ebmstate: An R Package For Disease Progression Analysis Under Empirical Bayes Cox Models

by Rui J. Costa and Moritz Gerstung

Abstract The new R package ebmstate is a package for multi-state survival analysis. It is suitable for high-dimensional data and allows point and interval estimation of relative transition hazards, cumulative transition hazards and state occupation probabilities, under clock-forward and clock-reset Cox models. Our package extends the package mstate in a threefold manner: it transforms the Cox regression model into an empirical Bayes model that can handle high-dimensional data; it introduces an analytical, Fourier transform-based estimator of state occupation probabilities for clock-reset models that is much faster than the corresponding, simulation-based estimator in mstate; and it replaces asymptotic confidence intervals meant for the low-dimensional setting by non-parametric bootstrap confidence intervals. Our package supports multi-state models of arbitrary structure, but the estimators of state occupation probabilities are valid for transition structures without cycles only. Once the input data is in the required format, estimation is handled automatically. The present paper includes a tutorial on how to use ebmstate to estimate transition hazards and state occupation probabilities, as well as a simulation study showing how it outperforms mstate in higher-dimensional settings.

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 seven R packages that are able to fit *general* multi-state models and, at the same time, feature some kind of regression model or algorithm: `flexsurv` (Jackson, 2016), **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 (as defined in Aalen et al., 2008, p. 133). The only exceptions are **survival**, which also fits Aalen's additive regression model (Aalen, 1989), and `flexsurv`, which also implements accelerated failure time models (see, for example, Aalen et al., 2008, p. 443).

Recall that a Cox regression model is a semi-parametric model in which every transition hazard is assumed to be the product of a baseline hazard function of unspecified form (the non-parametric component) and an exponential relative risk function (the parametric component) (Aalen et al., 2008, p. 133). Generally, the relative risk regression models implemented in these packages are Cox regression models. However, some models in `flexsurv`, as well as those in **msm** and **SemiMarkov**, also restrict the baseline hazards to specific parametric families, i.e. they are fully parametric. 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 standard, fixed effects relative risk regression models and do not include regularisation or variable selection features.

It is also illustrative to order the seven 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 `flexsurv`, **msm** and **mstate**, and has been under implementation in **survival** (version 3.0 or later).

The present paper provides an introduction to **ebmstate**, a new R package for multi-state survival analysis available for download on the Comprehensive R Archive Network (CRAN). The main goal of **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 **ebmstate** is that it re-implements and extends the analysis framework of **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 **mstate** package. This not only eliminates redundancies, but also makes our package more accessible to the numerous users of **mstate** (the three papers associated with **mstate** have jointly over 2000 citations).

To improve the performance of **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 **mstate** are applicable when the number of observations is much larger than the number of parameters to be estimated (see section Interval estimation below). To preserve the completeness of **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 **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 **mstate**. However, for clock-reset Cox models, only a simulation-based estimator is available in this package (see also the `flexsurv` package for a similar, simulation-based estimator). The FFT estimator in **ebmstate** was conceived as a faster alternative to this simulation-based estimator, but its scope is currently restricted to multi-state models with transition structures that have no cycles, i.e. in which a transition between two states is either not possible or follows a unique sequence of states. Figure 1 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 Estimator performance on estimator performance). Also, as a ridge-type regression method, it can be used as an alternative to the lasso method of **penMSM** 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 **penMSM** and **gamboostMSM**: point and interval estimates of state occupation probabilities under the regularised Cox model can also be computed.

2 Models

A multi-state Cox model is a continuous-time stochastic process with a finite (and usually small) state space $\mathcal{S}$. 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 $\{X(t)\}$ (an adapted covariate is one whose path until t is known once $\{X(u)\}$, $u \leq t$, 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

$$\alpha_{ij}(t|\mathbf{z}, s) := \lim_{h \downarrow 0} \frac{1}{h} \mathrm{P}\left[X(t+h) = j \,|\, X(t) = i, S(t) = s, \vec{Z}(t) = \mathbf{z}\right] \ , \ s \in [0, \infty) \ , \ t \in [s, \infty) \ .$$

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

$$\alpha_{ij}(t \,|\, \mathbf{z}, s) = \lambda_{ij}^{(0)}(s) \exp\left[\boldsymbol{\beta}_{ij}^{\mathsf{T}} \mathbf{z}\right] \quad , \tag{1}$$

and it is termed a 'clock-forward' model when

$$\alpha_{ij}(t \,|\, \mathbf{z}) = \alpha_{ij}^{(0)}(t) \exp\left[\boldsymbol{\beta}_{ij}^{\mathsf{T}} \mathbf{z}\right] \quad . \tag{2}$$

In both cases, $i, j \in \mathcal{S}$, with $i \neq j$; $\boldsymbol{\beta}_{ij}$ 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, $\alpha_{ij}(t|\mathbf{z}, s)$ is the same for all $t \geq s$, we simplify its notation to $\lambda_{ij}(s|\mathbf{z})$. 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 $\exp\left[\boldsymbol{\beta}_{ij}^{\mathsf{T}} \mathbf{z}\right]$, is termed the relative transition hazard. In **mstate** and **ebmstate**, estimating the relative transition hazard amounts to estimating the regression coefficient vector $\boldsymbol{\beta}_{ij}$. In **mstate**, these parameters are assumed to be non-random. With **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 $\{\boldsymbol{\beta}_{ij}\}$, for all $(i, j) \in \mathcal{P}$, be a partition of $\boldsymbol{\beta}$, a vector containing the regression coefficients for all direct transitions allowed. Each $\boldsymbol{\beta}_{ij}$ is further partitioned into $\{\boldsymbol{\beta}_{ijk}\}$, for $k \in \{1, 2, ..., n_{ij}\}$. In **ebmstate**, the most general model regarding the prior distribution of $\boldsymbol{\beta}$ makes two assumptions: a) the scalar components of $\boldsymbol{\beta}$ are independent and normally distributed; b) the scalar components of $\boldsymbol{\beta}_{ijk}$ have a common (and undetermined) mean μ_{ijk} and a common (and also undetermined) variance σ_{ijk}^2.

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 **mstate** a Breslow estimator of two types of cumulative transition hazard: one on a global time scale, defined as

$$\mathrm{A}_{ij}\left(t \mid \mathbf{z}\right) := \int_0^t \alpha_{ij}^{(0)}\left(u\right) \exp\left[\boldsymbol{\beta}_{ij}^{\mathsf{T}} \mathbf{z}\right] \mathrm{d}u \quad ,$$

and another on a sojourn time scale, defined as

$$\Lambda_{ij}(s \mid \mathbf{z}) := \int_0^s \lambda_{ij}^{(0)}\left(u\right) \exp\left[\boldsymbol{\beta}_{ij}^{\mathsf{T}} \mathbf{z}\right] \mathrm{d}u \quad .$$

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 $\{\vec{Z}(u)\}_{u\geq 0}$ is said to be *external* if, for all $t \in [0, \infty)$, each transition hazard at t, conditional on $\vec{Z}(t)$, is independent of $\{\vec{Z}(u)\}_{u>t}$ (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 $\vec{Z}$ 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 $\boldsymbol{\mu}_{ij}$, with $(i, j) \in \mathcal{P}$ (the set of direct transitions allowed), denote a vector whose scalar components are the parameters μ_{ijk}, $k \in \{1, 2, ..., n_{ij}\}$. Similarly, let $\boldsymbol{\sigma}_{ij}^2$ be composed of the parameters $\left\{\sigma_{ijk}^2\right\}_k$. The estimation of $\boldsymbol{\beta}$, $\boldsymbol{\mu} := \{\boldsymbol{\mu}_{ij}\}$ and $\boldsymbol{\sigma}^2 := \{\boldsymbol{\sigma}_{ij}^2\}$ relies on the restricted maximum-likelihood (REML) type algorithm described in Perperoglou (2014), and introduced by Schall (1991). The resulting estimate of $\boldsymbol{\beta}$ is a maximum *a posteriori* estimate; the estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}^2$ are empirical Bayes estimates. In **ebmstate**, the estimator based on this algorithm is implemented in the function `CoxRFX` . The results of a simulation study showing its consistency are included in the Supporting Scripts and Data (file ESM_1.html, section 1).

