COMPUTER METHODS AND PROGRAMS IN BIOMEDICINE 99 (2010) 261–274

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

The mstate package for estimation and prediction in non- and semi-parametric multi-state and competing risks models

Liesbeth C. de Wreede, Marta Fiocco, Hein Putter*

Department of Medical Statistics and Bioinformatics, Leiden University Medical Center, P.O. Box 9600, 2300 RC Leiden, The Netherlands

ARTICLE INFO

Article history:
Received 4 April 2009
Received in revised form
6 November 2009
Accepted 4 January 2010

Keywords:
Survival analysis
Multi-state models
Competing risks models
Markov models
Cox models
Software

ABSTRACT

In recent years, multi-state models have been studied widely in survival analysis. Despite their clear advantages, their use in biomedical and other applications has been rather limited so far. An important reason for this is the lack of flexible and user-friendly software for multi-state models.

This paper introduces a package in R, called 'mstate', for each of the steps of the analysis of multi-state models. It can be applied to non- and semi-parametric models. The package contains functions to facilitate data preparation and flexible estimation of different types of covariate effects in the context of Cox regression models, functions to estimate patient-specific transition intensities, dynamic prediction probabilities and their associated standard errors (both Greenwood and Aalen-type). Competing risks models can also be analyzed by means of mstate, as they are a special type of multi-state models. The package is available from the R homepage http://cran.r-project.org.

We give a self-contained account of the underlying mathematical theory, including a new asymptotic result for the cumulative hazard function and new recursive formulas for the calculation of the estimated standard errors of the estimated transition probabilities, and we illustrate the use of the key functions of the mstate package by the analysis of a reversible multi-state model describing survival of liver cirrhosis patients.

1. Introduction

In recent years, multi-state models have been studied widely as a means to extend classical survival analysis (see Section 2.1 and e.g. [1–3]). Broadly speaking, multi-state models are used for two purposes. Their first aim is to obtain more biological insight into the disease/recovery process of a patient. In particular it is of interest to see how certain prognostic factors (covariates) influence different phases of this process. The second purpose is prediction; they enable clinicians to obtain more accurate predictions of survival duration for e.g. cancer patients than standard models and also to adjust these predictions in the course of time by incorporating intermediate events.

Despite the clear advantages of multi-state models, biomedical and other researchers have not frequently applied them so far. We can partly explain this limited use by the observation that it is difficult to communicate these more detailed models to colleagues. But another reason might be equally important for the disappointing dissemination of multi-state models in the biomedical literature: the lack of flexible and user-friendly software for multi-state models. This software needs to be capable of setting up and restructur-

* Corresponding author. Tel.: +31 715269710.
E-mail address: l.c.de_wreede@lumc.nl (L.C. de Wreede).
0169-2607/$ – see front matter © 2010 Elsevier Ireland Ltd. All rights reserved.
doi:10.1016/j.cmpb.2010.01.001

ing data, estimating covariate effects and hazards, predicting transition probabilities and calculating associated standard errors. Although a number of packages written in R or other languages are available for their analysis, they all have some limitations (Section 3.1).

We have created a software package that can be used for each of these steps of the analysis of multi-state models. This package, written in R, is called `mstate` (Sections 3.2 and 3.3). It can be applied to non- and semi-parametric models. The package contains functions to facilitate data preparation and flexible estimation of covariate effects in the context of Cox regression models, functions to estimate patient-specific transition intensities and dynamic prediction probabilities and their associated standard errors. These calculations involve the implementation of some rather complicated formulas derived by means of martingale techniques. The main formulas will be presented briefly in Sections 2.2 and 2.3. Competing risks models can also be analyzed by means of `mstate`, as they are a special type of multi-state models.

The `mstate` package is designed in such a way that each of its functions can be used independently: i.e., their input does not necessarily come from other functions in the package. Since a multi-state analysis rarely follows a standard path, we believe that the flexibility that this philosophy offers is an important asset of `mstate`. An overview of the philosophy and features of `mstate` is given in Section 3.2.

We shall illustrate the software by means of an analysis of data on liver cirrhosis patients (Section 4).

2. Non- and semi-parametric approaches to multi-state models

2.1. Multi-state models

A multi-state model is a model for time-to-event data in which all individuals start in one or possibly more starting states (e.g. post diagnosis, transplant or surgery) and eventually may end up in one (or more) absorbing or final state(s) (e.g. death or relapse). In between, intermediate states can be visited, possibly more than once. Some individuals are censored before they reach an absorbing state.

Competing risks models are a sub-category of multi-state models: they have one starting state, at least two absorbing states and no intermediate states (see Fig. 1). Their transitions are often indicated as causes of failure.

An example of a multi-state model is illustrated in Fig. 2. This example is often referred to as the 'illness-death model',

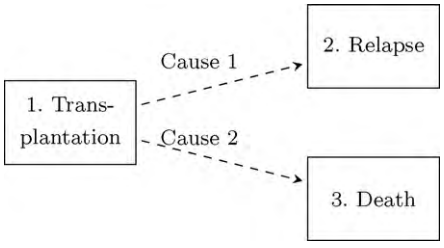

Fig. 1 – A competing risks model for patients with bone marrow transplantation.

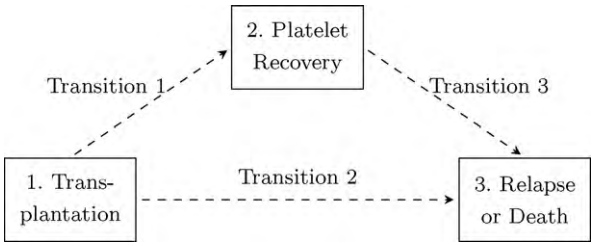

Fig. 2 – A multi-state model for patients with bone marrow transplantation.

and is the simplest true multi-state model. In this case, the so-called 'illness' state denotes platelet recovery, which is a beneficial development. This model will be the leading example in Section 2.

2.2. Notation, basic functions

In this section, we present a summary of the mathematical theory underlying the `mstate` package. Technical details are necessary for the understanding of the working and limitations of the functions in the `mstate` package.

Denote the states in the multi-state model with $\mathbb{S} = \{1, \ldots, S\}$, and let $X(t)$ be a random process taking values in $\mathbb{S}$. A fundamental concept in multi-state models is the transition intensity or hazard rate $\alpha_{gh}(t)$, which expresses the instantaneous risk of a transition from state g into state h at time t. It is defined as

$$\alpha_{gh}(t) := \lim_{\Delta t \to 0} \frac{P(X(t + \Delta t) = h | X(t) = g)}{\Delta t}. \quad (1)$$

The Markov assumption is implicitly present in definition (1). It says that the future depends on the history only through the present. This is formally: $P(X(t + \Delta t) = h | X(t) = g, \{X(s), s < t\}) = P(X(t + \Delta t) = h | X(t) = g)$. The cumulative transition hazard is defined as $A_{gh}(t) = \int_0^t \alpha_{gh}(u)du$. We will assume that there are $K > 0$ possible transitions within the multi-state model. If a direct transition between state g and state h is impossible, $A_{gh}(t) \equiv 0$.

These intensities can be gathered into a $S \times S$-matrix $\mathbf{A}(t)$, with diagonal elements $A_{gg}(t) = -\sum_{h \neq g} A_{gh}(t)$. This last equation expresses that individuals who have no transition remain in state g.

The transition probability matrix $\mathbf{P}(s, t)$ is our prime quantity of interest. It has elements

$$P_{hj}(s, t) = P(X(t) = j | X(s) = h), \quad (2)$$

which denote the transition probability from state h to state j in the time interval $(s, t]$. A single element $P_{hj}(s, t)$ combines both direct and indirect transitions from state h to state j. In Markov models, given $\mathbf{A}(t)$, the transition probability matrix $\mathbf{P}(s, t)$ can be calculated by means of a product integral [4, pp. 93, 288]:

$$\mathbf{P}(s, t) = \Pi_{(s,t]}(\mathbf{I} + d\mathbf{A}(u)). \quad (3)$$

In the case of competing risks models, the cumulative incidence function expresses the probability of failing of cause j before time t. It is given by $P_{1j}(0,t)$. Classical survival models, in which we consider only the transition from being alive to being dead, can be interpreted as two-state models, with $S(t) = 1 - P_{12}(0,t) = P_{11}(0,t)$.

2.3. Estimation and data format

In this subsection, we present standard non-parametric and semi-parametric approaches for multi-state modelling under independent right censoring and possibly left truncation. In the formulas given below, we consider time-continuous Markov models with finite state spaces. The restriction to Markov models implies that the time t refers to the time since the patient entered the initial state (this category is called 'Clock forward'-models). Our notation and ideas are based on [4], in particular Section VII.2.

We provide asymptotic theory. Technical details are given by [4], and an overview of both mathematical and practical issues is e.g. found in [1]. In Section 2.4 we will discuss several examples of models in which the Markov assumption is relaxed.

Let $N_{qi}(t)$ be the counting process for events of type q, where $q = 1, \ldots, Q$ ($Q \leq K$), for subject i ($i = 1, \ldots, n$). Events are of the same 'type' if they have a common baseline hazard. This use of 'type' is a slight extension from the meaning of [4], Sections III.1.3 and VII.1. Since we want to include Cox models in which several transitions share one baseline hazard, we formulate the theory in the more general terms of types instead of in terms of transitions. In the case when each transition has its own baseline hazard ($Q = K$), each type is equivalent to one transition from state g into state h. Terminology and notation then go back to the usual ones, with q replaced by gh. However, we will also consider models in which a type is defined by all transitions into a certain state originating from different starting states (see Section 2.3.2 for more details and examples).

