



Semi-Parametric Joint Modeling of Survival and Longitudinal Data: The R Package JSM

Cong Xu

Southern University of
Science and Technology

Pantelis Z. Hadjipantelis

University of
California, Davis

Jane-Ling Wang

University of
California, Davis

Abstract

This paper is devoted to the R package **JSM** which performs joint statistical modeling of survival and longitudinal data. In biomedical studies it has been increasingly common to collect both baseline and longitudinal covariates along with a possibly censored survival time. Instead of analyzing the survival and longitudinal outcomes separately, joint modeling approaches have attracted substantive attention in the recent literature and have been shown to correct biases from separate modeling approaches and enhance information. Most existing approaches adopt a linear mixed effects model for the longitudinal component and the Cox proportional hazards model for the survival component. We extend the Cox model to a more general class of transformation models for the survival process, where the baseline hazard function is completely unspecified leading to semi-parametric survival models. We also offer a non-parametric multiplicative random effects model for the longitudinal process in **JSM** in addition to the linear mixed effects model. In this paper, we present the joint modeling framework that is implemented in **JSM**, as well as the standard error estimation methods, and illustrate the package with two real data examples: a liver cirrhosis data and a Mayo Clinic primary biliary cirrhosis data.

Keywords: B-splines, EM algorithm, multiplicative random effects, semi-parametric models, transformation model.

1. Introduction

In medical studies it is common to record important longitudinal measurements (e.g., biomarkers) up to a possibly censored event time (or survival time) along with additional baseline covariates collected at the onset of the study. The objectives of investigations based on this kind of data often are to characterize the effects of the longitudinal biomarkers as well as baseline covariates on the survival time to understand the pattern of changes of the longitu-

dinal biomarkers, and to study the joint evolution of the longitudinal and survival processes. Examples that appear frequently in the literature are HIV (human immunodeficiency virus) clinical trials where the CD4 (cluster of differentiation 4) counts are collected at baseline and subsequent clinical visits and the event of interest is progression to AIDS (acquired immunodeficiency syndrome) or death. Many existing methods are well-suited to analyze the two outcomes separately, however, difficulties arise when: (1) the longitudinal measurements are only available intermittently or may be contaminated by measurement errors, and (2) occurrence of the event, such as death, results in patient's informative drop-out from the longitudinal study. To overcome these difficulties and provide valid inference, jointly modeling the survival and longitudinal processes is necessary and has become a mainstream research topic.

Under the joint modeling framework, the hazard for the survival time and the model for the longitudinal process are assumed to depend jointly on some latent random effects. Many different methods have been proposed for parameter estimation, e.g., the two-stage method (Tsiatis, Degruittola, and Wulfsohn 1995), the Bayesian approach (Faucett and Thomas 1996; Xu and Zeger 2001; Brown and Ibrahim 2003), the maximum likelihood approach (Wulfsohn and Tsiatis 1997; Song, Davidian, and Tsiatis 2002; Hsieh, Tseng, and Wang 2006) and the conditional score approach (Tsiatis and Davidian 2001). For the maximum likelihood approach, computations are typically performed by the EM (expectation and maximization) algorithm (Dempster, Laird, and Rubin 1977). Several excellent review papers, e.g., Tsiatis and Davidian (2004), Yu, Law, Taylor, and Sandler (2004), and Gould *et al.* (2015), are available. The edited volume by Verbeke and Davidian (2008), a 2012 book by Rizopoulos (2012), a special issue in *Statistical Methods in Medical Research* by Rizopoulos and Lesaffre (2014), and a recent book by Elashoff, Li, and Li (2016) provide additional perspectives of the joint modeling approaches.

Parametric assumptions on the baseline hazard functions of the survival times are common in the joint modeling literature to facilitate likelihood inference. These lead to tractable computation and are practical when the assumptions can be verified. The joint modeling implementation in SAS (SAS Institute Inc. 2013) and WinBUGS (Lunn, Thomas, Best, and Spiegelhalter 2000; Spiegelhalter, Thomas, Best, and Lunn 2003) as discussed by Guo and Carlin (2004) are based on such parametric assumptions, where the baseline hazard functions are assumed to follow Weibull distributions. To avoid the strong model assumptions and extra effort of goodness-of-fit tests, some later software tools proposed to approximate the baseline hazard function by piecewise constant or spline functions, e.g., the SAS macro JMFIt (Zhang, Chen, Ibrahim, Boye, and Shen 2016), R (R Core Team 2020) packages JM (Rizopoulos 2010), JMBayes (Rizopoulos 2016), lcmm (Proust-Lima, Philipps, and Lique 2017; Proust-Lima, Philipps, Diakite, and Lique 2019) and frailtypack (Rondeau, Mazroui, and Gonzalez 2012; Król, Mauguen, Mazroui, Laurent, Michiels, and Rondeau 2017), as well as Stata (StataCorp 2019) module stj (Crowther, Abrams, and Lambert 2013). Both piecewise constant and spline function approximations are in the spirit of non-parametrics and the survival component can indeed be modeled non-parametrically if the number of knots is allowed to increase with the sample size when selected by a model selection criterion. However, as currently implemented in these software packages, the tuning parameters (e.g., the number and locations of the knots) for the survival component are fixed and not data adaptive. Therefore, the approximations correspond to flexible parametric models, which have their merits but may contain non-ignorable bias. Some other software packages, e.g., R packages

joineR (Philipson, Sousa, Diggle, Williamson, Kolamunnage-Dona, and Henderson 2020) and **joineRML** (Hickey, Philipson, Jorgensen, and Kolamunnage-Dona 2018), adopted the Cox proportional hazards model (Cox 1972) for the survival times, leaving the baseline hazard completely unspecified. However, **joineR** and **joineRML** estimate the standard errors of the regression parameters through the bootstrap method of Efron (1994), which is computationally intensive and may suffer from convergence problems. The package **JM** also has an option to apply a completely unspecified baseline hazard, but it underestimates the standard errors of the regression parameters in that case.

In light of this and the difficulty to choose a parametric baseline hazard function, we follow the semi-parametric spirit of the Cox model to focus on joint modeling with semi-parametric models for the survival time and provide valid and computationally efficient standard error estimates in the **JSM** package (Xu, Hadjipantelis, and Wang 2020) which is freely available from the Comprehensive R Archive Network (CRAN) at <https://CRAN.R-project.org/package=JSM>. The acronym **JSM** stands for “joint statistical modeling” and reflects the fact that the package explores many different joint models. The semi-parametric models are not restricted to the Cox proportional hazards model and can be chosen from a class of transformation models presented in Section 2. For the longitudinal process, both a linear mixed effects model and a non-parametric multiplicative random effects model (Ding and Wang 2008) are incorporated. Since it might be difficult to detect the time trends in the linear mixed effects model, just like **JM**, **JSM** also allows users to employ B-spline basis functions as an option to model the time trend of the longitudinal data. This adds flexibility to an otherwise parametric approach for the longitudinal component. An additional feature of **JSM** is that we also include two ways to link the longitudinal and survival processes, one where the longitudinal process serves as a covariate of the survival outcome and another setting where the longitudinal and survival data each have their own model but are linked together by sharing some common random effects. This results in four different types of joint models between the survival and longitudinal components. Details are given in Section 2, where model fitting and estimation are presented. Precision estimation for the procedures is presented in Section 3, where two standard error estimation approaches are described. These are the profile likelihood method by Murphy and Van der Vaart (1999, 2000) and the method of numerically differentiating the profile Fisher score by Xu, Baines, and Wang (2014). Details about the implementation of the algorithms in **JSM** and two illustrating data examples (a liver cirrhosis data and a primary biliary cirrhosis data) are provided in Sections 4 and 5, respectively. Finally, Section 6 concludes with a short summary and discussion.

2. Joint modeling framework

2.1. Model setup

In Wulfsohn and Tsiatis (1997), the longitudinal and survival outcomes are modeled with a linear mixed effects model and a Cox proportional hazards model, respectively. We adopt a more flexible framework here as described below.

For subject i ($i = 1, 2, \dots, n$), let T_i be the event time which may be right censored by the censoring time C_i . Thus, instead of observing T_i , we are only able to observe $V_i = \min(T_i, C_i)$ and the censoring indicator $\Delta_i = I(T_i \leq C_i)$. Let $Y_i(t)$ denote the longitudinal

outcome for subject i evaluated at time t which is the sum of a “true” underlying value $m_i(t)$ and a measurement error. The longitudinal measurements are recorded intermittently at $\mathbf{t}_i = (t_{i1}, t_{i2}, \dots, t_{in_i})$ so that the observed data are $\mathbf{Y}_i = Y_i(\mathbf{t}_i) = (Y_{i1}, Y_{i2}, \dots, Y_{in_i})$ with $Y_{ij} = m_i(t_{ij}) + \varepsilon_{ij}$ and $\varepsilon_{ij} \stackrel{\text{iid}}{\sim} \mathcal{N}(0, \sigma_e^2)$. Two models for $m_i(t)$ are incorporated. The first model is the frequently used linear mixed effects model

$$m_i(t) = \mathbf{X}_i^\top(t)\boldsymbol{\beta} + \mathbf{Z}_i^\top(t)\mathbf{b}_i, \quad (1)$$

where $\mathbf{X}_i(t)$ and $\mathbf{Z}_i(t)$ are vectors of observed covariates for the fixed and random effects, respectively. Each of the covariates in $\mathbf{X}_i(t)$ and $\mathbf{Z}_i(t)$ can be either time-independent or time-dependent. The vector $\boldsymbol{\beta}$ denotes the unknown regression coefficients for the fixed effects while the vector $\mathbf{b}_i$ denotes the random effects. Model (1) is an additive model with users specifying what kind of covariates $\mathbf{X}_i$ and $\mathbf{Z}_i$ they prefer. We note here that users can choose to model both covariates through B-spline bases with their choice of knots or through a model selection criterion. Thus, model (1) can be adapted to be a non-parametric approach.

The second model incorporated in **JSM** is the non-parametric multiplicative random effects model proposed in [Ding and Wang \(2008\)](#)

$$m_i(t) = b_i \times \mathbf{B}^\top(t)\boldsymbol{\gamma} \text{ and } \mathbf{E}(b_i) = 1, \quad (2)$$

where $\mathbf{B}(t) = (B_1(t), \dots, B_L(t))^\top$ is a vector of B-spline basis functions and $\boldsymbol{\gamma}$ is the corresponding regression coefficient vector. The number of basis functions involved in $\mathbf{B}(t)$ (i.e., L) is not fixed and instead is chosen by some model selection criterion, e.g., AIC (Akaike information criterion) or BIC (Bayesian information criterion). Hence, this makes model (2) “non-parametric” in the sense that the mean function of $m_i(t)$ is modeled non-parametrically. Note that the b_i in (2) is different from the $\mathbf{b}_i$ in (1): $\mathbf{b}_i$ is a vector of random effects while b_i is a single scalar random effect which dramatically lowers the computational cost. In model (2), $\mathbf{B}^\top(t)\boldsymbol{\gamma}$ is the population mean and each subject is assumed to have a longitudinal profile proportional to it with b_i capturing the variation. Such a model may seem restrictive at first glance but is applicable in many real life examples and has the advantage that it allows for non-parametric fixed effects. For instance, it is applicable when the major mode of variation is along the population mean function and the first functional principal component ([Yao, Müller, and Wang 2005](#)) explains a high proportion of the total variation of the data. This is illustrated by the primary biliary cirrhosis data in Section 5, where the first functional principal component explains about 80% of the total variation. To make the model more applicable, we also allow for the inclusion of baseline covariates as columns of $\mathbf{B}(t)$ in addition to the B-spline basis functions.