The computation of cumulative hazard rates for given covariate values and an estimated regression coefficient vector relies on the function `msfit_generic`, which is essentially a wrapper for the function `mstate::msfit` (see section Computing cumulative transition hazard estimates). For the mathematical details of this computation, 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 $\hat{\mathrm{A}}_{ij}(\cdot \mid \mathbf{z})$ or $\hat{\Lambda}_{ij}(\cdot \mid \mathbf{z})$ 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 $\hat{\mathrm{A}}_{ij}(\cdot \mid \mathbf{z})$ 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 transition structure that has no cycles).

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), $i \in \{1, 2, ...\}$, 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 $\vec{Z}(u) = \mathbf{z}$ for all $u \geq 0$, as

$$\begin{aligned}&\mathrm{P}\left[X(t) = n \mid X(0) = 0\,, \vec{Z}(u) = \mathbf{z},\, u \geq 0\right] \\ &= \mathrm{P}\left[X_n = n, \tau_{0,n} < t, \tau_{n,n+1} \geq t - \tau_{0,n} | X_0 = 0\,, \vec{Z}(u) = \mathbf{z},\, u \geq 0\right]\,.\end{aligned}$$

Recall that $\lambda_{i,i+1}(s \mid \mathbf{z})$ 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 $\mathbf{z}$. The cumulative hazard for the same transition between sojourn times 0 and s, if the patient's covariate vector remains constant at $\mathbf{z}$, is represented by $\Lambda_{i,i+1}(s \mid \mathbf{z}) := \int_0^s \lambda_{i,i+1}(x \mid \mathbf{z})\,\mathrm{d}x$. Similarly, we let $\lambda_i(s \mid \mathbf{z})$ 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 $\mathbf{z}$. The cumulative hazard for the same event between sojourn times 0 and s, if the patient's covariate vector remains constant at $\mathbf{z}$, is represented by $\Lambda_i(s \mid \mathbf{z})$. The expressions $\hat{\Lambda}_i(s \mid \mathbf{z})$ and $\hat{\Lambda}_{i,i+1}(s \mid \mathbf{z})$ 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 $\vec{Z}(u) = \mathbf{z}$, for $u \geq 0$: this condition is omitted to simplify the notation.

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

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

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

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

and

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

and the finite difference

$$\Delta\hat{\Lambda}_{i,i+1}(t_k) := \hat{\Lambda}_{i,i+1}(t_{k+1}) - \hat{\Lambda}_{i,i+1}(t_k)\,,$$

the algorithm behind `probtrans_ebmstate` can be described as follows:

1. For $j = 1, 2, ..., n$, compute

$$\hat{q}_{j-1,j}(k) := \exp\left[-\hat{\Lambda}_{j-1}(t_k)\right]\Delta\hat{\Lambda}_{j-1,j}(t_k) \tag{3}$$

   for $k = 0, 1, ..., K-1$.

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

$$\hat{q}_{0j}(k) := \sum_{l=0}^{k-1} \hat{q}_{j-1,j}(k-l-1)\,\hat{q}_{0,j-1}(l) \tag{4}$$

   for $k = 0, 1, ..., K-1$.

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

$$\hat{p}_{0n}(k) := \sum_{l=0}^{k-1} \hat{r}_n(k-l-1)\,\hat{q}_{0,n}(l) \tag{5}$$

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

Substituting := for ≈ 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 Supporting Scripts and Data (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

$$\hat{p}_{0n} := \mathcal{F}^{-1}\left\{\hat{\mathfrak{q}}_{0,1} \cdot \hat{\mathfrak{q}}_{1,2} \cdot ... \cdot \hat{\mathfrak{q}}_{n-1,n} \cdot \hat{\mathfrak{r}}_n\right\},$$

where $\mathcal{F}$ denotes the discrete Fourier transform, $\hat{\mathfrak{q}}_{j-1,j} := \mathcal{F}(\hat{q}_{j-1,j})$ and $\hat{\mathfrak{r}}_n := \mathcal{F}(\hat{r}_n)$. Conversion to and from the frequency domain is carried out using the fast Fourier transform algorithm implemented in the `fft` function of the base package `stats`. The Supporting Scripts and Data 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 Supporting Scripts and Data (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 $\boldsymbol{\theta}$ (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**. Note that the non-parametric bootstrap can be given a Bayesian interpretation. Its interval estimates are approximately the same as those of a Bayesian model that assumes: a) a multinomial distribution for the data; and b) a non-informative Dirichlet prior distribution for the probability assigned to each category in the multinomial distribution. This is a specific case of the so-called Bayesian bootstrap (Hastie et al., 2009, p. 272). Further research is needed to determine the theoretical properties of the non-parametric bootstrap in the present setting, but this falls beyond the scope of the present paper. 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 $a \in$ {regression coefficients, relative hazards, state occupation probabilities} is the target of estimation, $m \in$ {standard Cox, empirical Bayes Cox, null} is the assumed hazard model, $G \in$ {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 $p\,(100) \in \{10, 100, 200, 300, 400, 500\}$. 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 `survival::coxph` and `ebmstate::CoxRFX` respectively; the estimation of state occupation probabilities is based on `mstate::probtrans` for the null model and on `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 $G =$'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 Supporting Scripts and Data (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 Supporting Scripts and Data). 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 `mstate` were illustrated in de Wreede et al. (2010) using a simple workflow. The starting point of this workflow is a data set in 'long format'. Such data set can be fed into `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 `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 `mstate::msfit` is fed into `mstate::probtrans`. In this section, we describe how **ebmstate** extends the scope of this workflow, i.e., how it uses the packages **survival** and **mstate** to generate estimates under a multi-state empirical Bayes Cox model. A diagram summarising the extension is shown in figure 8. In the Model assessment subsection, we give some recommendations on how to assess and compare models, but for more detailed tutorials on how to analyse multi-state data using models defined by transition hazards, we refer the reader to Putter et al. (2007) and Putter (2011).

The main steps of the **ebmstate** workflow are here illustrated using a data set of patients with myelodysplastic syndromes (MDS) which has been described and studied in Papaemmanuil et al. (2013). 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 9a 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 transition structure that has no cycles (i.e., 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 Supporting Scripts and Data (file ESM_2.html). For running only the R snippets shown below and reproduce their results, the best option is to use the R script in file ESM_3.R of the Supporting Scripts and Data.

id	from	to	trans	Tstart	Tstop	time	status	strata	ASXL1	DNMT3A	[...]
77	1	2	1	0	2029	2029	0	1	0	0	.
77	1	3	2	0	2029	2029	1	2	0	0	.
78	1	2	1	0	332	332	1	1	1	0	.
78	1	3	2	0	332	332	0	2	1	0	.
78	2	4	3	332	1449	1117	1	3	1	0	.

