Bone Marrow Transplantation (2010) 45, 1388–1395
© 2010 Macmillan Publishers Limited All rights reserved 0268-3369/10
www.nature.com/bmt

SPECIAL REPORT

Regression modeling of competing risk using R: an in depth guide for clinicians

L Scrucca[1], A Santucci[2] and F Aversa[2]

[1]*Statistics Section, Department of Economy Finance and Statistics, University of Perugia, Perugia, Italy and* [2]*Hematology and Clinical Immunology Section, Department of Clinical and Experimental Medicine, University of Perugia, Perugia, Italy*

We describe how to conduct a regression analysis for competing risks data. The use of an add-on package for the R statistical software is described, which allows for the estimation of the semiparametric proportional hazards model for the subdistribution of a competing risk analysis as proposed by Fine and Gray. *J Am Stat Assoc* 1999; 94: 496–509.

Bone Marrow Transplantation (2010) **45,** 1388–1395; doi:10.1038/bmt.2009.359; published online 11 January 2010
Keywords: regression model for competing risk analysis; model selection; R statistical software; instructions for use; sample data base; HSCT recipients

Introduction

Competing risks often occur in cases of transplant data, when the intent is to estimate the occurrence of one cause, for example relapse, but other events, such as transplant-related death, can compete and need to be taken into account appropriately.[1–3] This paper is the second part of an earlier published article by the same authors.[4] As the subject matter is complex, the paper is addressed to those clinicians with good statistical knowledge, who routinely analyze their own data.

In our earlier publication,[4] we presented a description of the univariate competing risk analysis performed using the R statistical software.[5] Furthermore, we discussed how to perform significance testing when different groups are involved. Here, we show how to perform a multivariable regression analysis in the presence of competing risks data.

During the last two decades, many authors have proposed different methods to analyze survival data in the presence of competing risk (see below for a brief review), but applications from clinicians, including those who are well trained in medical statistics, are still not currently performed. In fact, although multivariable survival analysis is a well-known tool, as evidenced by the popularity of the Cox model in the medical field, a different situation arises with competing risk analysis. In our opinion, one possible reason for this lack of adoption is related to the fact that most of the common statistical software used by clinicians has not implemented this type of analysis. To promote the use of regression modeling in the presence of competing risk events, we illustrate how to perform a multivariable regression analysis using the semiparametric proportional hazards model proposed by Fine and Gray.[6] The analysis is performed using the *crr* package for the R statistical software. A data set concerning hematopoietic stem cell transplantation is used as an example to evaluate the dependence of the cumulative incidence of relapse, in the presence of transplant-related death, on some predictive factors and covariates.

Regression models for competing risks data analysis

The most commonly used regression model for analyzing event-time data is the Cox proportional hazards model. In the presence of competing risks, the standard Cox proportional hazards model is not adequate because the cause-specific Cox model treats competing risks of the event of interest as censored observations. In addition, the cause-specific hazard function does not have a direct interpretation in terms of survival probability.

An adaptation of Cox regression requiring data augmentation has been proposed by Lunn and McNeil[7]. With k competing events, the data for each patient are duplicated k times, one row for each type of failure; then, $k-1$ indicator variables are created for identifying whether a certain event has occurred. A stratified Cox regression could also be applied to allow non-proportional hazards.

Direct regression modeling of the effect of covariates on the cumulative incidence function (CIF) for competing risks data has been proposed, among others, by Fine and Gray,[6] and by Klein and Andersen.[8] For a review of different approaches to regression modeling in the presence of competing risks see Klein and Zhang,[9] Moeschberger *et al.*,[10] and Logan *et al.*[11]

Fine and Gray[6] proposed a model for the subdistribution hazard of the CIF. The *subdistribution hazard* is a key

Correspondence: Professor F Aversa, Internal and Experimental Medicine Department, University of Perugia, Sant'Andrea delle Fratte, Perugia 06156, Italy.
E-mail: aversa@unipg.it

Received 28 April 2009; revised 9 October 2009; accepted 20 October 2009; published online 11 January 2010

concept in this approach, and it is defined as the hazard of failing from a given cause in the presence of competing events, given that a subject has survived or has already failed due to different causes. We can write the subdistribution hazard for cause r as

$$\begin{aligned}\lambda_r(t) &= \lim_{\Delta t \to 0} \frac{\Pr(t \leqslant T < t + \Delta t, R = r | T \geqslant t \cup (T \leqslant t \cap R \neq r))}{\Delta t} \\ &= -\frac{d}{dt} \log(1 - I_r(t))\end{aligned}$$

where $I_r(t) = \Pr(T \leqslant t,\ R = r)$ is the CIF for cause r $(r = 1, \ldots, k)$.

