Bone Marrow Transplantation (2007) 40, 381–387

www.nature.com/bmt

TECHNICAL REPORT

Competing risk analysis using R: an easy 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*

In the last decade with widespread use of quantitative analyses in medical research, close co-operation between statisticians and physicians has become essential from the experimental design through all phases of complex statistical analysis. On the other hand, easy-to-use statistical packages allow clinicians to perform basic statistical analyses themselves. Since the software they most commonly use does not perform in depth competing risk analysis, we recommend an add-on package for the R statistical software. We provide all the instructions for downloading it from internet and illustrate how to use it for analysis of a sample dataset of patients who underwent haematopoietic stem cell transplantation for acute leukaemia.

Bone Marrow Transplantation (2007) **40,** 381–387; doi:10.1038/sj.bmt.1705727; published online 11 June 2007
Keywords: competing risk analysis; R statistical software; sample data base; instructions for use

Introduction

In studies on HSCT Kaplan–Meier (KM) estimates of survival curves and Cox proportional hazard models are widely used to describe survival trends and identify significant prognostic factors. All these statistical analyses deal with only one type of event, for example death, independently of its cause.

A particular situation arises when interest is focused on a specific cause of failure in the presence of other different causes, which alter the probability of experiencing the event of interest. This is the case of competing risk events, which refers to a situation where an individual is exposed to two or more causes of failure, and its eventual failure can be attributed exactly to only one. In this case, the occurrence of one type of event hinders the occurrence of any other event.

In patients who underwent HSCT, failure events commonly studied are relapse of the original disease (REL) and death from causes related to the transplant (transplant related mortality (TRM)). If the interest is to estimate the probability of relapse, death from TRM is a competing risk event and the cumulative incidence function (CIF) must be calculated by appropriate accounting.[1,2]

Until recently, this analysis was erroneously performed as 1-KM, treating the competing events as censored at the time they occurred, but this censoring is inappropriate because after a competing event has occurred, failure from the cause of interest is no longer possible. 1-KM correctly estimates the probability of failure independently of any specific cause, while the probability of one type of competing event is correctly estimated using the CIF, which partition the probability of failure into the probability corresponding to each competing event: at any point in time, the overall 1-KM is equal to the sum of the CIFs for each type of event.[3,4] Moreover, to assess the statistical significance of a prognostic factor in a cumulative incidence analysis Gray's test[5] is one of the appropriate tests to perform.

Generally speaking, there are few statistical packages used today in medical statistics that implement CIF for competing risk events, and even fewer provide tools for comparing the cumulative incidence of a particular type of failure among different groups. A 'competing risks' analysis is provided by an add-on package of R.[6] R is an open source software for statistical computing and graphics, which is freely available at www.r-project.org. R performs many statistical analyses needed in practical applications: linear and generalized linear models, nonlinear regression models, time-series analysis, parametric and nonparametric tests, clustering, smoothing and survival analysis. R also provides additional packages for specific purposes.

We show how to perform a competing risk analysis in R by an example using a sample dataset containing survival times of anonymous patients affected by leukaemia who underwent HSCT. In the analysis we estimate CIF, perform test for equality of CIF curves in subgroups, and compute pointwise confidence intervals. A brief overview of the statistical concepts is reported in Appendix A.

Obtaining and installing R

R is an open source project that is distributed under the GNU (www.gnu.org) GPL (General Public License).

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 29 May 2006; revised 16 February 2007; accepted 19 April 2007; published online 11 June 2007

Sources, binaries, documentation and additional packages for R software can be obtained via CRAN, the Comprehensive R Archive Network, at the master site, TU Wien, Austria (http://cran.R-project.org). The R distribution comes with a set of manuals. We suggest beginners read the handbook *An Introduction to R*. Other books on R include introductory texts, such as Dalgaard,[7] Iacus and Masarotto[8] and data analysis books, such as Venables and Ripley.[9]

R software is currently developed for the Linux/Unix, Windows and Apple OS families of operating systems. The installing binary for Windows 95, 98, ME, NT4, 2000, and XP is available at http://cran.r-project.org/bin/windows/base/. The setup program for downloading is R-x.x.x-win32.exe, where x.x.x is the current version (at the time of this writing R-2.4.0 is available). After downloading the file, install as usual on the user's computer. Competing risk analysis is available in an add-on package called `cmprsk`.

