The R package ebmstate for disease progression analysis under empirical Bayes Cox models

Rui J. Costa¹ and Moritz Gerstung¹,²,³

Abstract

Statistical inference about multi-state random processes relies often on models defined by transition hazard rates. Inference based on such models necessarily involves the estimation of these rates, but it can also include estimates of state occupation probabilities. If the multi-state model incorporates a regression model, estimates of hazard rates and state occupation probabilities can be personalised, i.e. they can refer to an individual with specific covariate measurements. The software package mstate, in articulation with the package survival, provides not only a well-established multi-state survival analysis framework in R, but also one of the most complete, as it includes point and interval estimation of relative transition hazards, cumulative transition hazards and state occupation probabilities, both under clock-forward and clock-reset models; personalised estimates can also be obtained by fitting a Cox regression model. The new R package ebmstate, which we present in the current paper, is an extension of mstate and, to our knowledge, the first R package for multi-state model estimation that is suitable for higher-dimensional data and complete in the sense just mentioned. Its extension of mstate is threefold: it transforms the Cox model into a regularised, empirical Bayes model that performs significantly better with higher-dimensional data; it replaces asymptotic confidence intervals meant for the low-dimensional setting by non-parametric bootstrap confidence intervals; and it introduces an analytical, Fourier transform-based estimator of state occupation probabilities for clock-reset models that is substantially faster than the corresponding, simulation-based estimator in mstate. The present paper includes a detailed tutorial on how to use our package to estimate transition hazards and state occupation probabilities, as well as a simulation study showing how it improves the performance of mstate.

Keywords: multi-state survival analysis, regularised Cox model, R package, empirical Bayes, state occupation probabilities.

1 INTRODUCTION

Multi-state models based on transition hazard functions are often used in the statistical analysis of longitudinal data, in particular disease progression data [Hougaard, 1999]. The multi-state
model framework is particularly suitable to accommodate the growing level of detail of modern clinical data: as long as a clinical history can be framed as a random process which, at any moment in time, occupies one of a few states, a multi-state model is applicable. Another strong point of this framework is that it can incorporate a regression model, i.e., a set of assumptions on how covariates, possibly time-dependent ones, affect the risk of transitioning between any two states of the disease. Once estimated, multi-state models with regression features allow the stratification of patients according to their transition hazards. In addition, it is possible, under some models, to generate disease outcome predictions. These come in the form of state occupation probability estimates, meaning estimates of the probability of being in each state of the disease over a given time frame.

The survival analysis ‘task view’ of the Comprehensive R Archive Network lists six R packages that are able to fit general multi-state models and, at the same time, feature some kind of regression model or algorithm: msm (Jackson, 2011), SemiMarkov (Listwon and Saint-Pierre, 2015), survival (Therneau, 2015), mstate (de Wreede et al., 2010), mboost (Hothorn et al., 2020) – as extended by gamboostMSM (Reulen, 2014) – and penMSM (Reulen, 2015). All of them implement relative risk regression models, in particular the Cox model. The only exception is survival, which fits both the Cox model and Aalen’s additive regression model (Aalen, 1989).

Recall that in a semi-parametric Cox regression model each transition hazard is assumed to be the product of a baseline hazard function of unspecified form (the non-parametric component) and an exponential relative hazard function (the parametric component) (Aalen et al., 2008, p. 133). The Cox models implemented in these packages are all semi-parametric, with the exception of those in msm and SemiMarkov, which are fully parametric, i.e. they also restrict the baseline hazards to specific parametric families. In msm and SemiMarkov, the stronger assumptions regarding the functional form of the hazard are leveraged to do away with other common assumptions: SemiMarkov drops the usual Markov property to implement homogeneous semi-Markov models; msm is suitable for panel data, i.e., data in which the state of each individual is known only at a finite series of times.

Packages penMSM and gamboostMSM are the best suited to deal with higher-dimensional covariate data. The first of these packages relies on a structured fusion lasso method, while the second implements (jointly with mboost) a boosting algorithm. Both methods induce sparsity in the number of non-zero covariate effects, as well as equality among the different transition effects of each covariate, and are thus especially useful to reduce complicated multi-state models to more interpretable ones. The remaining packages assume the standard, fixed effects Cox model and do not include regularisation or variable selection features.

It is also illustrative to order the six packages mentioned according to how extensive their analysis workflow is. Packages SemiMarkov and penMSM are intended for the estimation of relative transition hazards only (i.e., for estimating the impact of covariates on each transition hazard). With the package mboost (as extended by gamboostMSM) it is also possible to estimate the baseline transition hazards. Finally, a more complete workflow including estimates of both
relative and cumulative transition hazards, as well as state occupation probabilities, is implemented in \texttt{msm} and \texttt{mstate}, and has been under implementation in \texttt{survival} (version 3.0 or later).

The present paper provides an introduction to \texttt{ebmstate}, a new R package for multi-state survival analysis available for download on the Comprehensive R Archive Network (CRAN). The main goal of \texttt{ebmstate} is to provide an analysis framework for the Cox model that performs better with higher-dimensional covariate data and is also complete, in the sense of being able to generate point and interval estimates of relative transition hazards, cumulative transition hazards and state occupation probabilities, both under clock-forward and clock-reset models. A fundamental characteristic of \texttt{ebmstate} is that it re-implements and extends the analysis framework of \texttt{mstate}, which is complete in the sense just mentioned. In fact, to a large extent, our package was built by importing, adapting and replacing functions from the \texttt{mstate} package. This not only eliminates redundancies, but also makes our package more accessible to the numerous users of \texttt{mstate} (the three papers associated with \texttt{mstate} have jointly over 2000 citations).

To improve the performance of \texttt{mstate}’s multi-state Cox model when dealing with higher-dimensional covariate data, a ridge-type regularisation feature was added. We allow the regression coefficients of the model to be partitioned into groups, with each group having its own Gaussian prior. A group can gather, for example, all the regression coefficients for a given transition. Or, within a given transition, coefficients can be grouped according to the covariate type they refer to (for example, demographic, clinical or genomic type). The resulting hierarchical Bayes model is empirical in that a full prior elicitation is not required (the mean and variance hyper-parameters of the Gaussian are estimated from the data). Model fitting relies on the iterative algorithm introduced by Schall (1991), which typically converges after a small number of steps. A simulation study showing that Schall’s algorithm performance compares well with that of other algorithms for ridge penalty optimisation, including one based on cross-validation, can be found in Perperoglou (2014).

The asymptotic confidence intervals generated by \texttt{mstate} are applicable when the number of observations is much larger than the number of parameters to be estimated (see section 3.3 below). To preserve the completeness of \texttt{mstate}’s framework in higher-dimensional settings, we therefore implemented non-parametric bootstrap intervals of regression coefficients, cumulative transition hazards and state occupation probabilities.

The high computational cost implied by the non-parametric bootstrap motivated a third extension to \texttt{mstate}. We developed an estimator of state occupation probabilities under clock-reset Cox models that is based on a convolution argument (as in Spitoni et al. 2012) and the Fast Fourier transform (FFT). At present, the estimation of such probabilities for clock-forward Cox models can be carried out using the efficient, product-limit based algorithm available in \texttt{mstate}. However, for clock-reset Cox models, only a simulation-based estimator is available in this package. The FFT estimator in \texttt{ebmstate} was conceived as a faster alternative to this
simulation-based estimator, but its scope is currently restricted to multi-state models with a tree-like transition structure, i.e. in which a transition between two states is either not possible or follows a unique sequence of states. Figure[] provides a short graphical summary of ebmstate, with the main inputs – a genomic-clinical data set and an empirical Bayes multi-state Cox model – and the main outputs – the estimates of relative hazards and state occupation probabilities (cumulative transition hazards are omitted).

