JM: An R Package for the Joint Modelling of Longitudinal and Time-to-Event Data

Dimitris Rizopoulos

Erasmus University Medical Center

Abstract

In longitudinal studies measurements are often collected on different types of outcomes for each subject. These may include several longitudinally measured responses (such as blood values relevant to the medical condition under study) and the time at which an event of particular interest occurs (e.g., death, development of a disease or dropout from the study). These outcomes are often separately analyzed; however, in many instances, a joint modeling approach is either required or may produce a better insight into the mechanisms that underlie the phenomenon under study. In this paper we present the R package **JM** that fits joint models for longitudinal and time-to-event data.

Keywords: attrition, dropout, longitudinal data, shared parameter models, survival data, time-dependent covariates.

1. Introduction

Longitudinal studies often produce two types of outcome, namely a set of longitudinal response measurements and the time to an event of interest, such as death, development of a disease or dropout from the study. Two typical examples of this setting are HIV and cancer studies. In HIV studies patients who have been infected are monitored until they develop AIDS or die, and they are regularly measured for the condition of the immune system using markers such as the CD4 lymphocyte count or the estimated viral load. Similarly in cancer studies the event outcome is the death or metastasis and patients also provide longitudinal measurements of antibody levels or of other markers of carcinogenesis, such as the PSA levels for prostate cancer.

These two outcomes are often separately analyzed using a mixed effects model for the longitudinal outcome and a survival model for the event outcome. However, in mainly two settings a joint modelling approach is required. First, when interest is on the event outcome and we wish to account for the effect of the longitudinal outcome as a time-dependent covariate, traditional approaches for analyzing time-to-event data (such as the partial likelihood for the Cox proportional hazards models) are not applicable. In particular, standard time-to-event models require that time-dependent covariates are external; that is, the value of this covariate at time point t is not affected by the occurrence of an event at time point u, with $t > u$ (Kalbfleisch and Prentice 2002, Section 6.3). However, the type of time-dependent covariates encountered in longitudinal studies do not satisfy this condition, due to the fact that they are the output of a stochastic process generated by the subject, which is directly related to the failure mechanism. Therefore, in order to produce valid inferences a model for the joint distribution of the longitudinal and survival outcomes is required instead. The second setting in which joint modelling is required is when interest is on the longitudinal outcome. In this case the occurrence of events causes dropout since no longitudinal measurements are available at and after the event time. When this dropout is nonrandom (i.e., when the probability of dropout depends on unobserved longitudinal responses), then bias may arise from an analysis that ignores the dropout process (Little and Rubin 2002, Chapter 15). To avoid this problem and obtain valid inferences the longitudinal and dropout process must be jointly modelled. One of the modelling frameworks that have been proposed in the missing data literature for handling nonrandom dropout is the shared parameter model (Follmann and Wu 1995), which postulates a mixed effects model for the longitudinal outcome and time-to-dropout model for the missingness process. In both settings the joint distribution of the event times and the longitudinal measurements is modelled via a set of random effects that are assumed to account for the associations between these two outcomes. Excellent overviews of this area of biostatistics and statistics research are given by Tsiatis and Davidian (2004) and Yu, Law, Taylor, and Sandler (2004).

Software for the separate analysis of longitudinal and event time data has been available for many years. For instance, in R (R Development Core Team 2010) several packages provide functions for fitting mixed effects models, such as package **nlme** (Pinheiro, Bates, DebRoy, Sarkar, and R Development Core Team 2009) and package **lme4** (Bates and Maechler 2010), and for survival models package **survival** (Therneau and Lumley 2009) – for a more comprehensive list of the available packages in the Comprehensive R Archive Network (CRAN) relevant to the analysis of event time data we refer to the CRAN task view "Survival Analysis" (Allignol and Latouche 2010). In this paper we focus in the joint modelling of longitudinal and time-to-event data, and we present the R package **JM** (available from CRAN at http://CRAN.R-project.org/package=JM) that can be used to fit such joint models. The implementation of joint models in SAS and **WinBUGS** has been discussed by Guo and Carlin (2004). The rest of the paper is organised as follows. In Section 2 we present the joint modelling framework, and discuss different choices for the time-to-event submodel. Furthermore, we describe the basics of maximum likelihood estimation for this type of models, with special mention to numerical integration techniques. In addition, we refer to several types of residuals and prediction of conditional probabilities of survival. Section 3 describes in detail the implementation of the theory described in Section 2 in package **JM**. Finally, Section 4 illustrates the capabilities of the package in a real data example from HIV research, and Section 5 gives some concluding remarks.

2. Joint modelling framework

2.1. Submodels specification

In the following we will present the joint modelling framework motivated by the time-to-event point of view (i.e., in the setting in which we want to incorporate a time-dependent covariate measured with error in a survival model) – a more direct connection with the missing data framework is made in Section 2.3. Let T_i denote the observed failure time for the i-th subject $(i = 1, \ldots, n)$, which is taken as the minimum of the true event time T_i^* and the censoring time C_i, i.e., $T_i = \min(T_i^*, C_i)$. Furthermore, we define the event indicator as $\delta_i = I(T_i^* \leq C_i)$, where $I(\cdot)$ is the indicator function that takes the value 1 if the condition $T_i^* \leq C_i$ is satisfied, and 0 otherwise. Thus, the observed data for the time-to-event outcome consist of the pairs $\{(T_i, \delta_i), i = 1, \ldots, n\}$. For the longitudinal responses, let $y_i(t)$ denote the value of the longitudinal outcome at time point t for the i-th subject. We should note here that we do not actually observe $y_i(t)$ at all time points, but only at the very specific occasions t_{ij} at which measurements were taken. Thus, the observed longitudinal data consist of the measurements $y_{ij} = \{y_i(t_{ij}), j = 1, \ldots, n_i\}$.

Our aim is to associate the *true* and *unobserved* value of the longitudinal outcome at time t, denoted by $m_i(t)$, with the event outcome T_i^*. Note that $m_i(t)$ is different from $y_i(t)$, with the latter being the contaminated with measurement error value of the longitudinal outcome at time t. To quantify the effect of $m_i(t)$ on the risk for an event, a standard option is to use a relative risk model of the form (Therneau and Grambsch 2000):

$$
\begin{aligned}
h_i(t \mid \mathcal{M}_i(t), w_i) &= \lim_{dt \to 0} \Pr\{t \leq T_i^* < t + dt \mid T_i^* \geq t, \mathcal{M}_i(t), w_i\}/dt \\
&= h_0(t) \exp\{\gamma^\top w_i + \alpha m_i(t)\},
\end{aligned}
\tag{1}
$$

where $\mathcal{M}_i(t) = \{m_i(u), 0 \leq u < t\}$ denotes the history of the true unobserved longitudinal process up to time point t, $h_0(\cdot)$ denotes the baseline risk function, and w_i is a vector of baseline covariates (such as a treatment indicator, history of diseases, etc.) with a corresponding vector of regression coefficients γ. Similarly, parameter α quantifies the effect of the underlying longitudinal outcome to the risk for an event; for instance, in the AIDS example mentioned in Section 1, α measures the effect of the number of CD4 cells to the risk for death. To avoid the impact of parametric assumptions, the baseline risk function $h_0(\cdot)$ is typically left unspecified. However, within the joint modelling framework Hsieh, Tseng, and Wang (2006) have recently noted that leaving this function completely unspecified leads to an underestimation of the standard errors of the parameter estimates. In particular, problems arise due to the fact that the nonparametric maximum likelihood estimate for this function cannot be obtained explicitly under the full joint modelling approach. To avoid this problem, we could either opt for the hazard function of a standard survival distribution (such as the Weibull or Gamma) or for more flexible models in which $h_0(t)$ is sufficiently approximated using step functions or spline-based approaches.

An alternative modelling framework for event time data, especially when the proportionality assumption in (1) fails, is the accelerated failure time model. In order to incorporate a time-dependent covariate within this framework, we let $\mathcal{S}_0$ denote an absolutely continuous baseline survival function, and we follow the formulation of Cox and Oakes (1984, Section 5.2) that

postulates

$$\left\{ \int_0^{T^*} \exp\{\gamma^\top w + \alpha m(s)\} \, ds \right\} \sim \mathcal{S}_0.$$

This can be reexpressed in terms of the risk rate function for subject i as:

$$h_i(t \mid \mathcal{M}_i(t), w_i) \quad = \quad h_0\{V_i(t)\} \exp\{\gamma^\top w_i + \alpha m_i(t)\}, \tag{2}$$

with

$$V_i(t) = \int_0^t \exp\{\gamma^\top w_i + \alpha m_i(s)\} \, ds.$$

Similarly, to (1), the baseline risk function $h_0(\cdot)$ can be assumed of a specific parametric form or modelled flexibly. An important difference of (2) compared to (1) is that in the former the entire covariate history $\mathcal{M}_i(t)$ is assumed to influence the subject-specific risk (due to the fact that $h_0(\cdot)$ is evaluated at $V_i(t)$), whereas in the latter the subject-specific risk depends only on the current value of the time-dependent covariate $m_i(t)$. The survival function for a subject with covariate history $\mathcal{M}_i(t)$ equals $\mathcal{S}_i\{t \mid \mathcal{M}_i(t)\} = \mathcal{S}_0\{V_i(t)\}$, which means that this subject ages on an accelerated schedule $V_i(t)$ compared to $\mathcal{S}_0$ – for more information we refer to Cox and Oakes (1984, Section 5.2). Within the joint modelling framework accelerated failure time models have been discussed by Tseng, Hsieh, and Wang (2005).

In the definitions of the survival models presented above we used $m_i(t)$ to denote the true value of the underlying longitudinal covariate at time point t. However and as mentioned earlier, longitudinal information is actually collected intermittently and with error at a set of few time points t_{ij} for each subject. Therefore, in order to measure the effect of this covariate to the risk for an event we need to estimate $m_i(t)$ and successfully reconstruct the complete longitudinal history $\mathcal{M}_i(t)$, using the available measurements $y_{ij} = \{y_i(t_{ij}), j = 1, \ldots, n_i\}$ of each subject and a set of modelling assumptions. For the remaining of this paper we will focus on normal data and we will postulate a linear mixed effects model to describe the subject-specific longitudinal evolutions. In particular, we have

$$\begin{aligned} y_i(t) &= m_i(t) + \varepsilon_i(t) \\ &= x_i^\top(t)\beta + z_i^\top(t)b_i + \varepsilon_i(t), \quad \varepsilon_i(t) \sim \mathcal{N}(0, \sigma^2), \end{aligned} \tag{3}$$

where β denotes the vector of the unknown fixed effects parameters, b_i denotes a vector of random effects, $x_i(t)$ and $z_i(t)$ denote row vectors of the design matrices for the fixed and random effects, respectively, and $\varepsilon_i(t)$ is the measurement error term, which is assumed independent of b_i, and with variance σ^2. We should note that care should be taken in the specification of $x_i(t)$ and $z_i(t)$ in order to produce a good estimate of $\mathcal{M}_i(t)$. The main reason for this is that, as we will see later in Section 2.2, in the definition of the likelihood of the joint model the complete longitudinal history is required for the computation of the survival function, and of the risk function under the accelerated failure time formulation. Therefore, in applications in which subjects show highly nonlinear longitudinal trajectories, it is advisable to consider flexible representations for $x_i(t)$ and $z_i(t)$ using a possibly high-dimensional vector of functions of time t, expressed in terms of high-order polynomials or splines (Ding and Wang 2008; Brown, Ibrahim, and DeGruttola 2005). Finally, in order to complete the specification of the longitudinal submodel a suitable distributional assumption for the random effects component is required. A standard choice for this distribution is the multivariate normal distribution; however, within the joint modelling framework and mainly

because of the nonrandom dropout (see also Section 2.3), there is the concern that relying on a standard parametric distribution may influence the derived inferences especially when this distribution differs considerably from the true random effects distribution. This motivated Song, Davidian, and Tsiatis (2002) to propose a more flexible model for the distribution of the random effects that is expressed as a normal density times a polynomial function. However, the findings of these authors suggested that the parameter estimates and standard errors of joint models fitted under the normal assumption for the random effects were rather robust to misspecification. This feature has been further theoretically corroborated by Rizopoulos, Verbeke, and Molenberghs (2008), who showed that as the number of repeated measurements per subject n_i increases, a misspecification of the random effects distribution has a minimal effect in parameter estimators and standard errors. Thus, here we will assume that $b_i \sim \mathcal{N}(0, D)$ and we will not further investigate this assumption.

2.2. Maximum likelihood estimation

The main estimation methods that have been proposed for joint models are (semiparametric) maximum likelihood (Hsieh *et al.* 2006; Henderson, Diggle, and Dobson 2000; Wulfsohn and Tsiatis 1997) and Bayes using MCMC techniques (Chi and Ibrahim 2006; Brown and Ibrahim 2003; Wang and Taylor 2001; Xu and Zeger 2001). Moreover, Tsiatis and Davidian (2001) have proposed a conditional score approach in which the random effects are treated as nuisance parameters, and they developed a set of unbiased estimating equations that yields consistent and asymptotically normal estimators. Here we give the basics of the maximum likelihood method for joint models as the one of the more traditional approaches.