To complete the model specification of the longitudinal processes, we need to pose proper distribution assumptions on $\mathbf{b}_i$ in (1) and b_i in (2). Assuming normality for $\mathbf{b}_i$ and b_i is a standard choice which provides computational advantages as will be described in Section 2.3. Although this is a parametric assumption, simulation studies reported in [Song et al. \(2002\)](#) suggest that the maximum likelihood estimates in these joint models are remarkably robust against departures from normal random effects. A theoretical explanation for such robustness properties was later provided in [Hsieh et al. \(2006\)](#). Therefore, we assume $\mathbf{b}_i \sim \mathcal{N}(\mathbf{0}, \Sigma_b)$ and $b_i \sim \mathcal{N}(1, \sigma_b^2)$ in our **JSM** package.

For the survival processes, the Cox proportional hazards model is used most frequently where the hazard functions are modeled as

$$\lambda(t|\mathbf{b}_i, m_i(t), \mathbf{W}_i(t)) = \lambda_0(t) \exp\{\mathbf{W}_i^\top(t)\boldsymbol{\phi} + \alpha m_i(t)\},$$

where $\mathbf{W}_i(t)$ is also a vector of observed covariates that can include both baseline covariates, which are constant overtime t , and other longitudinal covariates that were observed continuously without errors, such as interactions between treatments and time effects. In reality, the proportional hazards assumption may be violated. For example, it is a common phenomenon in clinical studies that patients under a more aggressive treatment may suffer elevated risk of death in the beginning but benefit in the long term if they manage to tolerate the treatment. In such cases, the proportional hazards assumption does not hold and more flexible models for the survival processes are needed. The transformation model proposed by [Dabrowska and Doksum \(1988\)](#) and studied in [Zeng and Lin \(2007\)](#) is a general class of models which includes both the Cox proportional hazards model and the proportional odds model ([Bennett 1983](#)) as special cases; so we consider such a general approach in **JSM**. We make a remark here about the terminology “transformation model”. There are several transformation models in the literature but we adopt the original name proposed in [Dabrowska and Doksum \(1988\)](#), which is the most commonly used designation in the survival literature. For a strictly increasing and continuously differentiable function $G(\cdot)$, the cumulative hazard functions in the transformation model ([Zeng and Lin 2007](#)) are modeled as:

$$\Lambda(t|\mathbf{b}_i, m_i(t), \mathbf{W}_i(t)) = G \left[\int_0^t \lambda_0(s) \exp\{\mathbf{W}_i^\top(s)\boldsymbol{\phi} + \alpha m_i(s)\} ds \right]. \quad (3)$$

Common choices for $G(\cdot)$ include the Box-Cox transformations and the logarithmic transformations. In our **JSM** package, we focus on the logarithmic transformations

$$G(x) = \frac{\log(1 + \rho x)}{\rho}, \quad \rho \geq 0, \quad (4)$$

where $\rho = 0$ corresponds to $G(x) = x$, yielding the Cox proportional hazards model, and $\rho = 1$ corresponds to $G(x) = \log(1 + x)$ yielding the proportional odds model. For general $\rho > 0$, it yields the “proportional γ odds model” ([Dabrowska and Doksum 1988](#)). Here we use ρ instead of γ since γ has already been used in (2). Note that we leave the baseline hazard $\lambda_0(\cdot)$ completely unspecified, as discussed in Section 1, so the resulting survival model is a very flexible semi-parametric model.

Lastly, we enlarge the scope of **JSM** by introducing another type of dependency between the survival and longitudinal outcomes, which differs from model (3). In (3), the entire longitudinal process (free of error) enters the survival model as a covariate. In **JSM** we also incorporate the dependency where the survival and longitudinal outcomes only share the same random effects:

$$\Lambda(t|\mathbf{b}_i, \mathbf{Z}_i(t), \mathbf{W}_i(t)) = G \left[\int_0^t \lambda_0(s) \exp\{\mathbf{W}_i^\top(s)\boldsymbol{\phi} + \alpha \mathbf{Z}_i^\top(s)\mathbf{b}_i\} ds \right], \quad (5)$$

and

$$\Lambda(t|b_i, \mathbf{W}_i(t)) = G \left[\int_0^t \lambda_0(s) \exp\{\mathbf{W}_i^\top(s)\boldsymbol{\phi} + \alpha b_i\} ds \right], \quad (6)$$

with (5) and (6) corresponding to longitudinal models (1) and (2), respectively. Such a shared random effects model is quite common and has been adopted by [Henderson, Diggle, and Dobson \(2000\)](#) among others. The choice between (3) and (5)/(6) depends on the primary focus of the analysis. If the focus is on the survival time conditional on the longitudinal measurements, then (3) is typically used. If the focus is on the joint evolution of the longitudinal and

survival processes, then (5)/(6) would be a reasonable choice. Altogether package **JSM** offers four different types of joint models between the survival and longitudinal data. Specifically, the four types are: (i) survival model (3) and longitudinal model (1), (ii) survival model (3) and longitudinal model (2), (iii) survival model (5) and longitudinal model (1), and (iv) survival model (6) and longitudinal model (2).

2.2. Likelihood formulation

In the **JSM** package, we focus on the maximum likelihood approach and follow the model setup in Section 2.1. In the following, we illustrate only one of the four joint models discussed in Section 2.1. The likelihood based on the joint model of form (1) and (3) and the corresponding EM algorithm are derived as an illustrating example, those based on other joint models follow analogously.

Let $\mathbf{O}_i = (\mathbf{Y}_i, \mathbf{X}_i, \mathbf{Z}_i, \mathbf{W}_i, V_i, \Delta_i)$ and $\mathbf{C}_i = (\mathbf{O}_i, \mathbf{b}_i)$ denote the observed and complete data, respectively. The parameter to be estimated is $\boldsymbol{\psi} = (\boldsymbol{\theta}, \Lambda_0)$ with $\boldsymbol{\theta} = (\boldsymbol{\phi}, \alpha, \beta, \sigma_e^2, \Sigma_b)$ being the finite dimensional (p -dimensional) parameter and Λ_0 being the non-parametric cumulative baseline hazard. To derive the joint-likelihood function, the following common assumptions are made: (i) the measurement errors ε_{ij} and the random effects $\mathbf{b}_i$ are independent; (ii) the censoring time and longitudinal measurement times are both non-informative; (iii) the random effect $\mathbf{b}_i$ fully captures the association between the longitudinal and survival outcomes (i.e., the longitudinal and survival outcomes are conditionally independent given $\mathbf{b}_i$). Then the joint likelihood for the observed data can be expressed as

$$\mathcal{L}_n(\boldsymbol{\psi}|\mathbf{O}) = f_n(\mathbf{O}|\boldsymbol{\psi}) = \prod_{i=1}^n \int f(\mathbf{Y}_i|\mathbf{b}_i; \boldsymbol{\theta}) f(V_i, \Delta_i|\mathbf{b}_i; \boldsymbol{\psi}) f(\mathbf{b}_i; \boldsymbol{\theta}) d\mathbf{b}_i, \quad (7)$$

with

$$f(\mathbf{Y}_i|\mathbf{b}_i; \boldsymbol{\theta}) = \prod_{j=1}^{n_i} \frac{1}{\sqrt{2\pi\sigma_e^2}} \exp \left\{ -\frac{(Y_{ij} - \mathbf{X}_{ij}^\top \boldsymbol{\beta} - \mathbf{Z}_{ij}^\top \mathbf{b}_i)^2}{2\sigma_e^2} \right\},$$

$$f(V_i, \Delta_i|\mathbf{b}_i; \boldsymbol{\psi}) = \left(\lambda_0(V_i) \exp\{\mathbf{W}_i^\top(V_i) \boldsymbol{\phi} + \alpha m_i(V_i)\} \right. \quad (8)$$

$$\left. G' \left[\int_0^{V_i} \exp\{\mathbf{W}_i^\top(t) \boldsymbol{\phi} + \alpha m_i(t)\} d\Lambda_0(t) \right] \right)^{\Delta_i} \\ \times \exp \left\{ -G \left[\int_0^{V_i} \exp\{\mathbf{W}_i^\top(t) \boldsymbol{\phi} + \alpha m_i(t)\} d\Lambda_0(t) \right] \right\}, \quad (9)$$

$$f(\mathbf{b}_i; \boldsymbol{\theta}) = \frac{1}{\sqrt{|2\pi\Sigma_b|}} \exp \left\{ -\frac{1}{2} \mathbf{b}_i^\top \Sigma_b^{-1} \mathbf{b}_i \right\}.$$

Direct maximization of (7) is computationally challenging due to the integral (possibly multi-dimensional) and the infinite dimensional parameter Λ_0 . Fortunately, by treating the random effects $\mathbf{b}_i$ as “missing data”, the EM algorithm can be readily applied to obtain the maximum likelihood estimates (MLEs). Note that the likelihood approach leads to non-parametric maximum likelihood estimates (NPMLEs; Kiefer and Wolfowitz 1956) for the parameters where $\hat{\Lambda}_0$, the estimate for Λ_0 , is a step function with jumps only at the uncensored survival times. Details of the algorithm are provided in the next section.

2.3. EM algorithm

Since first introduced systematically by [Dempster *et al.* \(1977\)](#), the EM algorithm has been extensively used to find the MLEs in statistical models where unobserved latent variables are involved. When a general class of transformation models is assumed for the survival outcomes, the EM algorithm is not as straightforward as that described in [Wulfsohn and Tsiatis \(1997\)](#), where the Cox proportional hazards model is assumed and the Breslow-type form suggested by [Breslow \(1972\)](#) is available for estimating Λ_0 in the M-step without relying on any numerical maximization method. The transformation $G(\cdot)$ would prohibit the simple Breslow-type estimator for Λ_0 in the M-step, instead a high dimensional non-linear optimization would be needed to estimate Λ_0 . To overcome this difficulty, we adopt the clever idea of [Zeng and Lin \(2007\)](#) to introduce an artificial random effect and treat it as “missing data” in the EM algorithm.

To elaborate on this idea let $\exp\{-G(x)\}$ be the Laplace transformation of some density function $\pi(x)$ (this function exists for the class of logarithmic transformations defined in (4)), i.e.,

$$\exp\{-G(x)\} = \int_0^\infty \exp(-xt)\pi(t)dt.$$

If ξ_i follows the distribution with density $\pi(x)$, $f(V_i, \Delta_i | \mathbf{b}_i; \boldsymbol{\psi})$ in (8) can be rewritten as:

$$\begin{aligned} f(V_i, \Delta_i | \mathbf{b}_i; \boldsymbol{\psi}) &= \int_0^\infty \left(\xi_i \lambda_0(V_i) \exp\{\mathbf{W}_i^\top(V_i)\boldsymbol{\phi} + \alpha m_i(V_i)\} \right)^{\Delta_i} \\ &\quad \times \exp \left\{ -\xi_i \int_0^{V_i} \exp\{\mathbf{W}_i^\top(t)\boldsymbol{\phi} + \alpha m_i(t)\} d\Lambda_0(t) \right\} \pi(\xi_i) d\xi_i, \end{aligned}$$

where the integrand is the likelihood function under the Cox proportional hazards model with conditional hazard function $\xi_i \lambda_0(t) \exp\{\mathbf{W}_i^\top(t)\boldsymbol{\phi} + \alpha m_i(t)\}$. Treating both $\mathbf{b}_i$ and ξ_i as “missing data”, the complete-data joint log-likelihood can be written as, up to a constant,

$$\begin{aligned} \ell_n^c(\boldsymbol{\psi}) &= -\frac{N}{2} \log \sigma_e^2 - \frac{1}{2\sigma_e^2} \sum_{i=1}^n \sum_{j=1}^{n_i} (Y_{ij} - \mathbf{X}_{ij}^\top \boldsymbol{\beta} - \mathbf{Z}_{ij}^\top \mathbf{b}_i)^2 + \sum_{i=1}^n \Delta_i [\log \xi_i + \log \lambda_0(V_i) \\ &\quad + \mathbf{W}_i^\top(V_i)\boldsymbol{\phi} + \alpha m_i(V_i)] - \sum_{i=1}^n \xi_i \int_0^{V_i} \exp\{\mathbf{W}_i^\top(t)\boldsymbol{\phi} + \alpha m_i(t)\} d\Lambda_0(t) \\ &\quad - \frac{n}{2} \log |\Sigma_b| - \frac{1}{2} \sum_{i=1}^n \mathbf{b}_i^\top \Sigma_b^{-1} \mathbf{b}_i, \quad (10) \end{aligned}$$

where $N = \sum_{i=1}^n n_i$. Let K be the number of distinct uncensored survival times ($\{V_i : \Delta_i = 1\}$) with $t^1 < t^2 < \dots < t^K$ the corresponding ordered statistics, then the NPMLE of Λ_0 is a step function with jumps at $\{t^1, t^2, \dots, t^K\}$. Replacing $\Lambda_0(\cdot)$ by the jump sizes $\{\Lambda^1, \Lambda^2, \dots, \Lambda^K\}$, the target function to be evaluated in the E-step and maximized in the

