

# Review and Comparison of Computational Approaches for Joint Longitudinal and Time-to-Event Models

Allison K.C. Furgal<sup>1</sup>, Ananda Sen<sup>1,2</sup>, and Jeremy M.G. Taylor<sup>1</sup>

<sup>1</sup>*Biostatistics Department, School of Public Health, University of Michigan, 1415 Washington Heights, Ann Arbor, MI 48109*

<sup>2</sup>*Department of Family Medicine, Michigan Medicine, University of Michigan, 1018 Fuller St, Ann Arbor, MI 48104*

## Summary

Joint models for longitudinal and time-to-event data are useful in situations where an association exists between a longitudinal marker and an event time. These models are typically complicated due to the presence of shared random effects and multiple submodels. As a consequence, software implementation is warranted that is not prohibitively time consuming. While methodological research in this area continues, several statistical software procedures exist to assist in the fitting of some joint models. We review the available implementation for frequentist and Bayesian models in the statistical programming languages R, SAS, and Stata. A description of each procedure is given including estimation techniques, input and data requirements, available options for customization, and some available extensions, such as competing risks models. The software implementations are compared and contrasted

This is the author's manuscript, accepted for publication and has undergone full peer review but has not been through the copyediting, typesetting, pagination and proofreading process, which may lead to differences between this version and the Version of Record. Please cite this article as doi:10.1111/example

*Key Words:* Joint models, longitudinal data, time-to-event data, survival data, software comparison

## 1 Introduction

Studies often measure both a longitudinal process and an associated event time. A case in point is a study of time to AIDS diagnosis and CD4 counts (Huang et al., 2011; Wang and Taylor, 2001; Wulfsohn and Tsiatis, 1997). In cancer, increasing PSA levels are known to be associated with prostate cancer recurrence (Proust-Lima and Taylor, 2009). Other examples include study of biomarkers in cancer vaccine trials, quality of life measurements in cancer, time-to-failure and degradation in engineering applications (Brown and Ibrahim, 2003; Ibrahim et al., 2010; Lehmann, 2009). In such contexts, separate models for the longitudinal and survival components that do not take into account the dependence between the longitudinal and survival processes produce inefficient results and are prone to bias. Using the longitudinal process as a time-dependent covariate in the survival model requires the unrealistic assumptions that the longitudinal measurement has negligible measurement error and that there are observations at every event time. Traditional Two-Stage models account for the measurement error in the survival model by fitting the longitudinal process first and subsequently using the estimated trajectory as a covariate in the survival model (Tsiatis et al., 1995). However, this strategy fails to account for the possible informative censoring in the longitudinal process (Mccrink et al., 2013). A fully specified joint model incorporates the dependence between the longitudinal and survival components properly and provides efficient inference that is less prone to bias.

Apart from the scenario when the inter-related longitudinal and survival processes are both of interest, joint models are also useful when the event time causes non-informative censoring on the observed longitudinal process that may be of primary interest (Rizopoulos, 2012, p.51,90). While there are differing approaches to formulate joint models, in this review we shall focus on random effects models as the software platform is developed almost exclusively in this framework.

Early work on joint models dates back to the mid-nineties (De Gruttola and Tu, 1994). Since then a body of work has steadily emerged in this area of research (Henderson et al., 2000; Wang and Taylor, 2001; Wulfsohn and Tsiatis, 1997). Some early Bayesian work on this topic includes those by Faucett and Thomas (1996), and Brown and Ibrahim (2003). More recently, there have been several extensions in the directions of dynamic predictions and prognostic tools (Proust-Lima and Taylor, 2009; Taylor et al., 2013; Rizopoulos et al., 2014), competing risks (Elashoff et al., 2007; Hu et al., 2009), recurrent events (Liu and Huang, 2009), multiple longitudinal variables (Li et al., 2007), cure rates (Yu et al., 2004), and diagnostics (Huang et al., 2009). A comprehensive review of early work on models and methods related to joint modeling of longitudinal and survival processes has appeared in Tsiatis and Davidian (2004).

With multiple submodels and shared random effects, joint models can become complicated and difficult to utilize. This motivates the need for creating an efficient computational platform to fit these models. Due to the complexity of joint models, implementation can be slow, but can still be useful for those wanting to fit relatively simple joint models for data analysis. Major statistical softwares such as R, SAS, and Stata include joint modeling functions. The purpose of this article is to provide a comprehensive review of associated implementation issues and explore new applications in related software platforms. There has been limited review of software implementation in the joint modeling literature. Some early reviews listed softwares available at the time (Gould et al., 2014; Mccrink et al., 2013), utilized a single software for data analysis (Rizopoulos, 2012) or compared a couple in the context of analyzing a dataset (Mccrink et al., 2013). Other reviews have compared selected softwares, such as WinBUGS and SAS PROC NLMIXED (Guo and Carlin, 2004). Documentation of the %JM macro in SAS compares the available features of the macro to the JM package in R (Garcia-Hernandez and Rizopoulos, 2015). A recent review of joint modeling literature by Sudell et al. (2016) discussed the frequency with which each joint modeling software was used. More recently, Yuen and Mackinnon (2016) compare some of the available

software with an application to a dataset documenting time to psychosis transition.

There are several ways in which our review differs from those in the existing literature. Most of the reviews focus primarily on the survival component of the joint analysis. By contrast, we provide findings from simulation that assess simultaneous performance of the time-to-event and the longitudinal model. Our documentation is also more comprehensive than most reviews covering specialized topics such as latent classes, competing risks, multiple longitudinal outcomes and more. Through extensive simulation, we present a comprehensive appraisal of the different implementations. In addition, the different softwares are contrasted by means of their performance when used to fit to data from an ongoing trial on adrenal cancer patients enrolled in the University of Michigan Comprehensive Cancer Center.

In Section 2, we will review the joint modeling framework and associated estimation methods along with the computational algorithms. The next section demonstrates the software capabilities with an example data analysis. We will discuss software implementation in Section 4. This includes the `JM` and `joiner` packages in R, the `%JM` macro in SAS, and the `stjm` command in Stata. In Section 5 we describe our simulations. Section 6 outlines implementation of a Bayesian joint model by means of the `JMbayes` package in R. Other models such as competing risks joint models and some specialized software functions are introduced in Section 7 and further described in the Supplemental Materials Section S4. In Section 8 we conclude with a discussion of our findings.

## 2 Modeling Framework

Joint models are useful within a few different scenarios. The main interest can be in the longitudinal process with an event causing informative dropout. Interest could be equally on a longitudinal and a survival process that are associated. A third common scenario occurs when a time-to-event process is modeled with a longitudinal covariate measured intermittently and with error. Such data require joint modeling to fully capture the association and reduce bias.

Consider  $i = 1, \dots, N$  subjects with repeated longitudinal measurements and a terminal

event measured. The repeated measurements and event time are assumed to be associated. Each subject  $i$  has  $n_i$  measurements taken at different times with the intervals between measurements possibly differing. The total number of measurements is  $n = \sum_{i=1}^N n_i$ .

## 2.1 Longitudinal Submodel

Let  $Y_i(t)$  denote the longitudinal process for subject  $i$  at time  $t$ . We assume there is a true underlying process, or trajectory,  $m_i(t)$  from which  $Y_i(t)$  is measured with error  $e_i(t)$ ,  $Y_i(t) = m_i(t) + e_i(t)$ . In practice, we do not observe the longitudinal process at all times. Instead for subject  $i$  we observe  $Y_i(t)$  at  $n_i$  times  $(t_{i1}, \dots, t_{in_i})$ . Let  $Y_i = (Y_{i1}, \dots, Y_{in_i}) = (Y_i(t_{i1}), \dots, Y_i(t_{in_i}))$  be the vector of observations. The trajectory  $m_i(t)$  is modeled with fixed (possibly time-dependent) covariates,  $X_{1ij}$  and parameter coefficients  $\beta$ , and random effects  $b_i$  with possibly time-dependent covariates  $Z_{ij}$ . The longitudinal submodel is shown in (1).

$$\begin{aligned} Y_{ij} &= m_i(t_{ij}) + e_i(t_{ij}) = X_{1ij}\beta + Z_{ij}b_i + e_{ij}, \\ \text{with } e_i &= (e_{i1}, e_{i2}, \dots, e_{in_i}) \sim N(0, \sigma^2 I_{n_i \times n_i}), \\ b_i &\sim N(0, \Sigma_b), \quad i = 1, \dots, N, j = 1, \dots, n_i; \end{aligned} \tag{1}$$

where  $\sigma^2$  and  $\Sigma_b$  are the dispersion parameters for the error and the random effects, respectively.

## 2.2 Time-to-Event Submodel

Let  $T_i^*$  denote the event time for subject  $i$ ,  $C_i$  the censoring time,  $T_i = \min\{T_i^*, C_i\}$  the observed time, and  $\delta_i = 1(T_i^* \leq C_i)$  the event indicator. A common time-to-event submodel is a relative risk model of the form (2).

$$\lambda_i(t|M_i(t)) = \lambda_0(t)\exp\{X_{2i}\gamma + \alpha m_i(t)\} \tag{2}$$

where  $M_i(t)$  is the history of the longitudinal process up to  $t$ ,  $M_i(t) = \{m_i(s), 0 \leq s < t\}$ . The matrix of fixed covariates,  $X_{2i}$ , can include the same variables as  $X_{i1}$  in the longitudinal

submodel but they do not necessarily overlap. The trajectory  $m_i(t)$  from the longitudinal model is included to link the two processes and  $\alpha$  measures the strength of this association. While proportional hazards models are common, accelerated failure time and other survival models have been implemented (Mccrink et al., 2013; Tseng et al., 2005)(Rizopoulos, 2012, p.137).

The model form in (2) will be referred to as the current value form since the association is through the current value of the longitudinal trajectory,  $m_i(t)$ , on the right-hand side of the equation. This form is most often used when the survival time is of interest and the longitudinal process is thought of as a time-dependent covariate measured with error. In (2),  $\lambda_i(t)$  is the event hazard and  $\lambda_0(t)$  is the baseline hazard. In ordinary Cox regression  $\lambda_0(t)$  is left unspecified, avoiding restrictions that come from specifying a parametric form for the baseline hazard (Yuen and Mackinnon, 2016). However if an unspecified hazard is applied in a joint model, it has been shown that standard errors of the parameter estimates can be underestimated (Hseih et al., 2006). This can be remedied by estimating the standard errors with an additional method such as bootstrapping(Rizopoulos, 2010; Yuen and Mackinnon, 2016), [profile likelihood \(Murphy and van der Vaart, 2000\)](#), or [numerically differentiating the EM operator \(Xu et al., 2014\)](#). To avoid this issue, parametric but flexible functions are often used for  $\lambda_0(t)$ , such as piecewise constant or spline models (Rizopoulos, 2012, p.53). These flexible functions can sufficiently approximate the baseline hazard and has been noted as the preferred choice for  $\lambda_0(t)$  by some (Rizopoulos, 2010; Yuen and Mackinnon, 2016)(Rizopoulos, 2012, p.53).