Maximum likelihood estimation for joint models is based on the maximization of the log-likelihood corresponding to the joint distribution of the time-to-event and longitudinal outcomes $\{T_i, \delta_i, y_i\}$. To define this joint distribution we will assume that the vector of time-independent random effects b_i underlies both the longitudinal and survival processes. This means that these random effects account for both the association between the longitudinal and event outcomes, and the correlation between the repeated measurements in the longitudinal process (conditional independence). Formally we have that,

$$
\begin{aligned}
p(T_i, \delta_i, y_i \mid b_i; \theta) &= p(T_i, \delta_i \mid b_i; \theta)\, p(y_i \mid b_i; \theta) & (4) \\
p(y_i \mid b_i; \theta) &= \prod_j p\{y_i(t_{ij}) \mid b_i; \theta\}, & (5)
\end{aligned}
$$

where $\theta = (\theta_t^\top, \theta_y^\top, \theta_b^\top)^\top$ denotes the parameter vector, with θ_t denoting the parameters for the event time outcome, θ_y the parameters for the longitudinal outcomes and θ_b the unique parameters of the random-effects covariance matrix, y_i is the $n_i \times 1$ vector of longitudinal responses of the i-th subject, and $p(\cdot)$ denotes an appropriate probability density function. Due to the fact that the survival and longitudinal submodels *share* the same random effects, joint models of this type are also known as shared parameter models. Under the modelling assumptions presented in the previous section, and these conditional independence assumptions the joint log-likelihood contribution for the i-th subject can be formulated as

$$
\log p(T_i, \delta_i, y_i; \theta) = \log \int p(T_i, \delta_i \mid b_i; \theta_t, \beta) \Big[\prod_j p\{y_i(t_{ij}) \mid b_i; \theta_y\} \Big] p(b_i; \theta_b)\, db_i, \qquad (6)
$$

where the likelihood of the survival part is written as

$$
p(T_i, \delta_i \mid b_i; \theta_t, \beta) = \{h_i(T_i \mid \mathcal{M}_i(T_i); \theta_t, \beta)\}^{\delta_i} \mathcal{S}_i(T_i \mid \mathcal{M}_i(T_i); \theta_t, \beta), \qquad (7)
$$

with $h_i(\cdot)$ given by either (1) or (2), and

$$
\begin{aligned}
\mathcal{S}_i(t \mid \mathcal{M}_i(t), w_i; \theta_t, \beta) &= \mathrm{Pr}(T_i^* > t \mid \mathcal{M}_i(t), w_i; \theta_t, \beta) \\
&= \exp\left\{ -\int_0^t h_i(s \mid \mathcal{M}_i(s); \theta_t, \beta)\, ds \right\},
\end{aligned}
\tag{8}
$$

$p\{y_i(t_{ij}) \mid b_i; \theta_y\}$ is the univariate normal density for the longitudinal responses, and $p(b_i; \theta_b)$ is the multivariate normal density for the random effects.

Maximization of the log-likelihood function corresponding to (6) with respect to θ is a computationally challenging task. This is mainly because both the integral with respect to the random effects in (6), and the integral in the definition of the survival function (8) do not have an analytical solution, except in very special cases. Standard numerical integration techniques such as Gaussian quadrature and Monte Carlo have been successfully applied in the joint modelling framework (Song *et al.* 2002; Henderson *et al.* 2000; Wulfsohn and Tsiatis 1997). Furthermore, Rizopoulos, Verbeke, and Lesaffre (2009) have recently discussed the use of Laplace approximations for joint models, that can be especially useful in high-dimensional random effects settings (e.g., when splines are used in the random effects design matrix). For the maximization of the approximated log-likelihood the Expectation-Maximization (EM) algorithm has been traditionally used in which the random effects are treated as 'missing data'. The main motivation for using this algorithm is the closed-form M-step updates for certain parameters of the joint model. However, a serious drawback of the EM algorithm is its linear convergence rate that results in slow convergence especially near the maximum. Nonetheless, Rizopoulos *et al.* (2009) have noted that a direct maximization of the observed data log-likelihood, using for instance, a quasi-Newton algorithm (Lange 2004), requires very similar computations to the EM algorithm. Therefore hybrid optimization approaches that start with EM and then continue with direct maximization can be easily employed.

2.3. Residuals

A traditional approach to check model assumptions is the inspection of residual plots. Properties and features of residuals, when longitudinal and survival outcomes are separately modelled, have been extensively studied in the literature. For instance, different types of residuals for linear mixed models are discussed in Nobre and Singer (2007) and Verbeke and Molenberghs (2000), whereas residuals for parametric and semiparametric survival models are presented in Harrell (2001) and Therneau and Grambsch (2000). However, calculating these residuals based on the fitted joint model and the observed data, and then using them to check the model assumptions may prove problematic. In particular, complications arise due to the nonrandom dropout in the longitudinal process caused by the occurrence of events. To show this more clearly, we define for each subject the observed and missing part of the longitudinal response vector. The observed part $y_i^o = \{y_i(t_{ij}) : t_{ij} < T_i, j = 1, \dots, n_i\}$ contains all observed longitudinal measurements of the i-th subject before the observed event time, whereas the missing part $y_i^m = \{y_i(t_{ij}) : t_{ij} \geq T_i, j = 1, \dots, n_i'\}$ contains the longitudinal measurements that would have been taken until the end of the study, had the event not occurred. Under these definitions and the assumptions of the joint modelling framework, we can derive the dropout mechanism, which is the conditional distribution of the time-to-dropout given the complete vector of longitudinal responses (y_i^o, y_i^m),

$$
p(T_i^* \mid y_i^o, y_i^m; \theta) = \int p(T_i^* \mid b_i; \theta)\, p(b_i \mid y_i^o, y_i^m; \theta)\, db_i.
\tag{9}
$$

We observe that the time-to-dropout depends on y_i^m through the posterior distribution of the random effects $p(b_i \mid y_i^o, y_i^m; \theta)$. This in practice implies that the observed data, upon which the residuals are calculated, are not a random sample of the target population, and therefore should not be expected to exhibit the standard properties of zero mean, constant variance, etc.

To overcome this problem and produce residuals that can be readily used in diagnostic plots for joint models, Rizopoulos, Verbeke, and Molenberghs (2010) have recently proposed to augment the observed data with randomly imputed longitudinal responses under the complete data model, corresponding to the longitudinal outcomes that would have been observed had the patients not dropped out. To present this idea briefly, we assume that the joint model has been fitted to the data set at hand, and that we have obtained the maximum likelihood estimates $\hat{\theta}$ and an estimate of their asymptotic covariance matrix, say $\hat{\mathcal{H}}$. Moreover, we assume that longitudinal measurements are planned to be taken at prespecified time points $t_0, t_1, \ldots, t_{max}$, and that for the i-th subject measurements are available up to the last prespecified visit time earlier than T_i. In the following we adopt a Bayesian point of view for the joint modelling framework, since multiple imputation has Bayesian grounds (Little and Rubin 2002, Chapter 10). Multiple imputation is based on repeated sampling from the posterior distribution of y_i^m given the observed data, averaged over the posterior of the parameters. Under joint model (6) and dropout mechanism (9), the density for this distribution can be expressed as

$$p(y_i^m \mid y_i^o, T_i, \delta_i) = \int p(y_i^m \mid y_i^o, T_i, \delta_i; \theta) \, p(\theta \mid y_i^o, T_i, \delta_i) \, d\theta. \tag{10}$$

The first part of the integrand in (10) can be derived from (9) taking also into account assumptions (4) and (5), i.e.,

$$
\begin{aligned}
p(y_i^m \mid y_i^o, T_i, \delta_i; \theta) &= \int p(y_i^m \mid b_i, y_i^o, T_i, \delta_i; \theta) \, p(b_i \mid y_i^o, T_i, \delta_i; \theta) \, db_i \\
&= \int p(y_i^m \mid b_i; \theta) \, p(b_i \mid y_i^o, T_i, \delta_i; \theta) \, db_i. \tag{11}
\end{aligned}
$$

For the second part, which is the posterior distribution of the parameters given the observed data, we use arguments of standard asymptotic Bayesian theory (Cox and Hinkley 1974, Section 10.6), and assume that the sample size n is sufficiently large such that $\{\theta \mid y_i^o, T_i, \delta_i\}$ can be well approximated by $\mathcal{N}(\hat{\theta}, \hat{\mathcal{H}})$. This assumption, combined with (10) and (11), suggests the following simulation scheme:

1. Draw $\theta^{(\ell)} \sim \mathcal{N}(\hat{\theta}, \hat{\mathcal{H}})$.

2. Draw $b_i^{(\ell)} \sim \{b_i \mid y_i^o, T_i, \delta_i, \theta^{(\ell)}\}$.

3. Draw $y_i^{m(\ell)}(t_{ij}) \sim \mathcal{N}\left\{\mu_i^{(l)}(t_{ij}), \hat{\sigma}_y^{2,(\ell)}\right\}$, for the prespecified visit times $t_{ij} \geq T_i$, $j = 1, \ldots, n_i'$ that were not observed for the i-th subject, where $\mu_i^{(l)}(t_{ij}) = x_i^\top(t_{ij})\hat{\beta}^{(\ell)} + z_i^\top(t_{ij})\hat{b}_i^{(\ell)}$.

4. Repeat Steps 1–3 for each subject, $\ell = 1, \ldots, L$ times, where L denotes the number of imputations.

Steps 1 and 2 account for uncertainties in the parameters and empirical Bayes estimates, respectively, whereas Step 3 imputes the missing longitudinal responses (more information

can be found in Rizopoulos *et al.* 2010). The benefit of using the simulated $y_i^{m(\ell)}(t_{ij})$ values together with y_i^o to calculate residuals is that these residuals inherit now the properties of the complete data model, and therefore they can be directly used in diagnostic plots without requiring to take dropout into account.

2.4. Expected survival

The calculation of predicted survival probabilities based on a fitted survival model on some reference population has received a lot of attention in the statistical literature – see for instance, Therneau and Grambsch (2000, Chapter 10) and references therein. In this section we focus on expected survival but within the joint modelling framework. In particular, based on a joint model fitted on sample of size n, we are interested in predicting survival probabilities for a new subject i that has provided a set of longitudinal measurements $\mathcal{Y}_i(t) = \{y_i(s), 0 \leq s \leq t\}$ (dependence on baseline covariates is assumed but is suppressed here for ease of exposition). Note that in the joint modelling framework, providing longitudinal measurements up to time point t in fact implies survival up to time t. Thus, we are actually interested in estimating the conditional probability

$$\pi_i(u \mid t) = \Pr(T_i^* \geq u \mid T_i^* > t, \mathcal{Y}_i(t), \mathcal{D}_n), \tag{12}$$

where $u > t$ and $\mathcal{D}_n$ denotes the sample on which the joint model was fitted. Similarly to Section 2.3, it is convenient to proceed using a Bayesian formulation of the problem. More specifically, (12) can be written as

$$\Pr(T_i^* \geq u \mid T_i^* > t, \mathcal{Y}_i(t), \mathcal{D}_n) = \int \Pr(T_i^* \geq u \mid T_i^* > t, \mathcal{Y}_i(t); \theta) \, p(\theta \mid \mathcal{D}_n) \, d\theta. \tag{13}$$

The first part of the integrand and using assumption (4) is given by

$$
\begin{aligned}
\Pr(T_i^* \geq u \mid T_i^* > t, \mathcal{Y}_i(t); \theta) &= \int \Pr(T_i^* \geq u \mid T_i^* > t, \mathcal{Y}_i(t), b_i; \theta) \, p(b_i \mid T_i^* > t, \mathcal{Y}_i(t); \theta) \, db_i \\
&= \int \Pr(T_i^* \geq u \mid T_i^* > t, b_i; \theta) \, p(b_i \mid T_i^* > t, \mathcal{Y}_i(t); \theta) \, db_i \\
&= \int \frac{\mathcal{S}_i\{u \mid \mathcal{M}_i(u, b_i, \theta); \theta\}}{\mathcal{S}_i\{t \mid \mathcal{M}_i(t, b_i, \theta); \theta\}} \, p(b_i \mid T_i^* > t, \mathcal{Y}_i(t); \theta) \, db_i, \tag{14}
\end{aligned}
$$

where $\mathcal{S}_i(\cdot)$ is given by (8), and furthermore we have explicitly noted that the longitudinal history $\mathcal{M}_i(\cdot)$, as estimated by the linear mixed effects model, is a function of both the random effects and the parameters. For the second part we again follow asymptotic Bayesian arguments and assume that $\{\theta \mid \mathcal{D}_n\}$ can be well approximated by a multivariate normal distribution with mean the maximum likelihood estimates $\hat{\theta}$ and covariance matrix $\hat{\mathcal{H}} = \mathrm{v\hat{a}r}(\hat{\theta})$. Combining (13) with (14) and $\{\theta \mid \mathcal{D}_n\} \sim \mathcal{N}(\hat{\theta}, \hat{\mathcal{H}})$, a Monte Carlo estimate of $\pi_i(u \mid t)$ can be obtained using the following simulation scheme:

1. Draw $\theta^{(\ell)} \sim \mathcal{N}(\hat{\theta}, \hat{\mathcal{H}})$.

2. Draw $b_i^{(\ell)} \sim \{b_i \mid T_i^* > t, \mathcal{Y}_i(t), \theta^{(\ell)}\}$.

3. Compute $\pi_i^{(\ell)}(u \mid t) = \mathcal{S}_i\{u \mid \mathcal{M}_i(u, b_i^{(\ell)}, \theta^{(\ell)}); \theta^{(\ell)}\}/\mathcal{S}_i\{t \mid \mathcal{M}_i(t, b_i^{(\ell)}, \theta^{(\ell)}); \theta^{(\ell)}\}$.