M-step can be written as

$$\begin{aligned}
Q(\boldsymbol{\psi}|\tilde{\boldsymbol{\psi}}) &= \mathbb{E}_{\tilde{\boldsymbol{\psi}}}(\ell_n^c(\boldsymbol{\psi})|\mathbf{O}) = -\frac{N}{2} \log \sigma_e^2 - \frac{1}{2\sigma_e^2} \sum_{i=1}^n \sum_{j=1}^{n_i} \mathbb{E}_{\tilde{\boldsymbol{\psi}}} \left[(Y_{ij} - \mathbf{X}_{ij}^\top \boldsymbol{\beta} - \mathbf{Z}_{ij}^\top \mathbf{b}_i)^2 | \mathbf{O}_i \right] \\
&+ \sum_{i=1}^n \Delta_i \left[E_{\tilde{\boldsymbol{\psi}}}(\log \xi_i | \mathbf{O}_i) + \sum_{k=1}^K \log \Lambda^k I(V_i = t^k) + \mathbf{W}_i^\top(V_i) \boldsymbol{\phi} + \alpha \mathbb{E}_{\tilde{\boldsymbol{\psi}}}(m_i(V_i) | \mathbf{O}_i) \right] \\
&- \sum_{i=1}^n \sum_{k=1}^K \Lambda^k I(V_i \geq t^k) \mathbb{E}_{\tilde{\boldsymbol{\psi}}} \left[\xi_i \exp\{\mathbf{W}_i^\top(t^k) \boldsymbol{\phi} + \alpha m_i(t^k)\} | \mathbf{O}_i \right] \\
&- \frac{n}{2} \log |\Sigma_b| - \frac{1}{2} \sum_{i=1}^n \mathbb{E}_{\tilde{\boldsymbol{\psi}}} \left(\mathbf{b}_i^\top \Sigma_b^{-1} \mathbf{b}_i | \mathbf{O}_i \right), \quad (11)
\end{aligned}$$

with $\tilde{\boldsymbol{\psi}} = (\tilde{\boldsymbol{\theta}}, \tilde{\Lambda}_0)$ being the current parameter estimate. By maximizing $Q(\boldsymbol{\psi}|\tilde{\boldsymbol{\psi}})$ w.r.t. Λ^k , we obtain the NPMLE:

$$\hat{\Lambda}^k = \frac{\sum_{i=1}^n \Delta_i I(V_i = t^k)}{\sum_{i=1}^n I(V_i \geq t^k) \mathbb{E}_{\tilde{\boldsymbol{\psi}}} \left(\xi_i \exp\{\mathbf{W}_i^\top(t^k) \boldsymbol{\phi} + \alpha m_i(t^k)\} | \mathbf{O}_i \right)}, \quad (12)$$

which is a Breslow-type estimate. Moreover, it is straightforward to get

$$\begin{aligned}
\hat{\sigma}_e^2 &= \frac{1}{N} \sum_{i=1}^n \sum_{j=1}^{n_i} \mathbb{E}_{\tilde{\boldsymbol{\psi}}} \left[(Y_{ij} - \mathbf{X}_{ij}^\top \boldsymbol{\beta} - \mathbf{Z}_{ij}^\top \mathbf{b}_i)^2 | \mathbf{O}_i \right], \\
\hat{\Sigma}_b &= \frac{1}{n} \sum_{i=1}^n \mathbb{E}_{\tilde{\boldsymbol{\psi}}} \left(\mathbf{b}_i \mathbf{b}_i^\top | \mathbf{O}_i \right).
\end{aligned}$$

However, there are no explicit expressions for the NPMLEs of $\boldsymbol{\beta}$, $\boldsymbol{\phi}$ and α so one-step Newton-Raphson updates in each EM iteration are needed. The score functions needed for the Newton-Raphson updates are computed by taking the derivative of (11) w.r.t. the corresponding parameters and are not shown here. Note that in the E-step, to evaluate $\mathbb{E}_{\tilde{\boldsymbol{\psi}}} \left(\xi_i \exp\{\mathbf{W}_i^\top(t^k) \boldsymbol{\phi} + \alpha m_i(t^k)\} | \mathbf{O}_i \right)$ with $m_i(t) = \mathbf{X}_i^\top(t) \boldsymbol{\beta} + \mathbf{Z}_i^\top(t) \mathbf{b}_i$, we are not integrating over both $\mathbf{b}_i$ and ξ_i . Instead we compute

$$\begin{aligned}
\mathbb{E}_{\tilde{\boldsymbol{\psi}}}(\xi_i | \mathbf{b}_i, \mathbf{O}_i) &= \int_{\xi} \xi_i f(\xi_i | \mathbf{b}_i, \mathbf{O}_i; \tilde{\boldsymbol{\psi}}) d\xi_i = \frac{\int_{\xi} \xi_i f(\mathbf{O}_i | \xi_i, \mathbf{b}_i; \tilde{\boldsymbol{\psi}}) \pi(\xi_i) d\xi_i}{f(\mathbf{O}_i | \mathbf{b}_i; \tilde{\boldsymbol{\psi}})} \\
&= G' \left[\sum_{k=1}^K \tilde{\Lambda}^k \exp \left\{ \eta(t^k | \mathbf{b}_i, \mathbf{O}_i; \tilde{\boldsymbol{\psi}}) \right\} I(t^k \leq V_i) \right] \\
&- \Delta_i \times \frac{G'' \left[\sum_{k=1}^K \tilde{\Lambda}^k \exp \left\{ \eta(t^k | \mathbf{b}_i, \mathbf{O}_i; \tilde{\boldsymbol{\psi}}) \right\} I(t^k \leq V_i) \right]}{G' \left[\sum_{k=1}^K \tilde{\Lambda}^k \exp \left\{ \eta(t^k | \mathbf{b}_i, \mathbf{O}_i; \tilde{\boldsymbol{\psi}}) \right\} I(t^k \leq V_i) \right]}, \quad (13)
\end{aligned}$$

with $\eta(t | \mathbf{b}_i, \mathbf{O}_i; \tilde{\boldsymbol{\psi}}) = \mathbf{W}_i^\top(t) \tilde{\boldsymbol{\phi}} + \tilde{\alpha} \tilde{m}_i(t) = \mathbf{W}_i^\top(t) \tilde{\boldsymbol{\phi}} + \tilde{\alpha} \left[\mathbf{X}_i^\top(t) \tilde{\boldsymbol{\beta}} + \mathbf{Z}_i^\top(t) \mathbf{b}_i \right]$, and subsequently obtain

$$\mathbb{E}_{\tilde{\boldsymbol{\psi}}} \left(\xi_i \exp\{\mathbf{W}_i^\top(t^k) \boldsymbol{\phi} + \alpha m_i(t^k)\} | \mathbf{O}_i \right) = \mathbb{E}_{\tilde{\boldsymbol{\psi}}} \left[\mathbb{E}_{\tilde{\boldsymbol{\psi}}}(\xi_i | \mathbf{b}_i, \mathbf{O}_i) \times \exp\{\mathbf{W}_i^\top(t^k) \boldsymbol{\phi} + \alpha m_i(t^k)\} | \mathbf{O}_i \right],$$

which is an integration over $\mathbf{b}_i$ only by plugging in (13). Therefore, the E-step involves integrations over $\mathbf{b}_i$ only. This suggests that by introducing an artificial random effect ξ_i , we greatly simplify the M-step while not complicating the E-step. We note that the integrations over $\mathbf{b}_i$ in the E-step do not have explicit solutions and have to be computed numerically. Fortunately, assuming a normal distribution for $\mathbf{b}_i$ enables the application of the (adaptive) Gauss-Hermite quadrature, providing more numerical precision and efficiency than other numerical integration methods, e.g., Monte Carlo integration and Laplace approximation.

3. Standard error estimation

For the standard error (SE) estimation under semi-parametric joint modeling settings, i.e., when the baseline hazard function $\lambda_0(t)$ is left completely unspecified, Xu *et al.* (2014) proposed two new methods and systematically examined their performance along with the profile likelihood method by Murphy and Van der Vaart (1999, 2000) and the bootstrap method of Efron (1994). According to their results, numerically differentiating the profile Fisher score with Richardson extrapolation (PRES) is recommended as the best choice. In our **JSM** package, we apply the PRES method as the default but also provide two alternative approaches: the PFDS (numerically differentiating the profile Fisher score with forward difference) method in Xu *et al.* (2014) and the PLFD (numerically deriving the second derivative of the profile likelihood function with forward difference) method in Murphy and Van der Vaart (1999). The bootstrap method, being computationally intensive, is not implemented in **JSM** but users can easily construct their own bootstrap SE estimators using our code. Details of the methods are described below.

3.1. PLFD: Numerically differentiating the profile likelihood function

Our interests are mainly to infer the finite dimensional parameter $\boldsymbol{\theta}$. The PLFD method obtains the observed-data information matrix I_o of $\boldsymbol{\theta}$ at $\boldsymbol{\theta}^*$ (here $\boldsymbol{\psi}^* = (\boldsymbol{\theta}^*, \Lambda_0^*)$ denotes the NPMLE of $\boldsymbol{\psi}$) by taking the second derivative of the profile likelihood function with forward difference. The profile likelihood function is

$$\text{pl}_n(\boldsymbol{\theta}) = \max_{\Lambda_0} \{\log \mathcal{L}_n(\boldsymbol{\theta}, \Lambda_0 | \mathbf{O})\}$$

and we have ($1 \leq i, j \leq p$)

$$I_o(i, j) \approx \frac{\text{pl}_n(\boldsymbol{\theta}^* + \delta \mathbf{e}_i + \delta \mathbf{e}_j) - \text{pl}_n(\boldsymbol{\theta}^* + \delta \mathbf{e}_i) - \text{pl}_n(\boldsymbol{\theta}^* + \delta \mathbf{e}_j) - \text{pl}_n(\boldsymbol{\theta}^*)}{\delta^2}, \quad (14)$$

where $\mathcal{L}_n$ is defined in (7), $\mathbf{e}_i$ is the i th coordinate vector and $\delta > 0$ is the increment used by forward difference. Then the variance-covariance matrix estimate of $\boldsymbol{\theta}$ at $\boldsymbol{\theta}^*$ is obtained by $V = I_o^{-1}$. Note that to evaluate $\text{pl}_n(\boldsymbol{\theta})$, we need to compute

$$\Lambda^*(\boldsymbol{\theta}) = \operatorname{argmax}_{\Lambda_0} \log \mathcal{L}(\boldsymbol{\theta}, \Lambda_0 | \mathbf{O}), \quad (15)$$

i.e., the estimate of Λ_0 given any $\boldsymbol{\theta}$.