As already mentioned, our empirical Bayes method improves estimator performance in models with larger numbers of covariates (see section 4 on estimator performance). Also, as a ridge-type regression method, it can be used as an alternative to the lasso method of penHSM in two particular cases: when the levels of correlation between covariates are high enough to compromise the stability of lasso-based covariate selection; or simply to improve prediction accuracy when interpretability is not essential and the number of covariates is not greater than the number of observations (Zou and Hastie 2005). In addition, and perhaps more importantly, ebmstate goes beyond the regularised estimation of transition hazards offered by penHSM and gamboostHSM: point and interval estimates of state occupation probabilities under the regularised Cox model can also be computed.

Figure 1: Summary of inputs and outputs of the package ebmstate. The input data set should be one that violates the assumption – commonly used in survival analysis – that the number of observations is much larger than the number of parameters to be estimated (a genomic-clinical data set is shown as a typical example). The input model is a multi-state Cox model defined by a transition structure and a prior distribution on the regression coefficients. This prior distribution is defined by partitioning the vector of regression coefficients into groups of regression coefficients, with each group having its own Gaussian prior with undetermined mean and variance. The outputs of ebmstate include estimates of the relative transition hazards associated with each covariate, as well as estimates of the probability that a specific patient (with specific covariate measurements) has of occupying each state of the model over some time period. Estimates of cumulative transition hazards are omitted from the figure.

\[
\lambda(t | \beta) = \beta \exp[\beta x] \\
\beta = \text{covariate vector} \\
\text{unknown coefficients} \\
\text{Prior:} \\
\beta = (\beta_1, \beta_2, \ldots) \\
\beta_{ij} \sim \mathcal{N}(\mu_\beta, \sigma^2_\beta) \\
\]

- Patient 1: state 1, age 23, male, gene 1 
- Patient 2: state 1, age 44, male, gene 1 
- Patient 3: state 1, age 47, male, gene 1, gene 2 

\[
\lambda_1(t) \quad \lambda_2(t) \quad \lambda_3(t) \\
\]

- state 1 
- state 2 
- state 3 
- state 4 

\[
0.0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1.0 \\
0 \quad 2 \quad 4 \quad 6 \quad 8 \quad 10 \\
\]

\[
\text{Probability} \\
0 \quad 0.2 \quad 0.4 \quad 0.6 \quad 0.8 \quad 1.0 \\
0 \quad 2 \quad 4 \quad 6 \quad 8 \quad 10 \\
\]

Figure 1: Summary of inputs and outputs of the package ebmstate. The input data set should be one that violates the assumption – commonly used in survival analysis – that the number of observations is much larger than the number of parameters to be estimated (a genomic-clinical data set is shown as a typical example). The input model is a multi-state Cox model defined by a transition structure and a prior distribution on the regression coefficients. This prior distribution is defined by partitioning the vector of regression coefficients into groups of regression coefficients, with each group having its own Gaussian prior with undetermined mean and variance. The outputs of ebmstate include estimates of the relative transition hazards associated with each covariate, as well as estimates of the probability that a specific patient (with specific covariate measurements) has of occupying each state of the model over some time period. Estimates of cumulative transition hazards are omitted from the figure.
2 MODELS

A multi-state Cox model is a continuous-time stochastic process with a finite (and usually small) state space \( S \). To better describe the models implemented in `ebmstate`, we define the following notation. We let \( t \) denote the time since some initiating event (usually diagnosis or disease onset). For \( t \in [0, \infty) \), we define the following random variables: \( X(t) \) represents the disease state of the patient, \( S(t) \) the time spent in the current state, and \( \vec{Z}(t) \) the value of a covariate vector. The realisation of each component of the process \( \{\vec{Z}(t)\} \) is a step function, possibly approximating the evolution in time of a continuous covariate. In addition, \( \{\vec{Z}(t)\} \) is assumed not-adapted to the filtration generated by \( \{X(u)\}, u \leq t, \) is known). The transition hazard rate of a patient from state \( i \) to state \( j \) \( (i \neq j) \) at time \( t \), conditional on the sojourn time and the covariate vector, is defined as

\[
\alpha_{ij}(t|z,s) := \lim \limits_{h \downarrow 0} \frac{1}{h} \mathbb{P}\left[ X(t+h) = j \mid X(t) = i, S(t) = s, \vec{Z}(t) = z \right], \quad s \in [0, \infty), \quad t \in [s, \infty).
\]

Independent right-censoring and left-truncation are assumed throughout (Aalen et al., 2008, p. 57). The purpose of the present section is to give a (not necessarily exhaustive) description of the scope of `mstate` and `ebmstate` with respect to the multi-state Cox model. Using the terminology in de Wreede et al. (2011), a Cox model is termed a ‘clock-reset’ model when

\[
\alpha_{ij}(t|z,s) = \lambda_{ij}^{(0)}(s) \exp\left[ \beta_{ij}^T z \right],
\]

and it is termed a ‘clock-forward’ model when

\[
\alpha_{ij}(t|z) = \alpha_{ij}^{(0)}(t) \exp\left[ \beta_{ij}^T z \right].
\]

In both cases, \( i, j \in S \), with \( i \neq j \); \( \beta_{ij} \) is an unknown vector of regression coefficient parameters, and both \( \lambda_{ij}^{(0)}(\cdot) \) and \( \alpha_{ij}^{(0)}(\cdot) \) are unknown (baseline hazard) functions, non-negative on \( \mathbb{R}^+ \). When, as in equation [1], \( \alpha_{ij}(t|z,s) \) is the same for all \( t \geq s \), we simplify its notation to \( \lambda_{ij}(s|z) \). As can be seen from equations [1] and [2] the ‘clock-reset’ and ‘clock-forward’ models are models for how the transition hazard rates are affected by time. In the former case, the only relevant time scale is the time \( s \) spent in the current state, whereas in the latter only the time \( t \) since the initiating event matters. While the ‘clock-forward’ model is arguably the default one in multi-state survival analysis (Andersen et al., 1993; Aalen et al., 2008), in some cases the ‘clock-reset’ model is more appropriate. For example, in some forms of cancer, it can be sensible to assume that the transition hazards from the state of complete remission depend on the sojourn time, rather than on the time since the initial diagnosis.
2.1 Relative transition hazards

The parametric component of the transition hazard from \(i\) to \(j\), written \(\exp \left[ \beta_i^T z \right]\), is termed the relative transition hazard. In \texttt{mstate}\ and \texttt{ebmstate}, estimating the relative transition hazard amounts to estimating the regression coefficient vector \(\beta_{ij}\). In \texttt{mstate}, these parameters are assumed to be non-random. With \texttt{ebmstate}, the following prior distributions can be imposed.

Define \(\mathcal{P}\) as the set of all pairs of states between which a direct transition is possible. Let \(\{\beta_{ij}\}\), for all \((i, j) \in \mathcal{P}\), be a partition of \(\beta\), a vector containing the regression coefficients for all direct transitions allowed. Each \(\beta_{ij}\) is further partitioned into \(\{\beta_{ijk}\}\), for \(k \in \{1, 2, ..., n_{ij}\}\). In \texttt{ebmstate}, the most general model regarding the prior distribution of \(\beta\) makes two assumptions: a) the scalar components of \(\beta\) are independent and normally distributed; b) the scalar components of \(\beta_{ijk}\) have a common (and undetermined) mean \(\mu_{ijk}\) and a common (and also undetermined) variance \(\sigma^2_{ijk}\).