Alternatively, the submodels can be linked through a shared parameters model. This is often used when the longitudinal process is the main interest with informative censoring or when the focus is on both processes equally. Assuming a longitudinal submodel as in (1), a survival submodel as shown in (3) is common

$$\lambda_i(t) = \lambda_0(t)\exp\{X_{i2}\gamma + \alpha Z_{2i}(t)b_i\} \quad (3)$$

where  $b_i$  is the same vector of random effects as in (1) and  $\alpha$  again measures the association. A typical example is a random coefficients model where  $Z_{2i}(t)b_i = b_{0i} + b_{1i}t$  with  $b_{0i}$  and  $b_{1i}$  correlated, often multivariate normal (Wulfsohn and Tsiatis, 1997). The association can be generalized so that  $\alpha Z_{2i}(t)b_i = \alpha_0 b_{0i} + \alpha_1 b_{1i}t$  and  $\alpha_0$  need not equal  $\alpha_1$  (Mccrink et al., 2013).

### 2.3 Estimation

It is commonly assumed that conditionally given the random effects, the longitudinal and time-to-event outcomes are independent and so are the longitudinal measurements taken on a single subject. Denoting the set of all parameters by  $\theta = (\beta, \gamma, \alpha)$ , the log-likelihood contribution from subject  $i$  is

$$l_i(\theta) = \log \int \left( \prod_{j=1}^{n_i} f(y_{ij}|b_i; \theta) \right) \lambda_i(T_i|b_i; \theta)^{\delta_i} S(T_i|b_i; \theta) f(b_i; \theta) db_i,$$

where  $f(\cdot)$  denotes the density function and  $S(\cdot)$  denotes the survival function. There is generally no closed form solution to the likelihood equations. Numerical integration and optimization techniques such as an EM algorithm treating the random effects as missing data, are used in practice. Due to high-dimensional integration and potential correlation induced by the random effects, the process of convergence can be slow. Other methods include Newton-type or hybrid-EM and quasi-Newton algorithms (Rizopoulos, 2012, p.64)(Henderson et al., 2000; Hsieh et al., 2006; Tsiatis and Davidian, 2004; Wulfsohn and Tsiatis, 1997; Yu et al., 2004).

## 3 An Application

We explore the available software platforms through analysis of a dataset of 176 adrenal cancer patients at the University of Michigan Comprehensive Cancer Center diagnosed between 1983 and 2011. Multiple measurements of adrenal tumor characteristics were collected from periodic CT scans. Time until death or last follow-up was also collected. We selected

a single scan measurement as the longitudinal response variable. The longitudinal outcome chosen was psoas density in Hounseld Units (HU) measured using the density of pixels in the scan (Holcombe et al., 2016). A previous study with similar data describes how psoas muscle density and size can be measures of patient frailty, and scan measurements can be associated with survival (Miller et al., 2012). Baseline covariates used were age, cancer stage, and tumor grade. For simplicity we performed a complete case analysis. There were 160 patients with psoas density measurements and all baseline covariates. There are between 1 and 45 scans for each patient with a mean of 5.5 and median of 3 scans. In this group 100 patients died. Time until death or censoring fell between 0.1 and 17.9 years with a median of 2.4 years. Table 1 summarizes the relevant variables. Joint models are fit to the data using three major softwares, namely R, SAS, and Stata. Both the current value (2) and the shared parameter (3) forms are implemented.

First, a current-value joint model was fit to this data using JM in R, SAS, and Stata. The longitudinal submodel included a random intercept and slope in each software.

$$\begin{aligned} \text{PsoasDensity}_{ij} = m_i(t_{ij}) + e_{ij} = & \beta_0 + \beta_1 t_{ij} + \beta_2 \text{Age}_i + \beta_3 \text{Stage}_i \\ & + \beta_4 \text{TumorGradeHigh}_i + \beta_5 \text{TumorGradeUnknown}_i + b_{0i} + b_{1i} t_{ij} + e_{ij} \end{aligned} \quad (4)$$

Survival submodels with flexible baseline hazards were chosen since the true model is unknown. Stata was unable to fit the data with a flexible baseline hazard so a specific parametric (Weibull) hazard was used. Specifically, the softwares fit survival submodels with the form

$$\lambda_i(t) = \lambda_0(t) \exp(\gamma_1 \text{Age}_i + \gamma_2 \text{Stage}_i + \gamma_3 \text{TumorGradeHigh}_i + \gamma_4 \text{TumorGradeUnknown}_i + \alpha m_i(t)) \quad (5)$$



with baseline hazards

R JM PWC:  $\lambda_0(t) \sim$  Piecewise constant function

SAS %JM PWC:  $\lambda_0(t) \sim$  Piecewise constant function

Stata stjw Weib:  $\lambda_0(t) \sim$  Weibull

For SAS and R JM, the program defaults were used to create the flexible baseline hazards. Both procedures by default use six equally spaced internal knots to partition the observed event times. For the SAS model, the random effects are assumed uncorrelated because we found fitting issues with an unstructured matrix (Garcia-Hernandez and Rizopoulos, 2015).

A Two-Stage model was fit for comparison.

$$\left\{ \begin{array}{l} \text{First: } \text{PsoasDensity}_{ij} = \beta_0 + \beta_1 t_{ij} + \beta_2 \text{Age}_i + \beta_3 \text{Stage}_i + \beta_4 \text{TumorGradeHigh}_i \\ \quad + \beta_5 \text{TumorGradeUnknown}_i + b_{0i} + b_{1i} t_{ij} + e_{ij}, \\ \text{Then: } \lambda_i(t) = \lambda_0(t) \exp(\gamma_1 \text{Age}_i + \gamma_2 \text{Stage}_i \\ \quad + \gamma_3 \text{TumorGradeHigh}_i + \gamma_4 \text{TumorGradeUnknown}_i + \alpha \hat{m}_i(t)), \\ \lambda_0(t) \text{ unspecified} \end{array} \right. \quad (6)$$

Bootstrapping was used to estimate the coefficients and bias-corrected bootstrapped confidence intervals (Efron and Tibshirani, 1993), as shown in Table 2.

We also fit a shared parameter model as in (3) with `joiner`, SAS, and Stata. The longitudinal submodel included a random intercept in `joiner` and Stata, while the default in SAS included a random slope. The model fit is

$$\begin{aligned} \text{PsoasDensity}_{ij} &= \beta_0 + \beta_1 t_{ij} + \beta_2 \text{Age}_i + \beta_3 \text{Stage}_i + \beta_4 \text{TumorGradeHigh}_i \\ &\quad + \beta_5 \text{TumorGradeUnknown}_i + b_i + e_{ij} \end{aligned}$$

$$\lambda_i(t) = \lambda_0(t) \exp(\gamma_1 \text{Age}_i + \gamma_2 \text{Stage}_i + \gamma_3 \text{TumorGradeHigh}_i + \gamma_4 \text{TumorGradeUnknown}_i + \alpha b_i)$$

with baseline hazards

R joiner:  $\lambda_0(t)$  Unspecified

SAS %JM PWC:  $\lambda_0(t) \sim$  Piecewise constant function

Stata stjw Weib:  $\lambda_0(t) \sim$  Weibull

The findings are contrasted with a shared parameter two-stage model similar to that in (6) but with  $\alpha\hat{m}_i(t)$  replaced with  $\alpha b_i$ .

Comparing Tables 2 and 3, the results are similar for the current-value and shared parameters models. Nearly all joint models estimate the coefficients of the longitudinal submodel very similarly. Time has a significant negative value showing that psoas density tends to decrease over time, which fits with previous knowledge (Miller et al., 2012). Older patients tend to have less psoas density. Stage 3 or 4 and High Tumor Grade tend to decrease the average psoas density compared to lower stages and grade, but these are not significant in any software or models. There are more differences in the results of the survival submodel since each software fits the survival model differently. The effects are estimated to be larger in the Two-Stage current-value models than in the true joint models. Age does not have much of an effect on death hazard. Most software and model forms find higher stage to slightly increase the death hazard, but significance depends on the software. There is a small and significant positive coefficient for High Tumor Stage. Finally, each software except the current-value Two-Stage model found a small but significant negative association. Log-likelihoods are shown in Table 4 and are similar for all software and both model forms.

## 4 Assessment of Implementation Platform

Fitting a joint model can be computationally intensive. As indicated in the data analysis section, there are a few software packages that are designed to fit such models in an efficient manner. However, the class of such models is quite large and the softwares use different iterative algorithms for estimation and integration. It is thus important to compare

and contrast the implementation platforms to assess and appraise the performance of the competing tools. Such assessment is difficult to make on the basis of fits to few specific datasets. We undertake this evaluation with simulations using various statistical software modules available for fitting joint models. In order to lay the groundwork for the simulation, we first need to briefly describe the available software packages. All packages described in this section implement (1) and either (2) or (3) under a frequentist framework.

#### 4.1 JM Package in R

The JM package in the R language was designed for fitting joint models with the `jointModel()` function. A full description of this package and the `jointModel()` function are available in the CRAN documentation (Rizopoulos, 2016a).

The main arguments for the `jointModel()` function include the output from a linear mixed model (from the R function `lme()` (Pinheiro et al., 2016)) and the output from a Cox proportional hazard model (usually from the R function `coxph()`) (Therneau and Lumley, 2016). Using these separate model fits, `jointModel()` fits a corresponding current-value joint model with submodels having the same covariates and forms as in the separate models and with the additional association added to the survival submodel.

The `jointModel()` function fits three different model forms using the `parameterization` argument. The default is `parameterization = "value"` and fits the current value model in (2). For `parameterization = "slope"`, the survival submodel is linked to the longitudinal process through the slope of the trajectory as in (7).

$$\lambda_i(t|M_i(t)) = \lambda_0(t)\exp\{X_{2i}\gamma + \alpha m'_i(t)\} \quad (7)$$

The final option is `parameterization = "both"` which fits a model with both the current value and the slope of the trajectory in the survival submodel, namely  $\lambda_i(t|M_i(t)) = \lambda_0(t)\exp\{X_{2i}\gamma + \alpha_1 m_i(t) + \alpha_2 m'_i(t)\}$

The `method` argument specifies the form of the baseline hazard, the form of the model,

and the method of numerical integration. The available options are: `weibull-PH-aGH`, `piecewise-PH-aGH`, `spline-PH-aGH`, `weibull-AFT-aGH`, `Cox-PH-aGH`, and `ch-Laplace`. The method `ch-Laplace` uses the fully exponential Laplace approximation described by Rizopoulos et al. (2009). All other options follow a similar format. The first word describes the baseline hazard: `weibull` uses a Weibull baseline hazard, `piecewise` a piece-wise constant baseline hazard, `spline` a B-spline approximation, and `Cox` an unspecified baseline risk. Options with `PH` fit a proportional hazards survival submodel. There is one option for an accelerated failure time model using a Weibull baseline (`weibull-AFT-GH`). Methods ending in `aGH` use pseudo-adaptive Gauss-Hermite quadrature for integral approximation, where the quadrature knots are reassigned once after the first iteration (Rizopoulos, 2010). Each method can instead end in `GH` which uses standard Gauss-Hermite quadrature. Adaptive quadrature is generally preferred due to a reduced computational load using fewer quadrature points while still achieving error on the same order of magnitude as with the standard quadrature (Yuen and Mackinnon, 2016). Quasi-Newton techniques are used if EM iterations do not achieve convergence quickly. Only EM is used with the unspecified baseline hazard. Stratification is allowed only with `method="spline-PH-aGH"` or `method="spline-PH-GH"`.