4. Repeat Steps 1–3 for each subject, $\ell = 1, \ldots, L$ times, where L denotes the number of Monte Carlo samples.

Step 1 is straightforward to perform; on the contrary, the posterior distribution for the random effects given the observed data for subject i in Step 2 is of a non-standard form, and thus a more sophisticated approach is required to sample from it. Here we use of a Metropolis-Hastings algorithm with independent proposals from a multivariate t distribution centered at the empirical Bayes estimates $\hat{b}_i = \arg\max_b\{\log p(T_i^* > t, \mathcal{Y}_i(t), b; \hat{\theta})\}$ (where $\log p(T_i^* > t, \mathcal{Y}_i(t), b; \hat{\theta}) = \log \mathcal{S}_i(t \mid b; \hat{\theta}) + \log p(\mathcal{Y}_i(t) \mid b; \hat{\theta}) + \log p(b; \hat{\theta})$), with scale matrix $\mathrm{var}(\hat{b}_i) = \{-\partial^2 \log p(T_i^* > t, \mathcal{Y}_i(t), b; \hat{\theta})/\partial b^\top \partial b \mid_{b=\hat{b}_i}\}^{-1}$, and four degrees of freedom. We expect the multivariate t proposals to work satisfactorily in this setting because, as it has been recently shown by Rizopoulos *et al.* (2008), as n_i (the number of longitudinal measurements for subject i) increases, the leading term of the log posterior distribution of the random effects is the logarithm of the density of the linear mixed model $\log p(\mathcal{Y}_i(t) \mid b_i; \theta^{(\ell)}) = \sum_j \log p\{y_i(t_{ij}) \mid b_i; \theta^{(\ell)}\}$, which is quadratic in b_i and will resemble the shape of a multivariate normal distribution. The realizations $\{\pi_i^{(\ell)}(u \mid t), \ell = 1, \ldots, L\}$ can be used to derive estimates of $\pi_i(u \mid t)$, such as

$$\hat{\pi}_i(u \mid t) = \mathrm{median}\{\pi_i^{(\ell)}(u \mid t), \ell = 1, \ldots, L\} \quad \text{or} \quad \hat{\pi}_i(u \mid t) = L^{-1}\sum_{\ell=1}^{L}\pi_i^{(\ell)}(u \mid t),$$

and confidence intervals using the Monte Carlo sample percentiles.

3. The **R** package JM

3.1. Design

The R package **JM** has been developed to fit a variety of joint models for normal longitudinal responses and time-to-event data under maximum likelihood. This package has a single model-fitting function called `jointModel()`, which accepts as main arguments a linear mixed effects object fit as returned by function `lme()` of package **nlme**, and a survival object fit as returned by either function `coxph()` or function `survreg()` of package **survival**. The `method` argument of `jointModel()` specifies the type of the survival submodel to be fitted and the type of numerical integration method; available options are:

`method = "weibull-AFT-GH"`: the Weibull model under the accelerated failure time formulation (i.e., model (2) with $h_0(\cdot)$ being the Weibull baseline risk function). The Gauss-Hermite integration rule is used to approximate integral (6).

`method = "weibull-PH-GH"`: the Weibull model under the relative risk formulation (i.e., model (1) with $h_0(\cdot)$ being the Weibull baseline risk function). The Gauss-Hermite integration rule is used to approximate integral (6).

`method = "piecewise-PH-GH"`: the relative risk model (1) with a piecewise-constant baseline risk function. In particular,

$$h_0(t) = \sum_{q=1}^{Q} \xi_q I(v_{q-1} < t \le v_q), \tag{15}$$

where $0 = v_0 < v_1 < \cdots < v_Q$ denotes a split of the time scale, with v_Q being larger than the largest observed time, and ξ_q denotes the value of the hazard in the interval $(v_{q-1}, v_q]$. The Gauss-Hermite integration rule is used to approximate integral (6).

`method = "Cox-PH-GH"`: the relative risk model (1) with an unspecified baseline risk function. The Gauss-Hermite integration rule is used to approximate integral (6). This option corresponds to the joint model proposed by Wulfsohn and Tsiatis (1997).

`method = "spline-PH-GH"`: the relative risk model (1) with a spline-approximated baseline risk function. In particular, the log baseline risk function $\log h_0(t)$ is expanded into B-spline basis functions as follows,

$$\log h_0(t) = \kappa_0 + \sum_{d=1}^{m} \kappa_d B_d(t, q),$$

where $\kappa^\top = (\kappa_0, \kappa_1, \ldots, \kappa_m)$ are the spline coefficients, q denotes the degree of the B-splines basis functions $B(\cdot)$, and $m = \ddot{m} + q - 1$, with $\ddot{m}$ denoting the number of interior knots. The Gauss-Hermite integration rule is used to approximate integral (6).

`method = "ch-Laplace"`: for this option a similar survival model is assumed as in `method = "spline-PH-GH"`, but the integration over the random effects in (6) is achieved using a fully exponential Laplace approximation – more details regarding this approach can be found in Rizopoulos *et al.* (2009). This option is more suitable in settings in which the subject-specific longitudinal profiles are highly nonlinear and are modelled using high-dimensional random effects structures (e.g., using splines or polynomials of time). In this setting the Gauss-Hermite can prove very time consuming due to the high dimensionality of the random effects, whereas the Laplace method provides a reasonable approximation in considerably less computing time.

Currently, `jointModel()` allows for linear mixed effects submodels with iid error terms and no special structure in the random effects covariance matrix. Therefore, in the call to `lme()` to produce the object supplied as first argument to `jointModel()`, users should not specify a correlation structure (`correlation` argument), a variance function (`weights` argument) or a `pdMat` in the `random` argument.

Several supporting functions are available in the package that extract or calculate several useful statistics based on the fitted joint model, such as model summary and statistical significance for the estimated coefficients, empirical Bayes estimates (and their standard error), fitted and residuals values, and the covariance matrix of the parameter estimates. In particular, the function `jointModel()` return objects of class `jointModel`, for which the following methods are available: `print()`, `coef()`, `fixef()`, `ranef()`, `fitted()`, `residuals()`, `summary()`, `plot()`, `vcov()`. The `residuals()` method for class `jointModel` computes residuals based on both the observed data alone and the multiple-imputation residuals described in Section 2.3. In addition, the Monte Carlo procedure described in Section 2.4 for the estimation of predicted survival probabilities is implemented in function `survfitJM()`. A detailed description of these functions is available in the online help files.

3.2. Implementation details

The maximum likelihood estimates are obtained by maximizing the log-likelihood function, in which the integral in (6) is approximated using either the Gauss-Hermite rule or the fully exponential Laplace approximation, and the integral in (8) is approximated using the Gauss-Kronrod rule. For all joint models except the one with the unspecified baseline risk function, the maximization of the log-likelihood is based on a hybrid optimization procedure, which starts with the EM algorithm for a fixed number of iterations, and if convergence is not achieved switches to a quasi-Newton algorithm until convergence is attained. Available quasi-Newton algorithms are the BFGS of `optim()` (default), and the PORT routines of `nlminb()`. For the `method = "Cox-PH-GH"` the semiparametric maximum likelihood estimates are obtained using only the EM algorithm. Initial values for the parameters are taken from the linear mixed and survival models that are supplied as the first two arguments in `jointModel()`. During the EM iterations convergence is declared whenever either of the following two commonly used criteria is satisfied

$$\max\{|\theta^{(it)} - \theta^{(it-1)}|/(|\theta^{(it-1)}| + \epsilon_1)\} < \epsilon_2,$$
$$\ell(\theta^{(it)}) - \ell(\theta^{(it-1)}) < \epsilon_3\{|\ell(\theta^{(it-1)})| + \epsilon_3\},$$

where $\theta^{(it)}$ denotes the parameters values at the it-th iteration, $\ell(\theta) = \sum_i \log p(T_i, \delta_i, y_i; \theta)$, and $\epsilon_1 = 10^{-3}$, $\epsilon_2 = 10^{-4}$, and $\epsilon_3 = $ `sqrt(.Machine$double.eps)` which is about 10^{-8}. During the quasi-Newton iterations typically only the latter criterion is used. A finer control of the optimization procedure is provided via the `control` argument of `jointModel()`. This specifies among others, the optimization routine in R for the quasi-Newton algorithm (i.e., `optim()` or `nlminb()`), the tolerance values for the convergence criteria, the number of quadrature points for the Gauss-Hermite rule that approximates (6) and the Gauss-Kronrod rule that approximates (8), the type of numerical derivative that calculates the Hessian matrix based on the score vector (i.e., forward or central difference approximation), the number of EM and quasi-Newton iterations, and the position and number of knots when `method` is either `"piecewise-PH-GH"`, `"spline-PH-GH"` or `"ch-Laplace"`.

An important aspect for the success of the optimization procedure described above in approximating the true maximum likelihood estimates, is the accuracy of the Gauss-Hermite quadrature rule. As it is known from the mixed models literature (Pinheiro and Bates 1995), the choice of the number of quadrature points may influence the parameter estimates, standard errors and the log-likelihood value. Thus, it is typically advisable to investigate the fit of the model with an increasing number of quadrature points and/or consider the adaptive Gauss-Hermite rule that appropriately re-scales the quadrature points in the area where the integrand has its main mass. It is obvious that both these procedures increase substantially the computing time required to fit joint models. However, fortunately, we can decrease computational intensity without sacrificing too much precision by taking into advantage the information we have available from the linear mixed effects model fit that is provided as first argument in `jointModel()`. To illustrate how this can be achieved, we first compare the logarithm of the integrands in the linear mixed effects and joint models, given respectively as:

$$\log p(y_i, b_i; \theta) = \sum_{j=1}^{n_i} \log p\{y_i(t_{ij}) \mid b_i; \theta_y\} + \log p(b_i; \theta_b),$$

$$\log p(T_i, \delta_i, y_i, b_i; \theta) = \sum_{j=1}^{n_i} \log p\{y_i(t_{ij}) \mid b_i; \theta_y\} + \log p(b_i; \theta_b) + \log p(T_i, \delta_i \mid b_i; \theta_t, \beta).$$

Following an argument similar to the one raised in Step 2 of the Monte Carlo scheme of Section 2.4, we observe that the leading term in both integrands is the density $p(y_i \mid b_i; \theta_y) = \prod_j p\{y_i(t_{ij}) \mid b_i; \theta_y\}$ (Rizopoulos *et al.* 2008). This means that, especially as n_i increases, we expect both integrands to have their main mass around the same location. The fitting algorithms behind `jointModel()` take this information into account and scale the Gauss-Hermite quadrature points accordingly. This procedure shares similarities with the adaptive Gauss-Hermite rule, but we implement it only once, at the start of the optimization, and we do not further update the quadrature points afterwards. Based on empirical evidence, we have observed that even though we do not update the quadrature points at each iteration of the optimization algorithm (as in the adaptive Gauss-Hermite rule), this procedure performs very well in practice. The computational advantages are twofold: first, we can use fewer quadrature points than we would have used in the standard (i.e., non-adaptive) Gauss-Hermite rule, and second, we avoid the computationally demanding relocation of the quadrature points at each iteration of the adaptive Gauss-Hermite rule. A numerical study regarding the performance of this approach and further discussion is provided in Appendix A.

4. Analysis of a real data example using JM

As an illustrative example of joint modelling we consider a longitudinal study on 467 HIV infected patients who had failed or were intolerant of zidovudine therapy. The aim of the study was to compare the efficacy and safety of two alternative antiretroviral drugs, namely didanosine (ddI) and zalcitabine (ddC). Patients were randomly assigned to receive either ddI or ddC, and CD4 cell counts were recorded at study entry, where randomization took place, as well as 2, 6, 12, and 18 months thereafter. By the end of the study 188 patients had died, resulting in 59.7% censoring. More details about this data set can be found in Goldman, Carlin, Crane, Launer, Korvick, Deyton, and Abrams (1996). Our main research question here is to test for a treatment effect on survival after adjusting for the CD4 cell count. Due to the fact that the CD4 cell count measurements are in fact the output of a stochastic process generated by the patients and is only available at the specific visit times the patients came to the study center, it constitutes a typical example of time-dependent covariate measured intermittently and with error for which joint modelling is required.