The purpose of the framework just described is to allow the clustering of covariate effects according to their prior distribution. If there is no prior knowledge about how this clustering should be done, a single Gaussian prior can be imposed on all regression coefficients at once. If prior knowledge allows the grouping of effects according to the transition they refer to, a different Gaussian prior can be assigned to the coefficients of each transition. Even within each transition, different groups of coefficients can be assigned different prior distributions. In the analysis of biomedical data, for example, there can be a split between genes which are known to affect the transition hazard, and other genes whose effect is unknown.

2.2 Cumulative transition hazard functions

Our package imports from \texttt{mstate} a Breslow estimator of two types of cumulative transition hazard: one on a global time scale, defined as

\[
A_{ij}(t \mid \mathbf{z}) := \int_0^t \alpha_{ij}^{(0)}(u) \exp \left[ \beta_{ij}^T \mathbf{z} \right] \, du ,
\]

and another on a sojourn time scale, defined as

\[
\Lambda_{ij}(s \mid \mathbf{z}) := \int_0^s \lambda_{ij}^{(0)}(u) \exp \left[ \beta_{ij}^T \mathbf{z} \right] \, du .
\]

Note that, in either case, the covariate vector is assumed to remain constant.

2.3 State occupation probabilities

By state occupation probability, we mean the probability that a patient in state \(i\) at time 0 finds herself in state \(j\) at time \(t\). The estimates of these probabilities can be seen as functionals of the estimated cumulative transition hazard functions. For this reason, the restriction to models
with time-fixed covariates, which was just seen to be applicable to the estimators of cumulative
transition hazards, carries over to the estimation of state occupation probabilities.

When conditioning on a given covariate path (time-fixed or not), state occupation probability
estimates are not valid unless the covariates are external (Cortese and Andersen, 2010; Aalen
et al., 2008, p. 142). Note that a vector of covariates \( \{ \mathbf{Z}(u) \}_{u \geq 0} \) is said to be external if, for all \( t \in [0, \infty) \), each transition hazard at \( t \), conditional on \( \mathbf{Z}(t) \), is independent of \( \{ \mathbf{Z}(u) \}_{u > t} \) (i.e. independent of the future path of the covariate). Otherwise, it is said to be internal (for more
details on the distinction between internal and external covariates, see Kalbfleisch and Prentice,
2002, chapter 6). When one does not wish (or is not possible due to \( \mathbf{Z} \) being internal) to condition
on a future covariate path of the covariate process, the uncertainty introduced by this process
needs to be accounted for. This can be done by extending the state space of the disease process,
so that it includes information on the disease and the covariate process (Andersen et al., 1993,
p. 170). For example, to include a dichotomous transplant covariate (an internal covariate) in
a simple survival model with two states, the state space is expanded from \{alive, deceased\} to
\{alive without transplant, alive with transplant, deceased\}. One can then either assume that
transplanted patients have a different baseline death hazard or, more simply, that transplantation scales the death hazard by some constant \( \exp(\gamma) \). A similar but more detailed example can
be found in de Wreede et al. (2010, section 2.3.2, ‘model 3’).

3 ESTIMATION

In the current section, we present the estimation methods underlying the extensions of mstate
implemented in ebmstate.

3.1 Relative and cumulative hazard functions

Let \( \mu_{ij} \), with \((i, j) \in \mathcal{P}\) (the set of direct transitions allowed), denote a vector whose scalar
components are the parameters \( \mu_{ijk} \), \( k \in \{1, 2, \ldots, n_{ij}\} \). Similarly, let \( \sigma^2_{ij} \) be composed of the
parameters \( \{ \sigma^2_{ijk} \}_k \). The estimation of \( \beta, \mu := \{ \mu_{ij} \} \) and \( \sigma^2 := \{ \sigma^2_{ij} \} \) relies on the restricted
maximum-likelihood (REML) type algorithm described in Perperoglou (2014), and introduced
by Schall (1991). The resulting estimate of \( \beta \) is a maximum \textit{a posteriori} estimate; the estimates of \( \mu \) and \( \sigma^2 \) are empirical Bayes estimates. In ebmstate, the estimator based on this algorithm is
implemented in the function \texttt{CoxRFX}. The results of a simulation study showing its consistency
are included in the Supplementary Materials (file ESM_1.html, section 1).

The computation of cumulative hazard rates for given covariate values and an estimated re-
gression coefficient vector relies on the function \texttt{msfit_generic}, which is essentially a wrapper
for the function \texttt{mstate::msfit} (see section 5.3). For the mathematical details of this computa-
tion, we refer therefore the reader to de Wreede et al. (2010).
3.2 State occupation probabilities

The package `mstate` includes a simulation-based estimator that can take as input either \( \hat{A}_{ij} (\cdot \mid z) \) or \( \hat{A}_{ij} (\cdot \mid z) \) to generate estimates of state occupation probabilities under the clock-forward or the clock-reset model respectively. Another available estimator, an Aalen-Johansen-type estimator based on product integration, is far more efficient computationally and takes as input \( \hat{A}_{ij} (\cdot \mid z) \) only. As the scope of this estimator has been restricted to clock-forward Cox models (Andersen et al., 1993; Aalen et al., 2008), in our package we implemented a convolution-based estimator as a computationally efficient alternative (for models with a tree-like transition structure).

Convolution-based methods for clock-reset Cox models

For convenience, let the sequence of states from 0 to \( n \) have the labels 0, 1, 2, ..., \( n \), where 0 is the initial state by definition, and \( n \) is some state that might (eventually) be reached by the process. In addition, define \( X_0 := X(0) \) and \( T_0 := 0 \), and let \( (X_i, T_i), i \in \{1, 2, \ldots\} \), denote the marked point process associated with \( \{X(t)\} \), so that \( T_i \) is the time of the \( i^{th} \) transition and \( X_i \) is the state the process jumps to at time \( T_i \). The inter-transition times are denoted by \( \tau_{ij} := T_j - T_i \), for \( j > i \).

We can write the probability that a patient in state 0 at time 0 finds herself in state \( n \) at time \( t \), conditional on \( Z(u) = z \) for all \( u \geq 0 \), as

\[
P \left[ X(t) = n \mid X(0) = 0, Z(u) = z, u \geq 0 \right] = P \left[ X_n = n, \tau_{0,n} < t, \tau_{n,n+1} \geq t - \tau_{0,n} \mid X_0 = 0, Z(u) = z, u \geq 0 \right].
\]

Recall that \( \lambda_{ij+1} (s \mid z) \) denotes the hazard rate of a transition to state \( i + 1 \) at time \( s \) since arrival in state \( i \), for a patient that has covariate vector \( z \). The cumulative hazard for the same transition between sojourn times 0 and \( s \), if the patient’s covariate vector remains constant at \( z \), is represented by \( \Lambda_{ij+1} (s \mid z) := \int_0^s \lambda_{ij+1} (x \mid z) \, dx \). Similarly, we let \( \lambda_i (s \mid z) \) represent the hazard rate of going to any state that can be reached directly from \( i \), at time \( s \) since arrival in state \( i \), for a patient with covariate vector \( z \). The cumulative hazard for the same event between sojourn times 0 and \( s \), if the patient’s covariate vector remains constant at \( z \), is represented by \( \Lambda_i (s \mid z) \). The expressions \( \hat{\Lambda}_i (s \mid z) \) and \( \hat{\Lambda}_{ij+1} (s \mid z) \) denote the Breslow estimators of the cumulative hazards just defined. In what follows, all references to probabilities, hazard rates and cumulative hazards are to be understood as conditional on \( \tilde{Z}(u) = z \), for \( u \geq 0 \): this condition is omitted to simplify the notation.