Installing the R package `cmprsk`

Start R in Windows by double clicking on the desktop icon. R issues the symbol >, then expects input commands.

On line windows users:
select 'Packages', from the main menu,
select 'Install package(s)…',
choose a CRAN site,
select the `cmprsk` package to download and install.

Windows users who are not on-line:
download the `cmprsk` package as a zip file from http://cran.r-project.org/src/contrib/PACKAGES.html,
select 'Install package from local zip file…' from the 'Packages' menu.

Other information and details of how to install packages for other operating systems are available in the *R Installation and Administration* manual.

Reading data for competing risk analysis in R

As an example of competing risk analysis in R, we analyze data from 35 patients with acute leukaemia who underwent HSCT. We estimate the cumulative risk of relapse and TRM. At the same time we test equality of cumulative incidence curves in patients affected by AML and ALL. Data can be read in R in a variety of formats (including data from SAS, MINITAB, STATA) and these are fully explained in the *R Data Import/Export* manual. As MS Excel is commonly used for creating clinical databases, we will show how R software reads an Excel file which in our example is named '`bmt.xls`'.

Instructions:
Go to the Excel Save menu,
Save your worksheet file as a CSV (comma separated values) file,
Close Excel.

Start R in Windows by double clicking on the desktop icon. R issues the symbol >, then expects input commands.

In English versions of Excel where the decimal point is used:

Type:

```
>bmt=read.csv("bmt.csv")
```

When the comma is used for the decimal point as in the Italian version:

Type:

```
>bmt=read.csv("bmt.csv", sep=";", dec=",")
```

If, as above, the filename does not contain a path, the file is assumed to be located in the current working directory.

If the file is NOT in the current working directory, the function *`file.choose()`* provides a graphical user interface from which users can search for a file within any folder.

Type:

```
>bmt=read.csv(file.choose(), sep=";", dec=",")
```

Once our sample data has been read, R software recognizes them as *`bmt`*.

To see the data of the 35 patients

Type

```
 >bmt
      dis   ftime   status
1       0     13       2
2       0      1       1
3       0     72       0
4       0      7       2
[...]
34      1     32       0
35      0     12       1
```

`dis` indicates disease coded 0 for ALL and 1 for AML; *`ftime`* indicates follow-up time in months, that is, the length of follow-up from transplant to relapse for patients who relapsed, to death for patients dead for TRM or to the last check-up in survivors; *`status`* is coded as 0 for a survivor (censored), 1 for death from TRM, and 2 for relapse.

For an alternative data presentation, which shows the data structure,

Type:

```
>str(bmt)
'data.frame': 35 obs. of 3 variables:
$ dis :    int 0  0 0  0 0 1  0 0 1  1...
$ ftime:   int 13 1 72 7 8 67 9 5 70 4...
$ status:  int 2  1 0  2 2 0  2 2 0  0...
```

Before starting any data analysis

Type:

```
>attach(bmt)
```

The function *factor()* may be used to redefine *dis* (disease) as a factor with two levels, because in our example there are two diseases (AML and ALL). Each is given its label: 0 is labelled ALL, and 1 is labelled AML.

Type:

```
>dis=factor(dis, levels=c(0,1), labels=
c("ALL", "AML"))
```

To perform some basic descriptive summaries:

Type:

```
>table(dis, status)
        status
dis    0    1    2
ALL    2    3   12
AML    9    6    3
```

The table above shows the number of observations for each combination of status and disease. A table that reports the mean follow-up times in months for each combination of status and disease is obtained as follows:

Type:

```
>tapply(ftime, list(dis, status), mean)
        0         1         2
ALL    53.50000  4.333333  9.333333
AML    32.55556  4.166667  6.333333
```

Fitting the cumulative incidence function

We wrote an R function, called *CumIncidence()*, which allows to fit cumulative incidence curves with minimum user interaction. Its source code is reported in Appendix B but one may prefer to download the file '*CumIncidence.R*' from http://www.stat.unipg.it/luca/R.