The longitudinal and survival information for these patients is available in **JM** in the data frames `aids` and `aids.id`, respectively. The CD4 cell counts are known to exhibit right skewed shapes of distribution, and therefore, for the remainder of this analysis we will work with the square root of the CD4 cell values. As a descriptive analysis we present in Figure 1 the subject-specific longitudinal profiles and the Kaplan-Meier estimate for the time-to-death:

```
R> library("JM")
R> library("lattice")
R> xyplot(sqrt(CD4) ~ obstime | drug, group = patient, data = aids,
+    xlab = "Months", ylab = expression(sqrt("CD4")), col = 1, type = "l")
R> plot(survfit(Surv(Time, death) ~ drug, data = aids.id), conf.int = FALSE,
+    mark.time = TRUE, col = c("black", "red"), lty = 1:2,
+    ylab = "Survival", xlab = "Months")
R> legend("topright", c("ddC", "ddI"), lty = 1:2, col = c("black", "red"),
+    bty = "n")
```
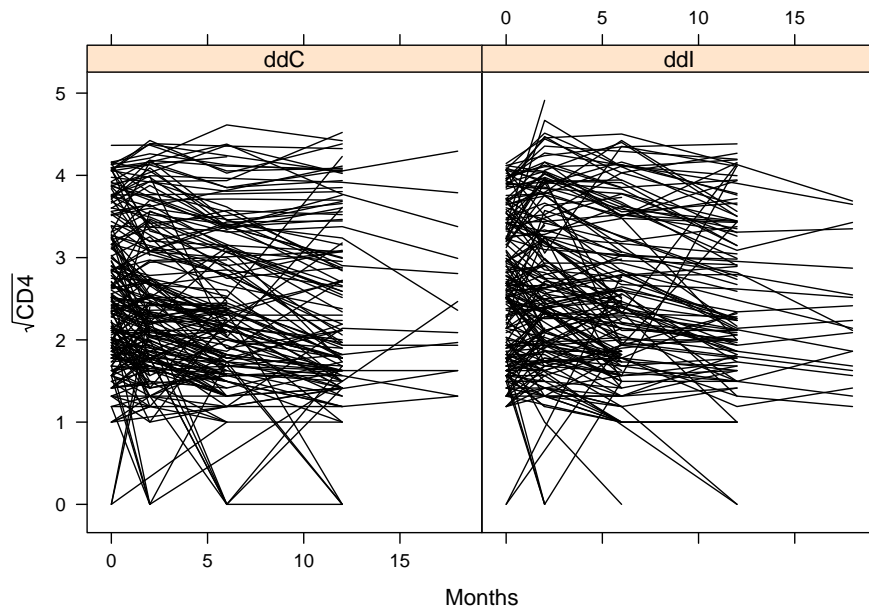
Figure 1: Subject-specific evolutions in time of the square root of the CD4 cell count measurements, separately for ddC and ddI.

We observe that in both groups patients show similar variability in their longitudinal profiles, whereas from the Kaplan-Meier estimate in Figure 2 it seems that the ddC group has slightly higher survival than the ddI group after the six month of follow-up.

To illustrate the virtues of the joint modelling approach, we will first start with a 'naive' analysis, in which we ignore the special characteristics of the CD4 cell counts and we fit a Cox model that includes the treatment indicator and CD4 as an ordinary time-dependent covariate (i.e., ignoring the measurement error). To fit this models we will use the standard counting process form of the Cox model:

```
R> td.Cox <- coxph(Surv(start, stop, event) ~ drug + sqrt(CD4), data = aids)
R> summary(td.Cox)

Call:
coxph(formula = Surv(start, stop, event) ~ drug + sqrt(CD4),
    data = aids)

  n= 1405


              coef exp(coef) se(coef)       z Pr(>|z|)
drugddI    0.32678   1.38650  0.14708   2.222   0.0263 *
sqrt(CD4) -0.72302   0.48528  0.07997  -9.042   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```
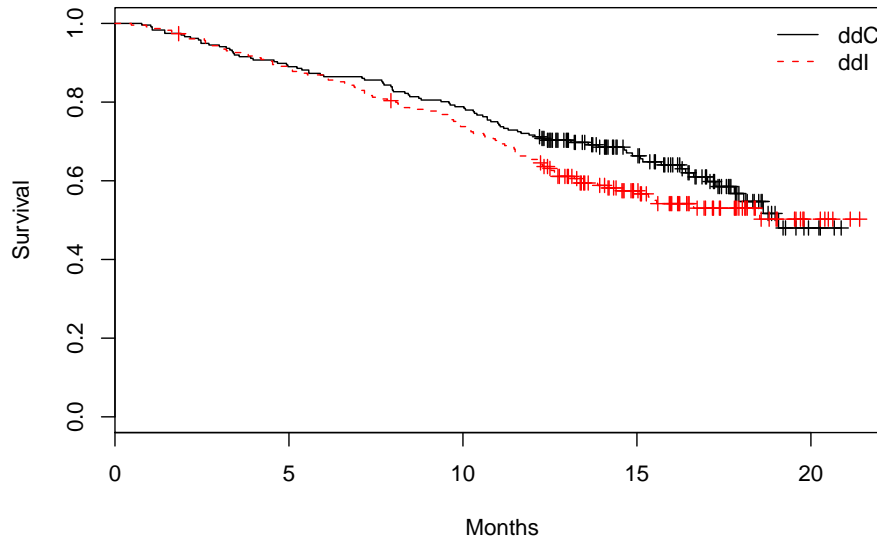
Figure 2: Kaplan-Meier estimates of the probability of survival for the ddC and ddI treatment groups.

```
          exp(coef) exp(-coef) lower .95 upper .95
drugddI      1.3865     0.7212    1.0393    1.8498
sqrt(CD4)    0.4853     2.0606    0.4149    0.5676

Rsquare= 0.059   (max possible= 0.786 )
Likelihood ratio test= 86.14  on 2 df,   p=0
Wald test            = 83.51  on 2 df,   p=0
Score (logrank) test = 83.25  on 2 df,   p=0
```

Variables `start` and `stop` denote the limits of the time intervals between visits in the study center, and `event` takes the value 1 when a patient died; for more details we refer to Therneau and Grambsch (2000, Section 3.7).

We observe that after adjusting for the square root of CD4 count in the Cox model, there is not a strong evidence for a treatment effect. We proceed by specifying and fitting a joint model that explicitly postulates a linear mixed effects model for the CD4 cell counts. In particular, taking advantage of the randomization set-up of the study, we include in the fixed-effects part of the longitudinal submodel the main effect of time and the interaction of treatment with time. In the random-effects design matrix we include an intercept and a time term. For the survival submodel and similarly to the Cox model above, we include as a time-independent covariate the treatment effect, and as time-dependent one the true underlying effect of CD4 cell count as estimated from the longitudinal model. The baseline risk function is assumed piecewise constant with six knots placed at equally spaced percentiles of the observed event

times. In particular, $\{v_1, \ldots, v_{Q-1}\}$ in (15) are specified as `quantile(Time, seq(0, 1, length = Q + 1))[-c(1, Q + 1)]`, where `Q` equals seven, and `Time` is the numeric vector that contains the observed event times T_i, $(i = 1, \ldots, n)$. In order to fit this joint model we need first to fit separately the linear mixed effects and Cox models, and then supply the returned objects as main arguments in `jointModel()`. More specifically, the joint model fitted by `jointModel()` has exactly the same structure for the longitudinal and survival submodels as these two separately fitted models, with the addition that in the survival submodel the effect of the estimated 'true' longitudinal outcome $m_i(t)$ is included in the linear predictor. Furthermore, due to the fact that `jointModel()` extracts all the required information from these two objects (e.g., response vectors, design matrices, etc.), in the call to `coxph()` we need to specify the argument `x = TRUE` in order for the design matrix of the Cox model to be included in the returned object, i.e.,

```
R> fitLME <- lme(sqrt(CD4) ~ obstime + obstime:drug,
+    random = ~ obstime | patient, data = aids)
R> fitSURV <- coxph(Surv(Time, death) ~ drug, data = aids.id, x = TRUE)
R> fit.JM <- jointModel(fitLME, fitSURV, timeVar = "obstime",
+    method = "piecewise-PH-GH")
R> summary(fit.JM)

Call:
jointModel(lmeObject = fitLME, survObject = fitSURV, timeVar = "obstime",
    method = "piecewise-PH-GH")

Data Descriptives:
Longitudinal Process                    Event Process
Number of Observations: 1405            Number of Events: 188 (40.3%)
Number of Groups: 467

Joint Model Summary:
Longitudinal Process: linear mixed effects model
Event Process: Relative risk model with piecewise-constant baseline risk
    function (knots at: 6.2, 11.1, 12.5, 13.9, 16, 17.8)

   log.Lik      AIC       BIC
 -2107.647 4247.295 4313.636

Variance Components:
            StdDev    Corr
(Intercept) 0.8660  (Intr)
obstime     0.0388  0.0681
Residual    0.3754

Coefficients:
Longitudinal Process
               Value Std.Err z-value p-value
(Intercept)   2.5558  0.0372 68.7962 <0.0001
```

```
obstime          -0.0423  0.0046 -9.1932 <0.0001
obstime:drugddI   0.0051  0.0065  0.7819  0.4343

Event Process
            Value Std.Err z-value p-value
drugddI     0.3510  0.1537  2.2829  0.0224
Assoct     -1.1019  0.1180 -9.3399 <0.0001
log(xi.1) -1.6483  0.2499 -6.5973
log(xi.2) -1.3388  0.2394 -5.5913
log(xi.3) -1.0226  0.2861 -3.5736
log(xi.4) -1.5795  0.3736 -4.2282
log(xi.5) -1.4716  0.3500 -4.2050
log(xi.6) -1.4375  0.4282 -3.3567
log(xi.7) -1.4767  0.5454 -2.7077


Integration:
method: Gauss-Hermite
quadrature points: 15


Optimization:
Convergence: 0
```

The main argument `timeVar` of `jointModel()` is used to specify the name of the time variable in the linear mixed effects model, which is required for the computation of $m_i(t)$. The `summary()` method returns a detailed output, including among others the parameter estimates, their standard errors, and asymptotic Wald tests for both the longitudinal and survival submodels. In the results of the event process the parameter labeled 'Assoct' is in fact parameter α in (1) that measures the effect of $m_i(t)$ (i.e., in our case of the true square root CD4 cell count) in the risk for death. The parameters `xi` are the ξ_q ($q = 1, \ldots, 7$) parameters for the piecewise constant baseline risk function in (15). A comparison between the standard time-dependent Cox model with the joint model reveals some interesting features. In particular, we observe that the regression coefficient for ddI is larger in magnitude in the joint model, which results in a slightly stronger treatment effect. Much stronger bias is observed for the effect of the CD4 cell count, with estimated regression coefficient -0.72 for the time-dependent Cox model and -1.10 for the joint model. As an alternative to the Wald test, a likelihood ratio test (LRT) can be also used to test for a treatment effect. To perform this test we need to fit the joint model under the null hypothesis of no treatment effect in the survival submodel, and then use the `anova()` method:

```
R> fitSURV2 <- coxph(Surv(Time, death) ~ 1, data = aids.id, x = TRUE)
R> fit.JM2 <- jointModel(fitLME, fitSURV2, timeVar = "obstime",
+    method = "piecewise-PH-GH")
R> anova(fit.JM2, fit.JM)