3.2. PRES/PFDS: Numerically differentiating the profile Fisher score

The key idea of the PRES and PFDS methods is that the observed-data information matrix

can be approximated by numerically differentiating the Fisher score defined as

$$S(\tilde{\psi}) = \frac{\partial Q(\psi|\tilde{\psi})}{\partial \psi} \Big|_{\psi=\tilde{\psi}}$$

with $Q(\psi|\tilde{\psi})$ defined by (11). Note that $S(\cdot)$ is a $(p + K)$ -dimensional vector where K is the number of distinct uncensored survival times, which is usually large and increases with the sample size. Luckily, differentiating the entire Fisher score vector is unnecessary and [Xu et al. \(2014\)](#) proposed a profile version of the Fisher score, i.e., the profile Fisher score with the form:

$$S_{\theta}(\tilde{\psi}) = \left[\frac{\partial Q(\psi|\tilde{\psi})}{\partial \theta} \Big|_{\Lambda^k = \hat{\Lambda}^k(\theta)} \right] \Big|_{\theta = \tilde{\theta}}, \quad (16)$$

which is a p -dimensional vector with $\hat{\Lambda}^k(\theta)$ corresponding to the $\hat{\Lambda}^k$ defined in (12). Then the step-by-step algorithm of the PRES and PFDS methods is:

1. For $i = 1, \dots, p$, let $\theta_1^{(i)} = \theta^* + \delta \mathbf{e}_i$, $\theta_2^{(i)} = \theta^* - \delta \mathbf{e}_i$, $\theta_3^{(i)} = \theta^* + 2\delta \mathbf{e}_i$, and $\theta_4^{(i)} = \theta^* - 2\delta \mathbf{e}_i$, obtain $\Lambda^*(\theta_1^{(i)})$, $\Lambda^*(\theta_2^{(i)})$, $\Lambda^*(\theta_3^{(i)})$ and $\Lambda^*(\theta_4^{(i)})$ defined by (15).
2. For $l = 1, 2, 3, 4$, let $\psi_l^{(i)} = (\theta_l^{(i)}, \Lambda^*(\theta_l^{(i)}))$ and evaluate $S_{\theta}(\psi_l^{(i)})$ defined by (16).
3. Compute the i th row of I_o :

$$\begin{aligned} \text{PRES: } I_o(i,) &= \frac{S_{\theta}(\psi_4^{(i)}) - 8S_{\theta}(\psi_2^{(i)}) + 8S_{\theta}(\psi_1^{(i)}) - S_{\theta}(\psi_3^{(i)})}{12\delta}. \\ \text{PFDS: } I_o(i,) &= \frac{S_{\theta}(\psi_1^{(i)}) - S_{\theta}(\psi^*)}{\delta}. \end{aligned} \quad (17)$$

4. Obtain V by $V = I_o^{-1}$.

In step 3, the difference between the PRES and PFDS method is in that PFDS adopts the forward differencing method to estimate a derivative while PRES adopts the stabled Richardson extrapolation method for derivatives. The PRES method costs more computing time but provides more reliable SE estimates based on the numerical experiments in [Xu et al. \(2014\)](#).

4. JSM: Implementation details

The **JSM** package is an add-on package to R which has two model-fitting functions named `jmodelTM()` and `jmodelMult()`. It requires additional packages from R: **nlme** ([Pinheiro, Bates, DebRoy, Sarkar, and R Core Team 2020](#)), **Rcpp** ([Eddelbuettel and François 2011](#)), **RcppEigen** ([Bates and Eddelbuettel 2013](#)), **splines** ([R Core Team 2020](#)), **statmod** ([Giner and Smyth 2016](#)) and **survival** ([Therneau and Grambsch 2000](#)). For the longitudinal outcome, `jmodelTM()` fits the linear mixed effects model in (1) while `jmodelMult()` fits the non-parametric multiplicative random effects model in (2) as described in Section 2.1. For the survival outcome, both `jmodelTM()` and `jmodelMult()` accommodate the transformation models with the class of logarithmic transformations defined by (4).

The `model` argument of `jmodelTM()` and `jmodelMult()` specifies the dependency between the survival and longitudinal outcomes adopted by the joint model. If `model = 1`, the entire longitudinal outcome enters the survival model as a covariate described by (3); otherwise, the survival model only shares the same random effects with the longitudinal model as in (5) (for `jmodelTM()`) and (6) (for `jmodelMult()`). In addition, the `rho` argument allows users to choose the ρ in the logarithmic transformations (4) to specify the survival models they would like to fit. For example, the default `rho = 0` indicates that a Cox proportional hazards model is fitted whereas a proportional odds model is fitted with `rho = 1`. Sophisticated users can also perform their own model selection method to choose the value of `rho` using the code in **JSM**, see the example shown in Section 5.1.

The `control` argument of `jmodelTM()` and `jmodelMult()` is a list of control values for the estimation algorithm with components: `tol.P`, `tol.L`, `max.iter`, `SE.method`, `delta` and `nknot`. Section 2.3 provides the EM algorithm, an iterative algorithm, used to obtain the parameter estimates. Naturally, a convergence criterion is needed to implement the algorithm and in **JSM** we claim convergence whenever

$$\max\{|\boldsymbol{\theta}^{(t)} - \boldsymbol{\theta}^{(t-1)}|/(|\boldsymbol{\theta}^{(t-1)}| + .Machine\$double.eps \times 2)\} < \text{tol.P}, \text{ or} \\ |\ell_n(\boldsymbol{\psi}^{(t)}) - \ell_n(\boldsymbol{\psi}^{(t-1)})|/(|\ell_n(\boldsymbol{\psi}^{(t-1)})| + .Machine\$double.eps \times 2) < \text{tol.L}$$

is satisfied. Here $\boldsymbol{\psi}^{(t)} = (\boldsymbol{\theta}^{(t)}, \Lambda_0^{(t)})$ denote the estimated parameters in the t th EM iteration, where $\ell_n(\boldsymbol{\psi}) = \log \mathcal{L}_n(\boldsymbol{\psi}|\mathbf{O})$ is defined by (7). The default values of `tol.P` and `tol.L` are 10^{-3} and 10^{-6} , respectively, which are the standard choices to claim convergence of iterative algorithms. The component `max.iter` states the maximum number of iterations for the EM algorithm with the default value being 250. The method to compute the standard error estimates is specified via `SE.method` and users could assign "PRES" (the default), "PFDS" or "PLFD" to it (refer to Section 3 for detailed explanation). Moreover, the increment δ used for numerical differentiation (as in (14) and (17)) can be customized through the component `delta`. By default, `delta = 10^{-5}` if `SE.method = "PRES"` and `delta = 10^{-3}` otherwise, according to the simulation studies performed in Xu *et al.* (2014).

On the other hand, `nknot` stands for the number of quadrature points used in the Gauss-Hermite rule for numerical integration. As mentioned at the end of Section 2.3, the integrations involved in the E-step of the EM algorithm are computed by applying the Gauss-Hermite quadrature rule. In **JSM**, we implement the adaptive Gauss-Hermite rule (Pinheiro and Bates 1995) which centers and scales the quadrature points in each EM iteration according to the conditional distribution of the random effects with current parameter $\boldsymbol{\psi}^{(t)}$, i.e., $f(\mathbf{b}_i|\mathbf{Y}_i; \boldsymbol{\psi}^{(t)})$. Although it seems to demand more computational effort, the adaptive rule is actually more efficient since it requires fewer quadrature points to reach the same level of precision. The default values of `nknot` differ between `jmodelTM()` and `jmodelMult()`. For `jmodelTM()`, `nknot = 9` for a one-dimensional random effect; `nknot = 7` for a two-dimensional random effect; `nknot = 5` for a higher-dimensional random effect. For `jmodelMult()`, `nknot = 11`. The reason why we assign a larger default value to `nknot` for `jmodelMult()` is that the model assumes a multiplicative random effect which requires more quadrature points to reach a satisfactory level of accuracy. These default values are selected based on a numerical study regarding the performance of the model-fitting functions with different values of `nknot`, which is provided in Appendix A.

The following commonly used methods are currently available for the objects returned by

`jmodelTM()` and `jmodelMult()`: `AIC()` and `BIC()` for computing AIC and BIC values; `confint()` for returning the confidence intervals of the parameter estimates; `fitted()` and `ranef()` for extracting the fitted values and random effects; `residuals()` for computing the residuals; `summary()` for outputting model-fitting details; `vcov()` for returning the variance-covariance matrix of the parameter estimates. Although the `residuals()` method is provided, we do not encourage its usage for diagnostic or model-assessment purposes under the joint modeling setting. An example is demonstrated in Section 5.1 to explain the reason.

The **JSM** package also provides a data preprocessing function called `dataPreprocess()` to prepare the data in a format that could be used to fit the joint model. An example is given in Section 5.1.

5. Real data analysis using JSM

The usage of `jmodelTM()` and `jmodelMult()` are illustrated through two examples.

5.1. Example I: Liver cirrhosis data analysis

Consider a randomized control trial involving 488 liver cirrhosis patients who were randomly allocated to prednisone (251) or placebo (237). The patients were followed until death or end of the study and variables such as prothrombin index (a measure of liver function) were recorded at study entry and several follow-up visits during the trial which differed among patients. Therefore, both survival and longitudinal data were collected and the main purpose is to investigate the development of prothrombin level over time as well as its relationship with the survival outcome. The data, available in **JSM** under the named `liver`, were reported in Andersen, Borgan, Grill, and Keiding (1993) and have been analyzed by Henderson, Diggle, and Dobson (2002). Figure 1 summarizes the data.

```
R> library("JSM")
R> liver.unq <- liver[!duplicated(liver$ID), ]
R> liver.pns <- liver[liver$Trt == "prednisone", ]
R> liver.pcb <- liver[liver$Trt == "placebo", ]
R> times.pns <- sort(unique(liver.pns$obstime))
R> times.pcb <- sort(unique(liver.pcb$obstime))
R> means.pns <- tapply(liver.pns$proth, liver.pns$obstime, mean)
R> means.pcb <- tapply(liver.pcb$proth, liver.pcb$obstime, mean)
R> par(mfrow = c(1, 2))
R> plot(lowess(times.pns, means.pns, f = 0.3), type = "l", col = "blue",
+       ylab = "Prothrombin index", xlab = "Time (years)", ylim = c(50, 150),
+       main = "Smoothed mean prothrombin index", lwd = 2)
R> grid()
R> lines(lowess(times.pcb, means.pcb, f = 0.3), col = "red", lty = 2,
+       lwd = 2)
R> legend("topright", c("Prednisone", "Placebo"), col = c("blue", "red"),
+       lty = 1:2, lwd = 2, bty = "n")
R> plot(survfit(Surv(Time, death) ~ Trt, data = liver.unq), lty = c(2, 1),
+       col = c("red", "blue"), mark.time = FALSE, ylab = "Survival",
+       xlab = "Time (years)", main = "Kaplan-Meier survival curves", lwd = 2)
```

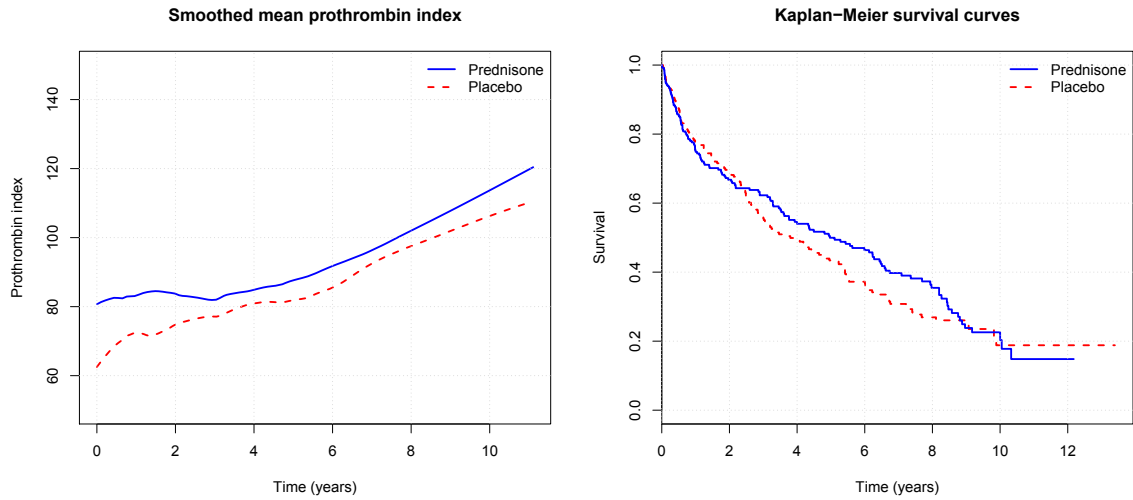


Figure 1: Smoothed mean prothrombin index (left) and Kaplan-Meier survival curves (right) for the liver cirrhosis patients, separately for the prednisone (treatment) and placebo groups.

```
R> grid()
R> legend("topright", legend = c("Prednisone", "Placebo"), lty = 1:2,
+       col = c("blue", "red"), lwd = 2, bty = "n")
```