The function *CumIncidence()* is simply a wrapper to the package `cmprsk`, and must be loaded in each work-session. Assuming it is contained in the file '`CumIncidence.R`' located in your working directory, it can be simply loaded as follows:

Type

```
>source ("CumIncidence.R")
```

If the file to be sourced is not in the current working directory, the function *file.choose()* can be used to select the file within any folder:

```
>source (file.choose())
```

The function *CumIncidence()* estimates cumulative incidence curves. *CumIncidence()* requires at least the following arguments: the follow-up time (*ftime* in our example); the indicator variable containing the competing events (*status*); the grouping variable (*dis*). The argument `cencode` indicates the code for censored observations, and it is set to *0* (default). Further arguments can be provided, such as *xlab* for the x-axis label, and will be described later. Finally, in our example the output estimates are assigned to an object denominated *fit*.

Type:

```
>fit=CumIncidence (ftime, status, dis, cencode =
0, xlab="Months")
```

The following output is shown on screen:

```
+- - - - - - - - - - - - - - - - - - - - +
| Cumulative incidence function estimates from |
|              competing risks data            |
+- - - - - - - - - - - - - - - - - - - - +

Test equality across groups:

   Statistic  P-value     df
1    1.302     0.253915    1
2    7.082     0.007785    1


Estimates at time points:
           0      10      20      30      40      50      60      70
ALL 1 0.05882 0.1176 0.1765 0.1765 0.1765 0.1765 0.1765 0.1765
AML 1 0.00000 0.3682 0.3682 0.3682 0.3682 0.3682 0.3682 0.3682
ALL 2 0.05882 0.4706 0.5882 0.7059 0.7059 0.7059 0.7059 0.7059
AML 2 0.00000 0.2057 0.2057 0.2057 0.2057 0.2057 0.2057 0.2057

Standard errors:
           0      10      20      30      40      50      60      70
ALL 1 0.05882 0.08056 0.09667 0.09667 0.09667 0.09667 0.09667 0.09667
AML 1 0.00000 0.12653 0.12653 0.12653 0.12653 0.12653 0.12653 0.12653
ALL 2 0.05882 0.12631 0.12667 0.12077 0.12077 0.12077 0.12077 0.12077
AML 2 0.00000 0.11558 0.11558 0.11558 0.11558 0.11558 0.11558 0.11558
```

First, the print out shows the results of Gray's test[5] for equality of CIFs across groups. In our example, cumulative incidence curves for ALL and AML are not statistically different for death due to TRM (coded as 1) (P-value = 0.253915), but they are highly significant for relapse (coded as 2) (P-value = 0.0077785).

The output reports cumulative incidence estimates, and corresponding standard errors, for each cause of failure (TRM or relapse) in each disease group (ALL or AML). A plot of estimated CIFs for each cause of failure–disease combination is also produced (see Figure 1).

By default cumulative incidence estimates are computed on a suitable grid of time points; in our example, time points are taken from 0 to 70 month by step of 10. These automatic time points can be easily customized by the user:

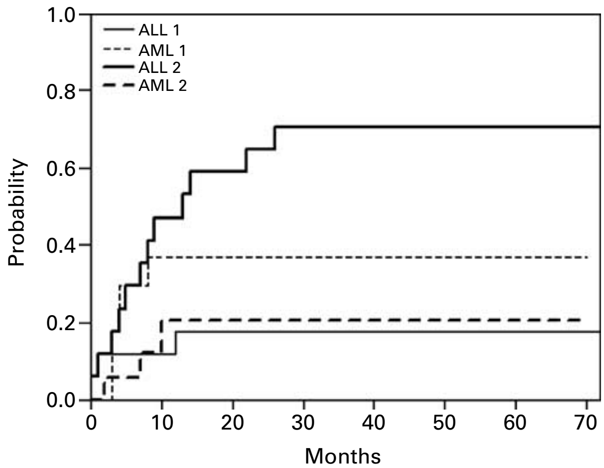

Figure 1 Estimated cumulative incidence curves with transplant related mortality (1) and relapse of the original disease (2) as competing events for each type of leukemia (acute lymphoblastic leukaemia or acute myeloid leukaemia).