Fine and Gray[6] adopted a semiparametric proportional hazards model for the subdistribution hazard of cause r for a subject with covariate vector $\boldsymbol{X}$ as follows

$$\lambda_r(t|\boldsymbol{X}) = \lambda_{r0}(t) \exp(\beta_r^{\mathrm{T}} \boldsymbol{X})$$

where $\lambda_{r0}(t)$ is the baseline subdistribution hazard of cause r, and β_r is the vector of coefficients for the covariates. Estimation for this model follows the partial likelihood approach used in a standard Cox model.

Using R for competing risks regression analysis

R is the statistical software available at http://www.R-project.org and distributed under the GNU (http://www.gnu.org) General Public License. It allows for many, if not all, types of statistical analyses, either in the base distribution or via additional packages.

Scrucca *et al.*[4] showed how to perform a competing risk analysis in R using an add-on package called cmprsk. They also discussed installation of the R software and the cmprsk package. In the following, a basic knowledge of R is assumed, especially with regard to reading a data file and manipulation of a vector or a matrix of values. Interested readers may consult the above-mentioned paper and the references therein for details.

To facilitate the analysis we wrote some simple functions, which are described in Appendix A and also contained in the file crr- addson.R available at http://www.stat.unipg.it/luca/R/. Assuming that this file is in your current working directory, you may load it as follows:

```
> source(“crr-addson.R”)
```

Data analysis example

We will analyze data from 177 patients who received a stem cell transplant for acute leukemia. The aim of the analysis was to estimate the cumulative incidence of relapse in the presence of transplant-related death, which deals with competing events. The effect on relapse of predictive factors and covariates such as Sex, Disease (lymphoblastic or myeloblastic leukemia), Phase at transplant (Relapse, CR1, CR2, CR3), Source of stem cells (bone marrow and peripheral blood, coded as BM + PB, or peripheral blood, coded as PB), and Age will be evaluated.

Table 1 Variables in the data file example

Variable	*Description*	*Statistical summary*[a]

Sex	Sex	M = Male (100)
		F = Female (77)

D	Disease	ALL (73) AML (104)
Phase	Phase	CR1 (47) CR2 (45) CR3 (12) Relapse (73)
Source	Type of transplant	BM + PB (21) PB (156)
Age	Age of patient (years)	4–62 30.47 (13.04)
Ftime	Failure time (months)	0.13–131.77 20.28 (30.78)
Status	Status indicator	0 = censored (46) 1 = relapse (56) 2 = competing event (75)

[a] For categorical variables, the counts for each level are reported; for quantitative variables, the minimum, the maximum, the mean, and standard deviation of the distribution are shown.

The data set is available at the URL http://www.stat.unipg.it/luca/R in the file ‘bmtcrr.csv’ and the contained variables are summarized in Table 1.

Assuming that the data file is located on the current working directory, it can be read as follows:

```
> bmt = read.csv(“bmtcrr.csv”)
```

Should the full path be needed, the function *file.choose()* provides a graphical user interface from which users can search for a file within any folder (see also Scrucca *et al.*[4]). If the data file is correctly read, then the user may look at the values for some of the first patients using:

```
> head(bmt)

  Sex D   Phase   Age Status Source ftime
1 M   ALL Relapse 48  2      BM+PB  0.67
2 F   AML CR2     23  1      BM+PB  9.50
3 M   ALL CR3     7   0      BM+PB  131.77
4 F   ALL CR2     26  2      BM+PB  24.03
5 F   ALL CR2     36  2      BM+PB  1.47
6 M   ALL Relapse 17  2      BM+PB  2.23
```

We aim to model the failure time ftime, with censoring and competing events provided by Status, as a function of the predictors Age, Sex, D, Phase, and Source. With the exception of age, the other covariates are categorical, so to include them into our model we need to carefully code them numerically. Several codings of factors are available, but the simplest is based on the so-called ‘baseline’ codification. For a factor or categorical variable made of *J* levels or categories, we must create *J-1* dummy variables or indicator variables, that is, variables coded as 1 in the presence of a given category and 0 otherwise. One category is treated as baseline and the corresponding dummy variable is dropped. For example, Sex has two levels (F and M) so we need only one variable to represent it; using males as baseline, we assign 1 to females and 0 to males. For the variable Phase, which has four levels (CR1, CR2, CR3, and Relapse), we need three dummy variables.