In the implementation of the Cox regression, the 'type' will act as a stratification variable. Thus, for practical purposes we can replace 'type' by 'stratum', because in fact it is not relevant for the analysis whether the strata are determined by covariate values or by groups of transitions, as long as each stratum is determined by its own baseline hazard.

Similarly, $Y_{qi}(t)$ denotes the at-risk process, i.e. $Y_{qi}(t) = 1$ if individual i is at risk for a type q transition at time $t-$, the time point just before time t. Let the aggregated processes $N_q(t) = \sum_{i=1}^{n} N_{qi}(t)$ and $Y_q(t) = \sum_{i=1}^{n} Y_{qi}(t)$ denote respectively the number of events of type q up to and including time t and the size of the risk set for type q at time t, which consists of all subjects who may experience a type q transition at time t.

Usually, data of survival studies are presented in a one-row-per-subject format, which we call 'wide format'. If that is the case, the data have to be recoded into 'long format' to make them suitable for the calculation of the relevant estimators in the multi-state framework. In this format, each subject has as many rows as the number of transitions for which she/he is at risk [1]. An example of data in long format is given in Table 1.

2.3.1. Transition hazards in non-parametric models

In the current subsection, we consider models without covariates. In these models, 'type' and 'transition' are equivalent. Because the risk sets for all transitions from state g are the same, we write $Y_g(t)$ instead of $Y_{gh}(t)$ for the risk set.

In non-parametric models, the hazard of transition $g \to h$ is estimated as follows

$$\Delta\hat{A}_{gh}(t) = dN_{gh}(t)/Y_g(t), \tag{4}$$

where we interpret $dN_{gh}(t)$ as the number of transitions from state g to state h at time t. By summing over all event times up to (and including) time t, we obtain the Nelson-Aalen estimator $\hat{A}_{gh}(t)$ of the cumulative hazard for transition $g \to h$.

In analogy to classical survival analysis, we consider two different calculations of the standard errors of the cumulative transition hazard estimates, the Aalen and the Greenwood estimators (see IV.4.1.3 of [4]). The former estimator is based on the assumption that the (true) cumulative hazard is continuous, whereas the latter allows jumps.

The Aalen estimator of the variance is calculated as

$$\widehat{\mathrm{var}}(\Delta\hat{A}_{gh}(t)) = dN_{gh}(t)/Y_g^2(t), \quad \text{if } g \neq h \tag{5}$$

and

$$\widehat{\mathrm{cov}}(\Delta\hat{A}_{gh}(t), \Delta\hat{A}_{g'h'}(t)) = 0, \quad \text{if } g \neq g' \text{ or } g \neq h \neq h' \text{ if } g = g'. \tag{6}$$

The Greenwood estimator of the variance is given by:

$$\widehat{\widehat{\mathrm{cov}}}(\Delta\hat{A}_{gh}(t), \Delta\hat{A}_{gh'}(t)) = \frac{(\delta_{hh'}Y_g(t) - dN_{gh}(t))dN_{gh'}(t)}{Y_g^3(t)}, \quad \text{if } g \neq h, g \neq h', \tag{7}$$

Table 1 – Data in long format.

id	from	to	trans	Tstart	Tstop	time	status	Z	type	PR	Z.1	Z.2	Z.3	age	T1
1	1	2	1	0	1	1	1	6	1	0	6	0	0	20	0
1	1	3	2	0	1	1	0	6	2	0	0	6	0	20	0
1	2	3	3	1	3	2	1	6	2	1	0	0	6	20	1
2	1	2	1	0	2	2	1	-2	1	0	-2	0	0	43	0
2	1	3	2	0	2	2	0	-2	2	0	0	-2	0	43	0
2	2	3	3	2	4	2	0	-2	2	1	0	0	-2	43	2
3	1	2	1	0	5	5	0	0	1	0	0	0	0	35	0
3	1	3	2	0	5	5	0	0	2	0	0	0	0	35	0

where $\delta_{hh'}$ is 1 if $h = h'$ and 0 otherwise, and

$$\widehat{\mathrm{cov}}(\Delta\hat{A}_{gh}(t), \Delta\hat{A}_{g'h'}(t)) = 0, \quad \text{if } g \neq g'. \tag{8}$$

In both cases the remaining (co)variances are defined by additivity of (co)variances.

The advantage of the Greenwood estimator is that it yields exact multinomial standard errors when no censoring takes place: for instance $\widehat{\mathrm{var}}P_{1j}(t) = P_{1j}(t)(1 - P_{1j}(t))/n$. For large data sets, the difference between these two estimators is small.

2.3.2. *Transition hazards in semi-parametric models, Cox regression*

The Cox model is the most commonly used model in classical survival analysis in the case when covariates are observed. We will consider a type-specific variant of the common Cox model, in which we include a basic (time-fixed for the time being) covariate vector $\mathbf{Z}$:

$$\alpha_q(t|\mathbf{Z}) = \alpha_{q0}(t)\exp(\boldsymbol{\beta}^\top \mathbf{Z}_q(t)), \tag{9}$$

where $\alpha_q(t|\mathbf{Z})$ indicates the hazard for type q ($q = 1, \ldots, Q$) for an individual with basic covariate vector $\mathbf{Z}$, α_{q0} is the baseline hazard for type q, $\boldsymbol{\beta}$ the vector of coefficients for the covariates, and $\mathbf{Z}_q(t)$ a vector of length p of type-specific covariates, possibly time-dependent, derived from $\mathbf{Z}$. How $\mathbf{Z}_q(t)$ is defined from the basic covariates $\mathbf{Z}$ depends on the specific model studied. We shall give a number of examples below.

Formulation (9) is well-known (see Section VII.2 in [4]) and convenient because it unifies several models that are very useful for practical applications and that are all completely covered both by the mathematical theory explained here and by our software. In the current subsection, we will explain these models and their formulation in some detail, and show what their data representation should look like.

The model formulation (9) is intimately related to a data representation in long format. In wide format, suppose the data for the first three patients look as follows, in which `pla` and `death` indicate time of platelet recovery and of death respectively in the case of an event, and time to censoring otherwise; `plastat` and `deathstat` are status indicators of platelet recovery and death respectively (1 in case of an event, 0 if the subject is censored), and `Z` and `age` are covariates at baseline:

	pat	pla	plastat	death	deathstat	Z	age
1	1	1	1	3	1	6	20
2	2	2	1	4	0	−2	43
3	3	5	0	5	0	0	35

For the multi-state model shown in Fig. 2, the data expanded in long format are shown in Table 1. We need one line for each individual for each transition for which he/she is at risk, containing data about her/his identity (`id`), the current transition (`from`, `to`, `trans`), the time of entry in the current state (`Tstart`), the time when he/she stops being at risk for the transition (`Tstop`), either due to an event or to censoring, as indicated by `status`. For the covariates, we need different columns of data for the three models as discussed below. For simplicity, in the examples below we will only illustrate the situation of one basic covariate Z for the multi-state model of Fig. 2 and the example data in Table 1.

Model 1:

We first consider a model in which each type corresponds to one transition ($Q = K$), the covariates are fixed at baseline and they have an identical effect for each transition. Eq. (9) now simplifies to

$$\alpha_q(t|\mathbf{Z}) = \alpha_{q0}(t)\exp(\boldsymbol{\beta}^\top \mathbf{Z}), \tag{10}$$

which is equal to $\alpha_{q0}(t)\exp(\beta Z)$ in the case of one covariate. In this case the covariate in $\exp(\beta Z)$ is time-fixed. In the data example of Table 1, covariate `Z` has the same value for all transitions. The model is fitted using a Cox proportional hazards model, stratified by transition, with a single, common, effect of the covariate Z. The data representation in long format enables to select only those rows that describe the data for the relevant transition.

Model 2:

We now consider a model in which again each type corresponds to one transition ($Q = K$) and the covariates are fixed at baseline, but now type-specific covariates are included:

$$\alpha_q(t|\mathbf{Z}) = \alpha_{q0}(t)\exp(\boldsymbol{\beta}^\top \mathbf{Z}_q). \tag{11}$$

In this case, we call $\mathbf{Z}_q$ a transition-specific covariate. Because transition-specific covariates are a special kind of type-specific covariates, the theory developed below in terms of type-specific covariates also covers the situation of transition-specific covariates.

Eq. (11) describes a stratified Cox model, in which each stratum represents one transition or type. Although the basic covariates $\mathbf{Z}$ have no common effect on all transitions, the transition-specific covariates $\mathbf{Z}_q$ do have common effects because their coefficients $\boldsymbol{\beta}$ are the same for all transitions.

If we consider just one basic covariate Z, the three transition-specific covariate vectors $\mathbf{Z}_1$, $\mathbf{Z}_2$, $\mathbf{Z}_3$ are defined as $\mathbf{Z}_1 = (Z, 0, 0)^\top$, $\mathbf{Z}_2 = (0, Z, 0)^\top$, $\mathbf{Z}_3 = (0, 0, Z)^\top$ and the regression vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3)^\top$ has length 3. With these definitions, (11) implies for the first transition intensity:

$$\alpha_1(t|Z) = \alpha_{10}(t)\exp(\beta^\top \mathbf{Z}_1) = \alpha_{10}(t)\exp(\beta_1 Z),$$

and similarly, $\alpha_2(t|Z) = \alpha_{20}(t)\exp(\beta_2 Z)$ and $\alpha_3(t|Z) = \alpha_{30}(t)\exp(\beta_3 Z)$. So we indeed have different effects for Z in transitions 1, 2, and 3, with regression coefficients β_1, β_2, and β_3, respectively.

