Error        Limits            Z    Pr > |Z|
Intercept     1154.997 219.2613  725.2530  1584.741     5.27   <.0001
disease   2   630.5407 185.4911  266.9848   994.0967    3.40   0.0007
disease   3   143.5041 216.8834 -281.580    568.5878    0.66   0.5082
fab          -518.600  169.5438 -850.900   -186.301    -3.06   0.0022
age           -11.5556   6.8876  -25.0551     1.9438   -1.68   0.0934
```

Here we see that AML low risk patients have the longest restricted mean life, namely 630.5 days longer than ALL patients and that AML patients with FAB class 4/5 have lifetimes 578.6 days shorter than the reference group.

The analogous R commands and output would be

```
a <- cbind(bmt,pseudo=pseudomean(bmt$tdfs,
          bmt$dfs,tmax=2000)$psumean,id=1:nrow(bmt))
library(geepack)
summary(fit <- geese(pseudo ~ disease+ fab + age, data = a, id=id,
        jack = T, family=gaussian, corstr="independence", scale.fix=F))
        round(cbind(mean = fit$beta,SD = sqrt(diag(fit$vbeta.ajs)),
        Z = fit$beta/sqrt(diag(fit$vbeta.ajs)),
        PVal=2-2*pnorm(abs(fit$beta/sqrt(diag(fit$vbeta.ajs))))),4)

                 mean        SD         Z   PVal
(Intercept) 1154.9972 223.1147  5.1767 0.0000
disease2     630.5407 187.2927  3.3666 0.0008
disease3     143.5041 220.7480  0.6501 0.5156
fab         -518.6004 172.8409 -3.0004 0.0027
age          -11.5556   7.0672 -1.6351 0.1020
```

This R output shows elevated standard deviations resulting in higher P-values than in SAS output.

The restricted mean pseudo-values with an identity link can also be used with the “gee” function from the “gee” package [14] as follows.

```
library(gee)
fit <- gee(pseudo ~ disease + fab + age, data = a, id=id,
       family=gaussian, corstr="independence", scale.fix=F)
round(cbind(mean = fit$beta,SD = sqrt(diag(fit$vbeta.ajs)),
      Z = fit$beta/sqrt(diag(fit$vbeta.ajs)),
      PVal = 2-2*pnorm(abs(fit$beta/sqrt(diag(fit$vbeta.ajs))))),4)

                 mean        SD         Z   PVal
(Intercept) 1154.9972 219.2613  5.2677 0.0000
disease2     630.5407 185.4911  3.3993 0.0007
disease3     143.5041 216.8834  0.6617 0.5082
fab         -518.6004 169.5438 -3.0588 0.0022
age          -11.5556   6.8876 -1.6777 0.0934
```

The function “gee” which used the “sandwich” estimator (3) to estimate the variance shows results identical to SAS. However, “gee” requires the use of a default link function (identity for the normal) and does not allow the selection of the complementary log–log as needed with the pseudo-value approach for survival and cumulative hazard functions.

4.3. *The competing risks cumulative incidence function*

For the cumulative incidence function we use the SAS macro and the R function “pseudoci” to compute the pseudo-values. To illustrate the SAS code we fit the complementary log–log model to the relapse cumulative incidence evaluated at 50, 105, 170, 280 and 530 days. Assuming the macro is in the file ‘pseudoci.txt’ the SAS code is

```
%include 'pseudoci.txt';
data times;
input tau;
cards;
50
105
170
280
530
;
run;

%pseudoci(one,tdfs,rel,trm,137,times,in.dataoutcr);

data two;
set in.dataoutcr;
dis2=0; if disease=2 then dis2=1;
dis3=0; if disease=3 then dis3=1;

proc print data=two round;

proc genmod;
class zid tpseudo;
FWDLINK LINK=LOG(-LOG(1-_MEAN_));
INVLINK ILINK=1-EXP(-EXP(_XBETA_));
model rpseudo= tpseudo dis2 dis3 fab /dist=normal noscale  noint;
repeated subject=zid / corr=ind ;
```

A partial listing of the SAS output is as follows:

```
                      d               r  d   t
                      i               p  p   p
                      s               s  s   s
     t                e      r        e  e   e   d  d
O    d  t  r  d       a  f   a    t   u  u   u   i  i
b    f  r  e  f   i   s  a   g    a   d  d   d   s  s
s    s  m  l  s   d   e  b   e    u   o  o   o   2  3