We observe from the left plot of Figure 1 that the average prothrombin level increases over time in both groups and that the trend is approximately linear. Therefore, we fit a linear mixed effects model for the mean prothrombin response Y assuming a linear trend in time with different intercept and slope for each treatment group. For the survival component we first adopt the proportional hazards model with a single treatment effect, as was done in Andersen *et al.* (1993) and Henderson *et al.* (2002), and then fit the joint model:

$$Y_{ij} = Y_i(t_{ij}) = m_i(t_{ij}) + \varepsilon_{ij} = \beta_1 + \beta_2 \text{Trt}_i + \beta_3 t_{ij} + \beta_4 \text{Trt}_i \times t_{ij} + b_{1i} + b_{2i} t_{ij} + \varepsilon_{ij},$$

$$\lambda(t|\mathbf{b}_i, \text{Trt}_i) = \lambda_0(t) \exp\{\phi \text{Trt}_i + \alpha m_i(t)\}. \quad (18)$$

We start by fitting the linear mixed effects model and proportional hazards model separately:

```
R> fitLME <- lme(proth ~ Trt * obstime, random = ~ obstime | ID,
+   data = liver)
R> fitCOX <- coxph(Surv(start, stop, event) ~ Trt, data = liver, x = TRUE)
```

Note that the variables `start` and `stop` denote the limits of the time intervals between visits, and `event` denotes whether death occurred by the end of each interval. When the user does not have the longitudinal and survival data in a single dataset, `dataPreprocess()` could help to merge the data and generate the corresponding `start`, `stop`, and `event` columns. For example, `liver.long` contains only the longitudinal data for the liver cirrhosis patients, with one row per longitudinal measurement, while `liver.surv` contains only the survival data, with one row per patient. Then `liver.join` returned by the following code could be used similarly as the `liver` data which we are using:

```
R> liver.join <- dataPreprocess(liver.long, liver.surv, "ID", "obstime",
+   "Time", "death")
```

The argument `x = TRUE` is required in the call to `coxph()` to include the design matrix of the fitted model in the returned object `fitCOX`. Then objects `fitLME` and `fitCOX` are supplied to `jmodelTM()` to fit the joint model:

```
R> fitJT.ph <- jmodelTM(fitLME, fitCOX, liver, timeVarY = "obstime")
R> summary(fitJT.ph)
```

Call:

```
jmodelTM(fitLME = fitLME, fitCOX = fitCOX, data = liver,
  timeVarY = "obstime")
```

Data Descriptives:

Longitudinal Process	Survival Process
Number of Observations: 2968	Number of Events: 292 (59.8%)
Number of Groups: 488	
AIC	BIC
logLik	
30130.18	30172.08
-15055.09	

Coefficients:

Longitudinal Process: Linear mixed-effects model

	Estimate	StdErr	z.value	p.value
(Intercept)	69.82578	1.39294	50.1284	< 2.2e-16 ***
Trtprednisone	7.08300	1.94243	3.6465	0.0002659 ***
obstime	0.67446	0.47773	1.4118	0.1580119
Trtprednisone:obstime	-0.27292	0.63254	-0.4315	0.6661296

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Survival Process: Proportional hazards model with unspecified
baseline hazard function

	Estimate	StdErr	z.value	p.value
Trtprednisone	0.1841093	0.1244622	1.4792	0.1391
proth	-0.0392174	0.0036416	-10.7693	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Variance Components:

	StdDev	Corr
(Intercept)	18.8699	
obstime	3.9396	0.0216
Residual	17.2195	

Integration: (Adaptive Gauss-Hermite Quadrature)

quadrature points: 7

StdErr Estimation:

method: profile Fisher score with Richardson extrapolation

```

Optimization:
convergence: success
iterations: 28

```

Note that `jmodelTM()` requires the argument `timeVarY` to specify the name of the time variable used in the linear mixed effects model. The results suggest that the linear trend in time of the longitudinal process is not statistically significant since the p value of β_3 is 0.158. This seems contradictory to the strong linear upward trend of the prothrombin level in the left plot of Figure 1 but is in fact a perfect example to demonstrate the bias incurred by a marginal, i.e., separate, analysis of the longitudinal data. This bias, as also pointed out by Henderson *et al.* (2002), is due to the fact that the observed mean profiles are based on the average values over all patients who remain alive at each time point so that the observed linear trend is not due to a substantive increase in mean prothrombin level, but to the loss of data from high-risk patients with low prothrombin levels as time increases. This bias is corrected in the joint analysis, by leveraging information from the survival component, as the fitted mean profiles for both groups now appear to be quite flat in the left plot of Figure 2 (generated by the R code below) which provides the observed and fitted mean longitudinal profiles.

```

R> fittedVal <- fitted(fitJT.ph)
R> fitted.pns <- fittedVal[liver$Trt == "prednisone"]
R> fitted.pcb <- fittedVal[liver$Trt == "placebo"]
R> residVal <- residuals(fitJT.ph)
R> par(mfrow = c(1, 2))
R> plot(lowess(times.pns, means.pns, f = 0.3), type = "l", col = "blue",
+       ylab = "Prothrombin index", xlab = "Time (years)", ylim = c(50, 150),
+       main = "Smoothed mean prothrombin index", lwd = 2)
R> grid()
R> lines(lowess(liver.pns$obstime, fitted.pns, f = 0.3),
+       lwd = 2, col = "deepskyblue")
R> lines(lowess(liver.pcb$obstime, fitted.pcb, f = 0.3),
+       lwd = 2, lty = 2, col = "lightsalmon")
R> plot(fittedVal, residVal, pch = 20, col = "gray", xlab = "Fitted Values",
+       ylab = "Residuals", main = "Residuals vs. Fitted Values")
R> grid()
R> lines(lowess(fittedVal, residVal, f = 0.5), lwd = 2, col = "red")
R> abline(h = 10, lty = 5, lwd = 2)

```

Caution about residual plots. The right plot of Figure 2 shows the residuals versus fitted values for the prothrombin index, which is commonly used to check misspecification of the mean structure. If the mean structure has been correctly specified, the residual plot should show a random behavior around zero. However, a caveat in the joint modeling setting is the bias induced by the informative dropout inherited in the longitudinal data when the event time is death. Thus, it is not surprising that the residual plot shows a systemic pattern. Therefore, a systemic behavior of the residuals does not necessarily indicate a lack-of-fit under the joint modeling setting and model diagnostics based on residual plots are not encouraged.

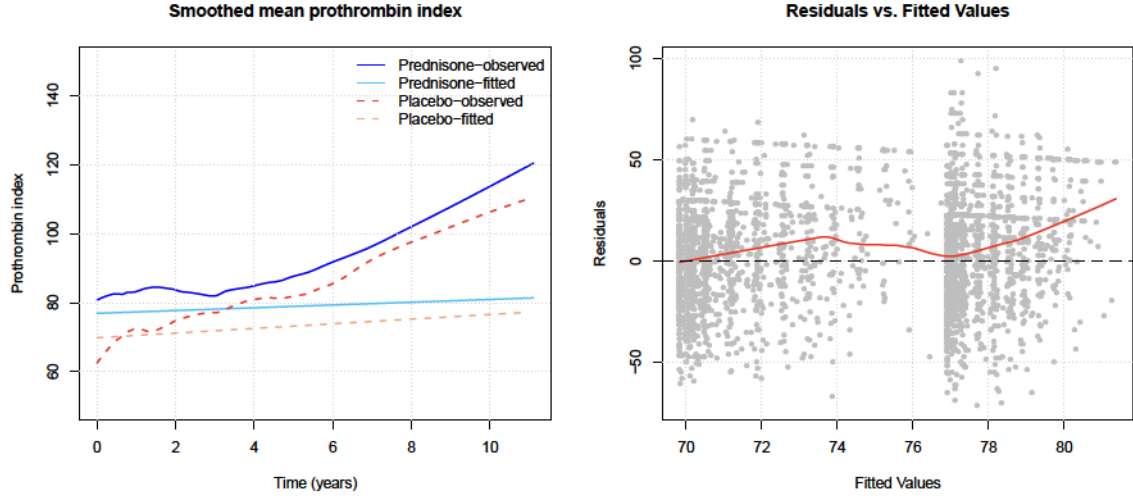


Figure 2: Observed and fitted mean prothrombin index (left); residuals versus fitted values for the prothrombin index (right).

Our next analysis is to fit a joint model with a proportional odds model for the survival outcome instead of a proportional hazards model. This is motivated by the fact that the two survival functions in the right plot of Figure 1 cross each other in the right tail, suggesting that the proportional odds model might be a better fit than the proportional hazards model for these data. Thus, we adopt the following survival model:

$$\Lambda(t|\mathbf{b}_i, \text{Trt}_i) = \log \left(1 + \int_0^t \lambda_0(s) \exp\{\phi \text{Trt}_i + \alpha m_i(s)\} ds \right)$$

and the same model as in (18) for the longitudinal outcome. This could be realized by specifying $\rho = 1$ in `jmodelTM()` and the corresponding R code is given below with the AIC values. `fitJT.po` has a smaller AIC value than `fitJT.ph`, suggesting that under the Akaike information criterion, the proportional odds model is preferred over the proportional hazards model for the survival outcome in the joint model for the liver cirrhosis data (AIC value 30130.2 against 30128.2).

```
R> fitJT.po <- jmodelTM(fitLME, fitCOX, liver, rho = 1, timeVarY = "obstime")
R> AIC(fitJT.ph); AIC(fitJT.po)
```

```
[1] 30130.18
```

```
[1] 30128.19
```

In addition to the proportional odds model, we also tried the logarithmic transformation model (4) with other values for ρ . Since a very huge sample size will be needed to estimate the optimal transformation parameter ρ and it is not necessary to have the exact optimal ρ in practice, we conduct a simple grid search over the candidate ρ values.

The results (Figure 3) strongly suggest that the value $\rho = 1$ offers the best model (i.e., the proportional odds model) according to AIC. While interpreting the covariate effects on the hazard function under the logarithmic transformation is straightforward when $\rho = 0$ or $\rho = 1$,

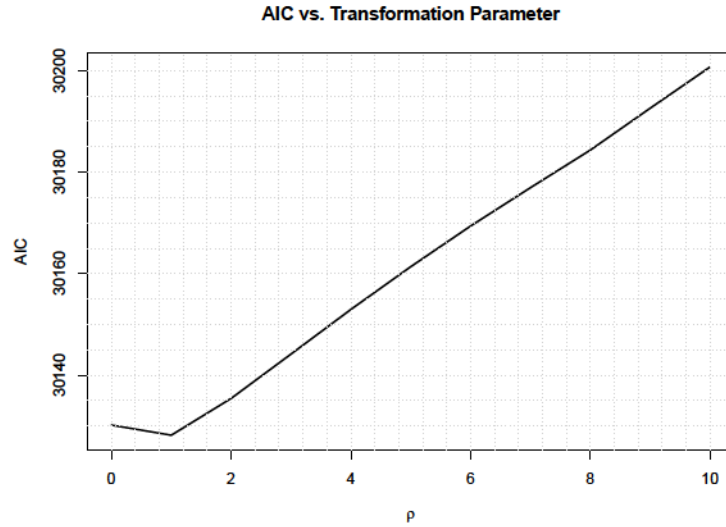


Figure 3: AIC curve for model fittings with different transformation parameter.

the interpretation under more general values of ρ is less transparent. Below we provide an interpretation.