The general model given by (11) is equivalent to one in which we perform Cox regressions for each of the transitions $g \to h$ separately [4, p. 478]:

$$\alpha_{gh}(t|\mathbf{Z}) = \alpha_{gh,0}(t)\exp(\boldsymbol{\beta}_{gh}^\top \mathbf{Z}), \tag{12}$$

where $\boldsymbol{\beta}_{gh}$ is the vector of regression coefficients corresponding to the transition from state g into state h. The vector $\boldsymbol{\beta}$ in (11) consists of the stacked transition-specific coefficients $\boldsymbol{\beta}_{gh}$.

Model 2 requires the same data as Model 1, apart from the covariates (see Table 1). We now need transition-specific covariates `Z.1`, `Z.2` and `Z.3` (`age` may be expanded analogously). Consider for instance `Z.1` for person 1. It has value 6 for transition 1, and 0 for the other transitions. The model

is fitted with a single Cox model, stratified by transition, and `Z.1`, `Z.2`, and `Z.3` as covariates.

Model 3:

Again the covariates are fixed at baseline and they have a different effect for each transition. An important additional assumption that is often made is that some of the baseline hazards of model (9) are proportional, for instance if transitions $h \to j$ and $k \to l$ are proportional,

$$\alpha_{hj,0}(t) = \tilde{\delta}\alpha_{kl,0}(t)$$

(see e.g. [1, pp. 2418–2421]). We call these 'proportional baseline hazards models'. Together, the transitions sharing the same baseline hazard define one type, which means that 'transition' and 'type' are not equivalent in this model (so we may have $Q < K$).

It is often assumed that transitions into the same state are proportional, e.g. in our example the two transitions into the death state. This case can be modelled by means of a time-dependent covariate $\tilde{Z}(t)$ in the regression model (9). Within a given type, this covariate distinguishes between different transitions into the same state: $\tilde{Z}(t)$ is 0 if the patient has not yet experienced a certain intermediate event (in our example, platelet recovery) and 1 afterwards. The proportionality is expressed by the coefficient $\tilde{\beta}$ of $\tilde{Z}(t)$: $\exp(\tilde{\beta}) = \tilde{\delta}$. $\tilde{Z}(t)$ is a special kind of endogenous or internal covariate that serves only to distinguish between transitions within a type (see Section 5.3 of [5] for the theory of different kinds of time-dependent covariates in survival analysis). Although prediction is in general no longer possible if endogenous covariates are used, this problem does not arise for this restricted class.

For the purpose of estimation and prediction transition-specific covariates are more suitable than type-specific covariates, also in this setting. For this reason, this model mixes transition-specific covariates with type-specific baseline hazards. Fortunately, transition-specific covariates can be equivalently represented as a certain type of time-dependent type-specific covariates; they are piecewise constant functions, changing over time when a transition occurs, i.e. when the patient becomes at risk for another transition within the same type. For this reason Model 3 can also be described by Eq. (9).

In our example, we assume that transitions 2 and 3 have the same baseline transition hazards. We thus have two types: the first transition belongs to type 1, and both transitions 2 and 3 belong to type 2 and share a common baseline hazard. We now need two type-specific covariate vectors $\mathbf{Z}_1(t)$ and $\mathbf{Z}_2(t)$. $\mathbf{Z}_1(t)$ is defined as $\mathbf{Z}_1 = (Z, 0, 0, 0)^\top$ for all t, and $\mathbf{Z}_2(t)$ either as $(0, Z, 0, 0)^\top$ for transition 2, or as $(0, 0, Z, 1)^\top$ for transition 3; the regression vector $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, \tilde{\beta})^\top$ has length 4. Note that $\mathbf{Z}_2(t)$ is now really time-dependent; if t_{PR} is the time of platelet recovery, then $\mathbf{Z}_2(t) = (0, Z, 0, 0)$ for $t \le t_{PR}$, and $\mathbf{Z}_2(t) = (0, 0, Z, 1)^\top$ for $t > t_{PR}$.

With these definitions, (9) implies for the first transition intensity:

$$\alpha_1(t|\mathbf{Z}) = \alpha_{10}(t)\exp(\boldsymbol{\beta}^\top \mathbf{Z}_1) = \alpha_{10}(t)\exp(\beta_1 Z),$$

and similarly, $\alpha_2(t|\mathbf{Z}) = \alpha_{20}(t)\exp(\beta_2 Z)$ for transition 2 and $\alpha_3(t|\mathbf{Z}) = \alpha_{20}(t)\exp(\beta_3 Z + \tilde{\beta}) = \tilde{\delta}\alpha_{20}(t)\exp(\beta_3 Z)$ for transition 3. Again we have different covariate effects for transitions 1, 2, and 3, with regression coefficients β_1, β_2, and β_3, respectively, and the hazard ratio between the baseline transition intensities of transitions 2 and 3 is given by $\exp(\tilde{\beta})$.

Apart from `Z.1`, `Z.2`, and `Z.3`, we now also need to include the time-dependent covariate $\tilde{Z}(t)$. This new covariate, `PR` in our data, is the fourth element in our type-specific covariate vectors. The model can be fitted with a Cox model, stratified by `type`, with `Z.1`, `Z.2`, `Z.3` and `PR` as covariates.

Not only models 1–3, but also any mixture of identical or different covariate effects across transitions and of stratified and proportional baseline hazards can be modelled by Eq. (9). We can for instance consider a model in which age has the same effect for all transitions and Z a different effect for each transition; the analysis of the data of Table 1 is then done on the basis of the covariates `age`, `Z.1`, `Z.2` and `Z.3`. It is also possible to stratify the data according to covariate values and add extra types according to this stratification, e.g. transition 2 for males and females separately. The advantage of Eq. (9) is that it is now possible to assume that (some) covariate effects are identical across (some) transitions and to test such an assumption.

By setting up the data in the way described above, standard statistical software can be used to estimate the regression coefficients and the baseline hazards, as long as delayed entry is allowed. We have found the approach of combining long data format, stratified Cox regression and transition/type-specific covariates very useful in many applications (see e.g. [6–8]).

In principle, other kinds of time-dependent covariates can also be considered if the data structure is adjusted. The estimation of regression coefficients and baseline hazards is still possible, and also the calculation of patient-specific transition hazards, provided the value of the covariates is fixed beforehand. However, in practical applications, prediction based on most types of time-dependent covariates gives problems of estimation and interpretation (see e.g. [9]).

Asymptotic results

Consider now again model (9) and define

$$S_q^{(0)}(\boldsymbol{\beta}, t) = \sum_{i=1}^{n} \exp(\boldsymbol{\beta}^\top \mathbf{Z}_{qi}(t))Y_{qi}(t),$$

$$\mathbf{S}_q^{(1)}(\boldsymbol{\beta}, t) = \sum_{i=1}^{n} \mathbf{Z}_{qi}(t)\exp(\boldsymbol{\beta}^\top \mathbf{Z}_{qi}(t))Y_{qi}(t),$$

$$\mathbf{S}_q^{(2)}(\boldsymbol{\beta}, t) = \sum_{i=1}^{n} \mathbf{Z}_{qi}(t)\mathbf{Z}_{qi}^\top(t)\exp(\boldsymbol{\beta}^\top \mathbf{Z}_{qi}(t))Y_{qi}(t),$$

scalar, p-vector, and $p \times p$ matrix, respectively. The term $S_q^{(0)}$ represents the weighted risk set.

Define also the p-vector and $p \times p$ matrix

$$E_q(\boldsymbol{\beta}, t) = \frac{\mathbf{S}_q^{(1)}(\boldsymbol{\beta}, t)}{S_q^{(0)}(\boldsymbol{\beta}, t)},$$

$$V_q(\boldsymbol{\beta}, t) = \frac{\mathbf{S}_q^{(2)}(\boldsymbol{\beta}, t)}{S_q^{(0)}(\boldsymbol{\beta}, t)} - E_q(\boldsymbol{\beta}, t)E_q^\top(\boldsymbol{\beta}, t).$$

These quantities can be seen as the mean and variance matrix respectively of $\mathbf{Z}_{qi}(t)$ under a particular probability distribution where individual i is selected for an event of type q with a probability proportional to $\exp(\boldsymbol{\beta}^\top \mathbf{Z}_{qi}(t))Y_{qi}(t)$, see [4, p. 486].

The regression coefficient $\boldsymbol{\beta}$ is estimated as the value that maximizes the partial log-likelihood

$$\sum_{q=1}^{Q}\sum_{i=1}^{n}\left[\int_0^\infty \boldsymbol{\beta}^\top \mathbf{Z}_{qi}(t)\mathrm{d}N_{qi}(t) - \int_0^\infty \log S_q^{(0)}(\boldsymbol{\beta}, t)\mathrm{d}N_{qi}(t)\right].$$

The baseline hazard of type q, denoted by $\alpha_{q0}(t, \boldsymbol{\beta})$, is estimated by:

$$\Delta\hat{A}_{q0}(t, \hat{\boldsymbol{\beta}}) = \frac{\Delta N_q(t)}{S_q^{(0)}(\hat{\boldsymbol{\beta}}, t)}, \tag{13}$$

where $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimator of $\boldsymbol{\beta}$. Eq. (13) is called the Breslow estimator. It is the weighted version of Eq. (4). The estimator of the cumulative baseline hazard $\hat{A}_{q0}(t, \hat{\boldsymbol{\beta}})$ of type q is the sum over the event times $u \le t$ of type q of the estimated hazard functions of Eq. (13). The dependence of $\hat{A}_{q0}(t)$ on $\hat{\boldsymbol{\beta}}$ will now be omitted in the notation. The Fisher information matrix $\mathcal{I}(\boldsymbol{\beta})$ is given by

$$\mathcal{I}(\boldsymbol{\beta}) = \sum_{q=1}^{Q}\int_0^t V_q(\boldsymbol{\beta}, t)\mathrm{d}N_q(t).$$

Our ultimate aim is to obtain an estimate and associated standard errors of the transition probability matrix $\mathbf{P}(s, t|\mathbf{Z}_0)$ for all models discussed above, where $\mathbf{Z}_0$ denotes the basic covariate vector for the person for whom we give a prediction. For this we need not only the variances but also the covariances of the estimated cumulative hazards of all transitions. In model (9), on which these estimates are based, the transitions $g \to h$ and $g' \to h'$ can be of the same or of different types. Furthermore, because of the use of transition-specific covariates and possibly proportionality constants for different transitions of the same type, the basic covariate vector $\mathbf{Z}_0$ may result in different expanded covariates for different transitions within the same type. Let $g \to h$ and $g' \to h'$ be transitions of type q and q' respectively (q may be equal to q'). Define $\hat{A}_{gh}(t|\mathbf{Z}_0) = \hat{A}_{q0}(t)\exp(\hat{\boldsymbol{\beta}}^\top \mathbf{Z}^*) = \hat{A}_q^*(t)$, where $\mathbf{Z}^*$ is the expanded covariate for transition $g \to h$, and $\hat{A}_{g'h'}(t|\mathbf{Z}_0) = \hat{A}_{q'0}(t)\exp(\hat{\boldsymbol{\beta}}^\top \mathbf{Z}^\dagger) = \hat{A}_{q'}^\dagger(t)$, where $\mathbf{Z}^\dagger$ is the expanded covariate for transition $g' \to h'$. This new notation is more convenient in the context of prediction.