in this case the call to the function *CumIncidence()* requires a further argument, *t*, which provides a vector containing the user-defined time points. Suppose, in our example, we chose 3-month intervals for the first year, and 12-month intervals for years 2, 3 and 4 after transplantation, then the call is the following:

Type:

```
>fit = CumIncidence(ftime, status, dis, cencode = 0,
xlab = "Months",
  t = c(0,   3,   6,   9,   12,   24,   36,   48))

+- - - - - - - - - - - - - - - - - - +
| Cumulative incidence function estimates from |
|            competing risks data              |
+- - - - - - - - - - - - - - - - - - +

  Test equality across groups:
     Statistic   P-value    df
  1    1.302     0.253915    1
  2    7.082     0.007785    1

Estimates at time points:
            0       3       6      9     12     24     36     48
ALL 1 0.05882 0.11765 0.11765 0.1176 0.1765 0.1765 0.1765 0.1765
AML 1 0.00000 0.17708 0.29514 0.3682 0.3682 0.3682 0.3682 0.3682
ALL 2 0.05882 0.17647 0.29412 0.4706 0.4706 0.6471 0.7059 0.7059
AML 2 0.00000 0.05556 0.05556 0.1205 0.2057 0.2057 0.2057 0.2057

Standard errors:
      0             3       6       9      12      24      36      48
ALL 1 0.05882 0.08056 0.08056 0.08056 0.09667 0.09667 0.09667 0.09667
AML 1 0.00000 0.09575 0.11478 0.12653 0.12653 0.12653 0.12653 0.12653
ALL 2 0.05882 0.09534 0.11429 0.12631 0.12631 0.12450 0.12077 0.12077
AML 2 0.00000 0.05556 0.05556 0.08379 0.11558 0.11558 0.11558 0.11558
```

The resulting output shows both estimates of cumulative incidence and s.e. evaluated at the given time points.

Pointwise confidence intervals for competing risk curves

Computing confidence intervals provides useful information about uncertainty related to parameter estimates. A pointwise confidence interval for CIF at some fixed time-point can be obtained using the method proposed by Choudhurt[10] (see Appendix A for a brief description).

The *CumIncidence()* function allows for the pointwise confidence intervals, by simply adding a further argument, `level`, where we specify the desired confidence level. For example, we may compute 95% pointwise confidence interval at our selected time points as follows:

Type:

```
>fit = CumIncidence(ftime, status, dis, cencode = 0,
xlab = "Months",
  t = c(0, 3, 6, 9, 12, 24, 36, 48), level = 0.95)

+- - - - - - - - - - - - - - - - - - +
| Cumulative incidence function estimates from |
|            competing risks data              |
+- - - - - - - - - - - - - - - - - - +

  Test equality across groups:
     Statistic   P-value    df
  1    1.302     0.253915    1
  2    7.082     0.007785    1

Estimates at time points:
              0         3         6       9      12      24      36      48
ALL 1   0.05882   0.11765   0.11765  0.1176  0.1765  0.1765  0.1765  0.1765
AML 1   0.00000   0.17708   0.29514  0.3682  0.3682  0.3682  0.3682  0.3682
ALL 2   0.05882   0.17647   0.29412  0.4706  0.4706  0.6471  0.7059  0.7059
AML 2   0.00000   0.05556   0.05556  0.1205  0.2057  0.2057  0.2057  0.2057

Standard errors:
              0         3         6       9      12      24      36      48
ALL 1   0.05882   0.08056   0.08056 0.08056 0.09667 0.09667 0.09667 0.09667
AML 1   0.00000   0.09575   0.11478 0.12653 0.12653 0.12653 0.12653 0.12653
ALL 2   0.05882   0.09534   0.11429 0.12631 0.12631 0.12450 0.12077 0.12077
AML 2   0.00000   0.05556   0.05556 0.08379 0.11558 0.11558 0.11558 0.11558

95% pointwise confidence intervals:
, , ALL  1
              0         3         6       9      12      24      36      48
Lower  0.003487   0.01819   0.01819 0.01819 0.03991 0.03991 0.03991 0.03991
Upper  0.242066   0.31884   0.31884 0.31884 0.39293 0.39293 0.39293 0.39293
, , AML  1
              0         3         6       9      12      24      36      48
Lower       Inf   0.04105    0.1024  0.1408  0.1408  0.1408  0.1408  0.1408
Upper       Inf   0.39118    0.5203  0.6010  0.6010  0.6010  0.6010  0.6010
, , ALL  2
              0         3         6       9      12      24      36      48
Lower  0.003487   0.04101    0.1023  0.2199  0.2199  0.3552  0.4017  0.4017
Upper  0.242066   0.38982    0.5185  0.6872  0.6872  0.8327  0.8755  0.8755
, , AML  2
              0         3         6       9      12      24      36      48
Lower       Inf  0.003365  0.003365 0.01778 0.04188 0.04188 0.04188 0.04188
Upper       Inf  0.230594  0.230594 0.32913 0.45472 0.45472 0.45472 0.45472
```