The `jointModel()` function allows for some extensions such as a competing risks model with the `CompRisk` argument. Other extensions can be formulated with the `interFact` and `derivForm` options as described in (Rizopoulos, 2010) and (Rizopoulos, 2016a).

## 4.2 `joinerR` Package in R

The `joinerR` package was created to analyze longitudinal studies, possibly with an event time causing informative censoring. Full description of this package is in the CRAN documentation (Philipson et al., 2012). The `joint()` function fits a joint model and requires data in a specific format that is the output from the function `jointdata()` in the same package. The user supplies the `joint()` function with the data and a formula object specifying the form for each of the longitudinal and survival submodels. This function fits a shared parameter joint model as in (1) and (3) with an unspecified baseline hazard. This function

is not capable of specifying a parametric form for the baseline hazard. The `model` argument determines the shared random effects. A random slope and intercept model is the default, whereas `model="int"` specifies a random intercept only, and `model="quad"` adds a quadratic time effect to the intercept and slope. The default settings fit a common association when there is more than one random effect terms ( $\alpha Z_{2i}(t)b_i = \alpha b_{0i} + \alpha b_{1i}t$ ). A separate association, as discussed in Section 2.2, can be implemented using the `sepassoc=TRUE` option ( $\alpha Z_{2i}(t)b_i = \alpha_0 b_{0i} + \alpha_1 b_{1i}t$ ,  $\alpha_0 \neq \alpha_1$ ). This function also uses an EM algorithm for estimation with some options available for control of this approximation. Since not specifying a baseline hazard may lead to underestimated standard errors, the `joiner` package includes a separate function to calculate bootstrap standard errors for the joint model (`jointSE()`).

### 4.3 %JM Macro in SAS

The %JM macro in the SAS language was written to fit several joint models in several possible forms. A full description of the macro was written by Garcia-Hernandez and Rizopoulos (2015). The %JM macro allows the longitudinal data to be fit to varying outcome types. The longitudinal data can conform to a Normal, binomial, or Poisson distribution, corresponding to continuous, categorical, or count outcomes, respectively. These can be specified with the `LongiType` option. The longitudinal model can also fit different random effects with the `LongiTimeModel` option, including a linear random intercept and slope model and multiple random splines. There are many options for the baseline hazard of the survival submodel using the `EventModel` option, such as exponential, Weibull, piecewise, and several spline options. The survival submodel can include stratification factors using the `EventStrata` argument. The association can be set with the `SharedParam` argument with the options including current value as in (2), slope (equivalent to our model (3) with  $Z_{2i}(t)b_i = b_{0i} + b_{1i}t$ ), cumulative ( $\lambda_i(t|M_i(t)) = \lambda_0(t)\exp\{X_{2i}\gamma + \alpha \int_0^t m_i(s)ds\}$ ), and “coefficients” in which the user can specify which random effects from the longitudinal submodel should be included in the survival submodel. Multiple associations can be used at the same time. The `SharedCoefficients` and `SharedLongitTerm` arguments can be used to create

even more joint model parameterizations. Estimation is carried out by the PROC NLMIXED procedure which is described in Supplemental Materials Section S4.1.

#### 4.4 `stjm` Command in Stata

In Stata, joint models can be fit with the `stjm` command. A detailed description is given by Crowther et al. (2013). A linear mixed model and a proportional hazards model can be fit as the submodels with several association structures available. The association between submodels can be of the current value or the current slope form, similar to the JM package in R. The association can also be through shared parameters with or without covariates. Estimation is carried out by Newton-Raphson method and numerical integration is implemented with standard or adaptive Gauss-Hermite quadrature. We found Stata to have more difficulties in fitting models with various parameter values in our simulations.

#### 4.5 Comparison of Software Functionality

In summary, all three major statistical softwares R, SAS and Stata fit joint models with comparable functionality. A detailed list of available options is provided in Table 5. All software will fit a Gaussian longitudinal submodel as in (1). Some software will fit a generalized linear model where the longitudinal outcome  $Y$  has a non-Gaussian distribution such as binomial or Poisson. Each software will also fit a standard proportional hazards survival submodel modeling the hazard function as in (2) or (3). Only JM in R will fit a survival model in an accelerated failure time (AFT) framework. Available association options are listed including current value (2), random intercept or slope (3) and separate associations like we saw in section 4.2. Model fit options describe different forms of the survival submodels that may be available such as a competing risks model, lagged effects, or stratification. Estimation options include choices for the model fitting algorithm and numerical integration. Finally, some other possibly useful options in the software are listed including built-in AIC or BIC calculations, plotting, predictions, and standard error calculations.

## 5 Simulations

The joint modeling software were compared through simulation. Data were generated under three scenarios, namely a random intercept only and a random intercept and slope model in the current-value form, and a random intercept shared-parameters model. Each scenario includes  $N = 500$  subjects and 100 simulated datasets. Data were generated in R version 3.2.2 and simulations were run in Windows 7 on a 3.2 GHz Intel Core i5 processor.

### 5.1 Scenario 1

Data for the current-value association random intercept only joint model, which we will call Scenario 1, were generated as follows. The longitudinal data were generated from the model

$$Y_{ij} = m_i(t_{ij}) + e_{ij} = \beta_0 + \beta_1 t_{ij} + \beta_2 X_{1i} + \beta_3 X_{2i} + b_i + e_{ij} \quad (8)$$

We have two covariates in our joint model, a binary group indicator  $X_1$  and a continuous  $X_2$ . Measurement times are between 0 and 3 years. The random effect and measurement error are normally distributed and independent. Survival times  $T_i^*$ , were generated from a relative risk model.

$$\lambda_i(t) = \exp(\gamma_0 + \gamma_1 X_{1i} + \gamma_2 X_{2i} + \alpha m_i(t)) \quad (9)$$

(See the Supplemental Material in Section S1 for details on generating the survival times).

The parameter values used are shown below.

$$\begin{array}{ll} X_{1i} \sim \text{Bernoulli}(0.5), & \beta_0 = 0.6 \quad \gamma_0 = -4.4 \\ X_{2i} \sim N(0, 1), & \beta_1 = 0.2 \quad \gamma_1 = 0.07 \\ (t_{i1}, t_{i2}, \dots, t_{i7}) = (0, 0.5, 1, 1.5, 2, 2.5, 3), & \beta_2 = -0.1 \quad \gamma_2 = 0.2 \\ e_{ij} \sim N(0, 4), & \beta_3 = 0.4 \quad \alpha = 0.5 \\ b_i \sim N(0, 2.25) & \end{array}$$

Independent censoring times  $C_i$  were drawn from a uniform distribution,  $C_i \sim \text{Unif}(0, 3.25)$ , and the observed time  $T_i$  is the minimum,  $T_i = \min(T_i^*, C_i)$ . Any longitudinal measurements  $Y_{ij}$  at times after the observed time  $T_i$  (where  $t_{ij} > T_i$  for subject  $i$ ) were dropped.

The R `JM`, `joinerR`, Stata `stjm`, and Two-Stage software fit (8) to the longitudinal part. The SAS macro by default fits a model with a random intercept and slope. The `%JM` macro has no built-in option for an intercept only association. Therefore we fit a misspecified longitudinal model in SAS, replacing  $b_i$  with  $b_{0i} + b_{1i}t_{ij}$  in (8). An unstructured covariance matrix was used for the SAS PWC model. For the SAS Weibull model, the default diagonal covariance matrix was used to achieve convergence.

For the survival part, a current-value association was used if available. The R `JM` package, Stata command, and the `%JM` macro in SAS fit a current-value survival submodel as specified in (9). The `joinerR` package only fits joint models with the shared parameters association conforming to the structure in (10).

$$\lambda_i(t) = \lambda_0(t) \exp(X_{1i}\gamma_1^* + X_{2i}\gamma_2^* + \alpha b_i) \quad (10)$$

This amounts to a re-parameterization of the current-value model where the coefficients  $\gamma_1^*$  and  $\gamma_2^*$  in the `joinerR` survival submodel converge to combinations of the true coefficients from both submodels, specifically  $\gamma_1^* = \gamma_1 + \alpha\beta_2$  and  $\gamma_2^* = \gamma_2 + \alpha\beta_3$ . For all results, the `joinerR` coefficients will be compared to these combinations. See Section S2 for more on this reparameterization.

The baseline hazards used for these software are in (11). Pseudo-adaptive Gauss-Hermite (GH) quadrature is implemented in R `JM Weib`, `PWC` and `Spl`. For the model with an unspecified baseline hazard (R `JM Unspec NA`) we used the standard (nonadaptive) GH quadrature, since we found using the adaptive algorithm led to poor convergence. In order to investigate whether there is a difference in the nonadaptive (NA) versus the pseudo-adaptive GH quadrature, we also ran models with each of Weibull, piecewise constant, and



spline based baseline hazards, using standard GH quadrature, labeled R JM Weib NA, R JM PWC NA, and R JM Spl NA.

$$\begin{aligned}
 & \text{R JM Weib: } \lambda_0(t) \sim \text{Weibull} \\
 & \text{R JM PWC: } \lambda_0(t) \sim \text{Piecewise constant function} \\
 & \text{R JM Spl: } \log(\lambda_0(t)) \sim \text{B-spline approximation} \\
 & \text{R JM Unspec NA: } \lambda_0(t) \text{ unspecified} \\
 & \text{SAS \%JM Weib: } \lambda_0(t) \sim \text{Weibull} \\
 & \text{SAS \%JM PWC: } \lambda_0(t) \sim \text{Piecewise constant function} \\
 & \text{Stata stjw Weib: } \lambda_0(t) \sim \text{Weibull} \\
 & \text{R joiner: } \lambda_0(t) \text{ unspecified}
 \end{aligned} \tag{11}$$

A Two-Stage model is also fit to compare this more simple technique to the true joint models. The Two-Stage model is fit as follows.

$$\left\{ \begin{array}{l}
 \text{First: } Y_{ij} = \beta_0 + \beta_1 t_{ij} + \beta_2 X_{1i} + \beta_3 X_{2i} + b_i + e_{ij}, \\
 \text{Then: } \lambda_i(t) = \lambda_0(t) \exp(\gamma_1 X_{1i} + \gamma_2 X_{2i} + \alpha \hat{m}_i(t))
 \end{array} \right.$$

We used the default options for numerical integration and for defining the flexible baseline hazard functions. Specifically, the R JM PWC and SAS \%JM PWC models both use a baseline hazard constructed from six equally spaced internal knots that partition the observed event times with the function being constant in between knots. For the R JM Spl model, the B-spline approximation is constructed with five internal knots.

Results are shown in Figure 1, plotting the bias of the estimated coefficients compared to the truth. Coverage probabilities are shown in Table 6. Tables with the bias and MSE for each parameter and the width of confidence intervals are in the Supplemental Material. Widths were calculated as the average of confidence interval lengths using a Normal approx-

imation, specifically the average of  $2 * 1.96 * (\text{standard error})$ . Table 7 shows the run-times for each software.