In `mstate`, the function `probtrans_ebmstate` generates a set of state occupation probability estimates at equally spaced time points:

\[
\{ \hat{P}_{0n} (k) \}_{k} := \{ \hat{P} [X_n = n, \tau_{0,n} < t_k, \tau_{n,n+1} \geq t_k - \tau_{0,n} \mid X_0 = 0] \}_{k}, k = 0, 1, 2, \ldots, K; t_k = k \times \Delta t.
\]

The number \( K \) of time intervals is 10,000 by default and \( t_k \) is a parameter set by the user.
Defining the functions

\[ q_{ij} (k) := P [X_j = j, \tau_{ij} \in [t_k, t_{k+1}) \mid X_i = i] \]

and

\[ r_i (k) := P [\tau_{i,i+1} > t_k \mid X_i = i] , \]

the algorithm behind probtrans_ebmstate can be described as follows:

1. For \( j = 1, 2, ..., n \), compute

\[ \hat{q}_{j-1,j} (k) := \exp \left[ -\hat{\Lambda}_{j-1} (t_k) \right] \Delta \hat{\Lambda}_{j-1,j} (t_k) \]

for \( k = 0, 1, ..., K - 1 \).

2. For \( j = 2, 3, ..., n \), compute (iteratively)

\[ \hat{q}_0 (k) := \sum_{l=0}^{k-1} \hat{q}_{j-1,j} (k - l - 1) \hat{q}_{0,j-1} (l) \]

for \( k = 0, 1, ..., K - 1 \).

3. Finally, use the estimates obtained in the last iteration of step 2 to compute

\[ \hat{p}_{0,n} (k) := \sum_{l=0}^{k-1} \hat{r}_n (k - l - 1) \hat{q}_{0,n} (l) \]

for \( k = 0, 1, ..., K \), where \( \hat{r}_n (\cdot) := \exp \left[ -\hat{\Lambda}_n (t_{\cdot}) \right] \).

Substituting \( := \approx \) for \( \approx \) and removing the ‘hats’ in definitions 3 to 5, we get the approximate equalities that justify the algorithm. These approximate equalities are derived in the Supplementary Materials (file ESM_1.html, section 2).

Apart from probtrans_ebmstate, the function probtrans_fft is also based on the convolution argument just shown. However, this function makes use of the convolution theorem, i.e., of the fact that the convolution of two (vectorized) functions in the time domain is equivalent to a pointwise product of the same functions in the frequency domain. The estimation of state occupation probabilities is thus simplified to

\[ \hat{p}_{0,n} := \mathcal{F}^{-1} \{ \hat{q}_{0,1} \cdot \hat{q}_{1,2} \cdot \ldots \cdot \hat{q}_{n-1,n} \cdot \hat{r}_n \} , \]

where \( \mathcal{F} \) denotes the discrete Fourier transform, \( \hat{q}_{j-1,j} := \mathcal{F} (\hat{q}_{j-1,j}) \) and \( \hat{r}_n := \mathcal{F} (\hat{r}_n) \). Conversion to and from the frequency domain is carried out using the fast Fourier transform algorithm implemented in the \texttt{fft} function of the \texttt{stats} package. The Supplementary Materials contain a
short simulation study checking that state occupation probabilities can be accurately estimated
with `probtrans_ebmstate` and `probtrans_fft` (see file ESM_1.html, sections 3 and 4).

Figure 2 consists of a grid of plots with estimated curves of state occupation probabilities. It
compares, in terms of speed and accuracy, the estimator in `probtrans_fft` with an estimator in
`mstate::mssample` that has the same target, but is simulation-based. Each plot contains a black
curve and a superimposed red curve. The red curves in any given column of the grid are all
based on the same run of a function: columns 1 to 3 are based on runs of `mssample` with the
number of samples $n$ equal to 100, 1000 and 10.000 respectively, while column 4 is based on a
run of `probtrans_fft`. Each column in the grid reproduces the same 4 black curves. These are
based on a single run of `mssample` with $n = 100.000$ and serve as benchmark. All function runs
are based on the same input: a set of cumulative transition hazard estimates for a multi-state
model with the ‘linear’ transition structure given in the leftmost diagram of figure 3. Plots in a
given row refer to the same state of the model. The running times on top of each column refer
to the estimation of red curves. The main conclusion suggested by this analysis of simulated
data is that `probtrans_fft` is as accurate as `mssample` with $n = 10.000$, but it is almost 100
times faster (columns 3 and 4). With $n = 1000$, `mssample` achieves a good approximation to
the true state occupation probabilities, but is still roughly 9 times slower. The details on how
figure 2 and its underlying data were generated are given in the Supplementary Materials (file
ESM_1.html, section 5).

### 3.3 Interval estimation

Under any model estimated by `ebmstate` – as in general under a Bayesian model –, one can,
if the sample size is large enough, approximate the posterior by a normal distribution with
mean equal to the maximum a posteriori estimate and covariance matrix equal to the inverse of
the generalised observed Fisher information (see, for example, Gelman et al., 2014, p. 83-84).
This approximation has first-order accuracy and is thus outperformed by Laplace’s method,
which has second-order accuracy (Carlin and Louis, 2009, p. 110-111). However, as Carlin
and Louis (2009, p. 112) observe, “for moderate- to high-dimensional $\theta$ (say, bigger than 10),
Laplace’s method will rarely be of sufficient accuracy[…].” Carlin and Louis (2009, p. 244-251)
also describe three methods of interval estimation in empirical Bayes settings, but all of them
are designed for fully parametric models. These reasons, along with the fact that regularised
methods such as the one implemented `ebmstate` are typically used to fit models with more than
a dozen covariates, led us to choose the non-parametric bootstrap as the interval estimation
method in `ebmstate`. Interval estimates of regression coefficients, cumulative hazards and state
occupation probabilities are implemented in the function `boot_ebmstate`.
4 Estimator performance

It is a well-documented fact in the statistical literature that standard least-squares or maximum-likelihood estimators can often be improved by regularisation or shrinkage (see, for example, [Samworth, 2012]). This improvement comes about when the model dimensionality is high enough that the bias introduced by regularisation is outweighed by the reduction in the estimator variance. In the current setting, one might therefore ask: what kind of dimensionality does a semi-parametric, multi-state Cox model need to have to be outperformed by its empirical Bayes counterpart? A simulation study we carried out offers a tentative answer to this question, by comparing estimators under both Cox models for an increasing number of covariates. The study also features a third method, based on a fully non-parametric model, as a null model method. This was included to give an idea of how many covariates the empirical Bayes model can deal with before it becomes no better than a simple non-regressive model.
4.1 Simulation setup

We assessed the performance of all estimators defined by the tuple \([a, m, G, n, p(n)]\), where \(a \in \{\text{regression coefficients, relative hazards, state occupation probabilities}\}\) is the target of estimation, \(m \in \{\text{standard Cox, empirical Bayes Cox, null}\}\) is the assumed hazard model, \(G \in \{\text{linear, competing risks, ‘m’ structure}\}\) is the transition structure of the model (illustrated in figure 3) and \(n \in \{100, 1000\}\) is the number of patients/disease histories in the training data set; the variable \(p\) denotes the number of coefficients/covariates per transition in the true model and its range depends on \(n\): \(p(100) \in \{10, 40, 70, 100\}\) whereas \(p(1000) \in \{10, 100, 200, 300, 400, 500\}\).