            AIC     BIC  log.Lik  LRT df p.value
fit.JM2 4250.53 4312.72 -2110.26
fit.JM  4247.29 4313.64 -2107.65 5.23  1  0.0222
```
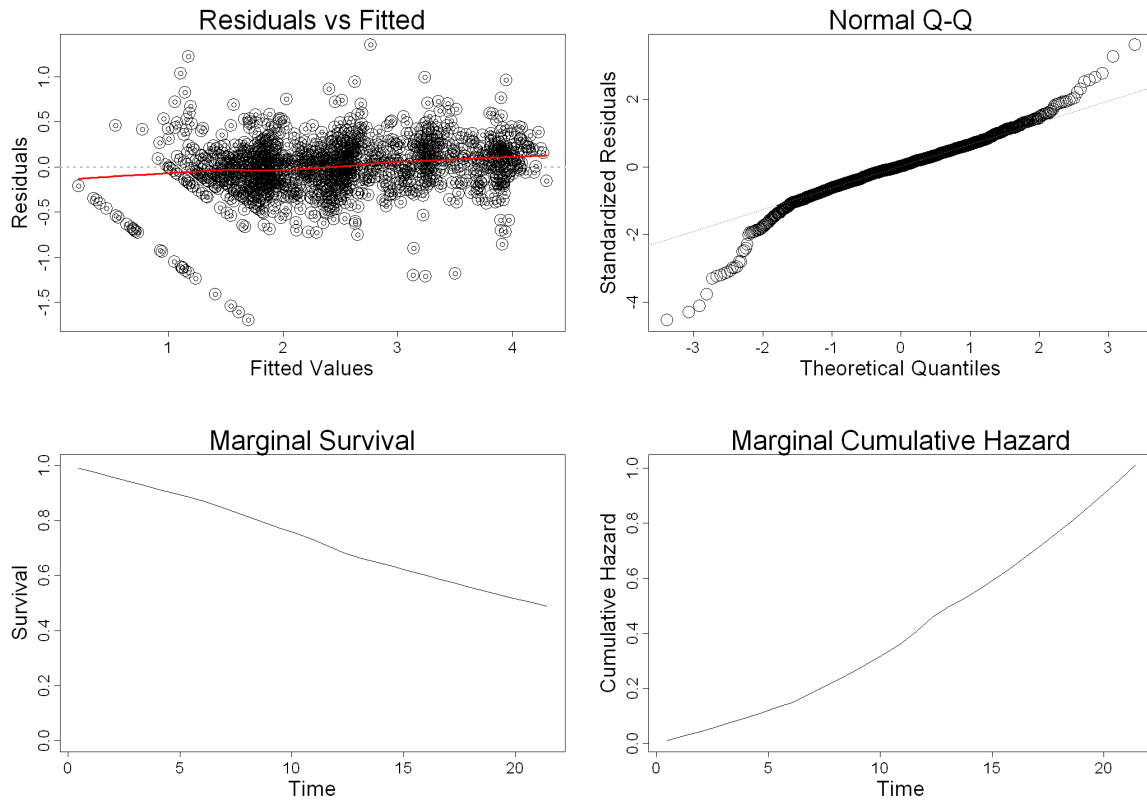
Figure 3: Diagnostic plots for the fitted joint model. The top-left panel depicts the subject-specific residuals for the longitudinal process versus their corresponding fitted values. The top-right panel depicts the normal Q-Q plot of the standardized subject-specific residuals for the longitudinal process. The bottom-left depicts an estimate of the marginal survival function for the event process. The bottom-right depicts an estimate of the marginal cumulative risk function for the event process.

We arrive at the same conclusion with an almost identical p value to the Wald test. The `anova()` method for class `jointModel` accepts as arguments two fitted joint models, with the first one always being the model under the null.

We proceed by checking the fit of the model using residuals plots. As it is standard in R, the residuals and fitted values can be extracted from a fitted model using the generic functions `residuals()` and `fitted()`, respectively. For joint models fitted by `jointModel()` we have several kinds of residuals and fitted values depending on the outcome (i.e., longitudinal or time-to-event) and the level of focus (i.e., marginalized over the subjects or subject-specific) – for more details we refer to Appendix B. Some model diagnostics are directly available by calling the `plot()` method for `jointModel` objects – for our fitted joint model this is illustrated in Figure 3.

```
R> par(mfrow = c(2, 2))
R> plot(fit.JM)
```

The estimated marginal survival function presented in bottom-left panel of Figure 3 is calcu-

lated according to the formula

$$\mathcal{S}(t) = \int \mathcal{S}_i(t \mid b_i; \hat{\theta}) \, p(b_i; \hat{\theta}) \, db_i \approx n^{-1} \sum_i \mathcal{S}_i(t \mid \hat{b}_i; \hat{\theta}),$$

where $\hat{b}_i$ denotes the empirical Bayes estimates for the random effects. The estimated marginal cumulative risk function is simply calculated as $H_i(t) = -\log \mathcal{S}(t)$. Additional residuals plots can be easily computed by first calculating the specific type of residuals of interest, and then plotting them against the corresponding fitted values or covariates. For example, the code below produces Figure 4 that contains the plots of the standardized subject-specific and standardized marginal residuals versus fitted values for the longitudinal process, and the martingale residuals versus $\hat{m}_i(T_i)$ and the Kaplan-Meier estimate of the Cox-Snell residuals for the event process (function `plotResid()` used to produce the following residuals plots can be found in Appendix C).

```
R> par(mfrow = c(2, 2))
R> resSubY <- residuals(fit.JM, process = "Longitudinal",
+    type = "stand-Subject")
R> fitSubY <- fitted(fit.JM, process = "Longitudinal", type = "Subject")

R> plotResid(fitSubY, resSubY, xlab = "Fitted Values", ylab = "Residuals",
+    main = "Subject-Specific Residuals vs Fitted Values")

R> resMargY <- residuals(fit.JM, process = "Longitudinal",
+    type = "stand-Marginal")
R> fitMargY <- fitted(fit.JM, process = "Longitudinal", type = "Marginal")
R> plotResid(fitMargY, resMargY, xlab = "Fitted Values", ylab = "Residuals",
+    main = "Marginal Residuals vs Fitted Values")

R> resMartT <- residuals(fit.JM, process = "Event", type = "Martingale")
R> fitSubY <- fitted(fit.JM, process = "Longitudinal", type = "EventTime")
R> plotResid(fitSubY, resMartT, xlab = "Fitted Values", ylab = "Residuals",
+    main = "Martingale Residuals vs Fitted Values")

R> resCST <- residuals(fit.JM, process = "Event", type = "CoxSnell")
R> sfit <- survfit(Surv(resCST, death) ~ 1, data = aids.id)

R> plot(sfit, mark.time = FALSE, conf.int = TRUE, lty = 1:2,
+    xlab = "Cox-Snell Residuals", ylab = "Survival Probability",
+    main = "Survival Function of Cox-Snell Residuals")
R> curve(exp(-x), from = 0, to = max(aids.id$Time), add = TRUE,
+    col = "red", lwd = 2)
```

We observe that the fitted loess curve in the plot of the standardized marginal residuals versus the fitted values shows a systematic trend with more positive residuals for small fitted values. However, as mentioned in Section 2.3, due to the nonrandom dropout in the longitudinal outcome caused by the occurrence of events, conclusions from residuals based on the observed data alone should be extracted with caution. For instance, in our example small numbers of
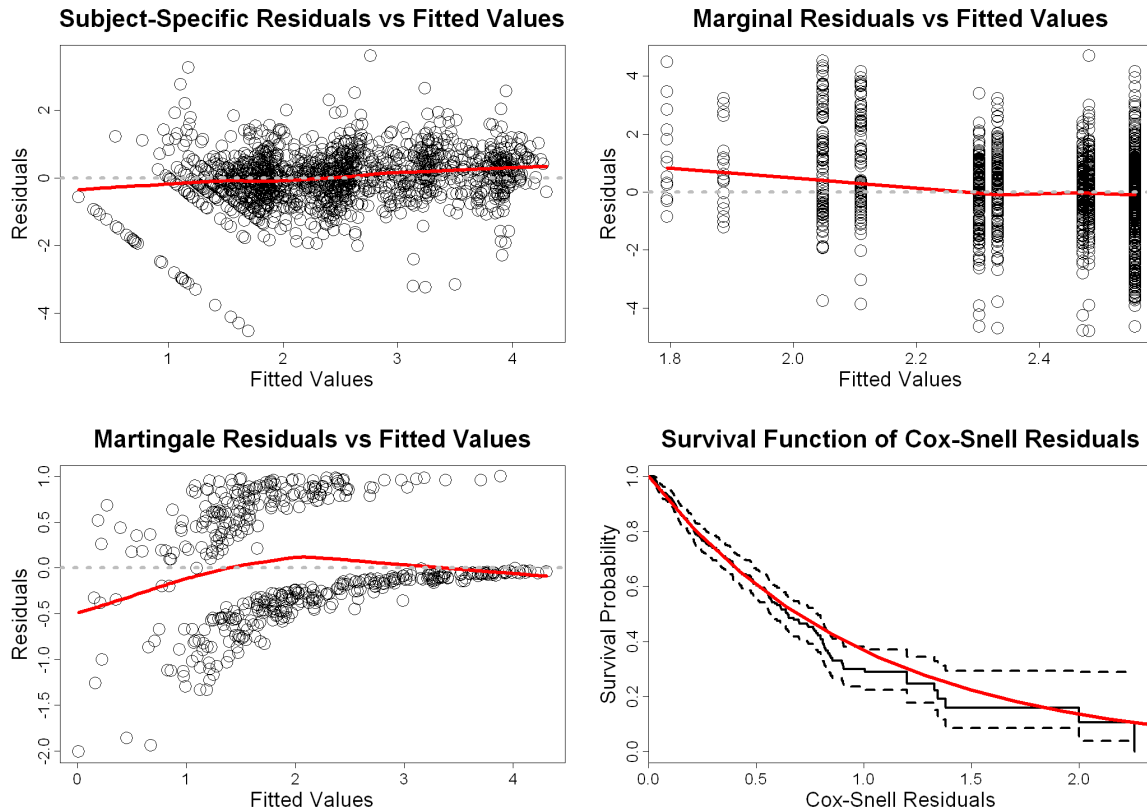
Figure 4: Diagnostic plots for the fitted joint model. The dashed lines in bottom-right panel denote the 95% confidence intervals for the Kaplan-Meier estimate of the Cox-Snell residuals.

CD4 cell counts indicate a worsening of patients' condition resulting in higher death rates (i.e., dropout). Thus, the residuals corresponding to small fitted values are only based on patients with a 'good' health condition, which results in the systematic trend. To take dropout into account we will use the multiply-imputed residuals introduced in Section 2.3. The simulation scheme described in Section 2.3 is available via the `residuals()` method for `jointModel` objects, and can be invoked using the logical argument `MI`. As an illustration, we calculate the multiply-imputed standardized marginal residuals (we set the seed for reproducibility):

```
R> set.seed(123)
R> res.MI <- residuals(fit.JM, process = "Longitudinal",
+    type = "stand-Marginal", MI = TRUE)
R> fitMargY.miss <- res.MI$fitted.valsM
R> resMargY.miss <- res.MI$resid.valsM
```

Contrary to the standard call to `residuals()` that returns a numeric vector of residuals (as illustrated above), setting `MI` to `TRUE` returns a list with several components useful in the further processing of the multiply-imputed residuals. The two components that we extract here are the fitted values and the multiply-imputed standardized marginal residuals that correspond to y_i^m, as defined in Section 2.3. Object `fitMargY.miss` is a numeric vector, whereas `resMargY.miss` is a numeric matrix with columns representing the realizations of
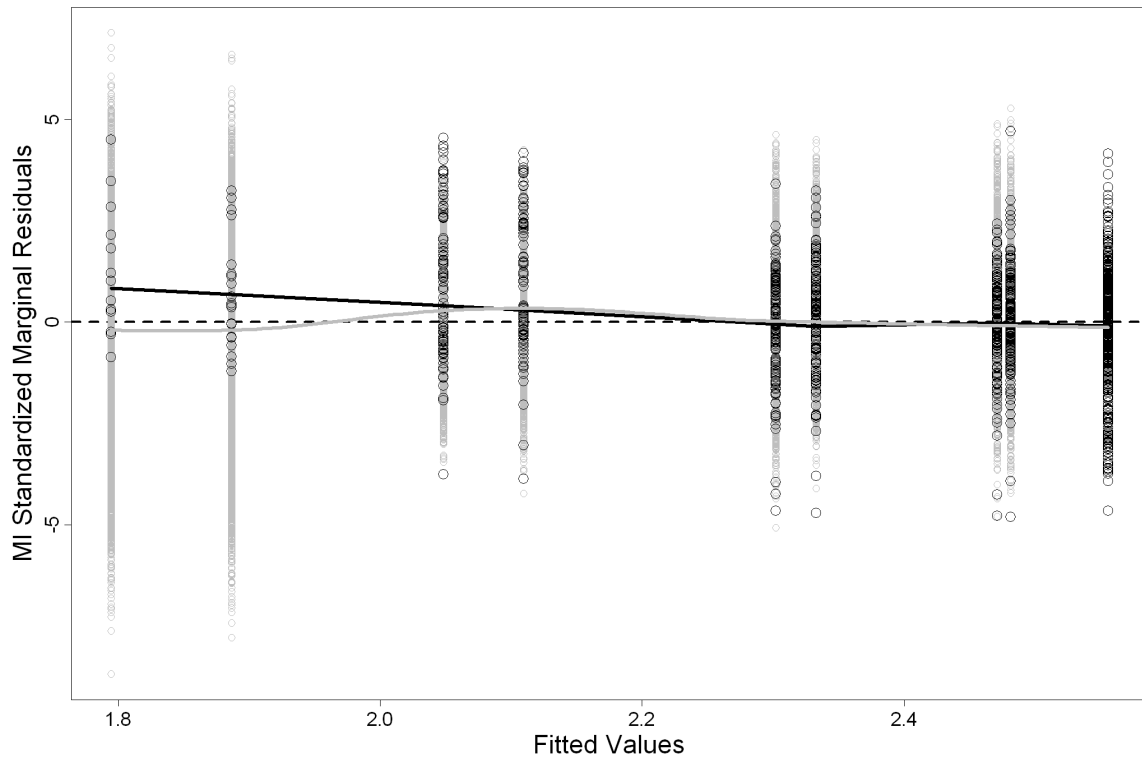
Figure 5: Observed standardized marginal residuals (black circles), augmented with all the multiply imputed residuals produced by the $L = 50$ imputations (grey points). The superimposed solid lines represent a loess fit based only on the observed residuals (black line), and a weighted loess fit based on all residuals (grey line).

the residuals based on the multiple-imputations for y_i^m (default is 50 multiple-imputations; this is controlled by argument M of `residuals()`). The following code produces Figure 5 that depicts the multiply-imputed residuals together with the observed residuals versus their corresponding fitted values:

```
R> M <- ncol(resMargY.miss)
R> resMargY.MI <- c(resMargY, resMargY.miss)
R> fitMargY.MI <- c(fitMargY, rep(fitMargY.miss, M))
R> plot(range(fitMargY.MI), range(resMargY.MI), type = "n",
+    xlab = "Fitted Values", ylab = "MI Standardized Marginal Residuals")
R> abline(h = 0, lty = 2)
R> points(rep(fitMargY.miss, M), resMargY.miss, cex = 0.5, col = "grey")
R> points(fitMargY, resMargY)
```

The black loess curve in Figure 5 is in fact the same curve as in the top-right panel of Figure 4 and is produced by

```
R> lines(lowess(fitMargY, resMargY), lwd = 2)
```

However, to produce the grey loess curve, which describes the relationship between the complete residuals (i.e., the multiply-imputed residuals together with the observed residuals) versus their corresponding fitted values, some extra steps are required. In particular, we need to take into account that for each of the time points the i-th subject did not appear in the study center we have M = 50 multiply-imputed residuals, whereas for the times that he did appear we only have one. Thus, in the calculation of the loess curve we will use case weights with the value 1 for the observed residuals and $1/M = 1/50 = 0.02$ for the multiply-imputed ones, i.e.,

```
R> dat.resid <- data.frame(
+    resid = resMargY.MI,
+    fitted = fitMargY.MI,
+    weight = c(rep(1, length(resMargY)), rep(1/M, length(resMargY.miss))))
+ )
R> fit.loess <- loess(resid ~ fitted, data = dat.resid, weights = weight)
R> nd <- data.frame(fitted = seq(min(fitMargY.MI), max(fitMargY.MI),
+    length.out = 100))
R> prd.loess <- predict(fit.loess, nd)
R> lines(nd$fitted, prd.loess, col = "grey", lwd = 2)
```

A comparison between the two loess smoothers reveals that indeed the systematic trends that were present in the residual plots based on the observed data alone are mainly attributed to the nonrandom dropout and not to a model lack-of-fit.

Following we will focus on the calculation of expected survival probabilities. The Monte Carlo scheme of Section 2.4 is implemented in function `survfitJM()` that accepts as main arguments a fitted joint model, and a data frame that contains the longitudinal and covariate information for the subjects for which we wish to calculate the predicted survival probabilities. Here we compute $\pi_i(u \mid t)$ for four patients in the data set who have not died by the time of loss to follow-up, using $L = 200$ Monte Carlo samples (as previously, we set the seed for reproducibility):

```
R> set.seed(123)
R> ND <- aids[aids$patient %in% c("7", "15", "117", "303"), ]
R> predSurv <- survfitJM(fit.JM, newdata = ND, idVar = "patient",
+    last.time = "Time")
R> predSurv

Prediction of Conditional Probabilities for Event based on 200 replications

$`7`
    times   Mean Median  Lower  Upper
1  14.6962 0.9863 0.9879 0.9696 0.9945
2  15.3147 0.9629 0.9673 0.9177 0.9853
3  15.9332 0.9393 0.9465 0.8655 0.9761
4  16.5518 0.9136 0.9209 0.8110 0.9651
5  17.1703 0.8877 0.8952 0.7628 0.9516
6  17.7888 0.8617 0.8715 0.7025 0.9421
```

```
7  18.4074 0.8359 0.8523 0.6502 0.9308
8  19.0259 0.8103 0.8302 0.6115 0.9245
9  19.6444 0.7849 0.8091 0.5475 0.9167
10 20.2629 0.7597 0.7897 0.4852 0.9107
11 20.8815 0.7349 0.7643 0.4159 0.8978
12 21.5000 0.7105 0.7425 0.3741 0.8905
...
```

By default `survfitJM()` computes $\pi_i(u \mid t)$ at a set of equally spaced time points, produced by a regular sequence of length 35 starting from the minimum observed event time to the maximum observed event time (i.e., `seq(min(Time), max(Time) + 0.1, length.out = 35)`); this default choice can be changed by specifying in argument `survTimes` a numeric vector of time points at which $\pi_i(u \mid t)$ is to be computed. We should note however that `survfitJM()` actually computes $\pi_i(u \mid t)$ for each subject for the time points that are later than the time of the last available longitudinal measurement (e.g., if a subject had provided longitudinal measurements up to time t, `survfitJM()` will compute $\pi_i(u \mid t)$ for all $u > t$). This is because for the time points that are earlier than the time of the last available longitudinal measurement we know that this subject was alive and therefore $\pi_i(u \mid t) = 1$ (see also Section 2.4). In addition, if the last time point at which a subject was still alive is later than the time point of his last available longitudinal measurement, then this can be specified via the optional argument `last.time`; for instance, in our example we specify `last.time = "Time"` since we know that all four patients were still alive at the censoring time, which is given by the variable `Time` in data frame `ND`. The printed output `survfitJM()` is rather self-explanatory – to avoid including here the whole lengthy output for all four patients, we only show the results for Patient 7. This patient provided CD4 cell count measurements up to 12 months from randomization and he was lost to follow-up after 14.3 months. He has 75% probability of surviving more than 21.4 months (which is the largest recorded follow-up time) with a 95% pointwise confidence interval ranging from 38% to 89%. The estimated $\pi_i(u \mid t)$'s can also be plotted using the `plot()` method for objects of class `survfitJM`. In particular, a simple call to `plot()` produces Figure 6:

```
R> par(mfrow = c(2, 2))
R> plot(predSurv, conf.int = TRUE)
```

Furthermore, the `fun` argument of the `plot()` method can be used to produce various transformations of $\pi_i(u \mid t)$. For instance, the following code includes in Figure 7 the plots of expected survival, natural logarithm of expected survival, and expected cumulative risk

```
R> par(mfrow = c(2, 2))
R> plot(predSurv, which = "7", conf.int = TRUE)
R> plot(predSurv, which = "7", conf.int = TRUE, fun = log,
+    ylab = "log Survival")
R> plot(predSurv, which = "7", conf.int = TRUE, fun = function (x) -log(x),
+    ylab = "Cumulative Risk")
```

An additional option provided by the `plot()` method for `survfitJM` objects, is to include in the same figure the estimated $\pi_i(u \mid t)$ and the longitudinal information for each patient. This is illustrated for Patient 7 in Figure 8 that is produced with following piece of code:
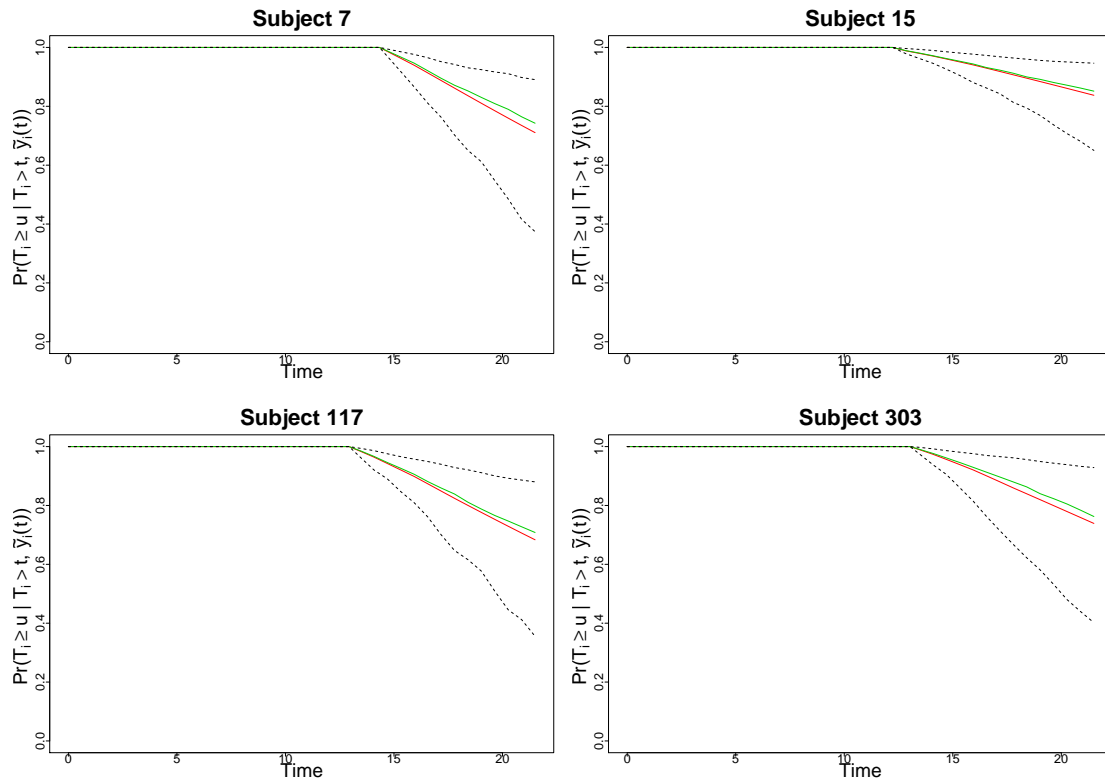
Figure 6: Predicted probabilities of survival for four patients who have not died by the end of the study, based on 200 Monte Carlo samples. The red and greed solid lines depict the mean and median, respectively, of $\pi_i(u \mid t)$ over the Monte Carlo samples. The black dashed lines depict a 95% pointwise confidence intervals based on the quantiles of $\pi_i(u \mid t)$ over the Monte Carlo samples.

```
R> plot(predSurv, conf.int = TRUE, estimator = "median", which = "7",
+    include.y = TRUE)
```

The `estimator` argument specifies which estimate of $\pi_i(u \mid t)$ to plot (i.e., mean, median or both), argument `which` can be used to specify for which specific subject we want to produce the plot, and logical argument `include.y` adds in the same figure the scatter plot of the longitudinal responses of this subject versus time. This figure is useful in investigating whether specific longitudinal profiles result in steeper survival curves, which could be insightful for the mechanisms that underly the medical condition under study.

Finally, to make the connection with the missing data framework we can compare the results of the linear mixed effects model ignoring the dropout process with the results of the longitudinal submodel from the joint model fit:

```
R> summary(fitLME)