The first part of the output is equal to that discussed above. For any combination of competing events (TRM/REL) and disease (ALL/AML), the estimated lower and upper confidence limits at given time points are reported. A set of graphs which plot the estimated curves with the corresponding confidence intervals is also produced (see Figure 2).

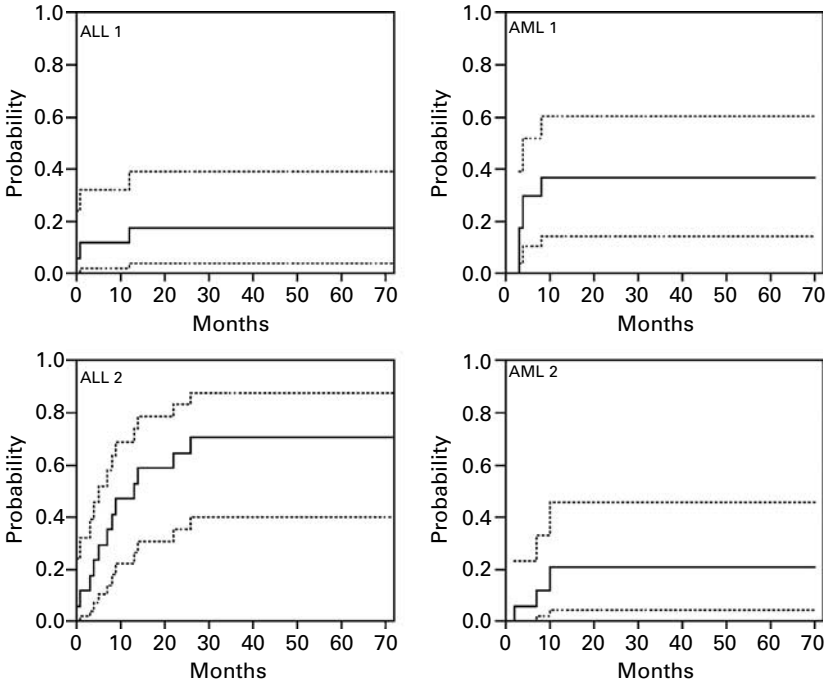

Figure 2 Estimated cumulative incidence curves (solid lines) for each combination of competing event (transplant related mortality/relapse of the original disease) and disease (acute lymphoblastic leukaemia/acute myeloid leukaemia) with 95% pointwise confidence intervals (broken lines).

Final remarks

Before analyzing your own data you might like to perform a competing risks analysis in R with our dataset, so as to confirm results and to practice the instructions we have given you here. The data file, both in CSV and XLS Excel format, used in the example is available at the web address http://www.stat.unipg.it/luca/R.

As you become confident and competent in using the R software, you will be able to exploit all its potential to estimate the effects of covariates in more complicated models of competing risk analysis, as proposed by Fine and Gray.[11]

Acknowledgements

We thank Dr Geraldine Anne Boyd, University of Perugia, for editing this paper.