1    1  1  0  1  35   1  0  42   50   0  1   50  0  0
2    1  1  0  1  35   1  0  42  105   0  1  105  0  0
3    1  1  0  1  35   1  0  42  170   0  1  170  0  0
4    1  1  0  1  35   1  0  42  280   0  1  280  0  0
4    1  1  0  1  35   1  0  42  530   0  1  530  0  0
```

```
                      Analysis Of GEE Parameter Estimates
                       Empirical Standard Error Estimates

                        Standard   95% Confidence
Parameter     Estimate    Error         Limits            Z Pr > |Z|
Intercept       0.0000   0.0000   0.0000   0.0000      .      .
tpseudo   50   -3.5543   0.8338  -5.1885  -1.9202   -4.26   <.0001
tpseudo   105  -2.5363   0.6538  -3.8177  -1.2548   -3.88   0.0001
tpseudo   170  -2.0702   0.6369  -3.3185  -0.8219   -3.25   0.0012
tpseudo   280  -1.7251   0.6204  -2.9411  -0.5090   -2.78   0.0054
tpseudo   530  -1.4339   0.6089  -2.6272  -0.2405   -2.36   0.0185
dis2           -1.7667   0.6647  -3.0694  -0.4639   -2.66   0.0079
dis3           -0.2447   0.5823  -1.3860   0.8965   -0.42   0.6743
fab             1.1327   0.5110   0.1311   2.1343    2.22   0.0267
rage            0.0143   0.0216  -0.0280   0.0566    0.66   0.5084
```

Here we are modeling $C_k(t|\mathbf{Z}) = 1 - \exp\{-\Lambda_0(t)e^{\beta\mathbf{Z}}\}$, where $\Lambda_0(t)$ is e to the power of the appropriate tseudo coefficient. In this model, positive values of β for a covariate suggest a larger cumulative incidence for patients with $\mathbf{Z} = 1$ or equivalently more relapse. The model suggests that the AML low risk patients have the least chance of relapse and the AML FAB 4/5 the highest chance of relapse. Note that here we are modeling the probability of having relapsed where for the Kaplan–Meier curves we are modeling the probability of the event occurring.

R implementation uses the function "pseudoci" which produces a dataset where the time and status variables are presented in the first two columns and the pseudo-values are located in columns starting from the third. Odd numbered columns correspond to the competing risk with indicator 1 and even numbered columns for the competing risk number 2. A pair of pseudo-values is given for each time point in "datatau." In the example, the third column represents the relapse pseudo-value at 50 days, the fourth the trm pseudo-value at 50 days, the fifth the relapse pseudo-value at 105 days, the sixth the trm pseudo-value at 105 days, and so forth. In order to use the "geese" function we need only relapse pseudo-values arranged in one column and in another column we need the pseudo-value's time points. The six lines of code in bold after the call to "pseudoci" merge the output of the function with the original data and prepare it for analysis using the function "geese." The program and output are given below:

```
cutoffs <- c(50,105,170,280,530)

bmt$icr <- bmt$relapse +  bmt$dfs
# the variable bmt$icr takes value 2 for relapse, 1 for death in
remission, and 0 for censoring

pseudo <- pseudoci(bmt$tdfs,bmt$icr,cutoffs)
# for each time cutoff the 'pseudoci' function generated several columns
of pseudovalues (one column for each competing risk); in our case we have
only two risks, 1 and 2 (in terms of the bmt$icr); since we are
specifically interested in the first one, the following variable will be
used to exclude risk 2 from further consideration

rel_mask <- c(50,-1,105,-1,170,-1,280,-1,530,-1)
b <- NULL
for(j in 3:ncol(pseudo)) b <- rbind(b,cbind(bmt,pseudo = pseudo[,j],
                              tpseudo = rel_mask[j-2],id=1:nrow(bmt)))
b <- b[order(b$id),]
b <- b[b$tpseudo != -1,]

library(geepack)
b$tpseudo    <- as.factor(b$tpseudo)

fit <- geese(pseudo ~ tpseudo + disease + fab + age - 1 ,
       data = b, id=id, jack = T, scale.fix=TRUE, family=gaussian,
       mean.link = "cloglog", corstr="independence")
round(cbind(mean = fit$beta,SD = sqrt(diag(fit$vbeta.ajs)),
      Z = fit$beta/sqrt(diag(fit$vbeta.ajs)),
      PVal =2-2*pnorm(abs(fit$beta/sqrt(diag(fit$vbeta.ajs))))),4)
```