By ‘relative hazards’ and ‘state occupation probabilities’, we mean here the relative transition hazards of an out-of-sample patient, and her state occupation probabilities at 7 chosen time points. We generated a batch of 300 independent absolute error observations (‘NA’ estimates included) for each estimator, where each observation is recorded after training the estimator on a newly simulated data set. Each boxplot in figures 6 \((n = 100)\) and 7 \((n = 1000)\) is based on one of these batches. As all estimators are vector estimators, each absolute error is actually an average absolute error, where the average is taken over the components of the vector.

All training data sets were simulated from clock-reset Cox models. Apart from \(G\) (the model transition structure), \(n\) and \(p\), also the true baseline hazards are held fixed within each batch of 300 training data sets. The coefficient vectors used in the simulation are always non-sparse and are scaled by \(\sqrt{\frac{10}{p}}\) to keep the log-hazard variance constant when the dimensionality grows. All covariates are dichotomous and mutually independent. To compute the coefficient errors for the non-parametric (null) model method, we think of it as a degenerate Cox model in which all regression coefficient estimates are fixed at zero. The estimation of regression coefficients under the standard Cox and the empirical Bayes Cox models was performed with \texttt{survival::coxph}\ and \texttt{ebmstate::CoxRFX} respectively; the estimation of state occupation probabilities is based on \texttt{mstate::probtrans} for the null model and on \texttt{ebmstate::probtrans_fft} for both the standard Cox and the empirical Bayes Cox models.

The reason we did not consider simulation scenarios with more than 500 covariates per transition, in data sets of 1000 patients, was simply computational cost. For example, generating the data and error observations for the scenario with \(n = 1000, p = 100\) and \(G = \text{‘m’ structure}\) took less than one hour to generate using 20 CPU cores in parallel; the same scenario but with \(p = 500\) took 6.5 days using 25 CPU cores. More details about the simulation setup can be found in the Supplementary Materials (file ESM_1.html, section 6, subsection ‘sample script’).

4.2 Missing values

Whenever an estimator was able to compute a valid estimate of its target for each training data set, i.e., when it did not return any ‘NA’ estimates, its boxplots are based on 300 valid error observations. This was always the case with non-parametric estimators: the estimates of regression coefficients and relative hazards of this type of estimators are trivial (fixed at zero and
one respectively) and hence it is also straightforward to compute absolute errors. It also happened that non-parametric estimators of state occupation probabilities had no ‘NA’ estimates (see file ESM_1.html, section 6, figure 6.3, in the Supplementary Materials). The situation was similar for the empirical Bayes Cox model estimators, which showed no more than 5% missing estimates in any of the simulation scenarios studied (ibid., figures 6.1 and 6.2). However, for the standard Cox model ones, the number of ‘NA’ estimates depends to a large extent on the number of patients in the data set, as well as on the dimensionality and transition structure of the model (figures 4 and 5). In data sets of 100 patients, it fares well in models with fewer than 10 covariates per transition, or in models with up to 40 covariates, if the transition structure is linear. Otherwise its failure rates range from roughly 25% to nearly 100%. In data sets of 1000 patients, the proportion of ‘NA’ estimates is never above 10%, if the transition structure is linear, but it can climb above 60% for other transition structures.

### 4.3 Comparison of estimators

With respect to the performance of the three methods studied, the boxplots in figures 6 and 7 suggest the following conclusions:

- As \( p/n \) grows, the empirical Bayes estimators quickly outperform the standard Cox model ones. They already fare substantially better at \( p/n = 0.1 \) for both \( n = 100 \) and \( n = 1000 \) and for all estimation targets. At the same time, the relative performance of the empirical Bayes method with respect to the null model one decreases. At \( p/n = 0.5 \), the difference between these two methods is already rather small for all simulation scenarios.

- The relative performance of the empirical Bayes method with respect to the null method decreases as the number of co-occurring transition hazards in the model grows. All other things equal, the empirical Bayes method has the best performance under the ‘linear’ structure model, which has no competing transitions; it performs less well under the ‘m’ structure transition model, where two transition hazards can co-occur; and has the worse
relative performances under the ‘competing risks’ model, where three transition hazards co-occur. This trend is clearer for $n = 100$ (figure 6) but can also be detected in the relative hazard errors for $n = 1000$ (figure 7). In any case, the empirical Bayes method seems to be far more robust than the standard Cox model against increases in the number of co-occurring transition hazards.

- Having as target the regression coefficients or the state occupation probabilities, instead of relative hazards, makes the empirical Bayes method better in comparison to the null method. In fact, as $p/n$ grows, the empirical Bayes method is never outperformed by the null method except in the estimation of relative hazards.

5 SURVIVAL ANALYSIS WORKFLOW

The features of \texttt{mstate} were illustrated in \cite{deWreede2010} using a simple workflow. The starting point of this workflow is a data set in ‘long format’. Such data set can be fed into \texttt{survival::coxph} to obtain estimates of the regression coefficients of a multi-state Cox model. The resulting model fit object can be passed on to \texttt{mstate::msfit}, along with a vector of covariates of a particular patient, to get personalised estimates of the cumulative hazard functions. Finally, state occupation probabilities for the same patient can be estimated if the object created by \texttt{mstate::msfit} is fed into \texttt{mstate::probtrans}. In this section, we describe how \texttt{ebmstate}...
Figure 5: Proportions of valid, infinite and missing ('NA') estimates for the standard Cox model estimators in the simulation study of figure 7 (1000 patients per simulated data set).

extends the scope of this workflow, i.e., how it uses the packages \texttt{survival} and \texttt{mstate} to generate estimates under a multi-state empirical Bayes Cox model. A diagram summarising the extension is shown in figure 8.

The main steps of the \texttt{ebmstate} workflow will be illustrated using a data set of patients with myelodysplastic syndromes (MDS) which has been described and studied in \cite{Papaemmanuil2013}. A myelodysplastic syndrome is a form of leukemia in which the bone marrow is not able to produce enough mature blood cells, and which sometimes develops into a cancer of white blood cells with a quick and aggressive progression, i.e., into acute myeloid leukemia (AML). Figure 9 illustrates an illness-death type model for MDS patients and also gives a breakdown of the number of transition events. The conversion to a model with a tree-like transition structure (that can be handled by our convolution-based estimators) is shown in figure 9b. The data set used for model estimation, obtained after a number of pre-processing steps, contains the disease history of 576 patients, as well as measurements on 30 covariates. Of these 30 covariates, 11 are mutation covariates and the remaining are clinical or demographic (see figure 9c). The running time for the estimation of relative transition hazards does not exceed 10 seconds in a standard laptop computer. The same holds for the estimation of cumulative transition hazards or state occupation probabilities for a given patient. The complete R code underlying the data analysis in the current section can be found in the Supplementary Materials (file ESM_2.html).
5.1 Input data