All software except Stata were able to fit models to a majority of the 100 datasets. Stata was only able to fit around one third of the datasets. Each software had relatively good coverage although Stata did not cover the intercept term well. The intervals around the association parameter from R JM Unspec NA are narrower than the other R JM models. Stata tends to have a larger bias but smaller MSE than the other software leading to narrow confidence intervals. Every software overestimated the random effect variance, with the Two-Stage and joineR models being most biased. Despite SAS including an extra random slope term, the random intercept variance estimates were similar to R JM and Stata. Empirical standard deviations were stable across software.

Models with a flexible baseline hazard took longer to run, as would be expected. Stata was exceptionally slow in this scenario. Comparing the models using NA (nonadaptive) quadrature (ex. R JM PWC NA) to the corresponding models using pseudo-adaptive quadrature (ex. R JM PWC) shows that the nonadaptive versions slightly increased the runtime but otherwise choice of numerical integration algorithm made very little difference in the results.

## 5.2 Scenario 2

Our Scenario 2 includes a random intercept and slope joint model, sometimes called the random coefficients model, generated from the following equations.

$$Y_{ij} = m_i(t_{ij}) + e_{ij} = \beta_0 + \beta_1 t_{ij} + \beta_2 X_{1i} + \beta_3 X_{2i} + b_{0i} + b_{1i} t_{ij} + e_{ij} \quad (12)$$

$$\lambda_i(t) = \exp(\gamma_0 + \gamma_1 X_{1i} + \gamma_2 X_{2i} + \alpha m_i(t)) \quad (13)$$

As in Scenario 1, there are two covariates, one binary and one continuous. The random effects are bivariate Normal. The parameter values used are shown below.

$$\begin{aligned}
 X_{1i} &\sim \text{Bernoulli}(0.5), & \beta_0 &= 1 & \gamma_0 &= -4.4 \\
 X_{2,i} &\sim N(0, 1), & \beta_1 &= 0.2 & \gamma_1 &= 0.1 \\
 (t_{i1}, t_{i2}, \dots, t_{i7}) &= (0, 0.5, 1, 1.5, 2, 2.5, 3), & \beta_2 &= -0.1 & \gamma_2 &= 0.25 \\
 e_{ij} &\sim N(0, 4), & \beta_3 &= 0.4 & \alpha &= 0.5 \\
 b_i = (b_{0i}, b_{1i})^T &\sim N \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.5 \\ 0.5 & 1 \end{pmatrix} \right)
 \end{aligned}$$

We draw independent censoring times  $C_i \sim \text{Unif}(1.25, 3.25)$  and the observed times are  $T_i = \min(T_i^*, C_i)$ . There was approximately 10% censoring.

Each software, JM, `joiner` in R, %JM in SAS, and `stjm` in Stata, fits (12) to the longitudinal part of the data. The default covariance structure was used in all models. As in Scenario 1, several survival submodels were evaluated. The R JM package, SAS, and Stata fit a current-value survival submodel as in (13). The shared-parameter model below was fit with `joiner` which is a reparameterization of the current-value model similar to Scenario 1.

$$\lambda_i(t) = \lambda_0(t) \exp(\gamma_1^* X_{1i} + \gamma_1^* X_{2i} + \alpha(b_{0i} + b_{1i}t))$$

The baseline hazards are the same as in (11) including models using both adaptive and nonadaptive (NA) in R JM Weib, PWC and Spl. Default settings were used for the piecewise constant and spline functions. Finally, a Two-Stage model was fit for comparison.

Results for Scenario 2 are in Figure 2, and Table 8. The Table 7 shows the run-times for this scenario. More results including bias, MSE and confidence interval widths are in the Supplemental Material.

The software were able to fit models to most of the 100 simulated datasets except for `joiner`. The `joiner` models could fit only around a quarter of the datasets. Coverage

probabilities were similar for all software with one exception. Coverage for the association parameter was noticeably lower for R JM Unspec NA. The confidence interval widths were generally very similar in each software. As in Scenario 1, on average the confidence intervals around the association estimate in R JM Unspec NA were considerably tighter.

In the boxplots, we see that all R JM models and Stata estimated the random effect covariance parameters well. In SAS, the covariance of the random effects was assumed zero according to the macro defaults and we see that the variance of the intercept and the slope were overestimated by SAS. The Two-Stage model estimated the random effect variances well but severely overestimated the covariance, possibly including additional variance in the longitudinal submodel random effects estimates that would be explained by the association with the survival submodel in a joint model. Empirical standard deviations were consistent in all software. We see that the random coefficients models in this scenario took longer to run on average than the models in Scenario 1 for all software except Stata. Using nonadaptive quadrature increased runtime more compared to Scenario 1.

### 5.3 Scenario 3

The final simulation scenario utilized a shared coefficients form as in (3), with a random intercept only. The data were generated from

$$Y_{ij} = \beta_0 + \beta_1 t_{ij} + \beta_2 X_{1i} + \beta_3 X_{2i} + b_{0i} + e_{ij} \quad (14)$$

$$\lambda_i(t) = \exp(\gamma_0 + \gamma_1 X_{1i} + \gamma_2 X_{2i} + \alpha b_i) \quad (15)$$

Parameter values are below.

$$\begin{array}{lll}
 X_{1i} \sim \text{Bernoulli}(0.5), & \beta_0 = 6 & \gamma_0 = -4.4 \\
 X_{2,i} \sim N(0, 0.5), & \beta_1 = 0.2 & \gamma_1 = 0.9 \\
 (t_{i1}, t_{i2}, \dots, t_{i7}) = (0, 0.5, 1, 1.5, 2, 2.5, 3), & \beta_2 = -0.1 & \gamma_2 = 1.2 \\
 e_{ij} \sim N(0, 4), & \beta_3 = 0.4 & \alpha = 0.5 \\
 b_i \sim N(0, 0.7) & & 
 \end{array}$$

For this scenario, independent censoring times were drawn  $C_i \sim \text{Unif}(1.25, 3.25)$ . There was approximately 20% censoring.

A random intercept only shared parameter survival submodel was used since the most software could fit the correct model. The R function `joiner` and Stata fit the same longitudinal submodel in (14). Again, the default in SAS is to fit a longitudinal model with both a random intercept and slope as was the case in Scenario 1 which in this case is an misspecified model. The `joiner`, SAS, and Stata software fit (15) to the survival part. The JM package in R only fits joint models with a current-value association, which amounts to a reparameterization of the shared parameter models. In this scenario R JM fits the survival submodel in (16). The survival coefficients will converge to a combination of true parameters, namely  $\gamma_1^* = \gamma_1 - \alpha\beta_2$  and  $\gamma_2^* = \gamma_2 - \alpha\beta_3$ . See Section S2 for details.

$$\lambda_i(t) = \lambda_0(t) \exp(\gamma_1^* X_{1i} + \gamma_2^* X_{2i} + \alpha m_i(t)) \quad (16)$$

Baseline hazards are listed in (11). Both adaptive and nonadaptive (NA) numerical integration was again utilized in R JM with Weibull, piecewise constant, and spline baseline hazard functions. We used the default settings to create piecewise constant and spline baseline hazard functions. A Two-Stage model was also fit with the survival model being a standard semi-parametric Cox model.

Figure 3 and Table S3 show the results for Scenario 3. Coverage probabilities are shown

in Table 9 and Table 7 shows the run-times for all scenarios. Additional results are included in the Supplemental Material.

Models could be fit to all or almost all of the 100 simulated datasets by each software in this scenario. R `joiner` had a considerably lower coverage probability for the  $X_2$  coefficient in the survival submodel. Nearly all software except R JM Weib (and R JM Weib NA) had relatively low coverage for the association parameter, with the coverage from R JM Unspec NA being especially poor. Again, the average width of confidence intervals around the association estimate from R JM Unspec NA was much smaller than all other software. The average confidence interval widths for  $X_1$  and  $X_2$  from R JM Spl are much larger than the other software, possibly due to poor model fitting on the simulated datasets corresponding to the outliers.

R JM and Stata estimated the random intercept variance best although all (except Two-Stage) underestimated this value. The Two-Stage model severely overestimated this value as in the first two scenarios. Interestingly fitting models to this data with R JM Spl produced estimates with larger bias for most covariates and considerably smaller MSE for some covariates compared to R JM with the other baseline options. We noted that SAS produced survival submodel  $X_2$  coefficient estimates with an unusually large MSE. Also, `joiner` generated more biased estimates for the survival submodel despite fitting a model in the same form as the data generating model. As in the first two scenarios, empirical standard deviations were very similar across all software. Lastly, SAS and R `joiner` ran for notably shorter times than in the first two Scenarios.

## 6 Bayesian Models and Associated Software

Joint modeling from a Bayesian perspective have also been implemented. The full posterior distribution conforms to the structure

$$f(\theta, b|T, \delta, y) \propto \prod_{i=1}^n (f(y_i|b_i; \theta)f(T_i, \delta_i|b_i; \theta)f(b_i; \theta)) \pi(\theta),$$

where  $\pi(\theta)$  denotes the joint prior for the model parameters. MCMC techniques such as Gibbs Sampling or Metropolis-Hastings algorithm can be used for inference (Bekele and Shen, 2005; Faucett and Thomas, 1996; Gould et al., 2014; Henderson et al., 2000; Rizopoulos and Ghosh, 2011; Tsiatis and Davidian, 2004; Yu et al., 2004). The Bayesian implementation was not compared to the simulations for the maximum likelihood methods in the last section. Instead, in this section we will discuss the single option for Bayesian joint modeling in the software and also use this to analyze the adrenal cancer data.

## 6.1 JMbayer Package in R

The `JMbayer` package in R was written to fit joint models in a Bayesian framework (Rizopoulos, 2016c). The function for fitting joint models is named `jointModelBayes()`. The arguments for this function are very similar to the `jointModel()` function in the `JM` package in R. A linear mixed model is fit to the longitudinal data unless the user specifies a different distribution with the `desLong` argument. A relative risk model is fit for the time-to-event data. The baseline hazard is estimated using splines, either penalized P-Splines (the default) or regression-splines (Rizopoulos, 2016b). The `param` argument determines the form of the association between the submodels. The default association is current value as in (2). Other options are association based on the current slope of the longitudinal trajectory analogous to the R `JM` package in (7), both the current value and slope, or shared parameters like in (3). A final association option is a combination of shared random effects and fixed effects,  $\beta^*$ , such as  $\lambda_i(t|M_i(t)) = \lambda_0(t)\exp\{X_{2i}\gamma + \alpha(\beta^* + b_i)\}$ . Available functionalities in the `JMbayer` package are listed in Table 5. Extra flexibility is available for the association structure since the user can define any transformation function using the `extraForm` and `transFun` arguments. Using this, the association can be defined to be any function of the current value or any function of the shared random effects.

Estimation is done using Markov Chain Monte Carlo (MCMC) sampling from the posterior conditional distributions of the random effects and the parameters. Usually a random walk Metropolis can be used, but in some cases Metropolis-Hastings or slice sampling are

needed (Rizopoulos, 2016c). Initial values can be set using the `init` argument but if left unspecified initial values are taken from the outputs from the separate models that are included as arguments to the `jointModelBayes()` function. Priors can also be specified by the user with the `priors` argument. If not specified, standard prior distributions are used: all the fixed parameters from both submodels as well as the association parameter are given independent diffuse Normal priors, a inverse Wishart prior is assumed for the covariance matrix of the random effects when fitting a Normally distributed longitudinal outcome, and an inverse Gamma prior for the error variance (Rizopoulos, 2016c). The `JMbayes` package also includes functions for plotting and running dynamic predictions (Rizopoulos, 2016b).