Table 1: A 5-row fragment of the MDS data set (in long format)

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 `Tstart[i]`, patient `id[i]` entered state `from[i]`, and thereby began to be at risk for transition `trans[i]`, i.e., at risk of going from state `from[i]` to state `to[i]`. If the first transition of patient `id[i]` after time `Tstart[i]` occurs before the last follow-up time for this patient, `Tstop[i]` records the time of this transition (regardless of whether the patient moved to state `to[i]` or not). Otherwise, `Tstop[i]` is set to the last follow-up time. The value of `status[i]` is set to 1 if and only if the first transition of patient `id[i]` after `Tstart[i]` is to state `to[i]` and occurs before the last follow-up (otherwise it is set to 0). The value of `time[i]` is defined simply as `Tstop[i]`−`Tstart[i]`, and `strata[i]` is the stratum of the baseline hazard for transition `trans[i]` (more about this variable in the following section). For `x` ∈ { `ASXL1`, `DNMT3A`, ... }, `x[i]` denotes the level of covariate `x` between `Tstart[i]` and `Tstop[i]` in patient `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 77 entered state 1 ('MDS') at time 0 and remained in this state until time 2029, when she moved to state 3 ('death before AML'). There are no rows to describe the evolution of patient 77 after entering state 3, as this state is an absorbing state. As to patient 78, she remained in state 1 until time 332, and moved from there to state 2 ('AML'). She lived with AML for 1117 days and moved to state 4 ('death after AML') at time 1449.

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:

```
outcome_covs <- c("id","from","to","trans","Tstart","Tstop","time","status",
                  "strata")
Z <- mstate_data[!names(mstate_data) %in% outcome_covs]
#(`mstate_data' has the data in long format)
```

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 are however ways 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.

```
# Columns `id' and `trans' from `mstate_data' together with the first
# two expanded covariates (patients 77 and 78):
    id trans ASXL1.1 ASXL1.2 ASXL1.3 DNMT3A.1 DNMT3A.2 DNMT3A.3  [...]
    77     1       0       0       0        0        0        0     .
```

```
77    2    0    0    0    0    0    0    .
78    1    1    0    0    0    0    0    .
78    2    0    1    0    0    0    0    .
78    3    0    0    1    0    0    0    .
```

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.

```
# 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_123 <- mstate::expand.covs(
    mstate_data,
    covs = names(mstate_data)[! names(mstate_data) %in% outcome_covs],
    append = F
)

# remove all covariates for transition 3 from `covariates_expanded_123'
# to fit a fully non-parametric model on this transition:
covariates_expanded_12 <- covariates_expanded_123[
    !grepl(".3",names(covariates_expanded_123),fixed = T)
]

#argument `Z' of coxrfx
Z_12 <- data.frame(covariates_expanded_12,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:

```
#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.

```
#argument `groups' of coxrfx
groups_12 <- paste0(rep("group",ncol(Z)-2),c("_1","_2"))

#fit random effects model
model_12 <- CoxRFX(Z_12,surv,groups_12,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 (≈ 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.

```
# Build `patient_data' data frame with the covariate values for which
# cumulative hazards are to be computed (covariate values of patient 78):
patient_data <- mstate.data[mstate.data$id == 78,,drop = F][rep(1,3),]
patient_data$strata <- patient_data$trans <- 1:3
patient_data <- mstate::expand.covs(
    patient_data,
    covs = names(patient_data)[ ! names(patient_data) %in% outcome_covs],
    append = T
)
patient_data <- patient_data[ ! grepl(".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  [...]
    78     1       1       0        0        0     .
    78     2       0       1        0        0     .
    78     3       0       0        0        0     .

# compute cumulative hazards
msfit_object_12 <- msfit_generic(model_12,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.

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 the computation of state occupation probability estimates under clock-reset models with a transition structure that has no cycles. When using `probtrans_fft` (the faster, but somewhat less stable, 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 the upper time limit for the generation of estimates (`max_time`). Both functions are based on a discrete-time approximation to a series of convolutions. The default argument `nr_steps` controls the number of (equally spaced) time steps used in this approximation. The arguments `max_time` and `nr_steps` 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_12 <- probtrans_fft("MDS",msfit_object_12, max_time = 4000)
```

Estimates are shown in figure 12, along with 95% non-parametric, bootstrap confidence intervals. For this particular patient, the estimated probability of being dead after AML remains below 0.4 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 following block of code shows how to compute confidence intervals with `boot_ebmstate`:

```
# Creating the object arguments for boot_ebmstate()

# `groups' arguments was already created, but we need to add names to it
names(groups_12) <- names(covariates_expanded_12)

# `mstate_data_expanded' argument (similar to `covariates_expanded' but
# including outcome variables)
mstate_data_expanded <- cbind(
  mstate_data[names(mstate_data) %in% outcome_covs],
  covariates_expanded_12
)

# create the non-parametric bootstrap confidence intervals
boot_ebmstate_object <- boot_ebmstate(
  mstate_data = mstate_data_expanded,
  which_group = groups_12,
  min_nr_samples = 100,
  patient_data = patient_data,
  tmat = tmat,
  initial_state = "MDS",
  time_model = "clockreset",
  input_file = NULL,
  coxrfx_args = list(max.iter = 200),
  probtrans_args = list(max_time = 4000)
)
```

5.5 Model assessment

For any model fitted with **ebmstate**, two performance metrics can be easily computed: the *concordance* statistic (Harrell et al., 1982; see also the help page of `survival::concordance` for the definition of concordance) and the *Bayesian Information Criterion* (BIC) score (Schwarz, 1978). As an example of how these two metrics can be obtained and used for model comparison, suppose we wish to compare

'model_12' fitted above – which consists of a Cox regression including all covariates for transitions 1 and 2 and a fully non-parametric model for transition 3 – with a model that combines Cox regressions of all covariates for each of the three transitions (denoted 'model_123' below). The following code snippet shows how to fit this second model.

```
# arguments `groups' and `Z' for fitting a Cox regression model on all transitions
Z_123 <- data.frame(
    covariates_expanded_123,
    strata = mstate_data$trans,
    trans = mstate_data$trans
)
groups_123 <- paste0(rep("group", ncol(Z_123) - 2), c("_1", "_2", "_3"))

# Fit a Cox regression model for all transitions
model_123 <- CoxRFX(Z = Z_123, surv = surv, groups = groups_123)
```

Running the concordance function in the **survival** package for each model yields the following output:

```
> concordance(model_12)
    Call:
    concordance.coxph(object = model_12)

    n= 1210
    Concordance= 0.8131 se= 0.01314
                concordant discordant tied.x tied.y tied.xy
    strata=1      18040       2783      0      1       0
    strata=2      37919       9678      0      7       0
    strata=3          0          0   1052      0       4

> concordance(model_123)
    Call:
    concordance.coxph(object = model_123)

    n= 1210
    Concordance= 0.8168 se= 0.01312
                concordant discordant tied.x tied.y tied.xy
    strata=1      18041       2782      0      1       0
    strata=2      37920       9677      0      7       0
    strata=3        784        268      0      4       0
```

The output shows that modelling transition 3 with a Cox model, instead of a fully parametric one, has a negligible impact on the overall concordance. However, this is due to the fact that there are far fewer observations for this transition. The concordance for transition 3 only, which corresponds to strata 3, is 0.5 under the fully parametric model (i.e., all patients are assigned the same transition hazard) and considerably higher under the Cox regression ($784/(784+268) = 0.75$). Ideally, the comparison of models of different complexity should be carried out on a test sample rather than on the training data. For this purpose, the test data can be input into to the concordance function (argument newdata). However, in the present case, only 61 patients were ever at risk of dying with AML (i.e. of undergoing transition 3), and of these only 41 actually died, so we might prefer to keep all patients in the training data, rather than saving a fraction of them for testing purposes. Such an option will yield more accurate coefficient estimates, at the expense of not allowing the computation of unbiased estimates of model performance. If the goal is only to compare models, we can make do without test data, by using an information score that penalises model complexity, such as the BIC. To facilitate model comparison, the BIC score is one of the attributes of the model fit object:

```
> model_12$BIC
    [1] 2508.37
> model_123$BIC
    [1] 2483.49
```

The best model is the one with the lowest score, so the choice of 'model_123' is confirmed.

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 Estimator performance, 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. However, 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.

Acknowledgements

The authors are supported by grant NNF17OC0027594 from the Novo Nordisk Foundation. We thank an anonymous reviewer for their constructive comments and helpful suggestions which led to a much-improved manuscript.

Supporting Scripts and Data

In the supporting Scripts and Data, 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'. The MDS data set is in files 'MDS.TPD.20Nov2012.csv' and 'mds.paper.clin.txt'. The file 'ESM_3.R' contains a simplified R script to run the code snippets in the present paper. The **ebmstate** package is available on CRAN.

7 Conflict of interest

The authors have declared no conflict of interest.

References

O. Aalen, O. Borgan, and H. Gjessing. *Survival and event history analysis*. Springer, 2008. URL https://link.springer.com/book/10.1007/978-0-387-68560-1. [p15, 17, 18, 19, 28]

O. O. Aalen. A linear regression model for the analysis of life times. *Statistics in Medicine*, 8(8):907–925, 1989. URL https://doi.org/10.1002/sim.4780080803. [p15]

P. Andersen, O. Borgan, R. Gill, and N. Keiding. *Statistical Models Based On Counting Processes*. Springer, 1993. URL https://link.springer.com/book/10.1007/978-1-4612-4348-9. [p17, 18, 19]

B. Carlin and T. Louis. *Bayesian Methods for Data Analysis*. CRC Press, 2009. URL https://doi.org/10.1201/b14884. [p20]

G. Cortese and P. K. Andersen. Competing risks and time-dependent covariates. *Biometrical Journal*, 52 (1):138–158, 2010. URL https://doi.org/10.1002/bimj.200900076. [p18]

L. C. de Wreede, M. Fiocco, and H. Putter. 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, 2010. ISSN 0169-2607. doi: https://doi.org/10.1016/j.cmpb.2010.01.001. URL http://www.sciencedirect.com/science/article/pii/S0169260710000027. [p15, 18, 22]

L. C. de Wreede, M. Fiocco, and H. Putter. mstate: An R package for the analysis of competing risks and multi-state models. *Journal of Statistical Software*, 38(7):1–30, 2011. URL http://www.jstatsoft.org/v38/i07/. [p17]

A. Gelman, J. Carlin, H. Stern, D. Dunson, A. Vehtari, and D. Rubin. *Bayesian Data Analysis*. CRC Press, 2014. URL https://doi.org/10.1201/b16018. [p20]

J. Harrell, Frank E., R. M. Califf, D. B. Pryor, K. L. Lee, and R. A. Rosati. Evaluating the Yield of Medical Tests. *JAMA*, 247(18):2543–2546, 05 1982. ISSN 0098-7484. doi: 10.1001/jama.1982.03320430047030. URL https://doi.org/10.1001/jama.1982.03320430047030. [p26]

T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer, 2009. URL https://link.springer.com/book/10.1007/978-0-387-84858-7. [p20]