Table 1 shows a fragment of the MDS data set. The data is in ‘long format’, which means that each row refers to a period of risk for a given transition and patient. For example, row \( i \) tells us that, at time \( T_{\text{start}}[i] \), patient \( \text{id}[i] \) entered state \( \text{from}[i] \), and thereby began to be at risk for transition \( \text{trans}[i] \), i.e., at risk of going from state \( \text{from}[i] \) to state \( \text{to}[i] \). If the first transition of patient \( \text{id}[i] \) after time \( T_{\text{start}}[i] \) occurs before the last follow-up time for this patient, \( T_{\text{stop}}[i] \) records the time of this transition (regardless of whether the patient moved to state \( \text{to}[i] \) or not). Otherwise, \( T_{\text{stop}}[i] \) is set to the last follow-up time. The value of \( \text{status}[i] \) is set to 1 if and only if the first transition of patient \( \text{id}[i] \) after \( T_{\text{start}}[i] \) is to state \( \text{to}[i] \) and occurs before the last follow-up (otherwise it is set to 0). The value of \( \text{time}[i] \) is defined simply as \( T_{\text{stop}}[i] - T_{\text{start}}[i] \), and \( \text{strata}[i] \) is the stratum of the baseline hazard for transition \( \text{trans}[i] \) (more about this variable in the following section). For \( x \in \{ \text{ASXL1, DNMT3A, ...} \} \), \( x[i] \) denotes the level of covariate \( x \) between \( T_{\text{start}}[i] \) and \( T_{\text{stop}}[i] \) in patient \( \text{id}[i] \). (In the MDS data set, we assume that the relative hazard of a patient is determined by her covariate vector...
at $t = 0$, i.e., we assume all covariates to be time-fixed.) If a patient enters a new state, and this state communicates directly with $n$ other states, then, as long as the patient actually spends time in the new state (i.e. the time of transition is not the same as the last follow-up time), $n$ rows must be added to the data set, with each row corresponding to a different possible transition.

From table 1, we know that patient 1 entered state 1 (‘MDS’) at time 0 and remained in this state until time 327, when she moved to state 3 (‘death before AML’). There are no rows to describe the evolution of patient 1 after entering state 3, as this state is an absorbing state. As to patient 2, she remained in state 1 until time 1613, and moved from there to state 2 (‘AML’). By the time of the last follow-up (1782), patient 2 was still in state 2 (and still at risk of going to state 4 – ‘death after AML’).

### 5.2 Fitting an empirical Bayes Cox model

Once the data is in ‘long format’, the estimation of an empirical Bayes model can be carried out using the function `CoxRFX`. A simple example of the first argument of `CoxRFX`, denoted ‘Z’, is a data frame gathering the `trans`, `strata` and covariate columns of the data in long format:
Figure 8: Extension of the mstate analysis framework by ebmstate. Arrows correspond to functions. Boxes correspond to inputs or outputs of functions. Functions CoxRFX and probtrans_fft from ebmstate compute point estimates only. Interval estimates can be obtained using the non-parametric bootstrap algorithm implemented in the function ebmstate::boot_ebmstate.

Table 1: The first five rows of the MDS data set (in long format)

| id | from | to | trans | Tstart | Tstop | time | status | strata | ASXL1 | DNMT3A |
|----|------|----|-------|--------|-------|------|--------|--------|-------|--------|
| 1  | 1    | 1  | 2     | 1      | 0     | 327  | 327    | 0      | 1     | 0      |
| 2  | 1    | 1  | 3     | 2      | 0     | 327  | 327    | 1      | 2     | 0      |
| 3  | 2    | 1  | 2     | 1      | 0     | 1613 | 1613   | 1      | 1     | 0      |
| 4  | 2    | 1  | 3     | 2      | 0     | 1613 | 1613   | 0      | 2     | 1      |
| 5  | 2    | 4  | 3     | 1613   | 1782  | 169  | 0      | 3      | 1     | 0      |

The strata column determines which baseline hazard functions are assumed to be equal. In table 1, each transition is assumed to have a (potentially) different baseline hazard. The model’s assumptions regarding how covariates affect the hazard are reflected on the format of the covariate columns of Z. When the Z argument is the one created in the previous block of code, CoxRFX returns a single regression coefficient estimate for each covariate. In other words, the impact of any covariate is assumed to be the same for every transition.

There is however a way of relaxing this assumption. One can replace the ASXL1 column in Z (or any other covariate column) by several ‘type-specific’ ASXL1 columns: the ASXL1 column specific for type i would show the mutation status of ASXL1 in rows belonging to transition of type i, and show zero in all other rows. This would force CoxRFX to estimate a (potentially) different ASXL1 coefficient for each transition type. This process of covariate expansion by type can be based on any partition of the set of transitions. When each type corresponds to a single transition, we refer to it simply as ‘covariate expansion by transition’. The output shown below illustrates the effect of expanding the covariates in ‘mstate.data’ by transition.
The example code given below shows how to use mstate to expand covariates by transition and how to create a Z argument that makes CoxRFX estimate a regression coefficient for each covariate for transitions 1 and 2, and assume a fully non-parametric hazard for transition 3.

```r
# To expand covariates by transition using mstate::expand.covs,
# first set the class of 'mstate.data' as
class(mstate.data)<-c("data.frame","msdata")

# then add the transition matrix as attribute:
attr(mstate.data,"trans")<-tmat #'tmat' is the output of mstate::transMat

# Expand covariates by transition:
covariates.expanded<-mstate::expand.covs(mstate.data,covs =
    names(mstate.data)[!names(mstate.data)%in%c("id","from","to","trans","Tstart","Tstop","time","status","strata")],
    append = F)

# remove all covariates for transition 3 from 'covariates.expanded'
# to fit a fully non-parametric model on this transition:
covariates.expanded<-covariates.expanded[!grepl("\.3",
    names(covariates.expanded),fixed = T)]

#argument 'Z' of coxrfx
Z<-data.frame(covariates.expanded,strata=mstate.data$trans,
    trans=mstate.data$trans)
```

The second argument of CoxRFX ('surv') is a survival object that can easily be built by feeding the outcome variable columns of the data to the function Surv (from the package survival). Whether CoxRFX fits a clock-forward model or a clock-reset model depends on the kind of survival object:

```r
#argument 'surv' for a clock-forward model
surv<-Surv(mstate.data$Tstart,mstate.data$Tstop,mstate.data$status)

#argument 'surv' for a clock-reset model
surv<-Surv(mstate.data$time,mstate.data$status)
```
The argument groups of CoxRFX is a vector whose length equals the number of covariates in the data. In other words, the length of groups is ncol(Z)-2, since the argument Z must include both the covariate data and the strata and trans columns. If, for \( i \neq j \), groups[i]=groups[j] = ‘foo’, this means that the regression coefficients of the \( i^{th} \) and \( j^{th} \) covariates of Z both belong to a group named ‘foo’ of coefficients with the same prior. For the Z object built above, the groups argument created in the following block of code embodies the assumption that all coefficients associated with a given transition have the same prior distribution. The final line of code fits the empirical Bayes model.

```r
# argument ‘groups’ of coxrfx
groups <- paste0(rep("group",ncol(Z)-2),c("_1","_2"))
# fit random effects model
coxrfx_object <- CoxRFX(Z,surv,groups,tmat)
```

Figure 10 shows regression coefficient point estimates for a clock-reset, empirical Bayes model fitted with the code above. Also shown are 95% non-parametric bootstrap confidence intervals computed using ebmstate::boot_ebmstate. The x-axis scale is logarithmic to allow estimates to be read as relative hazards more easily. For example, a mutation in RUNX1 is associated with a twofold increase in the hazard of progression from MDS to AML, and treatment centre 4 is associated with a 3-fold increase in the hazard of dying before progressing to AML, when compared to the baseline value of ‘treatment centre’ (treatment centre = 2 or 5). In covariates that have been log-transformed (age, platelet count and neutrophil count) or logit-transformed (proportions of myeloblasts and ring sideroblasts in the bone marrow), the interpretation of estimates is different. For example, an increase in age by a factor of \( e (\approx 2.72) \) almost triples the hazard of dying before AML; the same increase in the ratio \( bm\_blasts/(1-bm\_blasts) \) (where \( bm\_blasts \) is the proportion of myeloblasts in the bone marrow) is associated with an increment in the hazard of dying before AML of approximately 16%.