The following proposition on the asymptotic properties of the estimator of the cumulative hazard function in semi-parametric models with possibly related baseline hazards is a new extension of the results in Section VII.2 of [4]:

Proposition 1. *Define the true baseline hazards for types q and q' as $A_{q0}(t)$ and $A_{q'0}(t)$ respectively. Under the same conditions as in Theorem* VII.2.3 *of* [4], *the process*

$$\sqrt{n}\left(\exp(\hat{\boldsymbol{\beta}}^\top \mathbf{Z}^*)\hat{A}_{q0}(t) - \exp(\boldsymbol{\beta}^\top \mathbf{Z}^*)A_{q0}(t),\right.$$
$$\left.\exp(\hat{\boldsymbol{\beta}}^\top \mathbf{Z}^\dagger)\hat{A}_{q'0}(t) - \exp(\boldsymbol{\beta}^\top \mathbf{Z}^\dagger)A_{q'0}(t)\right)$$

converges weakly to a bivariate Gaussian process with mean zero and a covariance function which can be estimated uniformly consistently by

$$n\left\{\delta_{qq'}\int_0^\infty \frac{\exp(\hat{\boldsymbol{\beta}}^\top \mathbf{Z}^*)}{S_q^{(0)}(\hat{\boldsymbol{\beta}}, u)}\frac{\exp(\hat{\boldsymbol{\beta}}^\top \mathbf{Z}^\dagger)}{S_q^{(0)}(\hat{\boldsymbol{\beta}}, u)}\mathrm{d}N_q(u)\right.$$
$$+\int_0^\infty \left(E_q(\hat{\boldsymbol{\beta}}, u) - \mathbf{Z}^*\right)^\top \mathrm{d}A_q(u)\cdot\mathcal{I}(\hat{\boldsymbol{\beta}})^{-1}\cdot$$
$$\left.\int_0^\infty \left(E_{q'}(\hat{\boldsymbol{\beta}}, u) - \mathbf{Z}^\dagger\right)\mathrm{d}A'_q(u)\right\}, \tag{14}$$

in which $\delta_{qq'} = 1$ if $q = q'$ and 0 otherwise.

Proof. The proposition follows from Theorem VII.2.3 of [4], along the same lines as their Corollaries VII.2.4, VII.2.5, and VII.2.6. The only difference is that the covariance matrix of $(\hat{A}_q^*(t), \hat{A}_{q'}^\dagger(t))$ simultaneously is considered, which implies that the functional delta-method is applied to the triplet $(\hat{\boldsymbol{\beta}}, \hat{A}_{q0}(\cdot), \hat{A}_{q'0}(\cdot))$. □

This approximation takes into account the impact of two sources of error in the estimation of the cumulative hazards in the presence of covariates: the estimation of the coefficients $\boldsymbol{\beta}$, and the estimation of the cumulative baseline hazards. Proposition 1 implies that the covariance of $\hat{A}_q^*(t)$ and $\hat{A}_{q'}^\dagger(t)$ may be estimated by Eq. (14) without the factor n. A special case of the proposition yields the variance of $\hat{A}_q^*(t)$, which is also described in Corollary VII.2.6 of [4], see also equation (10.4) in [10, p. 267].

In accordance with current practice, we do not extend the Greenwood estimator of the variance of the cumulative hazard (7)–(8) to the covariate case. In particular, since proportional baseline hazards models need endogenous covariates for their description, the Greenwood approach is not suitable for them.

2.3.3. *Estimation of transition probabilities and their standard errors*

In this section, we describe how the estimators of the transition probability matrix $\hat{\mathbf{P}}(s, t)$ and of its standard errors depend on the estimators of the hazard matrices $\hat{\mathbf{A}}(t)$. This relation does not depend on the presence or absence of covariates, nor on how $\mathbf{A}$ is related to $\mathbf{Z}$, and holds also in the case of proportional baseline hazards models. The necessary ingredients for the calculation of the estimated transition probability matrix and its standard errors are just the estimates of the transition hazards $\hat{\mathbf{A}}(t)$ and of their variance-covariance matrices $\widehat{\mathrm{var}}(\hat{\mathbf{A}}(t))$, irrespective of how these were obtained. Therefore, in the remainder of this section, we suppress the covariates in the notation for simplicity and we consider only the matrix of the hazards $\mathbf{A}$ and no longer the A_{hj}'s separately.

When our software is applied, the cumulative hazard functions and their variances can either be calculated by means of `mstate` in the case of standard models, or by means of other functions (either available elsewhere or user-defined) in the case of alternative models. In either case, the calculation of the transition probability matrix and its associated variance-covariance matrix remains the same as long as the models are Markovian, and can be performed by `mstate`.

Since $\hat{\mathbf{A}}$ itself is a matrix, its estimated covariance matrix is defined as $\widehat{\text{var}}(\text{vec}(\hat{\mathbf{A}}))$, where $\text{vec}(\hat{\mathbf{A}})$ is defined as the vectorized ($S^2 \times 1$) matrix of $\mathbf{A}$ where the columns are stacked on top of each other. The dimension of $\widehat{\text{var}}(\hat{\mathbf{A}}(t))$ is $S^2 \times S^2$.

We estimate the transition matrix as

$$\hat{\mathbf{P}}(s,t) = \Pi_{u\in(s,t]}(\mathbf{I} + \Delta\hat{\mathbf{A}}(u)), \tag{15}$$

in which u indicates all event times in $(s,t]$. We call (15) the Aalen–Johansen-type estimator (the classical Aalen–Johansen estimator has no covariates, see [4, p. 288]). Note that the transition probability matrix in (3) is calculated by means of a product integral, while its estimator in (15) is based on a finite product, which only changes at event times.

We distinguish between two kinds of prediction: forward and fixed horizon. 'Forward' means that we give a prognosis for the future on the basis of the current time and history. In other words: in $\mathbf{P}(s,t)$, time s remains fixed, while time t varies. In the 'fixed horizon' option, the prediction is made from several starting points to one future time point. Now time t remains fixed and time s varies in $\mathbf{P}(s,t)$. From a computational point of view, the difference is in the order of multiplication of the matrix terms $(\mathbf{I} + \Delta\hat{\mathbf{A}}(u))$: in the forward-case, the terms are right-multiplied in ascending time order, in the fixed horizon-case, left-multiplied in descending time order.

The estimated standard errors of the estimators can for instance be used to construct pointwise confidence intervals around the estimated transition probability curves. They are the square roots of the diagonal elements of the estimated variance-covariance matrix

$$\widehat{\text{var}}(\hat{\mathbf{P}}(s,t)). \tag{16}$$

The entry $\widehat{\text{var}}(\hat{P}_{hj}, \hat{P}_{lm})$ of this $S^2 \times S^2$-matrix has coordinates $(S(j-1)+h, S(m-1)+l)$.

The variance-covariance matrix of (15) is again estimated by means of the delta-method. Estimator (16) can be calculated in two ways: directly and through a recursion formula. There are two types, Aalen and Greenwood, depending on which estimator of the (co)variances of the cumulative hazard functions is used. The Aalen-type is calculated directly as (cp. [4, p. 292]):

$$\widehat{\text{var}}(\hat{\mathbf{P}}(s,t)) = \sum_{u\in(s,t]} \{\hat{\mathbf{P}}(u,t)^\top \otimes \hat{\mathbf{P}}(s,u)\}\, \widehat{\text{var}}(\Delta\widehat{\mathbf{A}}(u)) \cdot \{\hat{\mathbf{P}}(u,t) \otimes \hat{\mathbf{P}}(s,u)^\top\}, \tag{17}$$

where $\cdot \otimes \cdot$ denotes the Kronecker product. This is a finite sum, because it has only non-zero contributions at event times. For the elements of $\widehat{\text{var}}(\Delta\widehat{\mathbf{A}}(u))$, see Eq. (5)–(6) (without covariates) and (14) (with covariates).