We provide a R function *factor2ind()*, which creates a matrix of indicator variables from a factor (see Appendix A). For example, to obtain the indicator variable for Sex using ‘Male’ as baseline we use:

```
> factor2ind(Sex, ‘‘M’’)

                                           Sex:F
[1,]                                           0
[2,]                                           1
[3,]                                           0
...
[177,]                                         0
```

The *factor2ind()* function returns a matrix with a single column and a number of rows equal to the number of observations. The values 1 indicate females, and 0 males. For instance, the second patient is a female, whereas the first and the third patients are males. If we apply this function to the categorical variable Phase with ‘Relapse’ as baseline, we obtain the following:

```
> factor2ind(Phase, ‘‘Relapse’’)

          Phase:CR1     Phase:CR2     Phase:CR3
[1,]          0             0             0
[2,]          0             1             0
[3,]          0             0             1
...
[177,]        0             0             0
```

The resulting matrix has three columns, that is, the number of levels of Phase minus one. Looking at the output we see that the first and last patients had a relapse (all zeroes appear in the corresponding rows), whereas the second patient CR2 and the third one CR3.

The matrix of fixed covariates, also called design matrix, can be constructed in R as follows

```
> x = cbind(Age, factor2ind(Sex, ‘‘M’’), factor2-
ind(D, ‘‘ALL’’), factor2ind(Phase, ‘‘Relapse’’),
factor2ind(Source))
```

Here, we use the function *cbind()* to concatenate by columns the variables Age, and the indicator variables for Sex, D, Phase, and Source. The first rows of the design matrix are:

```
> head(x)

     Age Sex: D:  Phase: Phase: Phase: Source:
         F    AML CR1    CR2    CR3    PB
[1,] 48  0    0   0      0      0      0
[2,] 23  1    1   0      1      0      0
[3,] 7   0    0   0      0      1      0
[4,] 26  1    0   0      1      0      0
[5,] 36  1    0   0      1      0      0
[6,] 17  0    0   0      0      0      0
```

The main function to fit regression models for competing risks data is *crr()*, which is contained in the cmprsk package. In the simplest form it requires a vector of follow-up times, a vector of status with a code for each failure type or censoring, and a matrix of fixed covariates. By default, the censoring code for status is set by the optional argument cencode = 0, and the code that denotes the failure type of interest is set by the optional argument failcode = 1. (In our example, transplant-related death, which is the competing event, is coded with 2.)

The first regression model for relapse can be produced by typing

```
> mod1 = crr(ftime, Status, x)
```

which returns an object assigned to the variable mod1. A summary of the estimation is simply obtained as follows:

```
> summary(mod1)

Competing Risks Regression
Call: crr(ftime = ftime, fstatus = Status, cov1 = x)
           coef      exp(coef)  se(coef)  z       P-value
Age        −0.0185   0.982      0.0119    −1.554  0.1200
Sex:F      −0.0352   0.965      0.2900    −0.122  0.9000
D:AML      −0.4723   0.624      0.3054    −1.547  0.1200
Phase:CR1  −1.1018   0.332      0.3764    −2.927  0.0034
Phase:CR2  −1.0200   0.361      0.3558    −2.867  0.0041
Phase:CR3  −0.7314   0.481      0.5766    −1.268  0.2000
Source:PB  0.9211    2.512      0.5530    1.666   0.0960

           exp(coef) exp(−coef) 2.5%      97.5%
Age        0.982     1.019      0.959     1.005
Sex:F      0.965     1.036      0.547     1.704
D:AML      0.624     1.604      0.343     1.134
Phase:CR1  0.332     3.009      0.159     0.695
Phase:CR2  0.361     2.773      0.180     0.724
Phase:CR3  0.481     2.078      0.155     1.490
Source:PB  2.512     0.398      0.850     7.426

Num. cases = 177
Pseudo Log-likelihood = − 267
Pseudo likelihood ratio test = 24.4 on 7 df
```

The first part of the output shows for each term in the design matrix the estimated coefficient $\hat{\beta}_j$, the relative risk $\exp(\hat{\beta}_j)$, the standard error, the *z*-value and the corresponding *P*-value for assessing significance. In our example, Sex is not significant, followed by Age and D, whereas Source is only marginally significant. Phase is a factor with relapse as baseline, so each *P*-value provides a test for the difference of each level with respect to the baseline. An overall *P*-value for Phase (the overall *P*-value is always required when modeling a factor with more than two levels), can be obtained through the Wald test. This is implemented in the R package aod and, assuming that such a package is already installed, by typing

```
> library(aod)
> wald.test(mod1$var, mod1$coef, Terms = 4:6)
```

```
Wald test:
  ----------
  Chi-squared test:

  X2 = 14.0, df = 3, P(>X2) = 0.0029
```