### 5.3 Computing cumulative transition hazard estimates

The function msfit.generic is the generic function in ebmstate that computes cumulative transition hazards for a given set of covariate values and an estimated Cox model. It calls a different method according to the class of its object argument. The default method corresponds to the original msfit function of the mstate package and is appropriate for objects of class coxph, i.e., objects that contain the fit of a Cox model with fixed effects. The other available method for msfit.generic, msfit.generic.coxrfx, is just the original msfit function, (slightly) adapted to deal with objects generated by CoxRFX. Quite importantly, msfit.generic.coxrfx does not allow the variance of the cumulative hazards to be computed, as this computation relies on asymptotic results which may not be valid for an empirical Bayes model. As a result, it only has two other arguments apart from the object of class coxrfx: a data frame with the covariate values of the patient whose cumulative hazards we want to compute; and a transition matrix describing the states and transitions in the model (such as the one that can be generated using
transMat from the package mstate). The following block of code exemplifies how these objects can be built and generates the msfit object containing the cumulative transition hazard estimates for a sample patient. Note that the object with the patient data must include a row for each transition, as well as a column specifying the transition stratum of each row of covariates.

```r
# Build 'patient_data' data frame with the covariate values for which cumulative hazards are to be computed (covariate values of patient 3):
patient_data <- mstate.data[mstate.data$id==3, , drop=F][rep(1,3),]
patient_data$strata <- patient_data$trans <- 1:3
patient_data <- mstate::expand.covs(patient_data, covs = names(patient_data)[!names(patient_data)%in%c("id", "from", "to", "trans", "Tstart", "Tstop", "time", "status", "strata")],
append = T)
patient_data <- patient_data[!grepl("\d.3", names(patient_data), fixed = T)]

# The 'patient_data' data frame has only 3 rows (one for each transition).
# The output below shows its 'id' and 'trans' columns and expanded covariates ASXL1 and DNMT3A:
# id trans ASXL1.1 ASXL1.2 DNMT3A.1 DNMT3A.2 [...]
# 1 3 1 1 0 0 0 .
# 2 3 2 0 1 0 0 .
# 3 3 3 0 0 0 0 .

# compute cumulative hazards
msfit_object <- msfit_generic(coxrfx_object, patient_data, tmat)

Figure 11 shows three plots of estimated cumulative transition hazards for the sampled patient, one for each transition in the model, along with 95% non-parametric bootstrap confidence intervals (computed with ebmstate::boot_ebmstate). Throughout the plotted period, the 'slope' of the cumulative hazard (i.e., the hazard rate) for the MDS to AML transition is lower than the one for the MDS to death transition, and this in turn is lower than the one for the AML to death transition. It should be recalled that the cumulative hazard estimate is strictly non-parametric for this last transition, i.e., it is the same for all patients. The central plot of figure 11 suggests that, as time since diagnosis goes by, the hazard of dying in MDS increases (possibly an effect of age). On the other hand, the hazard of dying in AML seems to decrease (slightly) with time (rightmost plot). Conclusions regarding the evolution of the AML hazard are hard to draw, since the confidence intervals for the corresponding cumulative hazard curve are very wide (leftmost plot).

If an object generated by msfit_generic is fed to plot, and the package mstate is loaded, the method mstate:::plot.msfit will be called. This is an efficient way of automatically plotting the cumulative hazard estimates for all transitions, but confidence interval lines (separately estimated) cannot be added.

Figure 11
5.4 Computing state occupation probability estimates

The functions `probtrans_mstate`, `probtrans_ebmstate` and `probtrans_fft` compute estimates of state occupation probabilities for a given `msfit` object. All three functions generate objects of class `probtrans` that can be fed to the `plot.probtrans` method from the package `mstate`. The first of these functions should only be used for clock-forward models, as it relies on product-limit calculations. It calls the method `probtrans_mstate.default`, if the `msfit` object was generated by `msfit_generic.default`, or the method `probtrans_mstate.coxrfx`, if it was generated by `msfit_generic.coxrfx`. Both methods are identical to the function `probtrans` in the `mstate` package, with the reserve that `probtrans_mstate.coxrfx` does not allow the computation of the variances or covariances of the state occupation probability estimator. The functions `probtrans_ebmstate` and `probtrans_fft` are the functions in `ebmstate` for computation of state occupation probability estimates under clock-reset models with a tree-like transition structure. When using `probtrans_fft` (the faster of these two functions), three arguments must be supplied: the initial state of the process whose state occupation probabilities one wishes to compute, the `msfit` object, and a vector of positive and increasing time points starting from and including zero. This last argument, denoted `time`, is crucial for precision: the density of time points and the upper time limit should be increased until the estimated curves become stable. The following line of code computes point estimates of state occupation probabilities for the sample patient.

```
probtrans_object <- probtrans_fft("MDS", msfit_object, time)
```

Estimates are shown in figure 12 along with 95% non-parametric, bootstrap confidence intervals (computed using `ebmstate::boot_ebmstate`). For this particular patient, the estimated probability of being dead after AML remains below 0.1 throughout a period of 10 years from the MDS diagnosis; if the patient does reach AML, death is expected to happen quickly thereafter, as reflected in the very low estimates for the probability of being in AML at any point in time.

The plots of figure 13 show leave-one-out personalised estimates of disease progression for a random sample of 196 patients, ordered by overall survival probability at 5 years since diagnosis. As shown in the following block of code, these were built using `ebmstate::loo_ebmstate`.

```
# Some arguments for the function loo_ebmstate:
patient_IDs <- sample(unique(mstate.data$id),14*14)
mstate.data.expanded <- mstate::expand.covs(mstate.data,
    covs = names(mstate.data)[!names(mstate.data)%in%c("id","from","to","trans","Tstart","Tstop","time","status","strata")], append = T)