T. Hothorn, P. Buehlmann, T. Kneib, M. Schmid, and B. Hofner. mboost: Model-based boosting. *R package version*, pages 2.9–3, 2020. URL https://CRAN.R-project.org/package=mboost. [p15]

P. Hougaard. Multi-state models: a review. *Lifetime data analysis*, 5(3):239–264, 1999. URL https://doi.org/10.1023/A:1009672031531. [p15]

C. Jackson. flexsurv: A platform for parametric survival modeling in R. *Journal of Statistical Software*, 70(8):1–33, 2016. doi: 10.18637/jss.v070.i08. [p15]

C. H. Jackson. Multi-state models for panel data: the msm package for R. *Journal of Statistical Software*, 38(8):1–29, 2011. URL http://www.jstatsoft.org/v38/i08/. [p15]

J. D. Kalbfleisch and R. L. Prentice. *The statistical analysis of failure time data*. John Wiley & Sons, 2002. doi: 10.1002/9781118032985. [p18]

A. Listwon and P. Saint-Pierre. SemiMarkov: An R Package for Parametric Estimation in Multi-State Semi-Markov Models. *Journal of Statistical Software*, 66(6):784, 2015. doi: 10.18637/jss.v066.i06. URL https://hal.archives-ouvertes.fr/hal-00860244. [p15]

E. Papaemmanuil, M. Gerstung, L. Malcovati, S. Tauro, G. Gundem, P. Van Loo, C. J. Yoon, P. Ellis, D. C. Wedge, A. Pellagatti, et al. Clinical and biological implications of driver mutations in myelodysplastic syndromes. *Blood*, 122(22):3616–3627, 2013. URL https://doi.org/10.1182/blood-2013-08-518886. [p22]

A. Perperoglou. Cox models with dynamic ridge penalties on time-varying effects of the covariates. *Statistics in Medicine*, 33(1):170–180, 2014. URL https://doi.org/10.1002/sim.5921. [p16, 18]

H. Putter. Tutorial in biostatistics: Competing risks and multi-state models analyses using the mstate package. *Companion file for the mstate package*, 2011. URL https://mirror.las.iastate.edu/CRAN/web/packages/mstate/vignettes/Tutorial.pdf. [p22]

H. Putter, M. Fiocco, and R. B. Geskus. Tutorial in biostatistics: competing risks and multi-state models. *Statistics in Medicine*, 26(11):2389–2430, 2007. URL https://doi.org/10.1002/sim.2712. [p22]

H. Reulen. gamboostmsm. *R package version*, page 1.1.87, 2014. URL https://CRAN.R-project.org/package=gamboostMSM. [p15]

H. Reulen. penmsm. *R package version*, page 0.99, 2015. URL https://CRAN.R-project.org/package=penMSM. [p15]

R. J. Samworth. Stein's paradox. *Eureka*, 62:38–41, 2012. URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7eebd55f569395544f2b5d367d6aee614901d2c1. [p21]

R. Schall. Estimation in generalized linear models with random effects. *Biometrika*, 78(4):719–727, 1991. doi: 10.1093/biomet/78.4.719. URL http://dx.doi.org/10.1093/biomet/78.4.719. [p16, 18]

G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978. URL https://www.jstor.org/stable/2958889. [p26]

C. Spitoni, M. Verduijn, and H. Putter. Estimation and asymptotic theory for transition probabilities in markov renewal multi-state models. *The International Journal of Biostatistics*, 8(1), 2012. doi: doi:10.1515/1557-4679.1375. URL https://doi.org/10.1515/1557-4679.1375. [p16]