The first argument to the function *wald.test()* is the estimated covariance matrix for the coefficients, followed by the vector of coefficients estimates, and the position of coefficients for which we want to assess significance (see help(wald.test) for a more detailed description of these and other available arguments). In our case, the *P*-value indicates that Phase is statistically significant.

The second part of the output for competing risks regression shows the relative risk for each term, $\exp(\beta_j)$, and a 95% confidence interval (the confidence level may be set at a different value using the argument conf.int in the *summary()* call—see *help(summary.crr)*). The relative risk or subdistribution hazard ratio for a categorical covariate is the ratio of subdistribution hazards for the actual group with respect to the baseline, with all other covariates being equal. If the covariate is continuous then the relative risk refers to the effect of a one unit increase in the covariate, with all other covariates being equal. In our data, $\exp(-0.0352)=0.965$ is the relative risk of a female with respect to a male, and $\exp(-0.0185)=0.982$ is the relative risk for a 1 year increase in age.

The last part of the output shows the pseudo log-likelihood at maximum and the pseudo likelihood ratio test, that is, the difference in the objective function at the global null and at the final estimates. As this objective function is not a true likelihood, this test statistic is not asymptotically distributed as a χ^2. As a consequence, model comparison based on likelihood ratio approach cannot be performed directly, but significance must be evaluated through simulations. However, a model selection criterion can be easily adopted as described in the following section.

Model selection

The likelihood of the data for a given model is a measure of the goodness of fit. However, the likelihood is increased when the number of parameters in the model is also increased. This may lead to overfitting. To avoid this, information criteria penalize the likelihood on the basis of the number of estimated parameters. Such criteria can then be used for the selection of a model among a set of candidate models.

Two of the most commonly used information criteria are the Akaike information criteria[12] (AIC) and the Bayesian information criteria[13] (BIC). The AIC is defined as

$$AIC = -2l + 2d,$$

where l is the maximized value of the log-likelihood for a given model and d is the number of free parameters to be estimated. For a regression model, d is usually equal to the number of estimated coefficients. Thus, AIC includes a penalty, which is an increasing function of the number of estimated parameters. In contrast, BIC is defined as

$$BIC = -2l + \log(n)d,$$

where n is the number of observations. The BIC penalizes models using a large number of free parameters more strongly than does the AIC.

From a practical point of view, the AIC and BIC differ in how the penalty term is defined. However, they have been derived from very different points of view: AIC provides an estimate of the expected relative Kullback–Leibler distance between the estimated model and the true model,[14] whereas BIC is derived from a Bayesian point of view, and it can be seen as an approximation to the logarithm of the Bayes factor for comparing two models with equal prior probability on each model.[15]

Both AIC and BIC do not provide a test on the model in the sense of hypothesis testing, rather they provide a tool for ranking the competing models according to the selected criterion. As the magnitude of any information criterion is not relevant, differences with respect to the smallest value are usually computed. Interpretation is then based on general rules of thumb. For example, if we define $\Delta BIC_i = BIC_i - \min(BIC)$ the BIC difference for model i with respect to the smallest value of BIC for a set of candidate models, Kass and Raftery[15] argued that values of $\Delta BIC_i > 10$ provide very strong evidence against the i-th model, but values of $0 < \Delta BIC_i < 2$ suggest that the i-th model has substantial support and should receive consideration in making inferences. Similar considerations apply for AIC.[14]

We return to our data analysis example. After we fitted our starting model, we realized that some covariates appeared not to be significant or only marginally significant, and so they were candidates for removal from the model. This problem can be recast as a model selection problem using one of the information criterion discussed.