Following the idea of [Dabrowska and Doksum \(1988\)](#), we define the “odds function”:

$$\Gamma_T(t) = \frac{1}{\rho} \left(\frac{1 - P^\rho(T > t)}{P^\rho(T > t)} \right), \quad \rho > 0. \quad (19)$$

When ρ is a positive integer, $\rho\Gamma_T(t)$ is the odds that ρ independent individuals (with the same covariate values) would not all survive to time t . By a simple derivation, our transformation model with transformation parameter ρ is equivalent to

$$\Gamma_T(t) = \int_0^t \lambda_0(s) \exp\{\phi T r t_i + \alpha m_i(s)\} ds, \quad (20)$$

so that the regression parameters under our transformation model can be interpreted similarly as those under the proportional hazards model, the only difference being that their effects are on the “odds function” instead of on the hazard function.

Therefore, if the data analysis has an emphasis on interpretation, the researcher can choose ρ as the value that results in the smallest AIC value among all possible integers. For the liver cirrhosis data, this leads to $\rho = 1$, i.e., the proportional odds model.

```
R> summary(fitJT.po)
```

Call:

```
jmodelTM(fitLME = fitLME, fitCOX = fitCOX, data = liver, rho = 1,
  timeVarY = "obstime")
```

Data Descriptives:

Longitudinal Process

Number of Observations: 2968

Survival Process

Number of Events: 292 (59.8%)

Number of Groups: 488

AIC	BIC	logLik
30128.19	30170.09	-15054.1

Coefficients:

Longitudinal Process: Linear mixed-effects model

	Estimate	StdErr	z.value	p.value
(Intercept)	69.70937	1.39046	50.1339	< 2.2e-16 ***
Trtprednisone	6.99834	1.93739	3.6123	0.0003035 ***
obstime	0.49102	0.48250	1.0176	0.3088475
Trtprednisone:obstime	-0.19597	0.63209	-0.3100	0.7565271

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Survival Process: Proportional odds model with unspecified baseline hazard function

	Estimate	StdErr	z.value	p.value
Trtprednisone	0.2805809	0.1838791	1.5259	0.127
proth	-0.0546661	0.0052977	-10.3188	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Variance Components:

	StdDev	Corr
(Intercept)	18.8399	
obstime	3.9620	0.0332
Residual	17.2232	

Integration: (Adaptive Gauss-Hermite Quadrature)

quadrature points: 7

StdErr Estimation:

method: profile Fisher score with Richardson extrapolation

Optimization:

convergence: success

iterations: 45

For the longitudinal process, the results indicate that the clinical trial was not well-randomized as the prednisone group had higher baseline prothrombin index than the placebo group, as $\hat{\beta}_2 = 6.998$ with p value < 0.05 and 95% confidence interval $[3.201, 10.796]$. The p values of β_3 and β_4 are 0.309 and 0.757, respectively, suggesting that the mean prothrombin index did not significantly vary over time for both treatment groups. For the survival process, the p value of ϕ is 0.127, indicating the lack of efficacy of prednisone. Lastly, $\hat{\alpha} = -0.055$ with p value < 0.05 and 95% confidence interval $[-0.065, -0.044]$ provides evidence that high prothrombin index is associated with long-term survival. Note that the confidence intervals reported above are obtain by running `confint(fitJT.po)`.

5.2. Example II: Mayo Clinic primary biliary cirrhosis data analysis

Here we consider a data set collected by the Mayo Clinic from a randomized control trial on PBC (primary biliary cirrhosis) patients during a ten-year interval, from 1974 to 1984. PBC is a rare chronic liver disease which develops slowly but would eventually lead to cirrhosis and even death. PBC patients usually display abnormalities in their blood values related to liver function, such as serum bilirubin, albumin and alkaline phosphatase, however, what triggers PBC remains unclear. A total of 312 PBC patients were included in the trial with 158 of them given the drug D-penicillamine and the other 154 given a placebo. Detailed descriptions of the clinical background and the variables recorded in the trial can be found in [Fleming and Harrington \(2011\)](#). Several baseline covariates, e.g., age at first diagnosis, gender and presence of some physical symptoms, were recorded at the study entry; some longitudinal covariates such as the blood values mentioned above were also collected at irregular subsequent visits until death or the end of the trial. Here, we focus on the serum bilirubin, the most significant biomarker, and include the drug type as the only time-independent covariate as in [Ding and Wang \(2008\)](#) did.

```
R> library("lattice")
R> xyplot(log(serBilir) ~ obstime | drug, group = ID, data = pbc,
+   type = "l", xlab = "Years", ylab = "log(serum bilirubin)",
+   col = "darkgray")
```

Figure 4 (generated by the R code above) shows the observed log-transformed serum bilirubin trajectories for all of the 312 patients. Apparently, the variations of the trajectories are large so a non-parametric multiplicative random effects model described by (2) may be more suitable than a traditional parametric model. Further support for the multiplicative random effects, based on functional principal component analysis ([Yao et al. 2005](#)), was provided in [Ding and Wang \(2008\)](#) and we refer the reader to that paper for more details.

We proceed by fitting a joint model where the mean log-transformed serum bilirubin Y follows

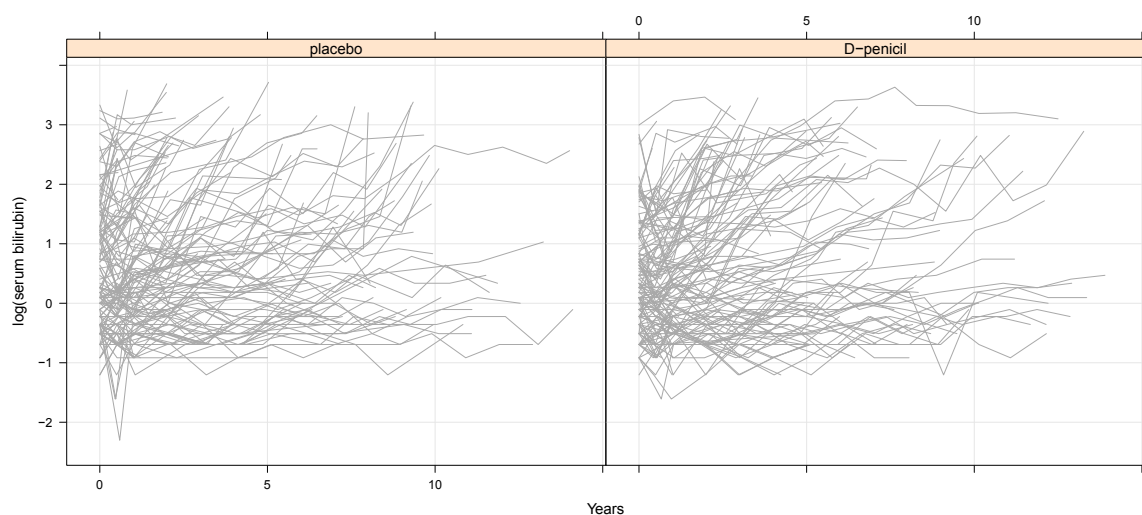


Figure 4: Subject-specific serum bilirubin trajectories in logarithmic scale, separately for the D-penicillamine and placebo groups.

a non-parametric multiplicative random effects model with two internal knots and quadratic B-spline basis. To show that baseline covariates can also be incorporated, drug is included in the model for Y . For the survival model we adopt the proportional hazards model, as it has been shown in the literature to fit these data well,

$$Y_{ij} = Y_i(t_{ij}) = m_i(t_{ij}) + \varepsilon_{ij} = b_i \times [\gamma_1 + \gamma_2 \text{drug}_i + \gamma_3 B_1(t_{ij}) + \gamma_4 B_2(t_{ij}) + \gamma_5 B_3(t_{ij}) + \gamma_6 B_4(t_{ij}) + \gamma_7 \text{drug}_i \times B_1(t_{ij}) + \gamma_8 \text{drug}_i \times B_2(t_{ij}) + \gamma_9 \text{drug}_i \times B_3(t_{ij}) + \gamma_{10} \text{drug}_i \times B_4(t_{ij})] + \varepsilon_{ij},$$

$$\lambda(t|b_i, \text{drug}_i) = \lambda_0(t) \exp\{\phi \text{drug}_i + \alpha m_i(t)\}. \quad (21)$$

The R code corresponding to the model is provided below. Note that we need `fitLME.pbc` as an input of `jmodelMult()` just to specify the $\mathbf{B}(t)$ in model (2) but not to fit a linear mixed effects model for the longitudinal outcome. The `Boundary.knots` argument is recommended and should include the longest follow-up period among all subjects in the trial.

```
R> fitLME.pbc <- lme(log(serBilir) ~ drug * bs(obstime, df = 4, degree = 2,
+   Boundary.knots = c(0, 15)), random = ~ 1 | ID, data = pbc)
R> fitCOX.pbc <- coxph(Surv(start, stop, event) ~ drug, data = pbc, x = TRUE)
R> fitJT.pbc <- jmodelMult(fitLME.pbc, fitCOX.pbc, pbc, timeVarY = "obstime")
R> summary(fitJT.pbc)
```

Call:

```
jmodelMult(fitLME = fitLME.pbc, fitCOX = fitCOX.pbc, data = pbc,
  timeVarY = "obstime")
```

Data Descriptives:

Longitudinal Process	Survival Process
Number of Observations: 1945	Number of Events: 140 (44.9%)
Number of Groups: 312	
AIC	BIC
5092.394	5144.796
logLik	-2532.197

Coefficients:

Longitudinal Process: Nonparametric multiplicative random effects model

	Estimate	StdErr	z.value	p.value
(Intercept)	0.640394	0.061081	10.4843	< 2.2e-16 ***
drugD-penicil	-0.101691	0.050756	-2.0035	0.04512 *
bs1	0.064151	0.038368	1.6720	0.09452 .
bs2	0.279446	0.054820	5.0975	3.441e-07 ***
bs3	1.418953	0.214824	6.6052	3.970e-11 ***
bs4	2.127941	0.469246	4.5348	5.766e-06 ***
drugD-penicil:bs1	-0.109590	0.053631	-2.0434	0.04101 *
drugD-penicil:bs2	0.066800	0.067901	0.9838	0.32522
drugD-penicil:bs3	-0.201956	0.235751	-0.8567	0.39164
drugD-penicil:bs4	-0.911742	0.529450	-1.7221	0.08506 .

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

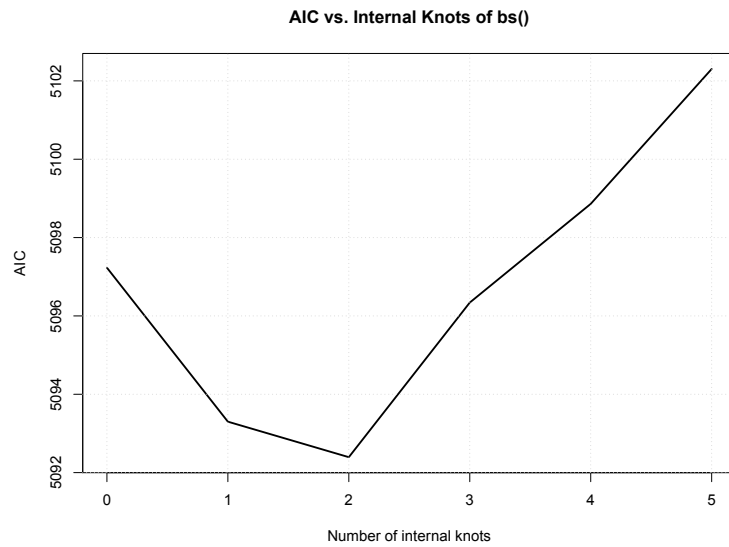


Figure 5: AIC curve for model fittings with different transformation parameter.