T. M. Therneau. *A Package for Survival Analysis in S*, 2015. URL https://CRAN.R-project.org/package=survival. version 2.38. [p15]

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005. doi: https://doi.org/10.1111/j.1467-9868.2005.00503.x. URL https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-9868.2005.00503.x. [p16]

Rui J. Costa
European Molecular Biology Laboratory
European Bioinformatics Institute (EMBL-EBI)
Hinxton, CB10 1SD
United Kingdom
ruibarrigana@hotmail.com

Moritz Gerstung
aff. 1: European Molecular Biology Laboratory
European Bioinformatics Institute (EMBL-EBI)
Hinxton, CB10 1SD
United Kindom
aff. 2: German Cancer Research Center (DKFZ)
Im Neuenheimer Feld 280
69120 Heidelberg
Germany
moritz.gerstung@dkfz.de

Figures

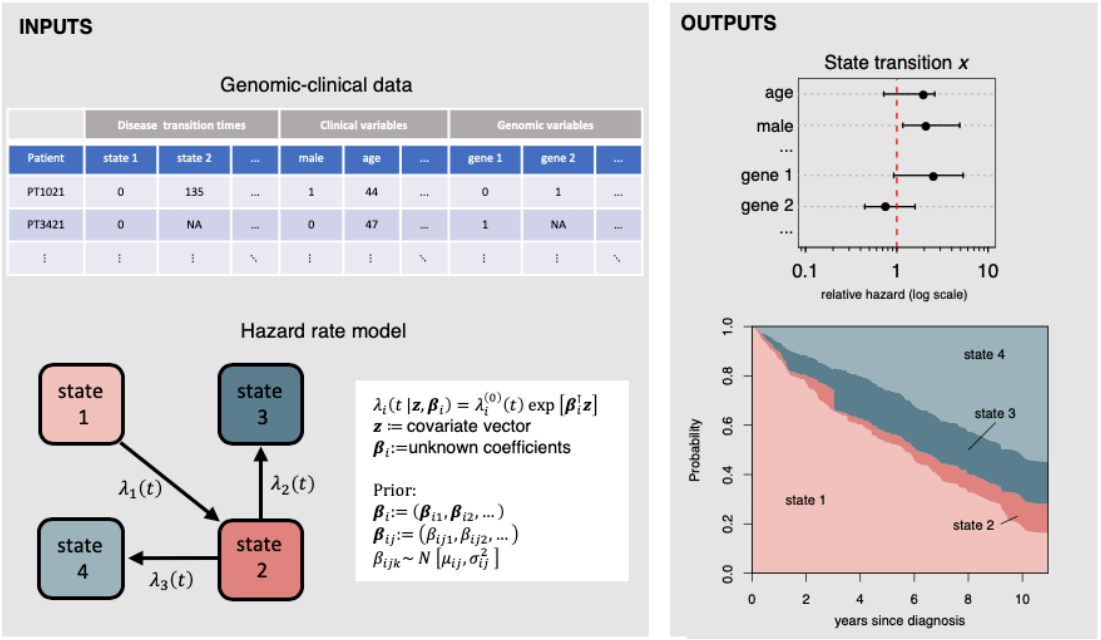


Figure 1: Summary of inputs and outputs of the package `ebmstate`. The input data set should be one that violates the assumption – commonly used in survival analysis – that the number of observations is much larger than the number of parameters to be estimated (a genomic-clinical data set is shown as 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.

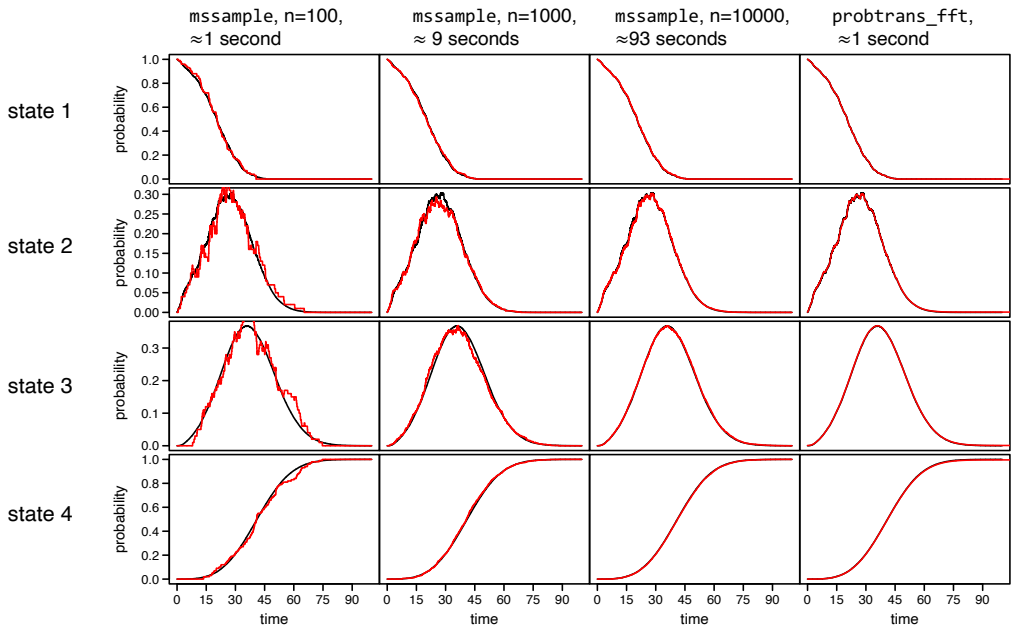


Figure 2: Comparison of running times and estimation accuracy of `mssample` and `probtrans_fft`. Each plot in the grid shows two estimated curves of state occupation probabilities. The black curves are based on a single run of `mstate::mssample` with $n = 100.000$ observations (approximately 17 minutes of running time) and are the same across columns. They serve as benchmark for precision assessment. In columns 1 to 3 of the grid, the superimposed red curves are based on a run of `mssample` with respectively 100, 1000, and 10.000 observations. In the rightmost column, the red curves are based on a run of `probtrans_fft`. All functions have as input the same set of cumulative transition hazards. These were estimated using a non-parametric multi-state model and a data set of 1000 patients generated according to a clock-reset Cox model with a 'linear' transition structure (leftmost diagram of figure 3). Plots in the same row refer to the same state of the model, while those in the same column refer to the same run of a function. Running times and, where appropriate, number of simulations (n) are given on top of each column.

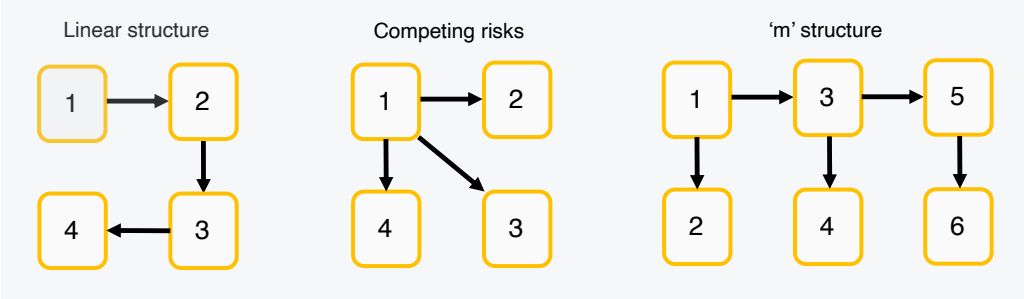


Figure 3: Model transition structures. We studied the performance of Cox model estimators, empirical Bayes Cox model estimators and fully non-parametric estimators with respect to these 3 transition structures.

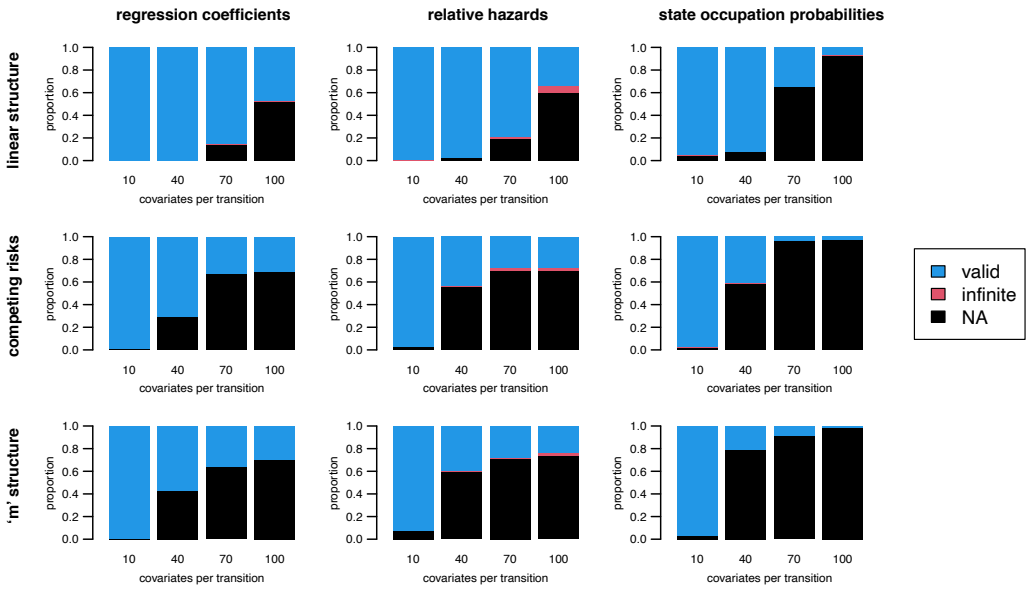


Figure 4: Proportions of valid, infinite and missing ('NA') estimates for the standard Cox model estimators in the simulation study of figure 6 (100 patients per simulated data set).

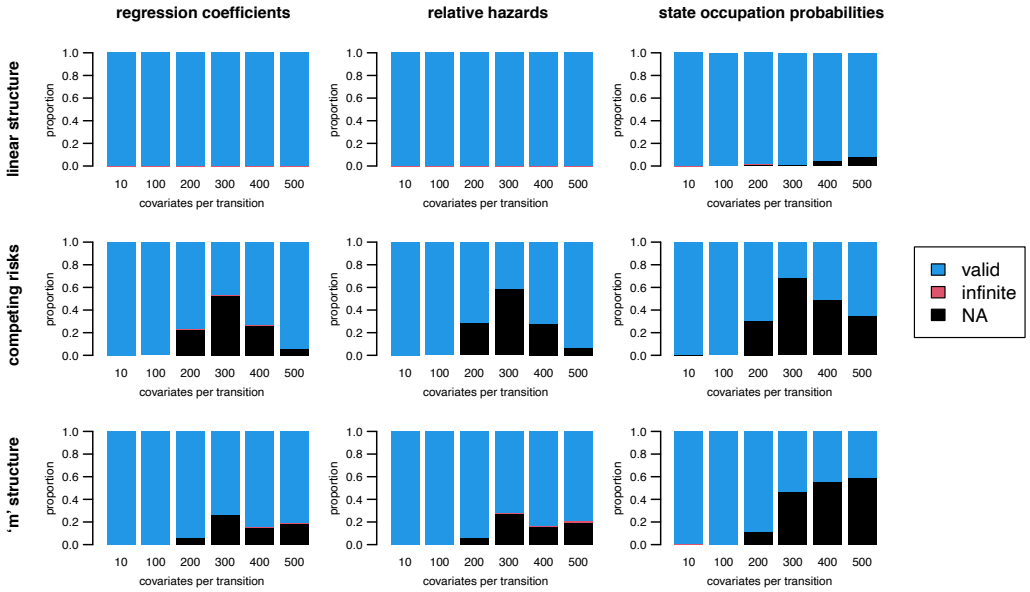


Figure 5: Proportions of valid, infinite and missing ('NA') estimates for the standard Cox model estimators in the simulation study of figure 7 (1000 patients per simulated data set).