We may start by fitting a set of candidate models for which we pursue model selection. Fitting all possible models is not viable as there are $(2^5-1)=31$ of them. We may consider a 'forward' approach, adding covariates on the basis of their significance in the full model and then comparing the resulting models. The first covariate is the factor Phase, followed by Source. Then there are three covariates, Age, Sex, and D, which are largely not significant and presumably should not all be included at the same time. For this reason, we may include one of them at a time. The models we are going to compare are the following (note that in R all the commands following the symbol # are commented out, hence below, we add comments to aid the reader):

```
> mod2 = crr(ftime, Status, x[,4:6]) # Phase
> mod3 = crr(ftime, Status, x[,c(4:6,7)])
  # Phase + Source
> mod4 = crr(ftime, Status, x[,c(4:6,7,1)])
  # Phase + Source + Age
> mod5 = crr(ftime, Status, x[,c(4:6,7,2)])
  # Phase + Source + Sex
> mod6 = crr(ftime, Status, x[,c(4:6,7,3)])
  # Phase + Source + D
```

The function *modsel.crr()* allows model selection on a list of candidate models (see Appendix A). It can be simply invoked as follows:

```
> modsel.crr(mod1, mod2, mod3, mod4, mod5, mod6)

Model selection table
Model 0: Null model
Model 1: crr(ftime = ftime, fstatus = Status, cov1
         = x)
Model 2: crr(ftime = ftime, fstatus = Status, cov1
         = x[, 4:6])
Model 3: crr(ftime = ftime, fstatus = Status, cov1
         = x[, c(4:6, 7)])
Model 4: crr(ftime = ftime, fstatus = Status, cov1
         = x[, c(4:6, 7, 1)])
Model 5: crr(ftime = ftime, fstatus = Status, cov1
         = x[, c(4:6, 7, 2)])
Model 6: crr(ftime = ftime, fstatus = Status, cov1
         = x[, c(4:6, 7, 3)])
    Num.obs  logLik    Df.fit  BIC     BIC diff
0   177.00   −278.71   0.00    557.41  0.00
1   177.00   −266.52   7.00    569.28  11.87
2   177.00   −271.53   3.00    558.59  1.18
3   177.00   −270.78   4.00    562.27  4.85
4   177.00   −267.81   5.00    561.49  4.08
5   177.00   −270.73   5.00    567.33  9.92
6   177.00   −267.82   5.00    561.53  4.11
```

For each model, we have included an argument in the call to the function, the output provides the sample size, the maximized log-likelihood, the number of estimated parameters (Df.fit), the BIC value and the BIC difference with respect to the minimum value observed from the set of candidate models. The null model (labeled as Model 0 in the above output) is also automatically included; this is the model with no covariates, so it may serve as a reference for the inclusion of any of the available predictors. The smallest BIC value is achieved by the null model, closely followed by the model with the covariate Phase included. Adopting the general rule-of-thumb discussed earlier, we use the following as our working model:

```
> summary(mod2)

Competing Risks Regression
Call: crr(ftime = ftime, fstatus = Status, cov1 = x[, 4:6])
           coef     exp(coef)  se(coef)  z      P-value
Phase:CR1  −1.113   0.329      0.371     −3.00  0.0027
Phase:CR2  −0.979   0.376      0.347     −2.82  0.0048
Phase:CR3  −0.789   0.455      0.602     −1.31  0.1900

           exp(coef) exp(-coef) 2.5%     97.5%
Phase:CR1  0.329     3.04       0.159    0.680
Phase:CR2  0.376     2.66       0.190    0.742
Phase:CR3  0.455     2.20       0.140    1.480

  Num. cases = 177
  Pseudo Log-likelihood = − 272
  Pseudo likelihood ratio test = 14.3 on 3 df
```

The coefficients associated with CR1 and CR2 are significantly different from zero, so the relative risk of these levels of Phase with respect to the baseline category given by Relapse is about 1/3. On the contrary, the coefficient for CR3 is not significant; therefore, the relative risk, which is about 1/2, has a confidence interval including the null hypothesis (0.14–1.48).

Model diagnostic

The output of the *crr()* function also provides a matrix of *Schoenfeld residuals*, which are analogous to the Schoenfeld residuals in ordinary survival models. Plotting the *j*-th column of this matrix against the vector of unique failure times allows for the evaluation of the lack of fit over time in the corresponding covariate.

In our case, we may display the matrix of residuals, and plot them against failure times as follows:

```
> mod2$res

           [,1]         [,2]         [,3]
[1,]    −0.1394277   −0.1525481   −0.04922084
[2,]    −0.1406975   −0.1539373   −0.04966909
[3,]    −0.1419906   −0.1553521   −0.05012558
...
[53,]   −0.1695314   −0.1810540   −0.06172888

> par(mfrow = c(1,3), mar = c(4.5,4,2,1))
> for(j in 1:ncol(mod2$res))

   scatter.smooth(mod2$uft, mod2$res[,j],
                  main = names(mod2$coef)[j],
                  xlab = “Failure time”,
                  ylab = “Schoenfeld residuals”)
```

Plots of Schoenfeld-type residuals against time failure for each term in the final model are shown in Figure 1. If the proportional hazard subdistribution assumption holds the residuals should have locally a constant mean across time. To check this, we added a scatterplot smoother to each plot; the resulting patterns do not show any evidence of a non-constant local average.