# Leave-one-out outcome predictions:
loo_object <- loo_ebmstate(mstate.data,mstate.data.expanded,
    groups,patient_IDs,initial_state="MDS",tmat,
    time_model="clock-reset")
```

The mosaic plot shows a wide range of prognoses. Quite importantly, a substantial number
of patients have a very low estimated probability of developing AML during the first 10 years following diagnosis, and should perhaps be spared the burden of aggressive treatments. The validity of such conclusions requires however that the model fares well in terms of prediction accuracy. Here further investigations would be needed.

6 DISCUSSION

We have shown that ebmstate is suitable for higher-dimensional, multi-state survival analysis, and that it is both efficient and easy-to-use. To a significant extent, the user-friendliness of ebmstate stems from the fact that it was not built ‘from the ground up’. Instead, we produced a package that is more easily accessible to the many users of mstate by taking advantage of whichever features of this package were useful to our method and by eliminating redundancies. The connection between ebmstate and mstate is based on the fact that the function CoxRFX takes the same type of input and produces the same type of output as coxph from the package survival, and the function probtrans_fft (or probtrans_ebmstate) has the same type of input and output as probtrans from mstate (as shown in figure 8).

We also sought to improve our package’s user-friendliness by making it as efficient as possible. The reduction of computational cost is based on two features. First, our empirical Bayes method relies on an expectation-maximisation algorithm that estimates both the parameters and the hyper-parameters of the model, i.e., no further tuning of the model is required. Second, in ebmstate, the computation of state occupation probability estimates relies on analytical results rather than on simulation: not only for clock-forward models, where we import from mstate a product-limit estimator, but also for clock-reset models, where we implement our own estimator based on a convolution argument and the fast Fourier transform.

To our knowledge, ebmstate is the first R package to put together a framework for multi-state model estimation that is complete and suitable for higher-dimensional data. It does so by implementing point and interval estimators of regression coefficients, cumulative transition hazards and state occupation probabilities, under regularised multi-state Cox models. In section 4 the results of the simulation study suggest that for data sets with 100 patients or more and a ratio of \( p \) (patients) to \( n \) (coefficients per transition) greater than 0.1, the standard Cox model estimator is clearly outperformed by the empirical Bayes one when it comes to the estimation of relative hazards and state occupation probabilities of an out-of-sample patient, or the regression coefficients of the model. However, the same study suggests that using an empirical Bayes method instead of a fully non-parametric one is of limited or no value in settings where \( p/n \geq 1 \). This loss of usefulness can already happen for \( p/n \leq 1/2 \) when it comes to the estimation of the relative hazards of an out-of-sample patient, especially for transition structures with multiple competing transitions.

As mentioned in previous sections, ebmstate imports a product-limit estimator from mstate that targets the state occupation probabilities of patients with time-fixed covariate vectors. How-
ever, these estimators are extendible to models with time-dependent covariates, as long as these are external and the estimates are conditional on specific covariate paths (Aalen et al., 2008, p. 142). For piecewise constant covariates, it is likely that such an adaptation could be obtained by combining transition probability estimates obtained for each period in which the covariates are fixed. While no significant theoretical obstacles are foreseen in this matter, the computer implementation for more than a single piecewise constant covariate is likely to be a laborious task. We have left it therefore for future work.

**Supplementary Materials**

The file ‘ESM_1.html’ contains additional simulation results and theoretical demonstrations. Additional details on the analysis of the MDS data set are given in the file ‘ESM_2.html’.

**Acknowledgements**

The authors are supported by grant NNF17OC0027594 from the Novo Nordisk Foundation.
Figure 9: a: transition model implied by the data set of patients with myelodysplastic syndromes, together with transition event numbers; b: conversion to a tree-like transition structure; c: transformations applied to the MDS covariate data and summary statistics for the data before transformation. MDS stands for myelodysplastic syndromes; AML stands for acute myeloid leukemia.
Figure 10: Point estimates of regression coefficients for the Cox model fitted to the MDS data, along with 95% non-parametric bootstrap confidence intervals. The x-axis scale is logarithmic so that coefficient estimates can be read as relative hazard estimates. If $\gamma_{ij}$ is the element of $\hat{\beta}_{ij}$ associated with a given covariate, $\exp(\gamma_{ij})$ is the estimated relative hazard for this covariate in transition $(i,j)$. In general, a relative hazard estimate $r$ for a covariate $z$ in transition $(i,j)$ means that a one-unit increase in $z$ is associated with an $r$-fold increase in the hazard of this transition. If $z$ was obtained by log-transformation (as in age, platelet counts and neutrophil counts), a one-unit increase in $z$ corresponds to scaling the original covariate by $e \approx 2.72$. In case $z$ was obtained by logit-transformation (as in bone marrow blasts and sideroblasts proportions), the same one-unit increase corresponds to scaling the odds of the original covariate by $e$. 
Figure 11: Point estimates of cumulative transition hazards for a sample patient with MDS (black curve), along with 95% non-parametric confidence intervals (dashed red lines).

Figure 12: Point estimates of state occupation probabilities for a sample patient with MDS (black curve), along with 95% non-parametric confidence intervals (dashed red lines).
Figure 13: Leave-one-out estimates of state occupation probabilities for a random sample of 196 individuals with MDS. The x-axis measures time since diagnosis and runs from 0 to 10 years.
References

Aalen, O., O. Borgan, and H. Gjessing (2008). *Survival and event history analysis*. Springer.

Aalen, O. O. (1989). A linear regression model for the analysis of life times. *Statistics in Medicine* 8(8), 907–925.

Andersen, P., O. Borgan, R. Gill, and N. Keiding (1993). *Statistical Models Based On Counting Processes*. Springer.

Carlin, B. and T. Louis (2009). *Bayesian Methods for Data Analysis*. CRC Press.

Cortese, G. and P. K. Andersen (2010). Competing risks and time-dependent covariates. *Biometrical Journal* 52(1), 138–158.

de Wreede, L. C., M. Fiocco, and H. Putter (2010). The mstate package for estimation and prediction in non- and semi-parametric multi-state and competing risks models. *Computer Methods and Programs in Biomedicine* 99(3), 261 – 274.

de Wreede, L. C., M. Fiocco, and H. Putter (2011). mstate: An R package for the analysis of competing risks and multi-state models. *Journal of Statistical Software* 38(7), 1–30.

Gelman, A., J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin (2014). *Bayesian Data Analysis*. CRC Press.

Hothorn, T., P. Buehlmann, T. Kneib, M. Schmid, and B. Hofner (2020). mboost: Model-based boosting. *R package version*, 2.9–3.

Hougaard, P. (1999). Multi-state models: a review. *Lifetime data analysis* 5(3), 239–264.

Jackson, C. H. (2011). Multi-state models for panel data: the msm package for R. *Journal of Statistical Software* 38(8), 1–29.

Kalbfleisch, J. D. and R. L. Prentice (2002). *The statistical analysis of failure time data*. John Wiley & Sons.

Listwon, A. and P. Saint-Pierre (2015). SemiMarkov: An R Package for Parametric Estimation in Multi-State Semi-Markov Models. *Journal of Statistical Software* 66(6), 784.

Papaemmanuil, E., M. Gerstung, L. Malcovati, S. Tauro, G. Gundem, P. Van Loo, C. J. Yoon, P. Ellis, D. C. Wedge, A. Pellagatti, et al. (2013). Clinical and biological implications of driver mutations in myelodysplastic syndromes. *Blood* 122(22), 3616–3627.

Perperoglou, A. (2014). Cox models with dynamic ridge penalties on time-varying effects of the covariates. *Statistics in Medicine* 33(1), 170–180.
Reulen, H. (2014). gamboostmsm. *R package version*, 1.1.87.

Reulen, H. (2015). pennsm. *R package version*, 0.99.

Samworth, R. J. (2012). Stein’s paradox. *Eureka* 62, 38–41.

Schall, R. (1991). Estimation in generalized linear models with random effects. *Biometrika* 78(4), 719–727.

Spitoni, C., M. Verduijn, and H. Putter (2012). Estimation and asymptotic theory for transition probabilities in markov renewal multi-state models. *The International Journal of Biostatistics* 8(1).

Therneau, T. M. (2015). *A Package for Survival Analysis in S*. version 2.38.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2), 301–320.