References

1 Klein JP, Moeschberger ML. *Survival Anal*, 2nd edn. Springer: New York, 2003, 536pp.
2 Pintilie M. *Competing Risks: A Practical Perspective*. John Wiley & Sons: New York, 2006, 240pp.
3 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.
4 Satagopan JM, Ben-Porat L, Berwick M, Robson M, Kutler D, Auerbach AD. A note on competing risks in survival data analysis. *Br J Cancer* 2004; **91**: 1229–1235.
5 Gray RJ. A class of K-sample tests for comparing the cumulative incidence of a competing risk. *Ann Stat* 1988; **16**: 1141–1154.
6 R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing 2006, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.
7 Dalgaard P. *Introductory Statistics with R*. Springer: New York, 2002, 267pp.
8 Iacus SM, Masarotto G. *Laboratorio Di Statistica Con R*. McGraw-Hill: Milano, 2002, 384pp.
9 Venables WN, Ripley BD. *Modern Applied Statistics with S*, 4th edn. Springer: New York, 2002, 495pp.
10 Choudhury JB. Non-parametric confidence interval estimation for competing risks analysis: application to contraceptive data. *Stat Med* 2002; **21**: 1129–1144.
11 Fine JP, Gray RJ. A proportional hazards model for the subdistribution of a competing risk. *J Am Stat Assoc* 1999; **94**: 496–509.
12 Marubini E, Valsecchi MG. *Analysing Survival Data from Clinical Trials and Observational Studies*. Wiley: New York, 1995, 430pp.
13 Pepe MS, Mori M. Kaplan–Meier, marginal or conditional probability curves in summarizing competing risks failure time data? *Stat Med* 1993; **12**: 737–751.
14 Lin DY. Non-parametric inference for cumulative incidence functions in competing risks studies. *Stat Med* 1997; **16**: 901–910.

Appendix A

Competing risk analysis is dedicated to the study of failure probabilities when each individual may fail due to one of several causes, called competing events. The *cumulative incidence function* is

defined as the probability of failing from cause r ($r = 1, \ldots, k$ where k is the number of causes of failure) up to a certain time point t. Formally, it may be written as

$$I_r(t) = \Pr(T \leqslant t, R = r) = \int_0^t \lambda_r(u) S(u) \mathrm{d}u \quad \text{for } r = 1, \ldots, k$$

where $\lambda_r(t)$ is the cause specific hazard rate and $S(t) = \Pr(T \geqslant t)$ is the survival function. Non-parametric MLE of (cause specific) CIF is computed as follows:

$$\hat{I}_r(t) = \sum_{j:t_j \leqslant t} \frac{d_{rj}}{n_j} \hat{S}(t_{j-1}) \quad \text{for } r = 1, \ldots, k$$

where d_{rj} is the number of failures at time t_j from cause r, n_j is the number of individuals at risk at time t_j, and $\hat{S}(t_j)$ is the Kaplan–Meier estimate of the overall survival function. It is interesting to note that $\sum_{r=1}^{k} \hat{I}_r(t) = 1 - \hat{S}(t_j)$, that is the sum of cumulative incidence from all causes is equal to 1 minus the Kaplan–Meier estimate of survival.

Confidence interval estimation can be derived[10] based on the ln(−ln) transformation, so the $(1-\alpha)100\%$ confidence interval for the cumulative incidence function at time t for cause r is given by

$$\hat{I}_r(t)^{\exp\left\{\frac{\pm z_{\alpha/2}\hat{\sigma}_r(t)}{\hat{I}_r(t)\ln \hat{I}_r(t))}\right\}}$$

where $z_{\alpha/2}$ is the upper $\alpha/2$ percentile of the standard normal distribution, and $\sigma_r(t)$ is the square root of the estimated variance of $\hat{I}_r(t)$. This can be calculated as follows (see Marubini and Valsecchi, p 341, eq. 10.12):[12]

$$\begin{aligned}\hat{\mathrm{Var}}\left(\hat{I}_r(t)\right) = &\sum_{j:t_j \leqslant t} \left[\left(\hat{I}_r(t) - \hat{I}_r(t_j)\right)^2 \frac{d_j}{n_j(n_j - d_j)}\right] \\ &+ \sum_{j:t_j \leqslant t} \hat{S}(t_{j-1})^2 \frac{n_j - d_{rj}}{n_j} \frac{d_{rj}}{n_j^2} - 2 \\ &\times \sum_{j:t_j \leqslant t} \left(\hat{I}_r(t) - \hat{I}_r(t_j)\right) \hat{S}(t_{j-1}) \frac{d_{rj}}{n_j^2}\end{aligned}$$

where $d_j = \sum_{r=1}^{k} d_{rj}$.