Time-varying covariates

Time-varying fixed effects are artificial time-dependent covariates, which represent the effect in different time periods of a covariate whose value is unchanging over time. These are typically adjustments for non-proportional hazards in the Cox model.[16]

The same arises with the Fine and Gray model, where one basic assumption that the subdistribution for an event of interest at a given covariate value is a constant shift on the complementary log–log scale from a baseline subdistribution function. This can be generalized by including interactions of one or more covariates with functions of time to allow the magnitude of the shift to change with follow-up time.

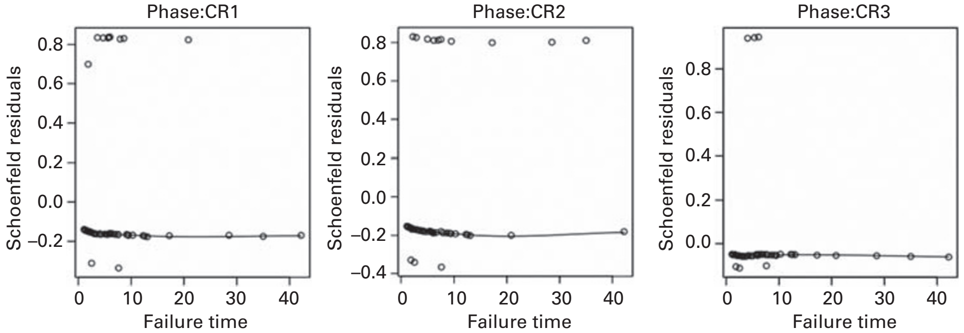

Figure 1 Plot of Schoenfeld-type residuals against time failure for each term in the final model. A locally weighted regression smoother has been added to each plot for assessing the proportional hazard subdistribution assumption.

Time-varying covariates can be modeled in *crr()* using two further arguments in the function call: cov2 a matrix of covariates to be multiplied with functions of time defined by the argument tf. Suppose we would like to fit a quadratic (in time) model for an Age, that is, β_1Age$+\beta_2$Age $t+\beta_3$Age t^2. This can be accomplished by using the following function call:

```
> mod7 = crr(ftime, Status, cov1 = x[,c(1,4:6)],
cov2 = cbind(Age,Age), tf = function(t)
cbind(t,t^2))
```

The argument cov1 contains the matrix of fixed covariates; in our case, we included the terms associated with the covariates Phase and Age. Then, cov2 provides a matrix with two columns, both having the values of Age, which are multiplied by the quadratic function defined in tf. The resulting output follows:

```
> summary(mod7)

Competing Risks Regression

Call: crr(ftime = ftime, fstatus = Status, cov1 = x[, c(1,
4:6)], cov2 = cbind(Age, Age), tf = function(t) cbind(t, t^2))

              Coef    exp(coef)  se(coef)      z   P-value
Age        −2.67e−02    0.974   2.00e−02  −1.335  0.1800
Phase:CR1  −1.09e+00    0.338   3.73e−01  −2.915  0.0036
Phase:CR2  −1.01e+00    0.364   3.48e−01  −2.907  0.0036
Phase:CR3  −8.85e−01    0.413   5.95e−01  −1.488  0.1400
Age*t       4.00e−04    1.000   3.17e−03   0.126  0.9000
Age*tf2     1.10e−05    1.000   7.09e−05   0.155  0.8800

          exp(coef) exp(−coef)   2.5%   97.5%
Age         0.974      1.03     0.936   1.013
Phase:CR1   0.338      2.96     0.163   0.701
Phase:CR2   0.364      2.75     0.184   0.719
Phase:CR3   0.413      2.42     0.129   1.324
Age*t       1.000      1.00     0.994   1.007
Age*tf2     1.000      1.00     1.000   1.000

  Num. cases = 177
  Pseudo Log-likelihood = − 269
  Pseudo likelihood ratio test = 19.5 on 6 df
```

Neither the linear nor the quadratic term expressing the interaction of time with Age are statistically significant. This provides further evidence that the proportional hazard subdistribution assumption is not violated.