## 6.2 Data Analysis

A joint model with the current value association as in (4) and (5) with a P-spline baseline hazard is fit to the adrenal cancer data from Section 3 using `JMbayes`. MCMC is run for 52,000 iterations with a burn-in of 2000. The results are shown in Table 10. The `JMbayes` estimates are generally similar to the frequentist current-value results in Table 2 and interpretations of the covariate effects on psoas density and survival are the same.

## 7 Extensions and Specialized Joint Models and their Implementation

The growing interest in joint modeling has led to many extensions, such as joint modeling with competing risks, recurrent events, or multiple longitudinal processes. The addition of many software to fit standard joint models has also encouraged the development of more specialized software. Here we briefly describe several model extensions and software available for implementation. See Section S4 in the Supplemental Material for more implementations via software. The extent of software implementation for specialized joint modeling is likely to increase in the future.

Up until this point we have focused on longitudinal measurements with a Gaussian distribution. Yet situations often arise that require a non-Gaussian outcome in the longitudinal



submodel, such as a logistic or Poisson model. The `JM` and `JMbayes` software in R as well as the `%JM` macro in SAS, described in section 4, can accommodate non-Gaussian longitudinal processes. The `Jointlcmm()` function in the `lcmm` package in R can implement different distributions in the longitudinal submodel of a latent class mixed model (Proust-Lima et al., 2016).

Joint latent class mixed modeling is an extension that can be used to investigate class-specific differences. These models typically include three submodels, a multinomial logistic model to determine the latent class, a class-specific linear (or latent process) mixed model, and a class-specific survival model. The number of latent classes must be set a priori based on knowledge of the situation from which the data were collected. The `lcmm` package in R was written for latent class mixed modeling and includes a function, `Jointlcmm()` which fits a joint latent class mixed model for longitudinal and time-to-event data (Proust-Lima et al., 2016). The baseline risk can be common or class-specific. Choice of initial values is important and it is preferred that the user specify initial values with the `B` argument over using the defaults.

In medical studies it is likely more than one biomarker is measured for each patient. This necessitates joint models with multiple longitudinal variables. Hickey et al. (2016) review developments in multivariate joint models including software implementations. Hickey et al. mentioned a new package for multivariate joint models, `sjmsoft` for R, available from the author's website (Brown, 2005). Another option is the R package `joinerML` Hickey et al. (2018).

An interesting but currently less studied extension is joint modeling with competing risks. The longitudinal submodel for a single longitudinal outcome is the same as a single risk joint model. Literature has almost exclusively focused on a survival submodel with proportional cause-specific hazards (Armero et al., 2016; Blanche et al., 2015; Elashoff et al., 2007; Hu et al., 2009; Huang et al., 2010; Williamson et al., 2008). Usually the survival submodel has

the following form, defining the cause specific hazard  $\lambda^{(k)}$  for cause  $k = 1, \dots, K$ .

$$\begin{aligned}\lambda^{(k)}(t; X_{2i}, b_i, \gamma^{(k)}, \alpha^{(k)}) &= \lim_{h \rightarrow 0} h^{-1} P(t \leq T_i < t + h, \delta_i = k | T_i \geq t, X_{2i}(t), b_i) \\ &= \lambda_0^{(k)}(t) \exp\{X_{2i}(t)\gamma^{(k)} + \alpha^{(k)}b_i\}\end{aligned}$$

The parameters are defined similar to those in Section 2.2,  $X_{2i}$  are fixed effects which have possibly cause-specific coefficients  $\gamma^{(k)}$ ,  $b_i$  are the random effects for subject  $i$  with possibly cause-specific association  $\alpha^{(k)}$ , and cause-specific baseline hazard  $\lambda_0^{(k)}(t)$ .

Some of the joint modeling software will fit a competing risks model. The JM package in R has an option `CompRisk` in the `jointModel()` function. The `joint()` function in the `joiner` package will fit a cause-specific hazard joint model if the event indicator has multiple levels but the documentation states this only works for two causes in addition to censoring, and no more (Philipson et al., 2012). The SAS macro `%JM` can also fit a competing risk model with the option `COMPETING` which is listed under `AdditionalOptions`. Documentation for the `stjm` Stata command states that extension to the competing risks setting is planned but there is no indication that this has been completed (Crowther et al., 2013). There is also no built-in option for fitting a competing risks model in `JMbayes`. Recently a SAS macro called `%SPM` was proposed specifically for fitting joint competing risks models (Wang et al., 2017). Finally, the `Jointlcmm()` function in `lcmm` in R can also handle competing risks (Proust-Lima et al., 2016).

Another common situation is data including recurrent events such as repeated hospitalizations or time between system breakdown in industry. The R package `frailtypack` will fit a standard joint model with function `longiPenal` and also joint models with longitudinal measurements, a terminal event plus recurrent events using the function `trivPenal` (Rondeau et al., 2017, 2012).

With the increasing number of studies utilizing joint modeling, it may be useful to analyze the results from multiple studies in a meta-analytic context. A package for joint modeling in a this context, `joinermeta` in R, has been developed which can pool model parameters from

multiple joint models using standard meta-analysis techniques or analyze the data from all studies simultaneously (Sudell et al., 2018; Sudell, 2018).

## 8 Concluding Remarks

Joint modeling is a growing field of statistical research and the available software encourages the use of these complicated models in applications. We have given an overview of joint modeling methodology and then compiled a comprehensive list of available software. We compared through simulation and data analysis the most common and user-friendly software: `JM`, `joineR`, and `JMbayes` in R, `%JM` in SAS, and `stjm` in Stata. We also included a short description of extensions to joint modeling such as those that accommodate competing risks, multiple longitudinal markers or recurrent events. The `%JMfit` macro developed for SAS, described in Supplemental Materials section S4, includes some goodness-of-fit calculations such as decomposition of AIC, BIC,  $\Delta$ AIC, and  $\Delta$ BIC (Zhang et al., 2016). Assessing model fit in a joint modeling framework has had limited study and further development of this area is likely in the future (Zhang et al., 2014).

Our simulations show joint modeling software is preferable to Two-Stage models when the longitudinal and survival processes are correlated, as the theoretical work finds. Ours is the first investigation that explored the performance of regression coefficients on both longitudinal and survival components. The packages in R fit only one type of association while SAS and Stata include many more options. All software we compared are similar in their performance when fitting the longitudinal submodel in simulations. There are more differences in the survival submodel options and performance. All except `joineR` included a flexible parametric baseline hazard which is most applicable to real data when the true model is unknown. A Cox-type survival submodel using an unspecified baseline hazard is currently available in the R packages `joineR` and `JM`. In each of our simulation scenarios we found that the average width of the confidence intervals around the association estimate were smaller than all other software. This could be evidence of underestimation of standard error. This is similar to what was found in simulations in Yuen and Mackinnon (2016).

Otherwise we did not see underestimation for the other covariates. Still, documentation of the R `JM` package does not indicate that any techniques are used to correct for the theoretical underestimation. Hence this should be considered when using the R `JM` package with a Cox type survival submodel. [A recently developed R package JSM fits semiparametric joint models while estimating standard errors with numerical differentiation in an EM algorithm as described by Xu et al. \(2014, 2018\)](#). We found convergence issues when using the adaptive Gauss-Hermite quadrature option only when using an unspecified baseline hazard in the R `JM` function. Additionally, we saw in our simulations that using a restrictive parametric baseline risk function that matches the truth does not aid in estimation but may shorten runtime. Unless there is a compelling reason to use a restrictive parametric function, such as Weibull, for the baseline hazard, we would recommend using a flexible hazard such as piecewise constant or spline based.

In our investigation through simulation, we found that the Stata `stjm` command can be very sensitive to the parameter values used in data generation. SAS can also have difficulties depending on the parameters chosen when using an unstructured covariance matrix, but the default model assuming uncorrelated random effects was always able to fit the models. The R functions were always able to fit models to at least some of the datasets, no matter the parameter values used.

Basic joint modeling capabilities are available in the main statistical programming languages, R, SAS, and Stata, giving the user an option to use the implementation in the language of their choice. Each software has good features as well as limitations. Overall, we would recommend R `JM` or SAS. The SAS macro `%JM` offers the most functionality, including submodel, association, and estimation options. R `JM` was consistently the fastest and includes almost as many options as SAS, so this may be a better option if runtime is a concern. The `joiner` function is limited in the type of models it can fit. But if a joint model with a Gaussian longitudinal response and a shared parameter survival submodel is appropriate for the data, `joiner` may be a good option because of its simplicity.

Bayesian models can be very powerful and are gaining traction in joint modeling literature but the implementation is relatively limited. Analyzing joint models in a general multi-purpose Bayesian statistical programming language such as OpenBUGS or WinBUGS is possible and attractive due to its flexibility. Defining a joint model can become complicated in OpenBUGS due to the lack of closed-form for the integrals encountered in estimation. A common simplifying assumption is that the survival process follows a parametric distribution, often Weibull (Guo, 2003; Guo and Carlin, 2004). Full proportional hazards survival submodels have been implemented in the literature and code is available in the supplemental materials of Rizopoulos and Ghosh (2011) for the case of multiple longitudinal outcomes and Andrinopoulou et al. (2013) for the case of two longitudinal variables and competing risks data. A competing risks joint model has been implemented in WinBUGS (Deslandes and Chevret, 2010).

Our investigation was focused on giving an overview of the available software and their features. Our simulations were limited to three scenarios that could be modeled in each common software. Other simulation scenarios were explored but not every software was able to fit models to the data. Running more simulations with different data forms and more focus on time slopes in longitudinal models, different forms of the submodels, and various associations would add to the knowledge of these software. Further development of this group of software is expected based on the continued research in this field and the applicability of these models. With increased generalization in the form of each part (longitudinal and survival submodels, and the association) the software could accommodate many more types of data. Some software are currently being developed and released to accommodate useful extensions such as competing risks and multiple longitudinal outcomes as we discussed. Further work on more robust estimation as well as faster algorithms would be useful. Fitting a simple joint model with only a few covariates is not prohibitively restrictive on time. The time required will increase with an increase in the number of covariates or a non-normal longitudinal models, especially if one has to rely on re-sampling techniques for inference

purposes. Such extensions may require creative enhancements and approximations that would be computationally efficient. An important area in joint modeling that is largely unexplored is model diagnostics. While some of the software offer some basic diagnostics (Rizopoulos, 2010, 2016c) and dynamic predictions (Crowther et al., 2013; Garcia-Hernandez and Rizopoulos, 2015; Rizopoulos, 2010, 2016c), there is room for further expansion.

## Acknowledgements

This work was partially supported by grant CA 083654 from the US National Institutes of Health.