```
              mean     SD       Z   PVal
tpseudo50  -3.5543 0.8522 -4.1707 0.0000
tpseudo105 -2.5362 0.6745 -3.7604 0.0002
tpseudo170 -2.0702 0.6587 -3.1430 0.0017
tpseudo280 -1.7251 0.6433 -2.6816 0.0073
tpseudo530 -1.4338 0.6323 -2.2677 0.0233
disease2   -1.7667 0.6749 -2.6177 0.0089
disease3   -0.2447 0.5942 -0.4119 0.6804
fab1        1.1327 0.5253  2.1562 0.0311
rage        0.0143 0.0227  0.6294 0.5291
```

Again the estimates are identical to those obtained in SAS but the bootstrap standard errors are slightly different.

5. Discussion

We have presented SAS macros and R functions to find pseudo-values for the survival function, the restricted mean and the cumulative incidence function. The SAS macros can be found at http://www.biostat.mcw.edu/software/SoftMenu.html. The R functions are available in the CRAN site.

The regression models for the survival function and cumulative incidence functions can be based on the functions at a single point in time or they can be for several points of the curves. When a regression model for the entire curve is to be studied we recommend, based on a Monte Carlo study found in [6], five to ten time points roughly evenly spaced on the event scale. In the examples we used an independent working covariance matrix for the GEE calculations. Another possibility is to use the empirical correlations between the pseudo-values [6].

The "geese" function from the R package "geepack" was used for GEE fitting. The "gee" function did not allow us to change mean link function to complementary log for the Gaussian family. However, "gee" "sandwich" variance estimates are identical to those in SAS, which is not true for "geese".

Conflict of interest

None.

Acknowledgements

This research is supported by a grant R01-54706-12 from the National Cancer Institute and by the Danish National Research Council, grant 272-08-0472 "Point processes and their statistical inference."

REFERENCES

[1] D.R. Cox, Regression models and life-tables (with discussion), J. Roy. Stat. Soc. B 34 (1972) 187–220.

[2] J.P. Klein, M.L. Moeschberger, Survival Analysis: Statistical Methods for Censored and Truncated data, 2nd ed., Springer–Verlag, New York, 2003.

[3] J.P. Klein, M.J. Zhang, in: Armitage, Colton (Eds.), Survival Analysis, Software: Encyclopedia of Biostatistics, vol. 8, 2nd ed., 2005, pp. 5377–5382.

[4] P.K. Andersen, J.P. Klein, S. Rosthøj, Generalized linear models for correlated pseudo-observations with applications to multi-state models, Biometrika 90 (2003) 15–27.

[5] P.K. Andersen, M.G. Hansen, J.P. Klein, Regression analysis of restricted mean survival time based on pseudo-observations, Life Time Data Anal. 10 (2004) 335–350.

[6] J.P. Klein, P.K. Andersen, Regression modeling of competing risks data based on pseudo-values of the cumulative incidence function, Biometrics 61 (2005) 223–229.

[7] J.P. Klein, Modeling competing risks in cancer studies, Stat. Med. 25 (2006) 1015–1034.

[8] P.K. Andersen, J.P. Klein, Regression analysis for multistate models based on a pseudo-value approach, with applications to bone marrow transplantation studies, Scand. J. Stat. 34 (2007) 3–16.

[9] J.P. Klein, P.K. Andersen, B.L. Logan, M.G. Harhoff, Analyzing survival curves at a fixed point in time, Stat. Med. 26 (2007) 4505–4519.

[10] K.-Y. Liang, S.L. Zeger, Longitudinal data analysis using generalized linear models, Biometrika 78 (1986) 13–22.

[11] E.L. Kaplan, P. Meier, Non-parametric estimation from incomplete observations, J. Am. Stat. Assoc. 53 (1958) 457–481.

[12] E.A. Copelan, J.C. Biggs, J.M. Thompson, P. Crilley, J. Szer, J.P. Klein, N. Kapoor, B.R. Avalos, I. Cunningham, K. Atkinson, K. Downs, G.S. Harmon, M.B. Daly, I. Brodsky, S.I. Bulova, P.J. Tutschka, Treatment for acute meyelocytic leukemia with allogeneic bone marrow transplantation following preparation with Bu/Cy, Blood 78 (1991) 838–843.

[13] http://cran.r-project.org/.

[14] J. Yan, J. Fine, Estimating equations for association structures, Stat. Med. 23 (2004) 859–874.