Model-based estimation of the CIF

The predicted CIF can be computed for cause r as

$$\hat{I}_r(t) = 1 - \exp(-\hat{H}_r(t)),$$

where $\hat{H}_r(t)$ is the estimated cumulative subdistribution hazard for the event of interest obtained for a specified covariate value, and calculated using a Breslow-type estimator.

In our example, we may obtain the predicted CIF for each level of the covariate Phase at each unique failure time by first creating a matrix of values for each level

```
> x0 = cbind(Phase = factor2ind(levels(Phase),
  ‘‘Relapse’’))
> x0

         Levels        Levels        Levels
     (Phase):CR1   (Phase):CR2   (Phase):CR3
[1,]       1             0             0
[2,]       0             1             0
[3,]       0             0             1
[4,]       0             0             0
```

and then using the generic function *predict()* to compute the predicted CIF as follows:

```
> pred = predict(mod2, x0)
> pred

        [,1]        [,2]         [,3]         [,4]         [,5]
[1,]    1.10  0.002962152  0.003384217  0.004093336  0.008984103
[2,]    1.20  0.005942385  0.006787645  0.008206966  0.017968205
[3,]    1.23  0.008940978  0.010210569  0.012341175  0.026952307
…
[53,]  42.17  0.196133976  0.220796778  0.260563415  0.485300353
```

The output of the function *predict()* is a matrix with unique failure times in the first column, and the other columns giving the estimated subdistribution function corresponding to the covariate combinations in the rows

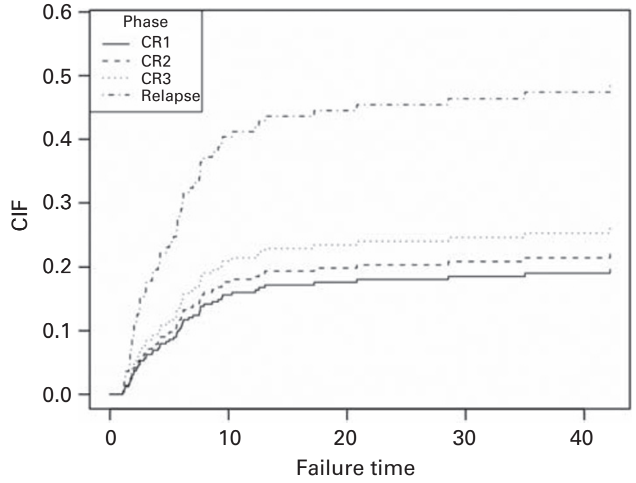

Figure 2 Predicted cumulative incidence curves for BMT Relapse at different Phase levels.

of the input matrix x0 at each failure time. Thus, in the above output, the second column refers to the predicted CIF for Phase = CR1, the third column for Phase = CR2, and so on.

Finally, we may plot the predicted CIF and obtain the graph in Figure 2 by typing

```
> plot(pred, lty = 1:4, xlab = ''Failure time'',
  ylab = ''CIF'')
> legend(''topleft'', legend = levels(Phase),
  lty = 1:4, title = ''Phase'')
```

Final remarks

In this paper, we discussed how to conduct a regression analysis for competing risks data using an add-on package for the R statistical software. We presented a typical transplant multivariable analysis, in which the cumulative incidence of relapse in the presence of the competitive event, transplant-related death, is studied.

As we did in our earlier paper, we provide the data set used as an example so that the readers may practice and feel more secure in their ability. We hope to have achieved our goal, which was to illustrate how to fit the models in R while, at the same time, supplying some useful statistical comments.

Conflict of interest

The authors declare no conflict of interest.

Acknowledgements

We thank Professor R Gray for his useful input on the cmprsk package, which lead to the improvement of displayed output of competing risk regression models in R.