The Greenwood-variant of the variances is calculated as (see [4, p. 294]):

$$\widehat{\widehat{\text{var}}}(\hat{\mathbf{P}}(s,t)) = \sum_{u\in(s,t]} \{\hat{\mathbf{P}}(u,t)^\top \otimes \hat{\mathbf{P}}(s,u-)\}\, \widehat{\widehat{\text{var}}}(\Delta\widehat{\mathbf{A}}(u)) \cdot \{\hat{\mathbf{P}}(u,t) \otimes \hat{\mathbf{P}}(s,u-)^\top\}, \tag{18}$$

with $\widehat{\widehat{\text{var}}}(\Delta\widehat{\mathbf{A}}(u))$ from Eqs. (7)–(8), and $t-$ the last event time before t. Note the two differences between the Greenwood and Aalen estimator: first, the potential discontinuity of the cumulative hazard in the Greenwood case implies the use of $\hat{P}_{hj}(s,u-)$ instead of $\hat{P}_{hj}(s,u)$ in the estimation; second, the estimators of the variance-covariance matrix of the cumulative hazards are different.

We have implemented the calculation of the standard errors by means of recursion formulas. The recursion formula to calculate the standard errors of **Greenwood-type** for **forward** prediction is derived in [4, p. 295] (formula (4.4.19); cp. Eq. (18)):

$$\widehat{\widehat{\text{var}}}(\hat{\mathbf{P}}(s,t)) = \{(I+\Delta\hat{\mathbf{A}}(t))^\top \otimes I\}\, \widehat{\widehat{\text{var}}}\,(\hat{\mathbf{P}}(s,t-))\{(I+\Delta\hat{\mathbf{A}}(t)) \otimes I\} + \{I \otimes \hat{\mathbf{P}}(s,t-)\}\, \widehat{\widehat{\text{var}}}(\Delta\hat{\mathbf{A}}(t))\, \{I \otimes \hat{\mathbf{P}}(s,t-)^\top\}, \tag{19}$$

where $t-$ indicates the last event time before t.

In analogy to this equation, we developed recursion formulas for the calculation of standard errors of Greenwood type, fixed horizon prediction, and of Aalen-type, forward and fixed horizon prediction. The **fixed horizon** version of the recursion formula for **Greenwood-type** is:

$$\widehat{\widehat{\text{var}}}(\hat{\mathbf{P}}(s,t)) = \{I \otimes (I+\Delta\hat{\mathbf{A}}(s+))\}\widehat{\widehat{\text{var}}}(\hat{\mathbf{P}}(s+,t)) \cdot \{I \otimes (I+\Delta\hat{\mathbf{A}}(s+))^\top\} + \{\hat{\mathbf{P}}(s+,t)^\top \otimes I\}\widehat{\widehat{\text{var}}}(\Delta\hat{\mathbf{A}}(s+)) \cdot \{\hat{\mathbf{P}}(s+,t) \otimes I\}, \tag{20}$$

where $s+$ is the first event time after s.

For **forward** prediction, **Aalen-type** standard errors, we obtain (cp. Eq. (17)):

$$\widehat{\text{var}}(\hat{\mathbf{P}}(s,t)) = \{(I+\Delta\hat{\mathbf{A}}(t))^\top \otimes I\}\, \widehat{\text{var}}\,(\hat{\mathbf{P}}(s,t-))\{(I+\Delta\hat{\mathbf{A}}(t)) \otimes I\} + \{I \otimes \hat{\mathbf{P}}(s,t)\}\, \widehat{\text{var}}(\Delta\hat{\mathbf{A}}(t))\, \{I \otimes \hat{\mathbf{P}}(s,t)^\top\}. \tag{21}$$

The **Aalen-type** standard errors of the **fixed horizon** probabilities are calculated as

$$\widehat{\text{var}}(\hat{\mathbf{P}}(s,t)) = \{I \otimes (I+\Delta\hat{\mathbf{A}}(s+))\}\, \widehat{\text{var}}\,(\hat{\mathbf{P}}(s+,t)) \cdot \{I \otimes (I+\Delta\hat{\mathbf{A}}(s+))^\top\} + \{\hat{\mathbf{P}}(s+,t)^\top \otimes (I+\Delta\hat{\mathbf{A}}(s+))\}\, \widehat{\text{var}}(\Delta\hat{\mathbf{A}}(s+)) \cdot \{\hat{\mathbf{P}}(s+,t) \otimes (I+\Delta\hat{\mathbf{A}}(s+))^\top\}. \tag{22}$$

In Eqs. (19)–(22), the starting value is $\widehat{\text{var}}(\hat{\mathbf{P}}(s,s), \hat{\mathbf{P}}(s,s)) = \widehat{\widehat{\text{var}}}(\hat{\mathbf{P}}(s,s), \hat{\mathbf{P}}(s,s)) = \mathbf{0}$ for forward prediction; for fixed horizon prediction, replace (s,s) by (t,t).

2.4. *Non-Markov models*

We consider several relaxations of the Markovian assumption of the previous subsections, and the way in which they fit in the theory above and can be handled by our software. First consider 'clock reset'-models, where the time t refers to the time spent in the current state (thus, the clock is 'reset' to 0 each time a patient enters a new state). These are semi-Markov models. Another relaxation is to include the time of entry in a previous or the current state as a time-dependent covariate in the model, leading to the category of what are called state arrival extended Markov models in [1]. With a

suitable reparametrization, these latter models can also be used to include sojourn time in previous states as a time-dependent covariate. Prediction is problematic in all these models, because the covariate history of the patient is not yet given at baseline, but only becomes known gradually. As a result, the value of the elements of $\hat{\mathbf{A}}(t)$ is not yet known in advance and (15) cannot be calculated directly.

The set-up of the data is largely the same as in the Markov models discussed before. In the case of clock reset models, we need a new time scale starting at zero when a new state is entered. In Table 1, this is `time`, equal to `Tstop-Tstart`. In the case of state arrival extended Markov models, a covariate describing time of entry in a state (in this case state 2, platelet recovery) is added (`T1` in Table 1). A regression model containing these new variables can again be handled by standard software.

For these models a simulation approach adapted to the context of survival data and multi-state models can be applied to estimate the transition probabilities and their standard errors. For the purpose of simulation, a model-based resampling method based on repeatedly sampling complete paths through the multi-state model is also implemented in the `mstate` library. Details concerning the algorithm used to generate such a path based on given cumulative hazard functions specified for each of the direct transitions can be found in [6].

Moreover a bootstrap procedure has also been implemented in the `mstate` library. This procedure is also described in detail in [6]. It can be applied to any general statistic of interest, not only to the estimation of the prediction probabilities or their standard errors. Although the bootstrap method implemented in the library yields reliable results, its disadvantage is that it is very computer-intensive and time-consuming. In [6] we explored the possibility of speeding up the computations by sampling a smaller number of multi-state trajectories and adjusting the simulation error afterwards.

3. Software

3.1. A short overview of the existing software

Some software for the analysis of multi-state models, written in different languages, is already available. When compared to our package, all of it has some limitations. We discuss first the software in other languages than `R`, and then the `R` packages. The programming language `R` has two major advantages over other languages: it is freely available, and an increasing number of statistically relevant packages has been developed in it (for an overview of survival packages see *CRAN Task View: Survival Analysis* in http://cran.r-project.org/web/views/Survival.html).

In SAS, two macros written by Rosthøj, Andersen and Abildstrom are available to calculate cumulative incidence functions and their associated standard errors in competing risks models [11]. They cannot be used in general multi-state models. We have tested our software with the help of theirs. MKVPCI, a program created by Alioum and Commenges in FORTRAN-77, fits multi-state Markov models, but only with piecewise constant intensities and covariates [12]. Their program is an extension of MARKOV, written by Marshall, Guo and Jones in FORTRAN, and designed for the analysis of Markov models with constant transition intensities [13].

The remaining software comes in the form of packages for `R`. `cmprsk` by Gray can analyze competing risks models, and implements Gray's test and Fine & Gray regression models, which is not yet possible in `mstate`. Standard errors of the estimated hazards and cumulative incidence functions are not available. Khomski contributed `CompetingRiskFrailty` for the analysis of competing risks models with frailties; however no standard errors are calculated. `changeLOS` by Wangler, Beyersmann and Schumacher is designed for computing the change in length of hospital stay by means of a multi-state model. The model does not include covariates and standard errors of the prediction probabilities are only calculated by means of a bootstrap procedure [14]. The package `tdc.msm` by Meira-Machado, Cadarso-Suárez and Uña-Álvarez covers a restricted category of multi-state models, namely the extended illness-death models [15]. Moreover, standard errors of the transition probabilities have not been implemented. Carstensen et al included some multi-state software in `Epi`. Allignol, Beyersmann and Schumacher have created two relevant packages for non-parametric estimation in multi-state models: `mvna` calculates the Nelson-Aalen estimator of the cumulative hazard and its standard errors in multi-state models without covariates, and `etm` calculates the accompanying Aalen–Johansen estimator of the transition probability matrix and its variance-covariance matrix (only Greenwood-type) [16].

All `R` packages discussed above are primarily meant for the analysis of non- and semi-parametric models, as is `mstate`. A fundamentally different approach was chosen by Christopher Jackson, the author of the `msm` package in `R`, which has been developed to analyze multi-state Markov and hidden Markov models [17]. His software is designed for parametric models: the transition hazards are (piecewise) constant functions, as in MKVPCI. The parametric assumptions make it possible to handle a number of features that are much harder or impossible to deal with in the context of non- or semi-parametric models, such as interval censoring or the possibility of misclassification. However, in some cases these assumptions may be a serious restriction (see [2]).

The limitations described above have been overcome in `mstate`, the first package that deals with a broad range of multi-state and competing risks models, both without and with covariates. Moreover, the estimated standard errors of all relevant estimators can be computed in different ways.