Linear mixed-effects model fit by REML
 Data: aids
```
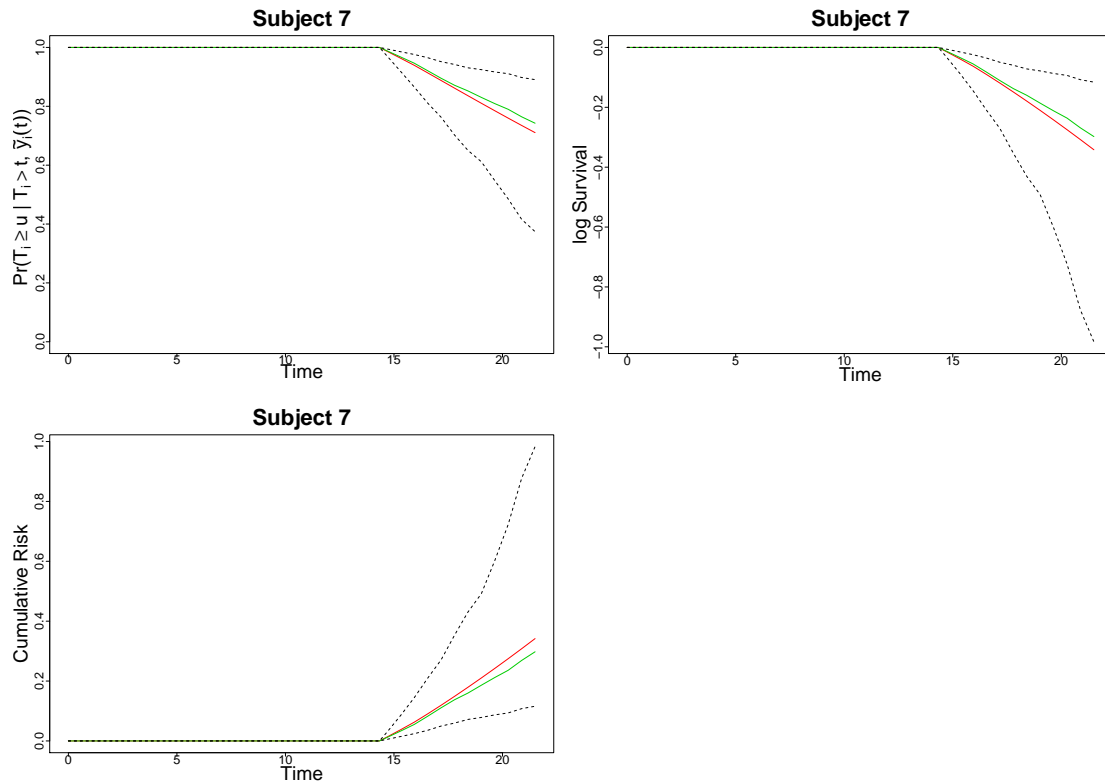
Figure 7: Transformations of the predicted survival probabilities for death for Patient 7, based on 200 Monte Carlo samples. The green solid line depicts the median of $\pi_i(u \mid t)$ over the Monte Carlo samples. The black dashed lines depict a 95% pointwise confidence interval.
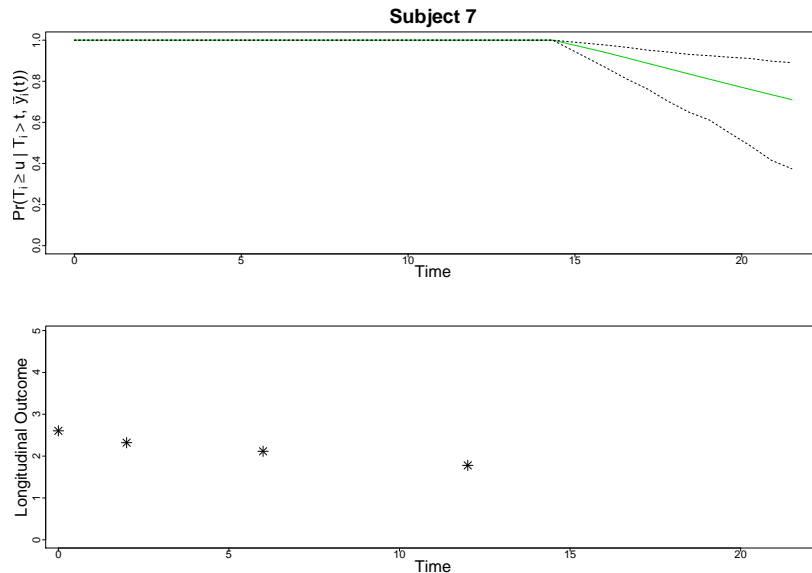
Figure 8: Predicted survival probabilities for Patient 7 (top panel) including also his available longitudinal $\sqrt{CD4}$ cell counts measurements (bottom panel), based on 200 Monte Carlo samples. The green solid line depicts the median of $\pi_i(u \mid t)$ over the Monte Carlo samples. The black dashed lines depict a 95% pointwise confidence interval.

```
       AIC      BIC    logLik
 2699.069 2735.789 -1342.535

Random effects:
 Formula: ~obstime | patient
 Structure: General positive-definite, Log-Cholesky parametrization
            StdDev     Corr
(Intercept) 0.87143264 (Intr)
obstime     0.03617033 -0.015
Residual    0.36844785



Fixed effects: sqrt(CD4) ~ obstime + obstime:drug
                    Value  Std.Error  DF  t-value p-value
(Intercept)      2.5118005 0.04258901 936 58.97766  0.0000
obstime         -0.0375070 0.00440225 936 -8.51997  0.0000
obstime:drugddI  0.0082141 0.00632277 936  1.29912  0.1942

Correlation:
               (Intr) obstim
obstime         -0.118
obstime:drugddI  0.000 -0.687
...
```

We observe a rather very small sensitivity, which is more easily noticeable by comparing the parameter estimates divided by their corresponding standard errors (i.e., `z-value` column in the results of the joint model fit and `t-value` column in the results of the linear mixed model fit). However, note that we have relaxed the MAR assumption only towards the MNAR mechanism implied by joint models (9) – we cannot definitely claim that there are no sensitivity issues.

5. Extra features and extensions

In this article we have illustrated the capabilities of package **JM** for the joint modelling of longitudinal and time-to-event data using shared parameter models. These models are applicable when either we wish to account for the effect of a time-dependent covariate measured with error in a survival analysis context or when we wish to correct for nonrandom dropout in the analysis of longitudinal outcomes.

Even though in Section 4 we have focused on joint models with a relative risk submodel with a piecewise constant baseline risk function for the event outcome, **JM** offers several other options for the survival submodel as described in Section 3.1. Moreover, for `method = "weibull-PH-GH"` and `method = "weibull-AFT-GH"` the `scaleWB` argument of `jointModel()` can be used to fix the scale parameter of the Weibull hazard function to a specific value (e.g., setting `scaleWB = 1`, the Weibull hazard reduces to the exponential hazard). In addition, for `method = "spline-PH-GH"`, it is also possible to include stratification factors, i.e., different coefficients for the B-spline approximation of the log baseline risk function are assumed for

different levels of categorical variable. To fit such a stratified joint model the user needs to include the stratification factors in the definition of the survival model, which must only be a Cox model. For instance, for the AIDS data set analyzed in Section 4, the following code includes as a stratification factor variable that distinguishes between patients who were intolerant or had failure in the standard treatment for AIDS:

```
R> fitSURV3 <- coxph(Surv(Time, death) ~ drug + strata(AZT), data = aids.id,
+    x = TRUE)
R> fit.JM3 <- jointModel(fitLME, fitSURV3, timeVar = "obstime",
+    method = "spline-PH-GH")
```

The supporting function `wald.strata()` can be used to test for equality of the spline coefficients across strata with a Wald test. Furthermore, for all types of joint models available in **JM** it is possible to postulate a lag effect for the longitudinal time-dependent covariate, that is

$$h_i(t \mid \mathcal{M}_i(t), w_i) = h_0(t) \exp[\gamma^\top w_i + \alpha m_i \{\min(t - k, 0)\}],$$

where $k > 0$. This is controlled by the argument `lag` of `jointModel()` that defaults to 0. Finally, a further issue in the calculation of the multiple-imputation residuals is the nature of the visiting process (i.e., the stochastic mechanism that generates the time points at which the longitudinal measurements are collected). In particular, in the AIDS study patients were supposed to provide CD4 cell count measurements at fixed follow-up times. However, in observational studies and in some randomized trials, the time points at which the longitudinal measurements are taken are not fixed by design but rather determined by the physician or even the patients themselves. The possibility of random visit times complicates the methodology presented in Section 2.3 due to the fact that the time points at which the i-th subject was supposed to provide measurements after the observed event time T_i are not available, and thus the corresponding rows $x_i^\top(t_{ij})$ and $z_i^\top(t_{ij})$, for $t_{ij} \geq T_i$, of the design matrices X_i and Z_i, respectively, required in Step 3 of the simulation scheme of Section 2.3, cannot be specified. To overcome this problem **JM** offers the option to model the visiting process (using a Weibull model with a Gamma frailty implemented in function `weibull.frailty()`), and use this visiting model to simulate future visit times for each individual. An example of this procedure as well as up-to-date information regarding the current and future features of the package, and R scripts, with detailed analysis of real data sets illustrating these features, can be found in the R wiki page of **JM**, accessible at http://rwiki.sciviews.org/doku.php?id=packages: cran:jm.

Package **JM** is still under active development. Future plans include among others, the handling of interval censored and grouped survival data, different types of parameterizations for the survival model, and the implementation of the fully exponential Laplace approximation for the other types of survival models available in `jointModel()`.

References

Allignol A, Latouche A (2010). *CRAN Task View: Survival Analysis*. Version 2010-05-26, URL http://CRAN.R-project.org/view=Survival.

Bates D, Maechler M (2010). ***lme4: Linear Mixed-Effects Models Using S4 Classes.*** R package version 0.999375-34, URL http://CRAN.R-project.org/package=lme4.

Brown E, Ibrahim J (2003). "A Bayesian Semiparametric Joint Hierarchical Model for Longitudinal and Survival Data." *Biometrics*, **59**, 221–228.

Brown E, Ibrahim J, DeGruttola V (2005). "A Flexible B-Spline Model for Multiple Longitudinal Biomarkers and Survival." *Biometrics*, **61**, 64–73.

Chi YY, Ibrahim J (2006). "Joint Models for Multivariate Longitudinal and Multivariate Survival Data." *Biometrics*, **62**, 432–445.

Cox D, Hinkley D (1974). *Theoretical Statistics.* Chapman & Hall, London.

Cox D, Oakes D (1984). *Analysis of Survival Data.* Chapman & Hall, London.

Ding J, Wang JL (2008). "Modeling Longitudinal Data with Nonparametric Multiplicative Random Effects Jointly with Survival Data." *Biometrics*, **64**, 546–556.

Follmann D, Wu M (1995). "An Approximate Generalized Linear Model with Random Effects for Informative Missing Data." *Biometrics*, **51**, 151–168.

Goldman A, Carlin B, Crane L, Launer C, Korvick J, Deyton L, Abrams D (1996). "Response of CD4+ and Clinical Consequences to Treatment Using ddI or ddC in Patients with Advanced HIV Infection." *Journal of Acquired Immune Deficiency Syndromes and Human Retrovirology*, **11**, 161–169.

Guo X, Carlin B (2004). "Separate and Joint Modeling of Longitudinal and Event Time Data using Standard Computer Packages." *The American Statistician*, **58**, 16–24.

Harrell F (2001). *Regression Modeling Strategies: With Applications to Linear Models, Logistic Regression, and Survival Analysis.* Springer-Verlag, New York.

Henderson R, Diggle P, Dobson A (2000). "Joint Modelling of Longitudinal Measurements and Event Time Data." *Biostatistics*, **1**, 465–480.

Hsieh F, Tseng YK, Wang JL (2006). "Joint Modeling of Survival and Longitudinal Data: Likelihood Approach Revisited." *Biometrics*, **62**, 1037–1043.

Kalbfleisch J, Prentice R (2002). *The Statistical Analysis of Failure Time Data.* 2nd edition. John Wiley & Sons, New York.

Lange K (2004). *Optimization.* Springer-Verlag, New York.

Little R, Rubin D (2002). *Statistical Analysis with Missing Data.* 2nd edition. John Wiley & Sons, New York.

Nobre J, Singer J (2007). "Residuals Analysis for Linear Mixed Models." *Biometrical Journal*, **6**, 863–875.

Pinheiro J, Bates D (1995). "Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model." *Journal of Computational and Graphical Statistics*, **4**, 12–35.

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Development Core Team (2009). ***nlme***: *Linear and Nonlinear Mixed Effects Models*. R package version 3.1-96, URL http://CRAN.R-project.org/package=nlme.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Rizopoulos D, Verbeke G, Lesaffre E (2009). "Fully Exponential Laplace Approximations for the Joint Modelling of Survival and Longitudinal Data." *Journal of the Royal Statistical Society B*, **71**, 637–654.

Rizopoulos D, Verbeke G, Molenberghs G (2008). "Shared Parameter Models under Random Effects Misspecification." *Biometrika*, **95**, 63–74.

Rizopoulos D, Verbeke G, Molenberghs G (2010). "Multiple-Imputation-Based Residuals and Diagnostic Plots for Joint Models of Longitudinal and Survival Outcomes." *Biometrics*, **66**, 20–29.

Song X, Davidian M, Tsiatis A (2002). "A Semiparametric Likelihood Approach to Joint Modeling of Longitudinal and Time-to-Event Data." *Biometrics*, **58**, 742–753.

Therneau T, Grambsch P (2000). *Modeling Survival Data: Extending the Cox Model*. Springer-Verlag, New York.

Therneau T, Lumley T (2009). ***survival***: *Survival Analysis Including Penalised Likelihood*. R package version 2.35-8, URL http://CRAN.R-project.org/package=survival.

Tseng YK, Hsieh F, Wang JL (2005). "Joint Modelling of Accelerated Failure Time and Longitudinal Data." *Biometrika*, **92**, 587–603.

Tsiatis A, Davidian M (2001). "A Semiparametric Estimator for the Proportional Hazards Model with Longitudinal Covariates Measured with Error." *Biometrika*, **88**, 447–458.

Tsiatis A, Davidian M (2004). "Joint Modeling of Longitudinal and Time-to-Event Data: An Overview." *Statistica Sinica*, **14**, 809–834.

Verbeke G, Molenberghs G (2000). *Linear Mixed Models for Longitudinal Data*. Springer-Verlag, New York.

Wang Y, Taylor J (2001). "Jointly Modeling Longitudinal and Event Time Data with Application to Acquired Immunodeficiency Syndrome." *Journal of the American Statistical Association*, **96**, 895–905.

Wulfsohn M, Tsiatis A (1997). "A Joint Model for Survival and Longitudinal Data Measured with Error." *Biometrics*, **53**, 330–339.

Xu J, Zeger S (2001). "Joint Analysis of Longitudinal Data Comprising Repeated Measures and Times to Events." *Applied Statistics*, **50**, 375–387.

Yu M, Law N, Taylor J, Sandler H (2004). "Joint Longitudinal-Survival-Cure Models and their Application to Prostate Cancer." *Statistica Sinica*, **14**, 835–832.

A. Study of the effect of the number of quadrature points

To investigate the effectiveness of the scaling procedure described in Section 3.2, we fitted three joint models of increasing complexity in the AIDS data set using 9, 11, 15, 21 and 25 Gauss-Hermite quadrature points, respectively (in order to be able to follow the context in which these joint models are specified please read first Section 4 up to p. 15). The specification of these models is as follows:

M1: In the longitudinal submodel we include in the fixed-effects part the main effect of time and the interaction of treatment with time, and in the random-effects part an intercept term. In the survival submodel we include as a time-independent covariate the treatment effect, and as time-dependent one the true underlying effect of CD4 cell count as estimated from the longitudinal model. The baseline risk function is assumed piecewise constant with six knots placed at equally spaced percentiles of the observed event times.

M2: The same specification as M1 but in the random-effects part we also include a random slopes term (this is the joint model presented in Section 4).

M3: The same specification as M2 but in both the longitudinal and survival submodels we include as extra baseline covariates, a dummy for AZT failure and a dummy for previous opportunistic infections. In addition, in the longitudinal submodel we also include the interaction effects of time with AZT failure and time with previous opportunistic infections.

The results, presented in Tables 1–4, have been produced in a Intel Core2Duo 2.50 GHz laptop, with 4GB RAM, running Windows Vista Home Premium, using R version 2.10.1 (2009-12-14) and **JM** (version 0.6-1). First from Table 1, and as expected, we observe that the computing time increases exponentially with respect to both model complexity (i.e., from M1 to M3) and the number of quadrature points. However, from Tables 2–4, we observe that with respect to parameter estimates, the scaling procedure described in Section 3.2 works satisfactorily. In particular, the joint model fits using 15 or even 11 quadrature points yields very similar parameter estimates as the more precise though more computational intensive 21 and 25 quadrature point joint model fits.

	GH_9	GH_{11}	GH_{15}	GH_{21}	GH_{25}
M1	0.26	0.28	0.44	0.42	0.61
M2	1.72	2.57	5.04	11.37	16.69
M3	3.90	4.95	6.56	14.47	23.10

Table 1: Elapsed time in minutes for the three joint models fitted to the AIDS data under different choices for the number of quadrature points for the Gauss-Hermite rule.

	GH$_9$	GH$_{11}$	GH$_{15}$	GH$_{21}$	GH$_{25}$
β_1	2.57	2.55	2.50	2.50	2.49
β_2	−0.04	−0.04	−0.04	−0.04	−0.04
β_5	0.01	0.00	0.00	0.01	0.00
$\log(\sigma)$	−0.81	−0.83	−0.86	−0.88	−0.89
β_1	2.61	2.56	2.56	2.50	2.50
β_2	−0.04	−0.04	−0.04	−0.04	−0.04
β_5	0.01	0.00	0.01	0.01	0.01
$\log(\sigma)$	−0.94	−0.96	−0.98	−1.00	−1.00
β_1	3.28	3.13	3.10	3.12	3.10
β_2	−0.03	−0.03	−0.04	−0.04	−0.04
β_3	−0.23	−0.10	−0.07	−0.09	−0.12
β_4	−1.05	−0.92	−0.83	−0.80	−0.80
β_5	0.00	0.00	0.01	0.01	0.01
β_6	−0.00	−0.00	0.00	0.00	−0.00
β_7	−0.01	−0.01	−0.01	−0.01	−0.01
$\log(\sigma)$	−0.94	−0.96	−0.99	−1.00	−1.00

Table 2: Parameter estimates for the longitudinal process for the three joint models fitted to the AIDS data under different choices for the number of quadrature points for the Gauss-Hermite rule. The top panel presents the results for the random intercepts joint model M1, the middle panel the results for the random intercepts and random slopes joint model M2, and the bottom panel the results for the random intercepts, random slopes + extra covariates joint model M3. β_1 denotes the intercept, β_2 the time effect, β_3 the effect of AZT failure, β_4 the effect of previous opportunistic infections, β_5 the interaction effect of time with treatment, β_6 the interaction effect of time with AZT, β_7 the interaction effect of time with previous opportunistic infections, and σ the standard deviation of the error terms.

	GH$_9$	GH$_{11}$	GH$_{15}$	GH$_{21}$	GH$_{25}$
D_{11}	0.795	0.782	0.774	0.776	0.773
D_{11}	0.780	0.783	0.750	0.765	0.758
D_{12}	0.000	0.001	0.002	0.002	0.003
D_{22}	0.002	0.001	0.002	0.001	0.001
D_{11}	0.618	0.602	0.586	0.572	0.578
D_{12}	−0.001	−0.001	−0.001	0.000	−0.000
D_{22}	0.001	0.001	0.001	0.001	0.001

Table 3: Parameter estimates for the random-effects covariance matrix for the three joint models fitted to the AIDS data under different choices for the number of quadrature points for the Gauss-Hermite rule. The top panel presents the results for the random intercepts joint model M1, the middle panel the results for the random intercepts and random slopes joint model M2, and the bottom panel the results for the random intercepts, random slopes + extra covariates joint model M3. D_{11} denotes the variance for the random intercepts, D_{22} the variance for the random slopes, and D_{12} their covariance.

	GH_9	GH_{11}	GH_{15}	GH_{21}	GH_{25}
γ_1	0.36	0.34	0.35	0.35	0.35
α	-1.08	-1.08	-1.09	-1.07	-1.08
$\log(\xi_1)$	-1.68	-1.67	-1.64	-1.68	-1.67
$\log(\xi_2)$	-1.31	-1.29	-1.27	-1.31	-1.29
$\log(\xi_3)$	-0.96	-0.94	-0.92	-0.96	-0.94
$\log(\xi_4)$	-1.50	-1.48	-1.45	-1.49	-1.47
$\log(\xi_5)$	-1.38	-1.36	-1.34	-1.38	-1.36
$\log(\xi_6)$	-1.39	-1.36	-1.33	-1.37	-1.35
$\log(\xi_7)$	-1.33	-1.31	-1.31	-1.35	-1.34
γ_1	0.36	0.35	0.35	0.34	0.35
α	-1.10	-1.10	-1.10	-1.09	-1.09
$\log(\xi_1)$	-1.66	-1.67	-1.65	-1.67	-1.67
$\log(\xi_2)$	-1.35	-1.35	-1.34	-1.35	-1.36
$\log(\xi_3)$	-1.03	-1.03	-1.02	-1.04	-1.04
$\log(\xi_4)$	-1.59	-1.58	-1.58	-1.59	-1.60
$\log(\xi_5)$	-1.48	-1.47	-1.47	-1.49	-1.49
$\log(\xi_6)$	-1.44	-1.44	-1.44	-1.45	-1.45
$\log(\xi_7)$	-1.47	-1.47	-1.48	-1.48	-1.49
γ_1	0.34	0.33	0.34	0.34	0.34
γ_2	0.09	0.08	0.09	0.09	0.09
γ_3	0.62	0.64	0.63	0.63	0.62
α	-0.99	-0.98	-0.98	-0.98	-0.99
$\log(\xi_1)$	-2.41	-2.44	-2.44	-2.42	-2.41
$\log(\xi_2)$	-2.06	-2.08	-2.09	-2.08	-2.06
$\log(\xi_3)$	-1.71	-1.74	-1.74	-1.73	-1.72
$\log(\xi_4)$	-2.25	-2.27	-2.28	-2.27	-2.26
$\log(\xi_5)$	-2.12	-2.15	-2.16	-2.15	-2.14
$\log(\xi_6)$	-2.06	-2.08	-2.10	-2.09	-2.08
$\log(\xi_7)$	-2.06	-2.07	-2.09	-2.09	-2.08

Table 4: Parameter estimates for the event process for the three joint models fitted to the AIDS data under different choices for the number of quadrature points for the Gauss-Hermite rule. The top panel presents the results for the random intercepts joint model M1, the middle panel the results for the random intercepts and random slopes joint model M2, and the bottom panel the results for the random intercepts, random slopes + extra covariates joint model M3. γ_1 denotes the treatment effect, γ_2 the effect of AZT failure, γ_3 the effect of previous opportunistic infections, α the association parameter, and $\xi_1, \ldots, \xi_7$ the parameters of the baseline risk function.

The code used to produce these results is copied below:

```
R> fitLME.int <- lme(sqrt(CD4) ~ obstime + obstime:drug,
+     random = ~ 1 | patient, data = aids)
```

```
R> fitLME.slp <- lme(sqrt(CD4) ~ obstime + obstime:drug,
+     random = ~ obstime | patient, data = aids)
```

```
R> fitLME.covs <- lme(sqrt(CD4) ~ obstime * (drug + AZT + prevOI) - drug,
+    random = ~ obstime | patient, data = aids)

R> fitSURV <- coxph(Surv(Time, death) ~ drug, data = aids.id, x = TRUE)

R> fitSURV.covs <- coxph(Surv(Time, death) ~ drug + AZT + prevOI,
+    data = aids.id, x = TRUE)

R> GHk <- c(9, 11, 15, 21, 25)
R> jmI <- jmII <- jmIII <- vector("list", length(GHk))
R> for (i in seq_along(GHk)) {
+    jmI[[i]] <- jointModel(fitLME.int, fitSURV, timeVar = "obstime",
+      method = "piecewise-PH-GH", GHk = GHk[i])
+    jmII[[i]] <- jointModel(fitLME.slp, fitSURV, timeVar = "obstime",
+      method = "piecewise-PH-GH", GHk = GHk[i])
+    jmIII[[i]] <- jointModel(fitLME.covs, fitSURV.covs, timeVar = "obstime",
+      method = "piecewise-PH-GH", GHk = GHk[i])
+ }
```

B. Types of residuals for joint models

For the longitudinal part of the joint model two frequently used types of residuals are the standardized marginal and standardized subject-specific residuals, which are defined as

$$r_i^{(ym)} = \hat{V}_i^{-1/2}(y_i - X_i\hat{\beta}), \text{ and}$$
$$r_i^{(ys)}(t_{ij}) = \{y_i(t_{ij}) - x_i^\top(t_{ij})\hat{\beta} - z_i^\top(t_{ij})\hat{b}_i\}/\hat{\sigma},$$

where $\hat{\beta}$, $\hat{\sigma}$, and $\hat{D}$ denote the maximum likelihood estimates under model (3), $\hat{b}_i$ are the empirical Bayes estimates for the random effects, and $\hat{V}_i = Z_i\hat{D}Z_i^\top + \hat{\sigma}^2\mathrm{I}$, with I denoting the identity matrix of appropriate dimensions. The marginal residuals $r_i^{(ym)}$ predict the marginal errors $y_i - X_i\beta = Z_ib_i + \varepsilon_{yi}$, and can be used to investigate misspecification of the mean structure $X_i\beta$ as well as to validate the assumptions for the within-subjects covariance structure V_i. The subject-specific residuals $r_i^{(ys)}(t_{ij})$ predict the conditional errors $\varepsilon_i(t)$, and can be used for checking the homoscedasticity and normality assumptions.

For the survival part of the joint model, a standard type of residuals is the martingale residuals defined as

$$r_i^{(tm)} = \delta_i - \int_0^{T_i} h_i(s \mid \hat{\mathcal{M}}_i(s); \hat{\theta}) \, ds.$$

These are commonly used for a direct assessment of excess events (i.e., to reveal subjects that are poorly fit by the model), and for evaluating whether the appropriate functional form for a covariate is used in the model. Another type of residuals for survival models, related to the martingale residuals, is the Cox-Snell residuals. These are calculated as the value of cumulative risk function evaluated at the observed event times T_i, i.e.,

$$r_i^{(tcs)} = \int_0^{T_i} h_i(s \mid \hat{\mathcal{M}}_i(s); \hat{\theta}) \, ds.$$

If the assumed model fits the data well, we expect $r_i^{(tcs)}$ to have a unit exponential distribution; however, when T_i is censored, $r_i^{(tcs)}$ will be censored as well. To take censoring into account in checking the fit of the model, we can compare graphically the Kaplan-Meier estimate of the survival function of $r_i^{(tcs)}$ with the survival function of the unit exponential distribution.

C. A useful function for residuals plots

```
R> plotResid <- function (x, y, ...) {
+    plot(x, y, ...)
+    lines(lowess(x, y), col = "red", lwd = 2)
+    abline(h = 0, lty = 3, col = "grey", lwd = 2)
+ }
```

Affiliation:

Dimitris Rizopoulos
Department of Biostatistics
Erasmus University Medical Center
PO Box 2040
3000 CA Rotterdam, The Netherlands
E-mail: d.rizopoulos@erasmusmc.nl
URL: http://eur.academia.edu/DimitrisRizopoulos