References

1 Pintilie M. *Competing Risks: A Practical Perspective*. John Wiley & Sons: New York, 2006. pp 24.
2 Klein JP, Rizzo JD, Zhang MJ, Keiding N. Statistical methods for the analysis and presentation of the results of bone marrow transplants. Part I: unadjusted analysis. *Bone Marrow Transplant* 2001; **28**: 909–915.
3 Kim HT. Cumulative incidence in competing risks data and competing risks regression analysis. *Clin Cancer Res* 2007; **13**: 559–565.
4 Scrucca L, Santucci A, Aversa F. Competing risk analysis using R: an easy guide for clinicians. *Bone Marrow Transplant* 2007; **40**: 381–387.
5 R Development Core Team. *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing 2009, Vienna, Austria. ISBN 3–900051–07–0, URL http://www.R-project.org.
6 Fine JP, Gray RJ. A proportional hazards model for the subdistribution of a competing risk. *J Am Stat Assoc* 1999; **94**: 496–509.
7 Lunn M, McNeil D. Applying Cox regression to competing risks. *Biometrics* 1995; **51**: 524–532.
8 Klein JP, Andersen PK. Regression modeling of competing risks data based on pseudo values of the cumulative incidence function. *Biometrics* 2005; **61**: 223–229.
9 Klein JP, Zhang MJ. Survival analysis. *Handbook of Statistics* 2007; **27**: 281–320.
10 Moeschberger ML, Tordoff KP, Kochar N. A Review of Statistical Analyses for Competing Risks. *Handbook of Statistics* 2007; **27**: 321–341.
11 Logan BR, Zhang MJ, Klein JP. Regression models for hazard rates versus cumulative incidence probabilities in hematopoietic cell transplantation data. *Biol Blood Marrow Transplant* 2006; **12**: 107–112.
12 Akaike H. A new look at the statistical model identification. *IEEE Trans Automat Contr* 1974; **19**: 716–723.
13 Schwartz G. Estimating the dimension of a model. *Ann Stat* 1978; **6**: 31–38.
14 Burnham KP, Anderson DR. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer-Verlag: New York, 2002. pp 488.
15 Kass RE, Raftery AE. Bayes factors. *J Am Stat Assoc* 1995; **90**: 773–795.
16 Klein JP, Rizzo JD, Zhang MJ, Keiding N. Statistical methods for the analysis and presentation of the results of bone marrow transplants. Part 2: Regression modeling. *Bone Marrow Transplant* 2001; **28**: 1001–1011.

Appendix A

```
# This ensures that the package is loaded requires(cmprsk)
  ''factor2ind'' <- function(x, baseline)
  {
  # Given a factor variable x, create an indicator matrix of
dimension
  #  length(x)  x  (nlevels(x)-1)  dropping  the  column
corresponding to the
  # baseline level (by default the first level is used as
baseline).
  # Example:
  # > x = gl(4, 2, labels = LETTERS[1:4])
  # > factor2ind(x)
  # > factor2ind(x, ''C'')
    xname <- deparse(substitute(x))
    n <- length(x)
    x <- as.factor(x)
    if(!missing(baseline)) x <- relevel(x, baseline)
    X <- matrix(0, n, length(levels(x)))
    X[(1:n) + n*(unclass(x) - 1)] <- 1
    dimnames(X) <- list(names(x), paste(xname, levels(x),
sep = “:”))
    return(X[,-1,drop=FALSE])
  }
  “modsel.crr” <- function (object, …, d = log(object$n))
  {
  if(class(object) != “crr”)
  stop(“object is not of class ‘crr’”)
  objects <- list(object, .)
  nmodels <- length(objects)
  modnames <- paste(“Model”, format(1:nmodels), “:”
                         lapply(objects, function(x) x$call),
                         sep = “”, collapse = “\n”)
  # add null model
  mod0 <- object
  mod0$loglik <- mod0$loglik.null
  mod0$coef <- mod0$call$cov1 <- mod0$call$cov2 <- NULL
  objects <- c(list(mod0), objects)
  nmodels <- nmodels + 1
  #
  modnames <- c(“Model 0: Null model”, modnames)
```

```
  ns <- sapply(objects, function(x) x$n)
  dfs <- sapply(objects, function(x) length(x$coef))
  if(any(ns != ns[1]))
    stop(“models were not all fitted to the same dataset”)
  out <- matrix(rep(NA, 5 * nmodels), ncol = 5)
  loglik <- sapply(objects, function(x) x$loglik)
  crit <- sapply(objects,
    function(x) -2*x$loglik + d*length(x$coef))
  out[,1] <- ns
  out[,2] <- loglik
  out[,3] <- dfs
  out[,4] <- crit
  out[,5] <- critmin(crit)
  if(d==log(object$n)) critname <- “BIC”
  else if(d == 2) critname <- “AIC”
  else critname <- “Criterion”
  colnames(out) <- c(“Num.obs”, “logLik”, “Df.t”, critname,
    paste(critname, “diff”))
  rownames(out) <- 0:(nmodels-1)
  title <- “Model selection table\n”
  topnote <- modnames
  structure(as.data.frame(out), heading = c(title, topnote),
    class = c(“anova”, “data.frame”))
  }
```