Finally, comparison of cause-specific CIFs in different groups can be performed using one of the tests proposed, among others, by Gray,[5] Pepe and Mori,[13] and Lin.[14]

Appendix B

```
CumIncidence <- function(ftime, fstatus, group, t, strata, rho = 0,
                         cencode = 0, subset, na.action = na.omit,
                         level, xlab = "Time", ylab = "Probability",
                         digits = 4)
{
  # check for the required package
  if(!require("cmprsk"))
    { stop("Package `cmprsk' is required and must be installed.\n
           See help(install.packages) or write the following command
           at prompt and then follow the instructions:\n
           > install.packages(\"cmprsk\")") }
  # collect data
  mf  <- match.call(expand.dots = FALSE)
  mf[[1]] <- as.name("list")
  mf$t <- mf$digits <- mf$level <- mf$xlab <- mf$ylab <- NULL
  mf <- eval(mf, parent.frame())
  g <- length(unique(mf$group))
  s <- length(unique(mf$fstatus))
  if(missing(t))
    { t <- pretty(c(0, max(mf$ftime)), 6)
      t <- t[t < max(mf$ftime)] }
  # fit model and estimates at time points
  fit   <- do.call("cuminc", mf)
  fit.t <- timepoints(fit, t)
  # print result
  cat("\n+-------------------------------------------------------------------------+")
  cat("\n| Cumulative incidence function estimates from competing
risks data |")
  cat("\n+-------------------------------------------------------------------------+\n")
  tests <- fit$Tests
  colnames(tests) <- c("Statistic", "p-value", "df")
  cat("\nTest equality across groups:\n")
  print(tests, digits = digits)
  cat("\nEstimates at time points:\n")
  print(fit.t$est, digits = digits)
  cat("\nStandard errors:\n")
  print(sqrt(fit.t$var), digits = digits)
  #
  if(missing(level))
    { # plot cumulative incidence functions
      time <- sort(unique(ftime))
      x <- timepoints(fit, time)
      matplot(time, t(x$est), type="s", ylim = c(0,1),
              xlab = xlab, ylab = ylab, xaxs="i", yaxs="i",
              col = rep(1:(s-1), rep(g,(s-1))), lty = rep(1:g, s-1))
      legend("topleft", legend =  rownames(x$est), x.intersp = 2,
             bty = "n", adj = 0.5, col = rep(1:(s-1), rep(g,(s-1))),
             lty = rep(1:g, s-1))
```

```
    out <- list(test = tests, est = fit.t$est, se = sqrt(fit.t$var))
  }
else
  { if(level < 0 | level > 1)
    error("level must be a value in the range [0,1]")
  # compute pointwise confidence intervals
  oldpar <- par(ask=TRUE)
  on.exit(par(oldpar))
  time <- sort(unique(c(ftime, t)))
  x <- timepoints(fit, time)
  z <- qnorm(1-(1-level)/2)
  lower <- x$est ^ exp(-z*sqrt(x$var)/(x$est*log(x$est)))
  upper <- x$est ^ exp(z*sqrt(x$var)/(x$est*log(x$est)))
  # plot pointwise interval confidence intervals
  for(j in 1:nrow(x$est))
     { matplot(time, cbind(x$est[j,], lower[j,], upper[j,]),
               type="s", xlab = xlab, ylab = ylab, xaxs="i",
               yaxs="i", ylim = c(0,1), col = 1, lty = c(1,3,3))
       legend("topleft", legend =  rownames(x$est)[j], bty = "n",adj = 0.5) }
  # print pointwise confidence intervals
  i <- match(t, time)
  ci <- array(NA, c(2, length(i), nrow(lower)))
  ci[1,,] <- t(lower[,i])
  ci[2,,] <- t(upper[,i])
  dimnames(ci) <- list(c("lower", "upper"), t, rownames(lower))
  cat(paste("\n", level*100, "% pointwise confidence intervals:\n\n", sep=""))
  print(ci, digits = digits)
  out <- list(test = tests, est = x$est, se = sqrt(fit.t$var), ci = ci)
  }
# return results
invisible(out)
}
```