## References

- Andrinopoulou, E.-R., D. Rizopoulos, J. J. Takkenberg, and E. Lesaffre (2013). Joint modeling of two longitudinal outcomes and competing risk data. *Statistics in Medicine* 33, 3167–3178.
- Armero, C., A. Forte, H. Perpiñán, M. J. Sanahuja, and S. Agustí (2016). Bayesian joint modeling for assessing the progression of chronic kidney disease in children. *Statistical Methods in Medical Research*, 1–17.
- Bekele, B. N. and Y. Shen (2005). A Bayesian Approach to Jointly Modeling Toxicity and Biomarker Expression in a Phase I/II Dose-Finding Trial. *Biometrics* 61, 344–354.
- Blanche, P., C. Proust-Lima, L. Loubère, C. Berr, J.-F. Dartigues, and H. Jacqmin-Gadda (2015). Quantifying and Comparing Dynamic Predictive Accuracy of Joint Models for Longitudinal Marker and Time-to-Event in Presence of Censoring and Competing Risks. *Biometrics* 71, 102–113.
- Brown, E. R. (2005). Software. Accessed: 2018-02-05.
- Brown, E. R. and J. G. Ibrahim (2003). A Bayesian Semiparametric Joint Hierarchical Model for Longitudinal and Survival Data. *Biometrics* 59(2), 221–228.
- Crowther, M. J., K. R. Abrams, and P. C. Lambert (2013). Joint modeling of longitudinal and survival data. *The Stata Journal* 13(1), 165–184.
- De Gruttola, V. and X. M. Tu (1994). Modelling Progression of CD4-Lymphocyte Count and Its Relationship to Survival Time. *Biometrics* 50(4), 1003.
- Deslandes, E. and S. Chevret (2010). Joint modeling of multivariate longitudinal data and the dropout process in a competing risk setting: application to ICU data. *Medical Research Methodology* 10(69), 1–13.
- Efron, B. and R. J. Tibshirani (1993). *An Introduction to the Bootstrap*. Chapman & Hall. pages 184-187.
- Elashoff, R. M., G. Li, and N. Li (2007). An approach to joint analysis of longitudinal measurements and competing risks failure time data. *Statistics in Medicine* 26(14), 2813–2835.

- Faucett, C. L. and D. C. Thomas (1996). Simultaneously Modelling Censored Survival Data And Repeatedly Measured Covariates: A Gibbs Sampling Approach. *Statistics in Medicine* 15(15), 1663–1685.
- Garcia-Hernandez, A. and D. Rizopoulos (2015). %JM Macro Reference Manual. Online; accessed December 1, 2015.
- Gould, A. L., M. E. Boye, M. J. Crowther, J. G. Ibrahim, G. Quartey, S. Micallef, and F. Y. Bois (2014). 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.
- Guo, X. (2003). WinBUGS 1.4 code for joint model (Model XI in Guo and Carlin paper). Online; accessed December 1, 2015.
- Guo, X. and B. P. Carlin (2004). Separate and Joint Modeling of Longitudinal and Event Time Data Using Standard Computer Packages. *The American Statistician* 58(1), 16–24.
- Henderson, R., P. Diggle, and A. Dobson (2000, Jan). Joint modelling of longitudinal measurements and event time data. *Biostatistics* 1(4), 465–480.
- Hickey, G. L., P. Philipson, and A. Jorgensen (2018). Package ‘joineRML’: Joint modeling of multivariate longitudinal data and time-to-event outcomes. Accessed: 2018-02-05.
- Hickey, G. L., P. Philipson, A. Jorgensen, and R. Kolamunnage-Dona (2016). Joint modelling of time-to-event and multivariate longitudinal outcomes: recent developments and issues. *BMC Medical Research Methodology* 16(117), 1–15.
- Holcombe, S., B. Enchakalody, P. Zhang, and S. Wang (2016). Morphomics Data Dictionary. Accessed: 2017-08-28.
- Hsieh, F., Y.-K. Tseng, and J.-L. Wang (2006). Joint modeling of survival and longitudinal data: a likelihood approach revisited. *Biometrics* 62, 1037–1043.
- Hu, W., G. Li, and N. Li (2009). A Bayesian approach to joint analysis of longitudinal measurements and competing risks failure time data. *Statistics in Medicine* 28(11), 1601–1619.
- Huang, X., G. Li, R. M. Elashoff, and J. Pan (2010). A general joint model for longitudinal measurements and competing risks survival data with heterogeneous random effects. *Lifetime Data Analysis* 17(1), 80–100.
- Huang, X., L. A. Stefanski, and M. Davidian (2009). Latent-Model Robustness in Joint Models for a Primary Endpoint and a Longitudinal Process. *Biometrics* 65(3), 719–727.
- Huang, Y., G. Dagne, and L. Wu (2011). Bayesian inference on joint models of HIV dynamics for time-to-event and longitudinal data with skewness and covariate measurement errors. *Statistics in Medicine* 30(24), 2930–2946.
- Ibrahim, J. G., H. Chu, and L. M. Chen (2010, Mar). Basic Concepts and Methods for Joint Models of Longitudinal and Survival Data. *Journal of Clinical Oncology* 28(16), 2796–2801.

- Lehmann, A. (2009). Joint modeling of degradation and failure time data. *Journal of Statistical Planning and Inference* 139(5), 1693–1706.
- Li, E., N. Wang, and N.-Y. Wang (2007). Joint Models for a Primary Endpoint and Multiple Longitudinal Covariate Processes. *Biometrics* 63(4), 1068–1078.
- Liu, L. and X. Huang (2009). Joint analysis of correlated repeated measures and recurrent events processes in the presence of death, with application to a study on acquired immune deficiency syndrome. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 58(1), 65–81.
- Mccrink, L. M., A. H. Marshall, and K. J. Cairns (2013). Advances in Joint Modelling: A Review of Recent Developments with Application to the Survival of End Stage Renal Disease Patients. *International Statistical Review* 81(2), 249–269.
- Miller, B. S., K. M. Ignatoski, S. Daignault, C. L. M. Doherty, P. G. Gauger, G. D. Hammer, S. C. Wang, G. M. Doherty, and T. U. of Michigan Analytical Morphomics Group (2012). Worsening Central Sarcopenia and Increasing Intra-Abdominal Fat Correlate with Decreased Survival in Patients with Adrenocortical Carcinoma. *World Journal of Surgery* 36, 1509–1516.
- Murphy, S. A. and A. W. van der Vaart (2000). On Profile Likelihood. *Journal of the American Statistical Association* 95(450), 449–465.
- Philipson, P., I. Sousa, P. Diggle, P. Williamson, R. Kolamunnage-Dona, and R. Henderson (2012). Package ‘joineR’: Joint modeling of repeated measurements and time-to-event data. Accessed: 2016-10-12.
- Pinheiro, J., D. Bates, S. DebRoy, D. Sarkar, EISPACK, S. Heisterkamp, and B. Van Willigen (2016). Package ‘nlme’: Linear and Nonlinear Mixed Effects Models. Accessed: 2016-10-12.
- Proust-Lima, C., V. Philips, and B. Liqueur (2016). Estimation of Extended Mixed Models Using Latent Classes and Latent Processes: The R Package lcmm. Online; accessed November 1, 2016.
- Proust-Lima, C. and J. M. G. Taylor (2009). Development and validation of a dynamic prognostic tool for prostate cancer recurrence using repeated measures of posttreatment PSA: a joint modeling approach. *Biostatistics* 10(3), 535–549.
- 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.
- Rizopoulos, D. (2012). *Joint models for longitudinal and time-to-event data: with applications in R*. CRC Press.
- Rizopoulos, D. (2016a). Package ‘JM’: Joint modeling of Longitudinal and Survival Data. Accessed: 2016-09-30.
- Rizopoulos, D. (2016b). Package ‘JMbayes’: Joint Modeling of Longitudinal and Time-to-Event Data under a Bayesian Approach. Accessed: 2016-10-30.
- Rizopoulos, D. (2016c). The R Package JMbayes for Fitting Joint Models for Longitudinal and Time-to-Event Data Using MCMC. *Journal of Statistical Software* 72(7), 1–46.



- Rizopoulos, D. and P. Ghosh (2011). A Bayesian semiparametric multivariate joint model for multiple longitudinal outcomes and a time-to-event. *Statistics in Medicine* 30(12), 1366–1380.
- Rizopoulos, D., L. A. Hatfield, B. P. Carlin, and J. J. M. Takkenberg (2014, Feb). Combining Dynamic Predictions From Joint Models for Longitudinal and Time-to-Event Data Using Bayesian Model Averaging. *Journal of the American Statistical Association* 109(508), 1385–1397.
- Rizopoulos, D., G. Verbeke, and E. Lesaffre (2009). Fully exponential Laplace approximations for the joint modelling of survival and longitudinal data. *Journal of the Royal Statistical Society, Series B* 71, 637–654.
- Rondeau, V., J. R. Gonzalez, Y. Mazroui, A. Mauguen, A. Krol, A. Diakite, A. Laurent, and M. Lopez (2017). General Frailty Models: Shared, Joint and Nested Frailty Models (Package 'frailtypack'). Online; accessed August 29, 2017.
- Rondeau, V., Y. Mazroui, and J. R. Gonzalez (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.
- Sudell, M. (2018). joineRmeta. Accessed: 2018-04-05.
- Sudell, M., R. Kolamunnage-Dona, and C. T. Smith (2018). joineRmeta: Joint Modelling for Meta-Analytic (Multi-Study) Data. Accessed: 2018-04-05.
- Sudell, M., R. Kolamunnage-Dona, and C. Tudur-Smith (2016). Joint models for longitudinal and time-to-event data: a review of reporting quality with a view to meta-analysis. *BMC Medical Research Methodology* 16(168), 1–11.
- Taylor, J. M. G., Y. Park, D. P. Ankerst, C. Proust-Lima, S. Williams, L. Kestin, K. Bae, T. Pickles, and H. Sandler (2013). Real-Time Individual Predictions of Prostate Cancer Recurrence Using Joint Models. *Biometrics* 69, 206–213.
- Therneau, T. M. and T. Lumley (2016). Package 'survival': Survival Analysis. Accessed: 2016-10-12.
- Tseng, Y.-K., F. Hsieh, and J.-L. Wang (2005). Joint modelling of accelerated failure time and longitudinal data. *Biometrika* 92(3), 587–603.
- Tsiatis, A. A. and M. Davidian (2004). Joint modeling of longitudinal and time-to-event data: an overview. *Statistica Sinica* 14, 809–834.
- Tsiatis, A. A., V. Degruittola, and M. S. Wulfsohn (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.
- Wang, W., W. Wang, T. H. Mosley, and M. E. Griswold (2017). A SAS macro for the joint modeling of longitudinal outcomes and multiple competing risk dropouts. *Computer Methods and Programs in Biomedicine* 138, 23–30.
- Wang, Y. and J. M. G. Taylor (2001). Jointly Modeling Longitudinal and Event Time Data With Application to Acquired Immunodeficiency Syndrome. *Journal of the American Statistical Association* 96(455), 895–905.