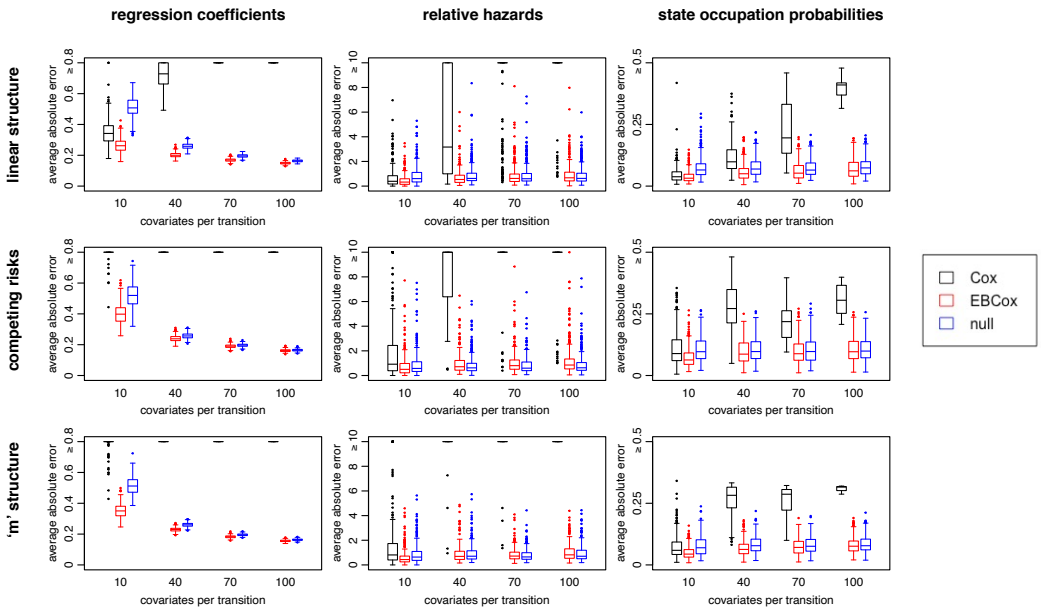


Figure 6: Performance comparison of standard Cox, empirical Bayes Cox, and fully non-parametric (null) estimators using training data sets with **100 observations** each. In the figure grid there is a boxplot corresponding to every tuple (a, m, G, p) such that $a \in \{$regression coefficients, relative hazards, state occupation probabilities$\}$ is the target of estimation, $m \in \{$standard Cox, empirical Bayes Cox, null$\}$ is the hazard model, $G \in \{$linear, competing risks, 'm' structure$\}$ is the transition structure of the model, and $p \in \{10, 40, 70, 100\}$ is the number of coefficients/covariates per transition. Each boxplot is based on at most 300 average absolute error observations. Figure 4, together with figures 6.1 and 6.3 in file ESM_1.html of the Supporting Scripts and Data, show the proportion of valid, missing and infinite estimates for each estimator. In each simulation scenario, the upper limit of the plot's y-axis defines a threshold above which observations are considered very large. Very large observations were replaced by the y-axis upper limit before the boxplots were built.

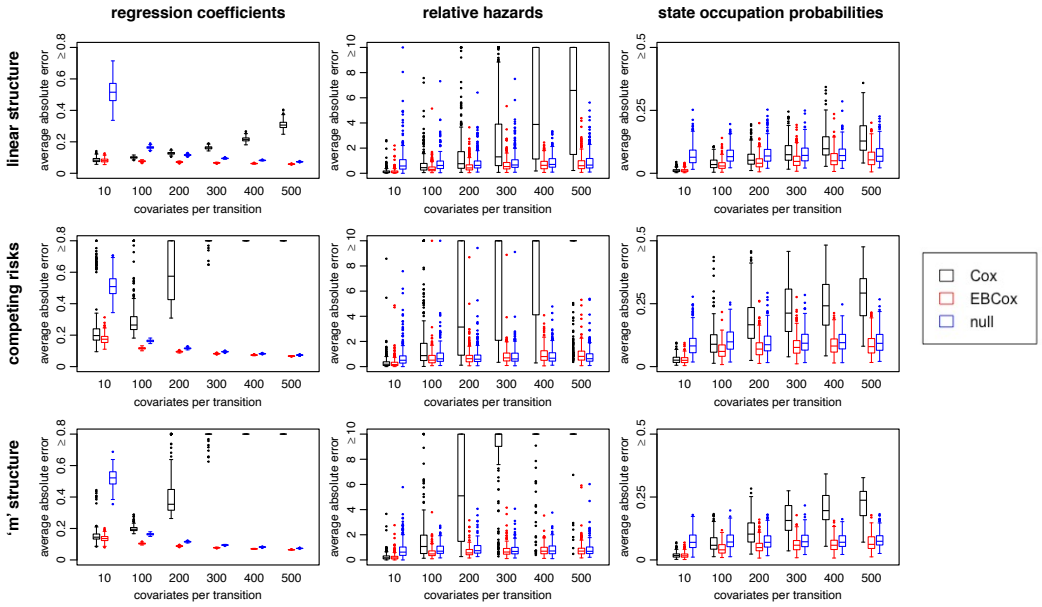


Figure 7: Performance comparison of standard Cox, empirical Bayes Cox, and fully non-parametric (null) estimators using training data sets with **1000 observations** each. In the figure grid there is a boxplot corresponding to every tuple (a, m, G, p) such that $a \in \{$regression coefficients, relative hazards, state occupation probabilities$\}$ is the target of estimation, $m \in \{$standard Cox, empirical Bayes Cox, null$\}$ is the hazard model, $G \in \{$linear, competing risks, 'm' structure$\}$ is the transition structure of the model, and $p \in \{10, 100, 200, 300, 400, 500\}$ is the number of coefficients/covariates per transition. Each boxplot is based on at most 300 average absolute error observations. Figure 5, together with figures 6.2 and 6.3 in file ESM_1.html of the Supporting Scripts and Data, show the proportion of valid, missing and infinite estimates for each estimator. In each simulation scenario, the upper limit of the plot's y-axis defines a threshold above which observations are considered very large. Very large observations were replaced by the y-axis upper limit before the boxplots were built.

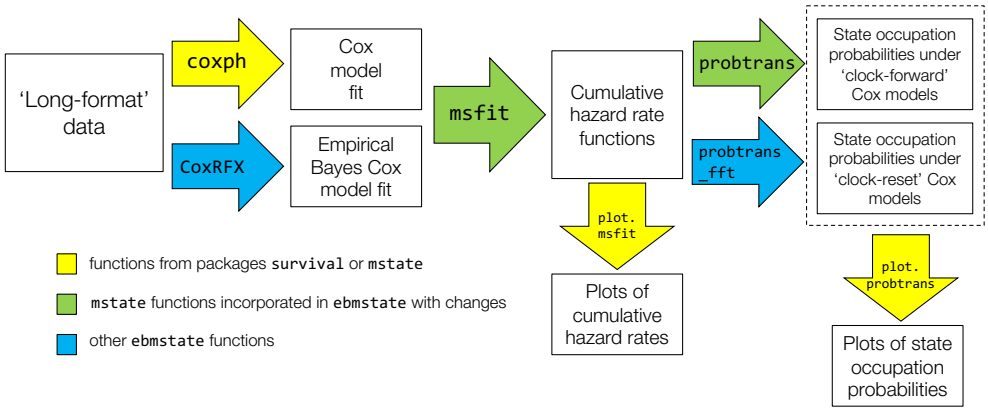
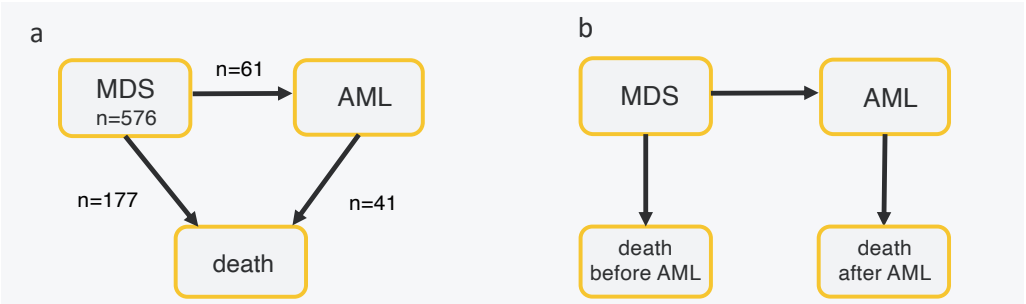


Figure 8: Extension of the `mstate` analysis framework by `ebmstate`. Arrows correspond to functions. Boxes correspond to inputs or outputs of functions. Functions `CoxRFX` and `probtrans_fft` from `ebmstate` compute point estimates only. Interval estimates can be obtained using the non-parametric bootstrap algorithm implemented in the function `ebmstate::boot_ebmstate`.



c

covariate	minimum	maximum	mean	std dev	data type	transformation