Survival Process: Proportional hazards model with unspecified baseline hazard function

	Estimate	StdErr	z.value	p.value
drugD-penicil	0.074350	0.180128	0.4128	0.6798
log(serBilir)	1.093954	0.087129	12.5555	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Variance Components:

	StdDev
Random	1.4245091
Residual	0.4501207

Integration: (Adaptive Gauss-Hermite Quadrature)

quadrature points: 11

StdErr Estimation:

method: profile Fisher score with Richardson extrapolation

Optimization:

convergence: success

iterations: 198

The choice of two internal knots is based on the AIC as it offers the smallest AIC value (Figure 5).

For the longitudinal outcome, we plot the expected logarithm serum bilirubin trajectories of six randomly selected patients in Figure 6 (R code given below). It demonstrates that the

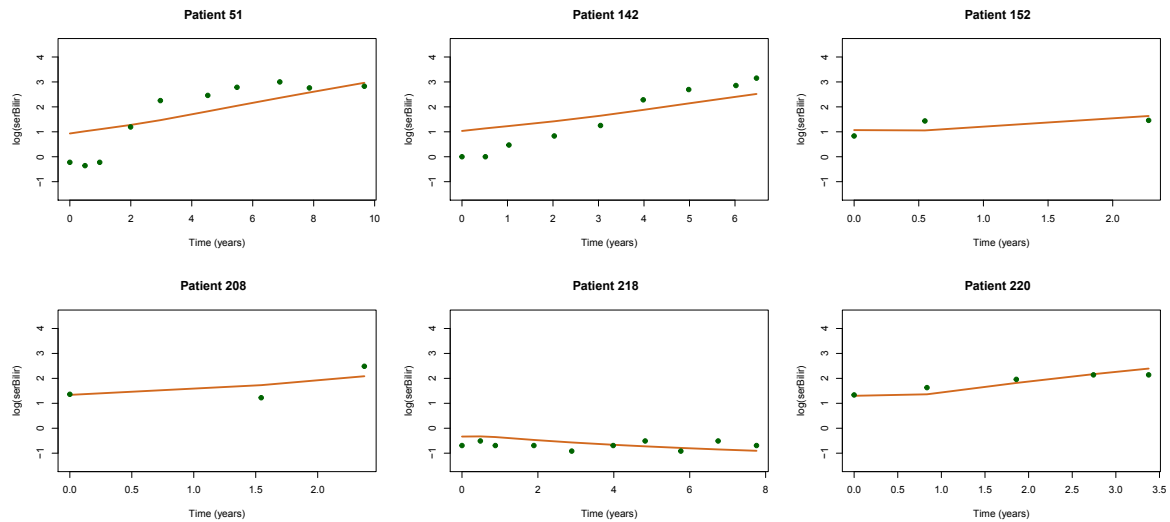


Figure 6: Expected log-transformed serum bilirubin trajectories (solid line) of six randomly selected patients. The dots represent the observed log-transformed serum bilirubin.

shape of the longitudinal trajectories are relatively well captured by the fitted joint model, given that only a one-dimensional random effect is assumed in the model. Had a linear mixed effects model been fitted, at least a random intercept as well as a random slope would be needed to capture the feature that some of the individual trajectories are increasing over time while some are not. Although there appears to be some lack of fit, one needs to bear in mind that this could be due to the bias issue discussed in Example 1, where the effect is more prominent for the fitted mean trajectory. For individual trajectories the effect of bias due to informative dropout is less prominent but still exists.

For the survival outcome, the estimated α is 1.094; the extremely small p value indicates a strong association between the serum bilirubin and time-of-death. While the estimated ϕ is 0.074 with p value 0.680 suggesting that there is no direct effect of D-penicillamine on survival after adjusting for serum bilirubin, there is an indirect effect through the association of D-penicillamine with serum bilirubin, as shown by the estimated γ_7 and γ_{10} . D-penicillamine is associated with lower serum bilirubin, which is in turn associated with improved survival.

```
R> set.seed(123)
R> patient <- sort(sample(replace = FALSE, size = 6, x = 312))
R> fittedPBC <- fitted(fitJT.pbc, type = "Conditional")
R> par(mfrow = c(2, 3))
R> for (i in 1:6) {
+   inds <- which(pbc$ID == patient[i])
+   times <- pbc$obstime[inds]
+   data <- pbc[inds, ]
+   mui <- fittedPBC[inds]
+   plot(times, mui, ylim = c(-1.5, 4.5), type = "l", col = "chocolate",
+       xlab = "Time (years)", ylab = "log(serBilir)", lwd = 2,
+       main = paste("Patient", patient[i], seq = " "))
+   points(data$obstime, log(data$serBilir), pch = 19, col = "darkgreen")
+ }
```

6. Summary

Joint modeling has emerged as a key tool in data analysis as evident by the addition of several recent software tools. We have briefly mentioned a number of software tools in Section 1 and here we present them with more details. The R package **JM** and **joinerR**, as well as Stata module **stjm** can be considered as the first generation of software tools for joint modeling, allowing the joint analysis of a single continuous longitudinal outcome and a single survival outcome based on a mixed effects model for the longitudinal outcome and a relative risk model for the survival outcome. Each of them has its own characteristics, e.g., **JM** and **joinerR** can model competing risks survival outcome while **JM** and **stjm** offers various association structures to link the survival and longitudinal processes. The more recent software tools try to explore the functionality of joint modeling from different aspects. For instance, the R package **JMBayes** fits joint models under a Bayesian approach using Markov chain Monte Carlo algorithms; the R package **joinerML** extends the **joinerR** package to incorporate the joint modeling of a survival outcome and multiple continuous longitudinal outcomes by a Monte Carlo EM algorithm. Other than these tools originally designed for joint modeling, some tools designed for other purposes have also been extended to fit joint models of survival and longitudinal outcomes. For example, the R package **frailtypack** has added two new functions, i.e., `longiPenal()` (joint model for longitudinal data and a terminal event) and `trivPenal()` (trivariate joint model for longitudinal data, recurrent events and a terminal event).

To summarize, this paper presents the R package **JSM**, which complements existing packages by offering a variety of joint models of survival and longitudinal data, most of which are not available elsewhere. The choice for the survival process is much broader compared to existing packages that typically are confined to the proportional hazards formulation. Package **JSM** is able to fit a whole class of transformation models, including the Cox proportional hazards model and the proportional odds model as special cases. Two different ways to model the longitudinal component (i.e., the linear mixed effects model and the non-parametric multiplicative random effects model) further distinguish **JSM** from existing packages. Moreover, a number of the package's internal functionalities are written in C++ in the background to improve computational speed. Last but not least, **JSM** provides reliable standard error estimates for valid statistical inference.

The **JSM** package is being actively curated and developed further to include additional features. Possible future developments are more sophisticated model selection methods, the capability to handle categorical and multiple longitudinal outcomes, and the incorporation of more complicated censoring patterns of the survival outcome.

Acknowledgments

The research and the development of the R package **JSM** were supported by the National Science Foundation (DMS-0906813 and DMS-1512975) and the National Institutes of Health (1R01 AG025218-01A2).

References

Andersen PK, Borgan Ø, Grill RD, Keiding N (1993). *Statistical Models Based on Counting Processes*. Springer-Verlag. doi:10.1007/978-1-4612-4348-9.

- Bates D, Eddelbuettel D (2013). “Fast and Elegant Numerical Linear Algebra Using the **RcppEigen** Package.” *Journal of Statistical Software*, **52**(5), 1–24. doi:10.18637/jss.v052.i05.
- Bennett S (1983). “Analysis of Survival Data by the Proportional Odds Model.” *Statistics in Medicine*, **2**(2), 273–277. doi:10.1002/sim.4780020223.
- Breslow NE (1972). “Discussion of the Paper by D.R. Cox.” *Journal of the Royal Statistical Society B*, **34**(2), 216–217.
- Brown ER, Ibrahim JG (2003). “A Bayesian Semiparametric Joint Hierarchical Model for Longitudinal and Survival Data.” *Biometrics*, **59**(2), 221–228. doi:10.1111/1541-0420.00028.
- Cox DR (1972). “Regression Models and Life Tables.” *Journal of the Royal Statistical Society B*, **34**(2), 187–202. doi:10.1111/j.2517-6161.1972.tb00899.x.
- Crowther MJ, Abrams KR, Lambert PC (2013). “Joint Modeling of Longitudinal and Survival Data.” *The Stata Journal*, **13**(1), 165–184. doi:10.1177/1536867x1301300112.
- Dabrowska DM, Doksum KA (1988). “Partial Likelihood in Transformation Models with Censored Data.” *Scandinavian Journal of Statistics*, **15**(1), 1–23.
- Dempster AP, Laird NM, Rubin DB (1977). “Maximum Likelihood from Incomplete Data via the EM Algorithm.” *Journal of the Royal Statistical Society B*, **39**(1), 1–22. doi:10.1111/j.2517-6161.1977.tb01600.x.
- Ding J, Wang JL (2008). “Modeling Longitudinal Data with Nonparametric Multiplicative Random Effects Jointly with Survival Data.” *Biometrics*, **64**(2), 546–556. doi:10.1111/j.1541-0420.2007.00896.x.
- Eddelbuettel D, François R (2011). “**Rcpp**: Seamless R and C++ Integration.” *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.
- Efron B (1994). “Missing Data, Imputation, and the Bootstrap.” *Journal of the American Statistical Association*, **89**(426), 463–475. doi:10.1080/01621459.1994.10476768.
- Elashoff RM, Li G, Li N (2016). *Joint Modeling of Longitudinal and Time-to-Event Data*. CRC Press.
- Faucett CL, Thomas DC (1996). “Simultaneously Modelling Censored Survival Data and Repeatedly Measured Covariates: A Gibbs Sampling Approach.” *Statistics in Medicine*, **15**(15), 1663–1685. doi:10.1002/(sici)1097-0258(19960815)15:15<1663::aid-sim294>3.0.co;2-1.
- Fleming TR, Harrington DP (2011). *Counting Processes and Survival Analysis*, volume 169. John Wiley & Sons. doi:10.1002/9781118150672.
- Giner G, Smyth GK (2016). “**statmod**: Probability Calculations for the Inverse Gaussian Distribution.” *The R Journal*, **8**(1), 339–351. doi:10.32614/rj-2016-024.