- Williamson, P., R. Kolamunnage-Dona, P. Philipson, and A. Marson (2008). Joint modelling of longitudinal and competing risks data. *Statistics in Medicine* 27(30), 6426–6438.
- Wulfsohn, M. S. and A. A. Tsiatis (1997). A Joint Model for Survival and Longitudinal Data Measured with Error. *Biometrics* 53(1), 330–339.
- Xu, C., P. D. Baines, and J.-L. Wang (2014). Standard error estimation using the EM algorithm for the joint modeling of survival and longitudinal data. *Biostatistics* 15(4), 731–744.
- Xu, C., P. Z. Hadjipantelis, and J.-L. Wang (2018). Semiparametric Joint Modeling of Survival and Longitudinal Data. Online; accessed February 11, 2019.
- Yu, M., N. J. Law, J. M. Taylor, and H. M. Sandler (2004). Joint Longitudinal-Survival-Cure Models and their Application to Prostate Cancer. *Statistica Sinica* 14, 835–862.
- Yuen, H. P. and A. Mackinnon (2016). Performance of joint modelling of time-to-event data with time-dependent predictors: an assessment based on transition to psychosis data. *PeerJ* 4, 1–20.
- Zhang, D., M.-H. Chen, J. G. Ibrahim, M. E. Boye, P. Wang, and W. Shen (2014). Assessing Model Fit in Joint Models of Longitudinal and Survival Data with Applications to Cancer Clinical Trials. *Statistics in Medicine* 33(27), 4715–4733.
- Zhang, D., M.-H. Chen, J. G. Ibrahim, M. R. Boye, and W. Shen (2016). JMfit: A SAS Macro for Joint Models of Longitudinal and Survival Data. *Journal of Statistical Software* 71(3), 1–24.

Variable		Mean or Count	SD or %
Psoas Density		54.4	8.5
Scan Time (years)		3.1	3.2
Age		46.1	13.6
Stage	1 or 2	71	44.4%
	3 or 4	89	55.6%
Tumor Grade	Low	79	49.4%
	High	58	36.3%
	Unknown	23	14.4%

Table 1: Description of adrenal cancer data.

Variable	Two-Stage*	JM PWC	SAS %JM PWC	Stata stjm Weib
N	89	87	100	100
Intercept	64.4 (59.5,68.2)	64.6 (59.2,68.3)	65.4 (59.8,69.7)	64.6 (60.6,68.6)
Time	-0.5 (-1.4,-0.1)	-0.6 (-1.7,-0.2)	-0.9 (-2.7,-0.4)	-0.6 (-1.9,-0.3)
Age	-0.2 (-0.3,-0.1)	-0.2 (-0.3,-0.1)	-0.2 (-0.3,-0.1)	-0.2 (-0.4,-0.2)
Stage 3 or 4	-0.6 (-1.7,0.2)	-0.6 (-1.7,0.3)	-0.6 (-1.8,0.05)	-0.6 (-2.2,-0.1)
Tumor Grade: High	-0.6 (-3.8,2.2)	-0.7 (-4.1,2.1)	-0.7 (-4.1,2.1)	-0.7 (-3.6,2.1)
Tumor Grade: Unknown	-2.9 (-7.5,0.3)	-3.2 (-8.4,-0.6)	-3.1 (-8.0,0.2)	-3.1 (-8.5,-0.4)
Surv-Age	0.1 (-0.009,0.6)	0.002 (-0.01,0.06)	-0.009 (-0.03,0.008)	-0.01 (-0.03,0.001)
Surv-Stage 3 or 4	0.7 (-0.06,3.4)	0.2 (0.02,0.3)	0.2 (-0.01,0.3)	0.2 (-0.06,0.4)
Surv-Tumor Grade: High	1.3 (-3.6,4.2)	0.6 (0.004,1.0)	0.5 (0.06,1.1)	0.4 (-0.04,1.1)
Surv-Tumor Grade: Unknown	3.4 (0.4,12.0)	1.0 (0.3,1.5)	0.9 (0.2,1.7)	1.0 (0.2,1.9)
Association	0.8 (0.1,3.0)	-0.07 (-0.1,-0.04)	-0.1 (-0.2,-0.06)	-0.09 (-0.1,-0.04)

Table 2: Current-value joint model parameter estimates and bias-corrected bootstrapped confidence intervals for the adrenal data. N is the number of models in 100 bootstraps that successfully converged.

\*Two-Stage intervals are the 2.5 and 97.5 quantiles of the bootstrapped values, not bias-corrected.

Variable	Two-Stage*	joineR	SAS %JM PWC	Stata stjm Weib
N	100	100	100	100
Intercept	63.7 (59.5,67.7)	63.1 (58.3,67.1)	64.5 (60.0,68.5)	64.4 (59.5,68.1)
Time	-0.3 (-0.6,-0.08)	-0.4 (-0.7,-0.2)	-0.6 (-1.4,-0.3)	-0.6 (-1.2,-0.3)
Age	-0.2 (-0.3,-0.1)	-0.2 (-0.3,-0.1)	-0.2 (-0.3,-0.1)	-0.2 (-0.3,-0.09)
Stage 3 or 4	-0.6 (-1.7,0.1)	-0.7 (-1.6,0.1)	-0.6 (-1.7,0.09)	-0.6 (-2.3,0.3)
Tumor Grade: High	-0.6 (-3.6,2.0)	0.6 (-2.0,3.7)	-0.7 (-3.7,2.1)	-0.6 (-4.0,2.2)
Tumor Grade: Unknown	-3.2 (-9.1,1.6)	-2.9 (-7.7,3.3)	-3.1 (-8.1,-0.09)	-3.2 (-7.5,-0.3)
Surv-Age	0.01 (-0.003,0.03)	0.01 (-0.004,0.02)	0.01 (-0.003,0.03)	0.006 (-0.02,0.02)
Surv-Stage 3 or 4	0.2 (0.06,0.4)	0.2 (0.01,0.4)	0.2 (0.04,0.4)	0.2 (-0.02,0.5)
Surv-Tumor Grade: High	0.5 (-0.01,1.0)	-0.5 (-1.0,0.02)	0.6 (0.04,1.2)	0.4 (-0.04,1.1)
Surv-Tumor Grade: Unknown	1.1 (0.6,1.8)	0.7 (-0.2,1.3)	1.2 (0.4,1.8)	1.3 (0.4,2.1)
Association	-0.09 (-0.1,-0.06)	-0.09 (-0.1,-0.05)	-0.1 (-0.1,-0.06)	-0.09 (-0.1,-0.05)

Table 3: Shared parameter joint model parameter estimates and bootstrapped confidence intervals for the adrenal data. N is the number of models in 100 bootstraps that successfully converged.

\*Intervals are the 2.5 and 97.5 quantiles of the bootstrapped values, not bias-corrected.

	Current-value	Shared Parameter
JM PWC	-3130.6	NA
joiner	NA	-3392.1
SAS %JM PWC	-3147.5	-3147.0
Stata stjm Weib	-3100.9	-3083.3

Table 4: Log-likelihood values for the adrenal data models.

Functionality	JM R package	joiner R package	%JM SAS Macro	%JMfit SAS Macro	stjm Stata Command	JMbayes R package
Longitudinal Submodel						
Gaussian linear model	x	x	x	x	x	x
Generalized linear models	x		x			x
Covariance matrix options			x		x	
Survival Submodel						
Relative Risk model	x	x	x	x	x	x
AFT Weibull model	x					
Unspecified baseline hazard	x	x				
Piecewise baseline hazard	x		x	x		
Spline baseline hazard	x		x	x	x	x
Association						
Current value assoc.	x		x		x	x
Current slope assoc.	x		x		x	x
Random intercept assoc.		x	x	x	x	x
Random slope association		x	x	x	x	x
Separate associations		x		x		x
Interactions in associations	x		x	x	x	x
Model fit options						
Lagged effects	x		x			x
Competing risks models	x		x			
Initial value options	x		x	x		x
Stratification in survival submodel	x		x			
Piecewise/Spline hazard customization	x		x	x		x
Estimation options						
EM only	x	x				
Quasi-Newton	x		x	x		
Bayesian MCMC						x
Number of iteration control	x	x	x	x	x	x
Convergence tolerance control	x	x	x	x	x	
Piecewise/Spline knots control	x		x	x		x
Quadrature points control	x	x	x	x	x	
Adaptive Gauss-Hermite quadrature			x	x	x	
Pseudo-adaptive Gauss-Hermite quadrature	x					
Laplace approximation	x		x			
Gauss-Kronrod Rule	x		x		x	
Other options						
AIC or BIC			x	x		x
Plotting	x	x			x	x
Predictions	x					x
Approx SE default	x		x	x	x	
Bootstrap SE option		x	x		x	

Table 5: Overview of available functionalities for joint modeling in each software; 'x' means the software in that column has the feature in the corresponding row.

Software	n models	Intercept	Time	X1	X2	Surv X1	Surv X2	Assoc.
Two-Stage	100	88	94	93	94	<b>97</b>	91	93
R JM Weib	93	86	91	92	<b>96</b>	<b>97</b>	90	<b>98</b>
R JM Weib NA	92	86	91	91	<b>97</b>	<b>97</b>	91	<b>98</b>
R JM PWC	100	87	92	93	<b>96</b>	<b>97</b>	91	<b>97</b>
R JM PWC NA	100	86	92	92	<b>97</b>	<b>97</b>	91	<b>97</b>
R JM Spl	93	86	91	92	<b>96</b>	<b>97</b>	90	<b>97</b>
R JM Spl NA	93	85	91	91	<b>97</b>	<b>97</b>	90	<b>97</b>
R JM Unspec NA	100	86	92	92	<b>96</b>	92	91	79
SAS % JM Weib*	100	87	94	93	<b>96</b>	<b>97</b>	91	<b>97</b>
SAS %JM PWC*	100	83	91	93	<b>96</b>	94	89	92
Stata Weib	33	58	85	82	88	<b>97</b>	91	<b>97</b>
R joineR**	100	85	94	92	92	<b>99</b>	<b>96</b>	89

Table 6: Coverage probabilities for Scenario 1 (in %)

\* The SAS longitudinal model is misspecified, including a random slope not in the data generation model.

\*\* The Surv X1 and Surv X2 estimates are compared to the value to which they converge and standard errors are estimated through bootstrapping. Details in Section S2.

Software	Average run-times (seconds)		
	Scenario 1	Scenario 2	Scenario 3
Two-Stage	0.5	1.5	0.1
R JM Weib	3.1	10.9	5.6
R JM Weib NA	5.3	44.5	8.2
R JM PWC	7.8	20.6	11.7
R JM PWC NA	10.5	85.2	15.8
R JM Spl	21.1	28.5	15.4
R JM Spl NA	23.7	104.9	20.6
R JM Unspec NA	6.6	223.8	93.7
SAS %JM Weib	35.6	49.7	10.6
SAS %JM PWC	94.5	392.5	49.3
Stata Weib	1201.3	609.6	661.9
R joineR*	96.9	121.1	48.5

Table 7: Average runtime for Scenarios 1, 2, and 3.

\* joineR run-times include estimating bootstrapped standard errors with 50 bootstrapped samples.

Software	n models	Intercept	Time	X1	X2	Surv X1	Surv X2	Assoc.
Two-Stage	100	<b>98</b>	87	<b>97</b>	89	<b>95</b>	94	<b>97</b>
R JM Weib	88	<b>95</b>	93	<b>97</b>	91	93	90	90
R JM Weib NA	95	<b>98</b>	<b>96</b>	<b>98</b>	91	94	93	92
R JM PWC	87	<b>97</b>	94	<b>98</b>	90	<b>95</b>	92	<b>98</b>
R JM PWC NA	94	<b>97</b>	<b>96</b>	<b>98</b>	91	<b>96</b>	91	<b>95</b>
R JM Spl	83	<b>95</b>	93	<b>95</b>	87	93	92	<b>95</b>
R JM Spl NA	94	<b>97</b>	<b>96</b>	<b>98</b>	91	<b>95</b>	93	<b>96</b>
R JM Unspec NA	85	<b>96</b>	<b>95</b>	<b>98</b>	92	91	92	64
SAS %JM Weib	100	92	91	91	86	88	89	86
SAS %JM PWC	100	<b>98</b>	<b>97</b>	<b>97</b>	91	<b>95</b>	92	89
Stata Weib	99	<b>97</b>	<b>97</b>	<b>97</b>	90	94	93	90
R joineR	25	<b>100</b>	88	<b>100</b>	<b>96</b>	<b>96</b>	80	92

Table 8: Coverage probabilities for Scenario 2 (in %).