3.2. Philosophy and features of the `mstate` package

The package `mstate` has been designed with the purpose of giving the users maximum flexibility in fitting their models. It contains both functions that help users to rearrange their data into a format suitable for multi-state analysis, and functions to analyze them, and it can easily be combined with other `R` functions, for example for model selection and plotting. It covers models without and with covariates: in the former case, the models are non-parametric (see Section 2.3.1) and in the latter case, the models considered are semi-parametric (see Section 2.3.2). The most prominent example of a semi-parametric model is the stratified Cox regression model where

the regression coefficients represent the parametric part and the baseline hazards the non-parametric part. In the analysis we have included both models in which each transition has its own baseline hazard and proportional baseline hazards models.

The user determines whether the effect of covariates differs for all transitions (transition-specific covariates), or is equal for some. In the case of categorical time-dependent covariates with a limited number of changes (e.g. number of infections), their predictive performance can be measured by adding to the model an extra state for each new value of the covariate.

A major goal of the package is to enable dynamic prediction, where an initial prediction may be updated by the incorporation of information about the arrival in other states at later time points. This helps to answer the clinical question: how to give a prognosis for a given patient with known covariate values at baseline, and how to update this prediction if intermediate events take place?

The flexibility of `mstate` requires that users do some programming in `R` themselves, for which purpose they need some knowledge of the `survival` package (see e.g. [10] or [18]). Yet the advantage of our approach is that they can easily experiment with many different models to study the same data, for instance models with more or less states, or models with more or less covariates, or with equal or different covariate effects across transitions. Moreover, users can also study models with a different structure than that described in Section 2, for instance models in which the same state can be visited more than once or additive hazard models. In these cases, they can prepare their own data and then apply several of the functions of the `mstate` package, without having to write down and calculate explicit formulas for the different transition probabilities and their standard errors. Another flexible feature of the package is the fact that it offers exact calculations of most relevant quantities in the models presented above, and simulation-based computations for other cases.

3.3. *The functions of* `mstate`

If a competing risks model without covariates is considered, the cumulative incidence functions and their standard errors can be calculated by means of `Cuminc`, possibly stratified by a grouping variable.

Otherwise, a multi-state analysis starts with the description of the states and transitions between states in matrix format. For competing risks models and the illness-death model, this can be done by means of the functions `trans.comprisk` and `trans.illdeath`; for all other models, the transition matrix can be defined by means of the function `transMat`, contributed by Steven McKinney. The function `paths` gives an overview of all possible trajectories through the model on the basis of this matrix.

The next step is data preparation. The function `msprep` rearranges the data from wide format into long format. The user can select covariates to be replicated in the new format. These covariates (or some of them) can then be expanded into transition-specific covariates, which are added to the data set, by means of `expand.covs`.

The total numbers of transitions are given by the function `events`, both in absolute numbers and in percentages. Its input is a data set in long format, either created by means of `msprep` or otherwise.

A stratified Cox model can now be fit by means of the function `coxph` from the `survival` package, both for models with and without covariates. This fitted model is the input of `msfit`, a function calculating person-specific transition hazards and associated standard errors at all event times. In the case of semi-parametric models, the values of the covariates at baseline of the person under consideration have to be included in the function call as well.

The function `msfit` yields the input for `probtrans`, which function calculates transition probabilities from different starting states and time points, thus enabling dynamic prediction. `probtrans` gives extensive numerical output, which can be studied *per se* or be used as the basis for graphs. It also calculates the associated standard errors. In models without covariates, they can be calculated either by the Aalen or by the Greenwood estimator. For models with covariates, only the Aalen-option is available. Both estimators can in principle be calculated in two ways: directly and through a recursion formula (see Section 2.3.3). For computational speed, we have chosen to implement the recursion formulas. Whenever appropriate, we have borrowed ingredients from the `survival` package. The `probtrans` function can calculate all (co)variances of the transition probabilities, but the user decides to have as output only transition probabilities, or also standard errors or all covariance matrices. The standard errors can be used to construct pointwise confidence intervals for the estimated transition probability curves.

In standard cases `msfit` and `probtrans` suffice to calculate all transition probabilities and their associated standard errors on the basis of the data and the model. In other cases, users can construct their own person-specific transition hazards and their associated standard errors, and still use `probtrans` for the calculation of the transition probabilities and their (co)variances.

The functions `mssample` and `msboot` deal with model-based resampling methods. The former generates a specified number of paths through the multi-state model, and computes probabilities of states and paths given the starting state and time for any multi-state model. Moreover, `mssample` can give all sampled paths stored in a data frame to be used later for semi-parametric bootstrap procedures. The function `msboot` can be employed for any vector-valued statistic that a user may wish to use as an estimator in a multi-state model. This function samples randomly with replacement subjects from the original data set. Bootstrap data sets are then produced by concatenating all selected rows of these individuals. In [6] these functions have been used to compute prediction probabilities and their confidence intervals and for obtaining standard errors of the regression coefficients in multi-state reduced rank models. Finally, the reduced-rank procedures of Fiocco, Putter and Van Houwelingen (see [19,6]) are implemented in the function `redrank`.

Several data sets suitable for competing risks or multi-state analysis have also been included into the package. They contain HIV/AIDS data, bone marrow transplantation data (from [20]), three data sets from the European Registry for Blood and

Marrow Transplantation and liver cirrhosis data. We use these last data in the following example.

4. A multi-state model for liver cirrhosis data

In this paper, we will focus on what we consider to be the most important functions in `mstate`. A more extensive analysis of these data can be found online in De Wreede, Fiocco and Putter, *A multi-state model for liver cirrhosis data*, on http://www.msbi.nl/multistate.

4.1. Introduction

We illustrate the key functions of the package by the analysis of a data set made available by Per Kragh Andersen and discussed in [4], where details concerning the data can be found.

The data describe patients with liver cirrhosis. They originate from a clinical trial held in Copenhagen, in which patients were randomized to receive either the hormone prednisone or placebo. The primary goal was to investigate whether prednisone prolongs the survival of liver cirrhosis patients. The data set presented here contains 251 patients who received prednisone and 237 who received placebo.

A first analysis of these data showed no clear difference in survival between the two treatment groups. However, a further analysis in which covariates were taken into account showed interactions between the treatment and some of the prognostic factors. Some of these factors were measured during follow-up visits. An example of such a time-dependent covariate is the prothrombin index, an indication of the functioning of the liver. The index is coded as a binary variable: normal and abnormal (less than 70% of normal values). We assume that the change of this variable takes place at the moment when it is measured. Patients with a low level of prothrombin can recover and obtain a normal level again. Their levels may decrease and increase several times.

The effect of prednisone on survival in interaction with the prothrombin index can be studied by means of a multi-state model. We consider a model with two transient states: normal prothrombin index and low prothrombin index. These are also the starting states. There is one absorbing state: death. Patients can travel between the two transient states and visit each of them more than once. A model with this feature is called reversible. The model is illustrated in Fig. 3.

The purpose of the present analysis of these data is to demonstrate how our software works and to show that it can reproduce the results presented in [4]. We do not concern ourselves with questions about which model is the best one to answer clinical questions.

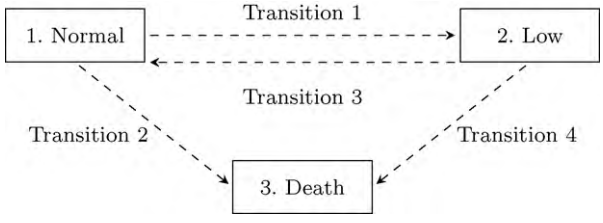

Fig. 3 – A multi-state model for liver cirrhosis patients.

4.2. Data preparation and description

The data are included in the `mstate` package. They are already in long format.

```
> library(mstate)
> data(prothr)
> prothr[prothr$id < 3, ]
```

	id	from	to	trans	Tstart	Tstop	status	treat
1	1	2	1	3	0	151	0	Placebo
2	1	2	3	4	0	151	1	Placebo
3	2	2	1	3	0	251	1	Placebo
4	2	2	3	4	0	251	0	Placebo
5	2	1	2	1	251	434	1	Placebo
6	2	1	3	2	251	434	0	Placebo
7	2	2	1	3	434	729	1	Placebo
8	2	2	3	4	434	729	0	Placebo
9	2	1	2	1	729	1735	1	Placebo
10	2	1	3	2	729	1735	0	Placebo
11	2	2	1	3	1735	2088	1	Placebo
12	2	2	3	4	1735	2088	0	Placebo
13	2	1	2	1	2088	2467	0	Placebo
14	2	1	3	2	2088	2467	1	Placebo

Consider for instance individual 2. At the start of the study, this patient had low prothrombin index, so he entered the study at time 0 in state 2, where he was at risk for transitions to state 1 and 3 (transitions 3 and 4). At time 251, his prothrombin index became normal again; therefore, he travelled to state 1 where he became at risk for transitions 1 and 2. At time 434, his prothrombin index became low again; at 729, normal again; at 1735, low; at 2088, normal. Finally he died after 2467 days.