- Gould AL, Boye ME, Crowther MJ, Ibrahim JG, Quartey G, Micallef S, Bois FY (2015). “Joint Modeling of Survival and Longitudinal Non-Survival Data: Current Methods and Issues. Report of the DIA Bayesian Joint Modeling Working Group.” *Statistics in Medicine*, **34**(14), 2181–2195. doi:[10.1002/sim.6141](https://doi.org/10.1002/sim.6141).
- Guo X, Carlin BP (2004). “Separate and Joint Modeling of Longitudinal and Event Time Data Using Standard Computer Packages.” *The American Statistician*, **58**(1), 16–24. doi:[10.1198/0003130042854](https://doi.org/10.1198/0003130042854).
- Henderson R, Diggle P, Dobson A (2000). “Joint Modelling of Longitudinal Measurements and Event Time Data.” *Biostatistics*, **1**(4), 465–480. doi:[10.1093/biostatistics/1.4.465](https://doi.org/10.1093/biostatistics/1.4.465).
- Henderson R, Diggle P, Dobson A (2002). “Identification and Efficacy of Longitudinal Markers for Survival.” *Biostatistics*, **3**(1), 33–50. doi:[10.1093/biostatistics/3.1.33](https://doi.org/10.1093/biostatistics/3.1.33).
- Hickey GL, Philipson P, Jorgensen A, Kolamunnage-Dona R (2018). **joineRML**: *Joint Modelling of Multivariate Longitudinal Data and Time-to-Event Outcomes*. R package version 0.4.2, URL <https://CRAN.R-project.org/package=joineRML>.
- Hsieh F, Tseng YK, Wang JL (2006). “Joint Modeling of Survival and Longitudinal Data: Likelihood Approach Revisited.” *Biometrics*, **62**(4), 1037–104. doi:[10.1111/j.1541-0420.2006.00570.x](https://doi.org/10.1111/j.1541-0420.2006.00570.x).
- Kiefer J, Wolfowitz J (1956). “Consistency of the Maximum Likelihood Estimator in the Presence of Infinitely Many Nuisance Parameters.” *The Annals of Mathematical Statistics*, **27**(4), 887–906. doi:[10.1214/aoms/1177728066](https://doi.org/10.1214/aoms/1177728066).
- Król A, Mauguen A, Mazroui Y, Laurent A, Michiels S, Rondeau V (2017). “Tutorial in Joint Modeling and Prediction: A Statistical Software for Correlated Longitudinal Outcomes, Recurrent Events and a Terminal Event.” *Journal of Statistical Software*, **81**(3), 1–52. doi:[10.18637/jss.v081.i03](https://doi.org/10.18637/jss.v081.i03).
- Lunn DJ, Thomas A, Best N, Spiegelhalter D (2000). “WinBUGS – A Bayesian Modelling Framework: Concepts, Structure, and Extensibility.” *Statistics and Computing*, **10**(4), 325–337. doi:[10.1023/a:1008929526011](https://doi.org/10.1023/a:1008929526011).
- Murphy SA, Van der Vaart AW (1999). “Observed Information in Semi-Parametric Models.” *Bernoulli*, **5**(3), 381–412. doi:[10.2307/3318710](https://doi.org/10.2307/3318710).
- Murphy SA, Van der Vaart AW (2000). “On Profile Likelihood.” *Journal of the American Statistical Association*, **95**(450), 449–465. doi:[10.1080/01621459.2000.10474219](https://doi.org/10.1080/01621459.2000.10474219).
- Philipson P, Sousa I, Diggle P, Williamson P, Kolamunnage-Dona R, Henderson R (2020). **joineR**: *Joint Modelling of Repeated Measurements and Time-to-Event Data*. R package version 1.2.5, URL <https://CRAN.R-project.org/package=joineR>.
- Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2020). **nlme**: *Linear and Nonlinear Mixed Effects Models*. R package version 3.1-144, URL <https://CRAN.R-project.org/package=nlme>.

- Pinheiro JC, Bates DM (1995). “Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model.” *Journal of Computational and Graphical Statistics*, **4**(1), 12–35. doi:10.2307/1390625.
- Proust-Lima C, Philipps V, Diakite A, Lique B (2019). *lcmm: Extended Mixed Models Using Latent Classes and Latent Processes*. R package version 1.8.1, URL <https://CRAN.R-project.org/package=lcmm>.
- Proust-Lima C, Philipps V, Lique B (2017). “Estimation of Extended Mixed Models Using Latent Classes and Latent Processes: The R Package **lcmm**.” *Journal of Statistical Software*, **78**(2), 1–56. doi:10.18637/jss.v078.i02.
- R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Rizopoulos D (2010). “**JM**: An R Package for the Joint Modelling of Longitudinal and Time-to-Event Data.” *Journal of Statistical Software*, **35**(9), 1–33. doi:10.18637/jss.v035.i09.
- Rizopoulos D (2012). *Joint Models for Longitudinal and Time-to-Event Data: With Applications in R*. Chapman & Hall/CRC, Boca Raton. doi:10.1201/b12208.
- Rizopoulos D (2016). “The R Package **JMbayes** for Fitting Joint Models for Longitudinal and Time-to-Event Data Using MCMC.” *Journal of Statistical Software*, **72**(7), 1–45. doi:10.18637/jss.v072.i07.
- Rizopoulos D, Lesaffre E (2014). “Introduction to the Special Issue on Joint Modelling Techniques.” *Statistical Methods in Medical Research*, **23**(1), 3–10. doi:10.1177/0962280212445800.
- Rondeau V, Mazroui Y, Gonzalez JR (2012). “**frailtypack**: An R Package for the Analysis of Correlated Survival Data with Frailty Models Using Penalized Likelihood Estimation or Parametrical Estimation.” *Journal of Statistical Software*, **47**(4), 1–28. doi:10.18637/jss.v047.i04.
- SAS Institute Inc (2013). *The SAS Software, Version 9.4*. Cary. URL <http://www.sas.com/>.
- Song X, Davidian M, Tsiatis AA (2002). “A Semiparametric Likelihood Approach to Joint Modeling of Longitudinal and Time-to-Event Data.” *Biometrics*, **58**(4), 742–753. doi:10.1111/j.0006-341x.2002.00742.x.
- Spiegelhalter DJ, Thomas A, Best NG, Lunn D (2003). *WinBUGS Version 1.4 User Manual*. MRC Biostatistics Unit, Cambridge. URL <http://www.mrc-bsu.cam.ac.uk/bugs/>.
- StataCorp (2019). *Stata Statistical Software: Release 16*. StataCorp LLC, College Station. URL <http://www.stata.com/>.
- Therneau TM, Grambsch PM (2000). *Modeling Survival Data: Extending the Cox Model*. Springer-Verlag.
- Tsiatis AA, Davidian M (2001). “A Semiparametric Estimator for the Proportional Hazards Model with Longitudinal Covariates Measured with Error.” *Biometrika*, **88**(2), 447–458. doi:10.1093/biomet/88.2.447.

- Tsiatis AA, Davidian M (2004). “Joint Modeling of Longitudinal and Time-to-Event Data: An Overview.” *Statistica Sinica*, **14**(3), 809–834.
- Tsiatis AA, Degruttola V, Wulfsohn MS (1995). “Modeling the Relationship of Survival to Longitudinal Data Measured with Error. Applications to Survival and CD4 Counts in Patients with AIDS.” *Journal of the American Statistical Association*, **90**(429), 27–37. doi:[10.1080/01621459.1995.10476485](https://doi.org/10.1080/01621459.1995.10476485).
- Verbeke G, Davidian M (2008). “Joint Models for Longitudinal Data: Introduction and Overview.” In G Fitzmaurice, M Davidian, G Verbeke, G Molenberghs (eds.), *Longitudinal Data Analysis*, Chapman & Hall/CRC Handbooks of Modern Statistical Methods, pp. 319–326. Chapman & Hall/CRC.
- Wulfsohn MS, Tsiatis AA (1997). “A Joint Model for Survival and Longitudinal Data Measured with Error.” *Biometrics*, **53**(1), 330–339. doi:[10.2307/2533118](https://doi.org/10.2307/2533118).
- Xu C, Baines PD, Wang JL (2014). “Standard Error Estimation Using the EM Algorithm for the Joint Modeling of Survival and Longitudinal Data.” *Biostatistics*, **15**(4), 731–744. doi:[10.1093/biostatistics/kxu015](https://doi.org/10.1093/biostatistics/kxu015).
- Xu C, Hadjipantelis PZ, Wang JL (2020). **JSM: Semiparametric Joint Modeling of Survival and Longitudinal Data**. R package version 1.0.0, URL <https://CRAN.R-project.org/package=JSM>.
- Xu J, Zeger SL (2001). “Joint Analysis of Longitudinal Data Comprising Repeated Measures and Times to Events.” *Journal of the Royal Statistical Society C*, **50**(3), 375–387. doi:[10.1111/1467-9876.00241](https://doi.org/10.1111/1467-9876.00241).
- Yao F, Müller HG, Wang JL (2005). “Functional Data Analysis for Aparse Longitudinal Data.” *Journal of the American Statistical Association*, **100**(470), 577–590. doi:[10.1198/016214504000001745](https://doi.org/10.1198/016214504000001745).
- Yu M, Law NJ, Taylor JM, Sandler HM (2004). “Joint Longitudinal-Survival-Cure Models and Their Application to Prostate Cancer.” *Statistica Sinica*, **14**(3), 835–862.
- Zeng D, Lin D (2007). “Maximum Likelihood Estimation in Semiparametric Regression Models with Censored Data.” *Journal of the Royal Statistical Society B*, **69**(4), 507–564. doi:[10.1111/j.1369-7412.2007.00606.x](https://doi.org/10.1111/j.1369-7412.2007.00606.x).
- Zhang D, Chen MH, Ibrahim JG, Boye ME, Shen W (2016). “**JMFit**: A SAS Macro for Joint Models of Longitudinal and Survival Data.” *Journal of Statistical Software*, **71**(3), 1–24. doi:[10.18637/jss.v071.i03](https://doi.org/10.18637/jss.v071.i03).

A. Effect of the number of quadrature points

Here we present a numerical study to explore the performance of the two model-fitting functions, i.e., `jmodelTM()` and `jmodelMult()`, running with different number of quadrature points. The results, obtained by running on an Intel Core i7 4.0 GHz iMac with 16GB DDR3, are summarized in Tables 1 and 2. Note that `nknot` is always odd as an odd number of quadrature points provides higher degree of precision. From Table 1, we observe that the parameter estimates and log-likelihood values are quite stable under varying `nknot`. Specifically, the model fits with `nknot = 7` and `code = 11` yield exactly the same parameter estimates. On the other hand, Table 2 indicates that although not as stable as `jmodelTM()`, the model fits by `jmodelMult()` with `nknot = 11` and `nknot = 15` yield very similar results.

Parameter	<code>nknot = 3</code>	<code>nknot = 5</code>	<code>nknot = 7</code>	<code>nknot = 11</code>
β_1	69.82709	69.82579	69.82578	69.82578
β_2	7.08302	7.08299	7.08300	7.08300
β_3	0.67439	0.67445	0.67446	0.67446
β_4	-0.27302	-0.27292	-0.27292	-0.27292
ϕ	0.18419	0.18411	0.18411	0.18411
α	-0.03922	-0.03922	-0.03922	-0.03922
log-likelihood	-15055.08	-15055.09	-15055.09	-15055.09
run time	13.031	24.554	43.425	86.864

Table 1: Parameter estimates, log-likelihood values, and elapsed time in seconds when fitting model (18) using `jmodelTM()` with different values of `nknot`.

Parameter	<code>nknot = 3</code>	<code>nknot = 7</code>	<code>nknot = 11</code>	<code>nknot = 15</code>
γ_1	0.63386	0.63968	0.64039	0.64037
γ_2	-0.10009	-0.10156	-0.10169	-0.10169
γ_3	0.06325	0.06408	0.06415	0.06415
γ_4	0.27655	0.27913	0.27945	0.27944
γ_5	1.40227	1.41714	1.41895	1.41891
γ_6	2.10635	2.12563	2.12794	2.12788
γ_7	-0.10822	-0.10946	-0.10959	-0.10959
γ_8	0.06663	0.06676	0.06680	0.06680
γ_9	-0.19700	-0.20155	-0.20196	-0.20197
γ_{10}	-0.90054	-0.91095	-0.91174	-0.91174
ϕ	0.07542	0.07427	0.07435	0.07434
α	1.09165	1.09348	1.09395	1.09394
log-likelihood	-2532.285	-2532.217	-2532.197	-2532.198
run time	10.761	22.102	29.012	36.707

Table 2: Parameter estimates, log-likelihood values, and elapsed time in seconds when fitting model (21) using `jmodelMult()` with different values of `nknot`.

Affiliation:

Cong Xu
Department of Statistics and Data Science
Southern University of Science and Technology
Shenzhen, China
E-mail: helenxu1112@gmail.com