ASXL1 (point mutation)	0	1	0.13	0.33	binary (yes/no)	none
DNMT3A (point mutation)	0	1	0.09	0.28	binary (yes/no)	none
RUNX1 (point mutation)	0	1	0.07	0.26	binary (yes/no)	none
SF3B1 (point mutation)	0	1	0.27	0.45	binary (yes/no)	none
SRSF2 (point mutation)	0	1	0.14	0.35	binary (yes/no)	none
TET2 (point mutation)	0	1	0.25	0.43	binary (yes/no)	none
TP53 (point mutation)	0	1	0.05	0.22	binary (yes/no)	none
U2AF1 (point mutation)	0	1	0.06	0.23	binary (yes/no)	none
del5q mutation	0	1	0.1	0.31	binary (yes/no)	none
complex mutations	0	1	0.11	0.31	binary (yes/no)	none
other mutations	0	1	0.06	0.24	binary (yes/no)	none
age	17	95	67.42	12.61	numeric (years)	log
sex	0	1	0.44	0.5	binary (yes/no)	none
peripheral blood cytopenias	0	1	0.5	0.5	binary (yes/no)	none
hemoglobin level	3.2	17.6	9.88	1.96	numeric (g/dL)	none
absolute neutrophil count	0	38.2	3.08	3.65	numeric (count per nanolitre)	log
platelet count	5	1478	197	187.32	numeric (count per nanolitre)	log
myeloblasts in bone marrow	0.03	0.64	0.07	0.07	numeric (proportion)	logit
ring sideroblasts in bone marrow	0.03	0.93	0.17	0.23	numeric (proportion)	logit
International Prognostic Scoring System	1	4	1.79	0.84	ordinal (score)	none
WHO class RA	0	1	0.18	0.38	binary (yes/no)	none
WHO class RARS	0	1	0.13	0.34	binary (yes/no)	none
WHO class RCMD	0	1	0.16	0.37	binary (yes/no)	none
WHO class RCMD.RS	0	1	0.1	0.3	binary (yes/no)	none
WHO class RAEB	0	1	0.23	0.42	binary (yes/no)	none
WHO class CMML	0	1	0.1	0.3	binary (yes/no)	none
treatment center 1	0	1	0.12	0.33	binary (yes/no)	none
treatment center 3	0	1	0.54	0.5	binary (yes/no)	none
treatment center 4	0	1	0.28	0.45	binary (yes/no)	none
date of diagnosis	15/04/1981	18/01/2012	04/08/2004	NA	numeric	time since 15/04/1981 in units of 5 years

Figure 9: **a**: transition model implied by the data set of patients with myelodysplastic syndromes, together with transition event numbers; **b**: conversion to a transition structure without cycles; **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*.

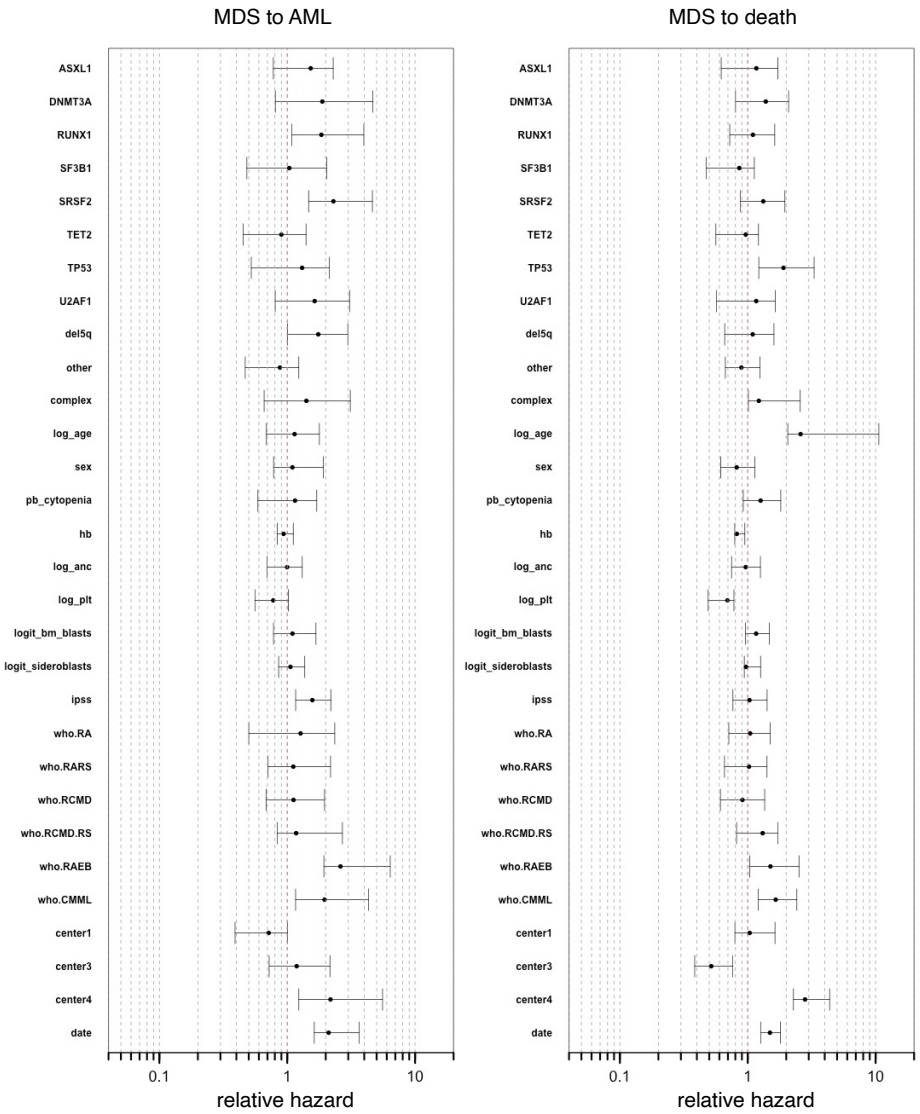


Figure 10: Point estimates of regression coefficients for the Cox model fitted to the MDS data, along with 95% non-parametric bootstrap confidence intervals. The x-axis scale is logarithmic so that coefficient estimates can be read as relative hazard estimates. If γ_{ij} is the element of $\hat{\boldsymbol{\beta}}_{ij}$ associated with a given covariate, $\exp\left(\gamma_{ij}\right)$ 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.

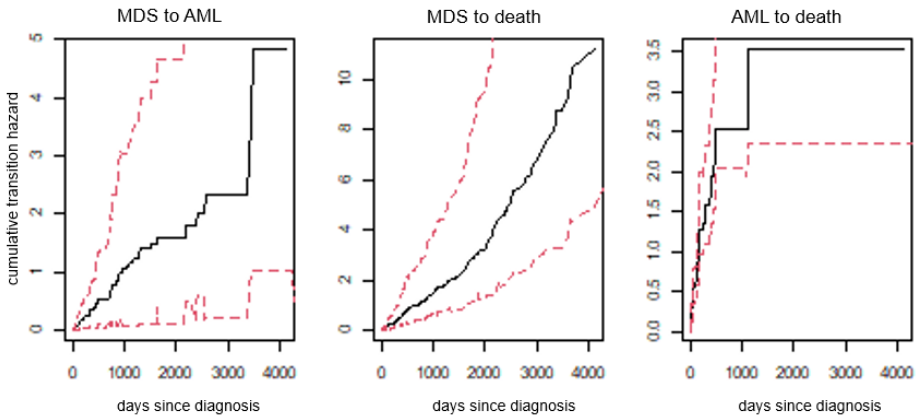


Figure 11: Point estimates of cumulative transition hazards for a sample patient with MDS (black curve), along with 95% non-parametric confidence intervals (dashed red lines).

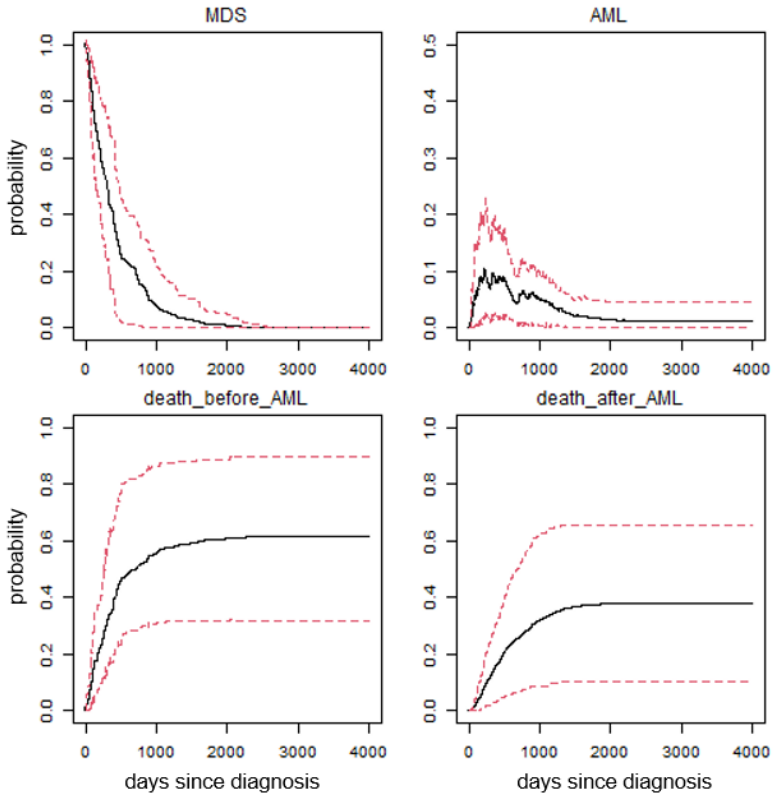


Figure 12: Point estimates of state occupation probabilities for a sample patient with MDS (black curve), along with 95% non-parametric confidence intervals (dashed red lines).