The following transition matrix describes the possible transitions between the states in the model:

```
tmat <- transMat(x = list( c(2, 3), c(1, 3), c() ),
        names = c("Normal", "Low", "Death"))
> tmat
```

from \ to	Normal	Low	Death
Normal	NA	1	2
Low	3	NA	4
Death	NA	NA	NA

We can now either split the data into two parts according to treatment for a non-parametric analysis (`prothr0` and `prothr1`), or analyze the data of the two treatment groups together, considering treatment as a covariate at baseline. If we want to study the impact of the prednisone on each transition separately, we have to expand the treatment variable into transition-specific covariates by means of the function `expand.covs` (more precisely, in case of categorical covariates, `expand.covs` expands the dummy variables associated with these covariates).

```
> prothr <- expand.covs(prothr, covs = "treat")
> prothr0 <- prothr[prothr$treat == "Placebo",]
> prothr1 <- prothr[prothr$treat == "Prednisone",]
> head(prothr1)
```

```
   id from to trans Tstart Tstop status      treat treatPrednisone.1
29  7    2  1     3      0   202      0 Prednisone                 0
30  7    2  3     4      0   202      1 Prednisone                 0
31  8    2  1     3      0   211      1 Prednisone                 0
32  8    2  3     4      0   211      0 Prednisone                 0
33  8    1  2     1    211  2770      0 Prednisone                 1
34  8    1  3     2    211  2770      0 Prednisone                 0
   treatPrednisone.2 treatPrednisone.3 treatPrednisone.4
29                 0                 1                 0
30                 0                 0                 1
31                 0                 1                 0
32                 0                 0                 1
33                 0                 0                 0
34                 1                 0                 0
```

4.3. *The non-parametric model*

We start by analyzing the two treatment groups separately, as discussed in Section 2.3.1. We estimate the transition hazards using `coxph` without any covariates. By adding the stratum variable `trans` to the model, we estimate separate transition hazards for each transition. Then we use the function `msfit` to build data frames containing hazards and their covariances for two patients, one receiving prednisone and one receiving placebo. Because no covariates are involved in the analysis, the user can select either Greenwood or Aalen standard errors. We choose the Aalen-option to enable easy comparison between the standard errors of the non-parametric and the semi-parametric model.

The code is as follows (first for placebo and then for prednisone patients):

```
> c0 <- coxph(Surv(Tstart,Tstop,status)~strata
   (trans), data=prothr0, method="breslow")
> msf0 <- msfit(object=c0, vartype="aalen")
> c1 <- coxph(Surv(Tstart,Tstop,status)~strata
   (trans), data=prothr1, method="breslow")
> msf1 <- msfit(object=c1, vartype="aalen")
```

The calculations of this call of `msfit` are based on Eqs. (4)–(6).

Fig. 4 shows the estimated cumulative hazards for the transitions into the death state, both for prednisone and for placebo patients (cp. Fig. IV.4.10 of [4]). We see that the cumulative death hazards are higher when the patient has low prothrombin index.

The output of the function `msfit` is the input for the function `probtrans`. By means of the latter we calculate the transition probabilities $P_{gh}(s,t)$ from all starting states to all possible states and their standard errors, between the starting time `predt` = 0 and all event times successively. These calculations are based on Eqs. (15) and (21). The output of `probtrans` consists of three lists (one for each starting state g in the multistate model), each containing the transition probabilities and standard errors calculated on the assumption that the patient was in that state at time `predt`.

The function call for `probtrans` is as follows (again first for a patient treated with placebo and then with a patient treated with prednisone):

```
> pt0 <- probtrans(msf0, predt = 0,
   method = "aalen")
> pt1 <- probtrans(msf1, predt = 0,
   method = "aalen")
```

We can summarize all transition probabilities in two plots (see Fig. 5).

4.4. *The semi-parametric model*

Instead of using treatment as a stratifying variable, we can also create a semi-parametric model in which treatment is considered as a covariate at baseline. We take a Cox model in which all baseline hazards are assumed to be unrelated (see Eq. (11)). Because we expect the treatment to have a different effect on each transition, we need transition-specific covariates as given in the expanded data set.

The code is as follows:

```
> c2 <- coxph(Surv(Tstart, Tstop, status) ~
   treatPrednisone.1 +
+    treatPrednisone.2 + treatPrednisone.3 +
   treatPrednisone.4 +
+    strata(trans), data = prothr,
   method = "breslow")
> c2
```

```
Call:
coxph(formula = Surv(Tstart, Tstop, status) ~ treatPrednisone.1 +
    treatPrednisone.2 + treatPrednisone.3 + treatPrednisone.4 +
    strata(trans), data = prothr, method = "breslow")

                    coef exp(coef) se(coef)     z      p
treatPrednisone.1 -0.256     0.774    0.121 -2.11 0.0350
treatPrednisone.2 -0.249     0.780    0.197 -1.26 0.2100
treatPrednisone.3  0.308     1.361    0.114  2.71 0.0067
treatPrednisone.4  0.248     1.281    0.148  1.68 0.0940

Likelihood ratio test=16.2  on 4 df, p=0.00278  n= 2152
```

The output shows that none of the treatment variables has a significant effect on the direct transitions to the death state (transitions 2 and 4), which explains why no clear differences

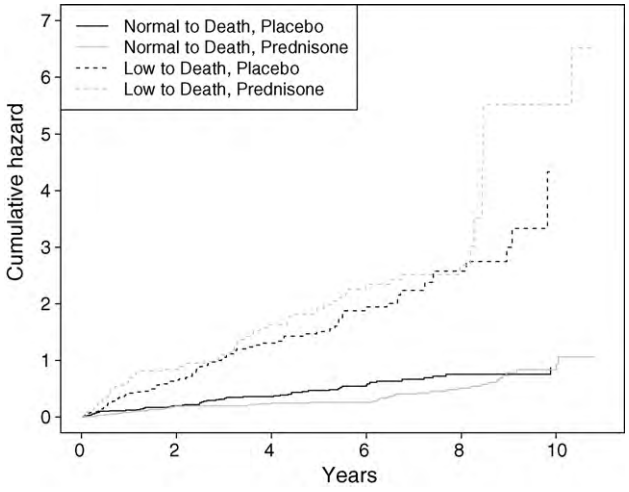

Fig. 4 – Estimated cumulative hazards.

in survival between the two treatments groups can be discerned. Rather, the effect of the treatment seems to be that it increases the probability of a Low to Normal (3) and lowers the probability of a Normal to Low (1) transition.

If we want to calculate transition probabilities according to this semi-parametric model, we need to set up new data frames describing two patients representing the two treatment groups. These dataframes are the input for `msfit`. In such a dataframe, we need one line for each of the transitions in the model. Each line contains the values of the covariates at baseline for the patient for whom a prediction is made, a transition indicator and a type indicator (called '`strata`'). The fastest way to do this is to select rows from the data with the required properties and then add the type indicator. Because of the presence of the covariate, `msfit` automatically calculates Aalen-type standard errors. The hazard-part of `msfit` is calculated by means of Eq. (13), its (co)variance by means of Eq. (14).

In analogy with the previous subsection, the output from the function `msfit` is denoted with `msf0C` and `msf1C` for a patient treated with placebo and one treated with prednisone respectively. The function `probtrans` again estimates transition probabilities and standard errors (Eqs. (15) and (21)).

```
> plac <- prothr[c(5:8),]
> plac$strata <- 1:4
> pred <- prothr[c(35:38),]
> pred$strata <- 1:4
> msf0C <- msfit(c2, plac, trans=tmat)
> msf1C <- msfit(c2, pred, trans=tmat)
> pt0C <- probtrans(msf0C, predt = 0)
> pt1C <- probtrans(msf1C, predt = 0)
```

The transition probabilities from time 0 of the two patients are again summarized in two plots (see Fig. 6; cp. Fig. 5). The plots suggest that prednisone somewhat prolongs survival for patients with normal prothrombin index but not for patients with low prothrombin level. The plots of the transition probabilities between the low and normal states show that prednisone patients tend to obtain normal prothrombin level more frequently than placebo patients. Although this seems to be a positive effect, still their survival from the low prothrombin state is not much better than for the placebo group.

4.5. *Comparison of the models*

We can compare the standard errors of model I (the non-parametric, stratified model) and II (the Cox model), both of Aalen-type, e.g. for the transition from state 1 (Normal) to state 3 (Death). Fig. 7 shows that the model assumption of model II

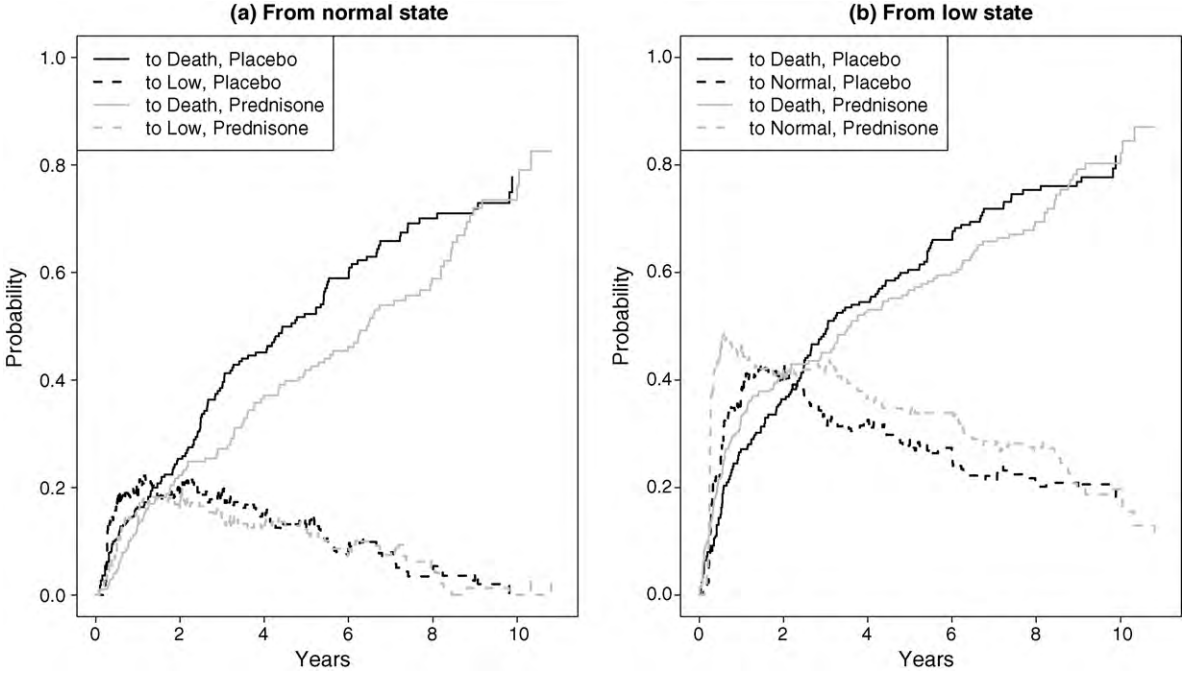

Fig. 5 – Estimated transition probabilities, non-parametric model.

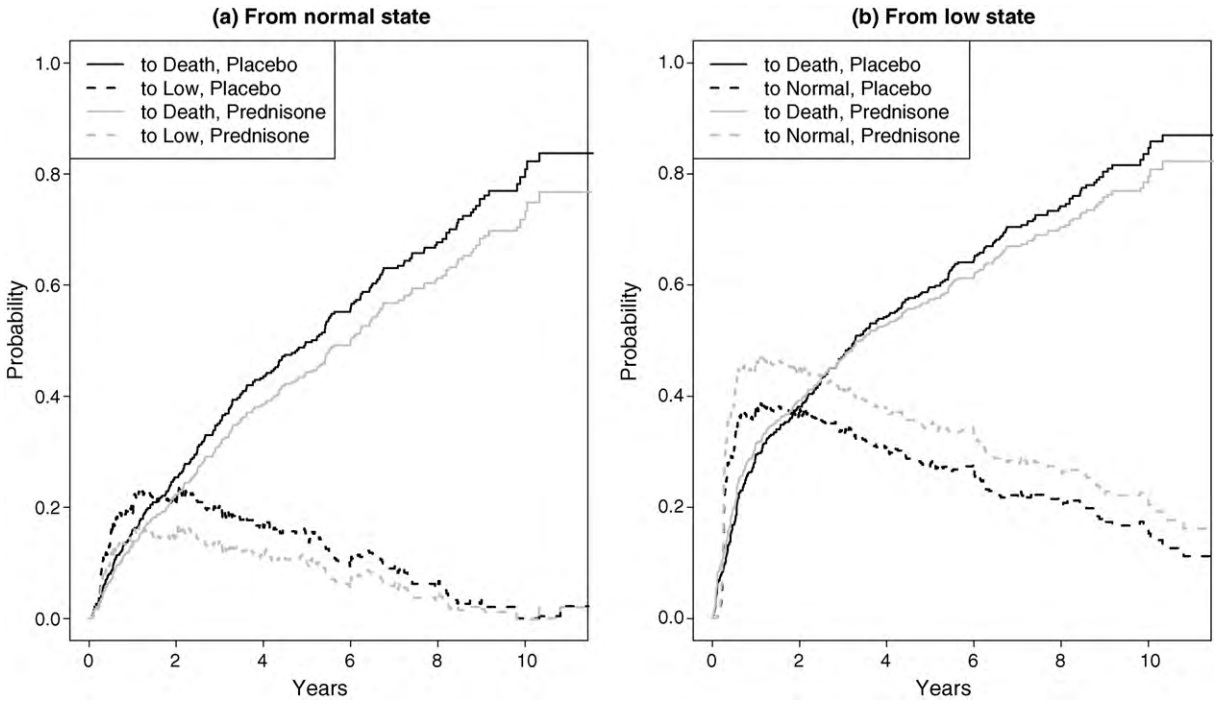

Fig. 6 – Estimated transition probabilities, semi-parametric model.

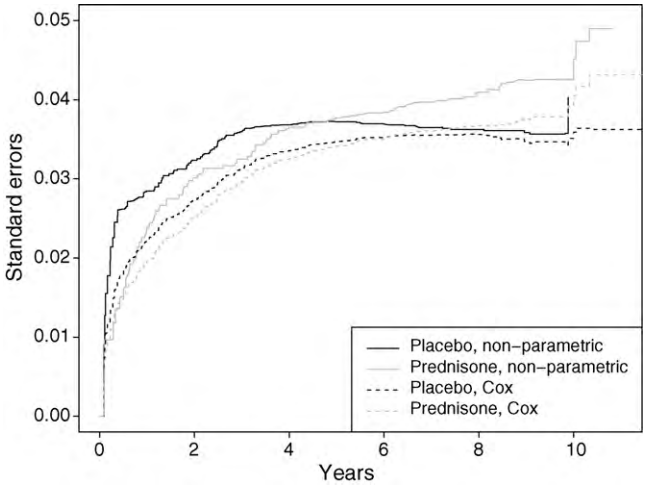

Fig. 7 – Estimated standard errors (Aalen-type) for transition probabilities from normal level of prothrombin to dead.

of proportional hazards tends to decrease the standard errors somewhat (see Fig. 7, and cp. Figs. IV.4.16 and VII.2.12 in [4]).

5. Discussion

Multi-state models are a useful extension of classical survival analysis for several reasons. Firstly, they help to give more biological insight into the disease/recovery process of a patient. Secondly, they enable clinicians to obtain more accurate predictions of survival probabilities and to calculate dynamic predictions.

The `mstate` package, written in `R`, is meant to help scientists to actually use multi-state and competing risks models. It is primarily meant as a tool in survival analysis, but it can also be applied in other contexts. It offers a variety of functions for data preparation, description and estimation, both facilitating standard and non-standard analyses. In particular, we have formulated and implemented a fast way to calculate prediction probabilities and their standard errors, both for fixed horizon and for forward prediction. The package is available from http://cran.r-project.org.

We restrict ourselves to non- and semi-parametric models, and we consider only independent right censoring and left truncation, and no other observational schemes such as right truncation, left censoring or interval censoring. However, once patient-specific hazards along with (co)variances have been obtained in such cases by means of other software, the function `probtrans` may still be used to obtain transition probabilities and their standard errors.

The development of a function calculating transition probabilities and standard errors without the use of simulations in the case of semi-Markov or non-Markov models requires the development of more mathematical theory first. We aim to investigate this in the future.

Conflict of interest statement

None declared.

Acknowledgements

Research leading to this paper was supported by the Netherlands Organization for Scientific Research Grant ZONMW-912-07-018 "Prognostic modeling and dynamic prediction for competing risks and multi-state models". We are grateful to Per Kragh Andersen for making available the liver cirrhosis data.

REFERENCES

[1] H. Putter, M. Fiocco, R.B. Geskus, Tutorial in biostatistics: competing risks and multi-state models, Statistics in Medicine 26 (11) (2007) 2389–2430.
[2] P.K. Andersen, M.P. Perme, Inference for outcome probabilities in multi-state models, Lifetime Data Analysis 14-4 (2008) 405–431.
[3] L. Meira-Machado, J. de Uña-Álvarez, C. Cadarso-Suárez, P.K. Andersen, Multi-state models for the analysis of time-to-event data, Statistical Methods in Medical Research 18 (2) (2009) 195–222.
[4] P.K. Andersen, Ø. Borgan, R.D. Gill, N. Keiding, Statistical Models Based on Counting Processes, 2nd ed., Springer Series in Statistics, Springer, 1993.
[5] J.D. Kalbfleisch, R.L. Prentice, The Statistical Analysis of Failure Time Data, Wiley, 1980.
[6] M. Fiocco, H. Putter, H.C. van Houwelingen, Reduced rank proportional hazards regression and simulation-based prediction for multi-state models, Statistics in Medicine 27 (21) (2008) 4340–4358.
[7] H. Putter, J. van der Hage, G. de Bock, R. Elgalta, C. van de Velde, Estimation and prediction in a multi-state model for breast cancer, Biometrical Journal 48 (3) (2006) 366–380.
[8] H.C. van Houwelingen, H. Putter, Dynamic predicting by landmarking as an alternative for multi-state modeling: an application to acute lymphoid leukemia data, Lifetime Data Analysis 14 (4) (2008) 447–463.
[9] G. Cortese, P.K. Andersen, Competing risks and time-dependent covariates, Biometrical Journal 52 (1) (2010) 138–158.
[10] T.M. Therneau, P.M. Grambsch, Modeling Survival Data: Extending the Cox Model, Springer, 2000.
[11] S. Rosthøj, P.K. Andersen, S.Z. Abildstrom, SAS macros for estimation of the cumulative incidence functions based on a Cox regression model for competing risks survival data, Computer Methods and Programs in Biomedicine 74 (1) (2004) 69–75.
[12] A. Alioum, D. Commenges, MKVPCI: a computer program for Markov models with piecewise constant intensities and covariates, Computer Methods and Programs in Biomedicine 64 (2) (2001) 109–119.
[13] G. Marshall, W. Guo, R.H. Jones, MARKOV: A computer program for multi-state Markov models with covariables, Computer Methods and Programs in Biomedicine 47 (2) (1995) 147–156.
[14] M. Wangler, J. Beyersmann, M. Schumacher, changeLOS: An R-package for change in length of hospital stay based on the Aalen–Johansen estimator, R News 6 (2006) 31–35.
[15] L. Meira-Machado, C. Cadarso-Suárez, J. de Uña-Álvarez, tdc.msm: An R library for the analysis of multi-state survival data, Computer Methods and Programs in Biomedicine 86 (2) (2007) 131–140.
[16] A. Allignol, J. Beyersmann, M. Schumacher, mvna: An R package for the Nelson-Aalen estimator in multistate models, R News 8 (2) (2008) 48–50.
[17] C. Jackson, Multi-state modelling with R: the `msm` package. http://cran.r-project.org/.
[18] W. Venables, B. Ripley, Modern Applied Statistics with S-PLUS, Springer, 2000 (corrected second printing).
[19] M. Fiocco, H. Putter, J.C. van Houwelingen, Reduced rank proportional hazards model for competing risks, Biostatistics 6 (3) (2005) 465–478.
[20] J.P. Klein, M.L. Moeschberger, Survival Analysis: Techniques for Censored and Truncated Data, Springer, 1997.