\* The Surv X1 and Surv X2 estimates are compared to the value to which they converge and standard errors are estimated through bootstrapping. Details in Section S2.

Software	n models	Intercept	Time	X1	X2	Surv X1	Surv X2	Assoc.
Two-Stage	100	91	94	<b>98</b>	93	<b>97</b>	92	94
R JM Weib	82	<b>95</b>	93	<b>99</b>	91	88	<b>95</b>	<b>95</b>
R JM Weib NA	80	91	90	<b>99</b>	91	<b>96</b>	94	<b>96</b>
R JM PWC	88	<b>95</b>	93	<b>99</b>	93	<b>95</b>	<b>98</b>	82
R JM PWC NA	89	94	93	<b>98</b>	92	<b>95</b>	<b>97</b>	80
R JM Spl	97	94	<b>95</b>	<b>98</b>	92	<b>97</b>	<b>95</b>	87
R JM Spl NA	100	94	94	<b>98</b>	92	<b>97</b>	<b>95</b>	88
R JM Unspec NA	45	93	89	<b>100</b>	89	93	88	11
SAS %JM Weib	100	87	<b>96</b>	<b>96</b>	94	<b>100</b>	<b>97</b>	84
SAS %JM PWC	100	88	<b>96</b>	<b>96</b>	94	<b>98</b>	<b>100</b>	80
Stata Weib	89	51	85	<b>98</b>	91	<b>99</b>	<b>100</b>	80
R joineR*	100	93	94	<b>95</b>	89	80	51	91

Table 9: Coverage probabilities for Scenario 3 (in %).

\* The SAS longitudinal model is misspecified, including a random slope not in the data generation model.

\*\* The Surv X1 and Surv X2 estimates are compared to the value to which they converge and standard errors are estimated through bootstrapping. Details in Section S2.

Variable	Estimate
Intercept	62.0 (57.4,66.5)
Time	-0.5 (-0.9,-0.2)
Age	-0.2 (-0.24,-0.07)
Stage 3 or 4	-0.4 (-1.4,0.5)
Tumor Grade: High	-0.5 (-3.2,2.2)
Tumor Grade: Unknown	-3.2 (-6.7,0.4)
Surv-Age	-0.007 (-0.02,0.01)
Surv-Stage 3 or 4	0.2 (0.01,0.3)
Surv-Tumor Grade: High	0.40 (-0.03,0.9)
Surv-Tumor Grade: Unknown	0.8 (0.2,1.4)
Association	-0.09 (-0.1,-0.06)

Table 10: Parameter estimates and credible intervals for the joint models fit to the adrenal data with JMbayes.



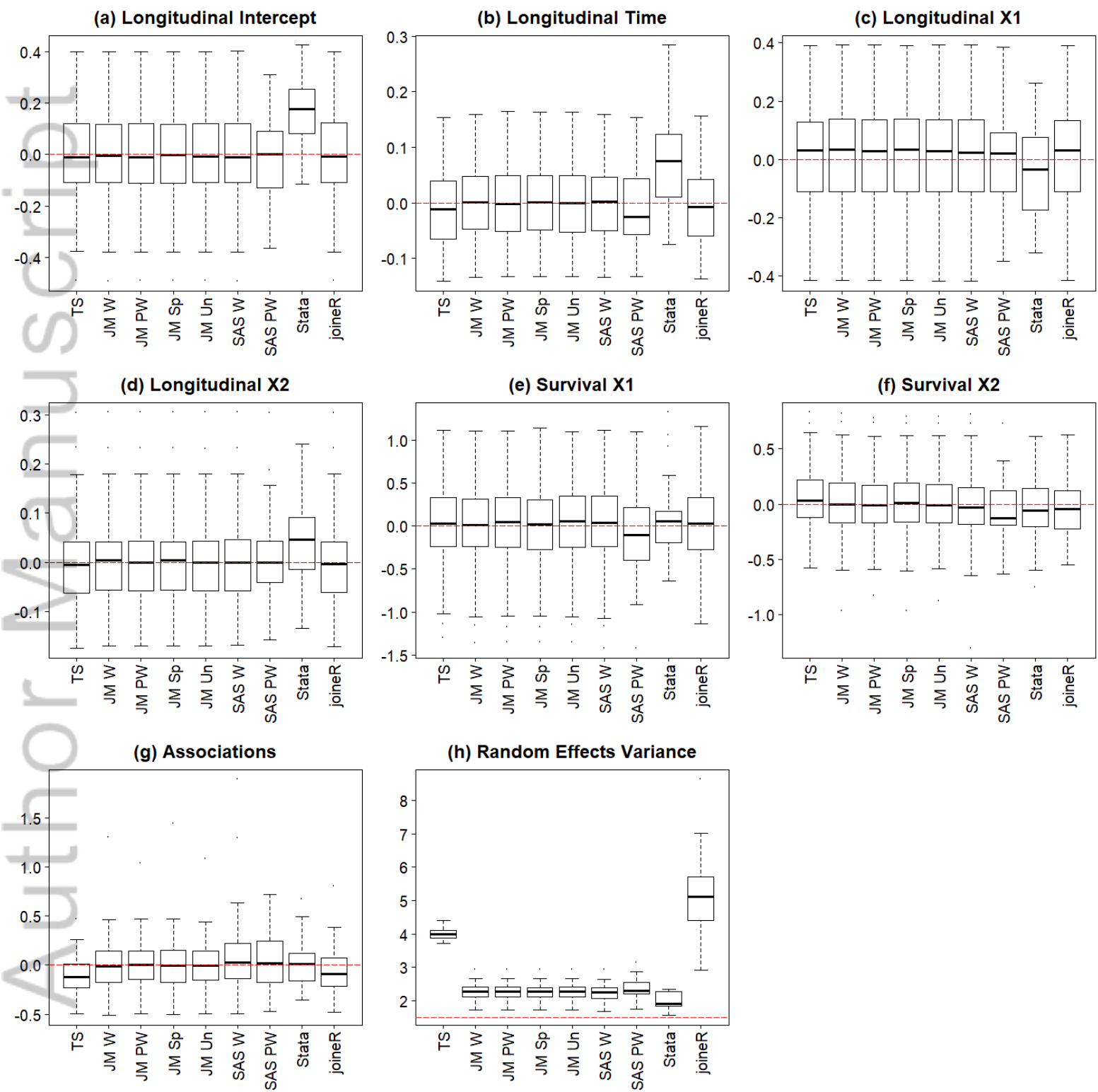
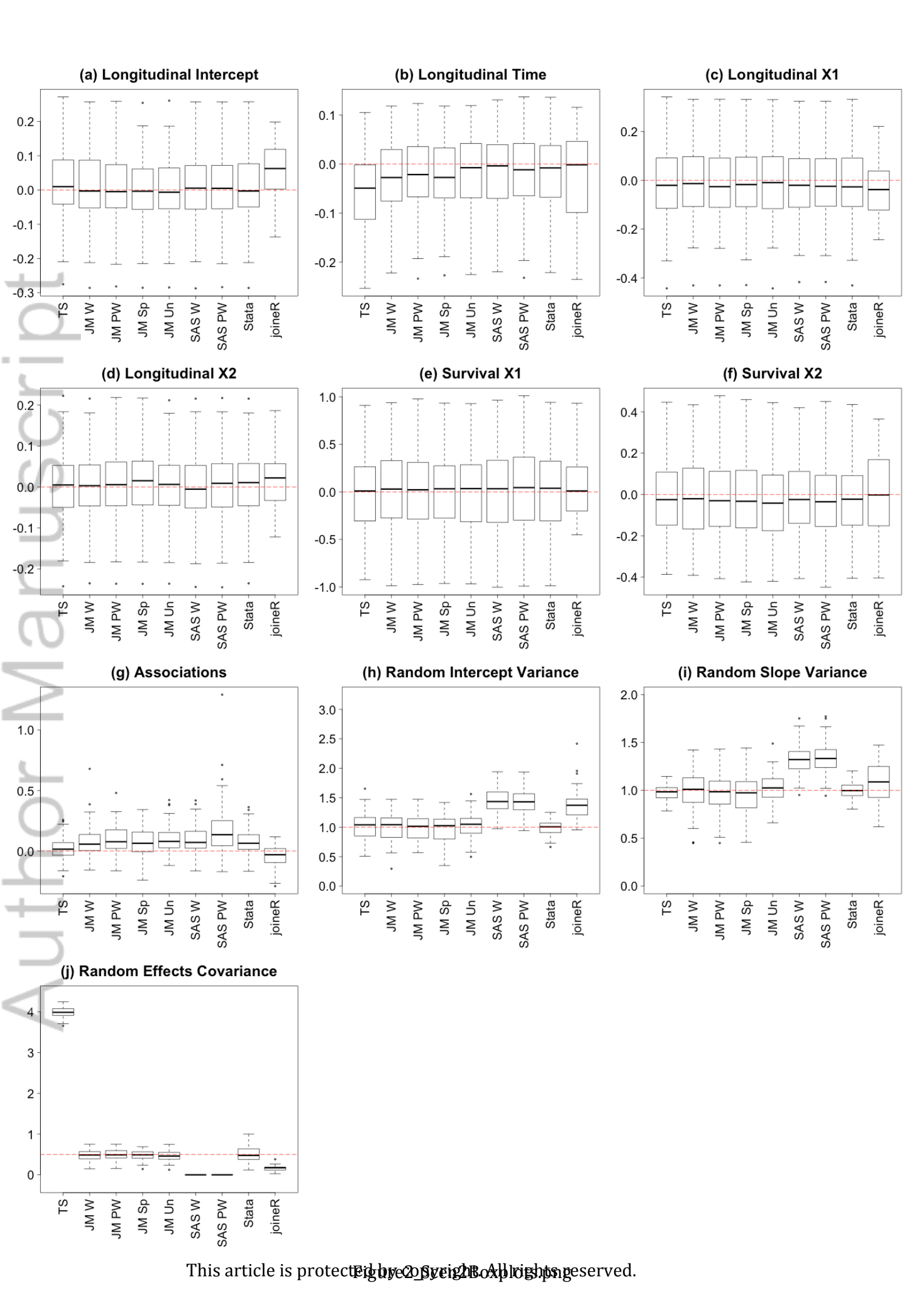


Figure1\_Scen1Boxplots.png



(a) Longitudinal Intercept

(b) Longitudinal Time

(c) Longitudinal X1

(d) Longitudinal X2

(e) Survival X1

(f) Survival X2

(g) Associations

(h) Random Intercept Variance

(i) Random Slope Variance

(j) Random Effects Covariance

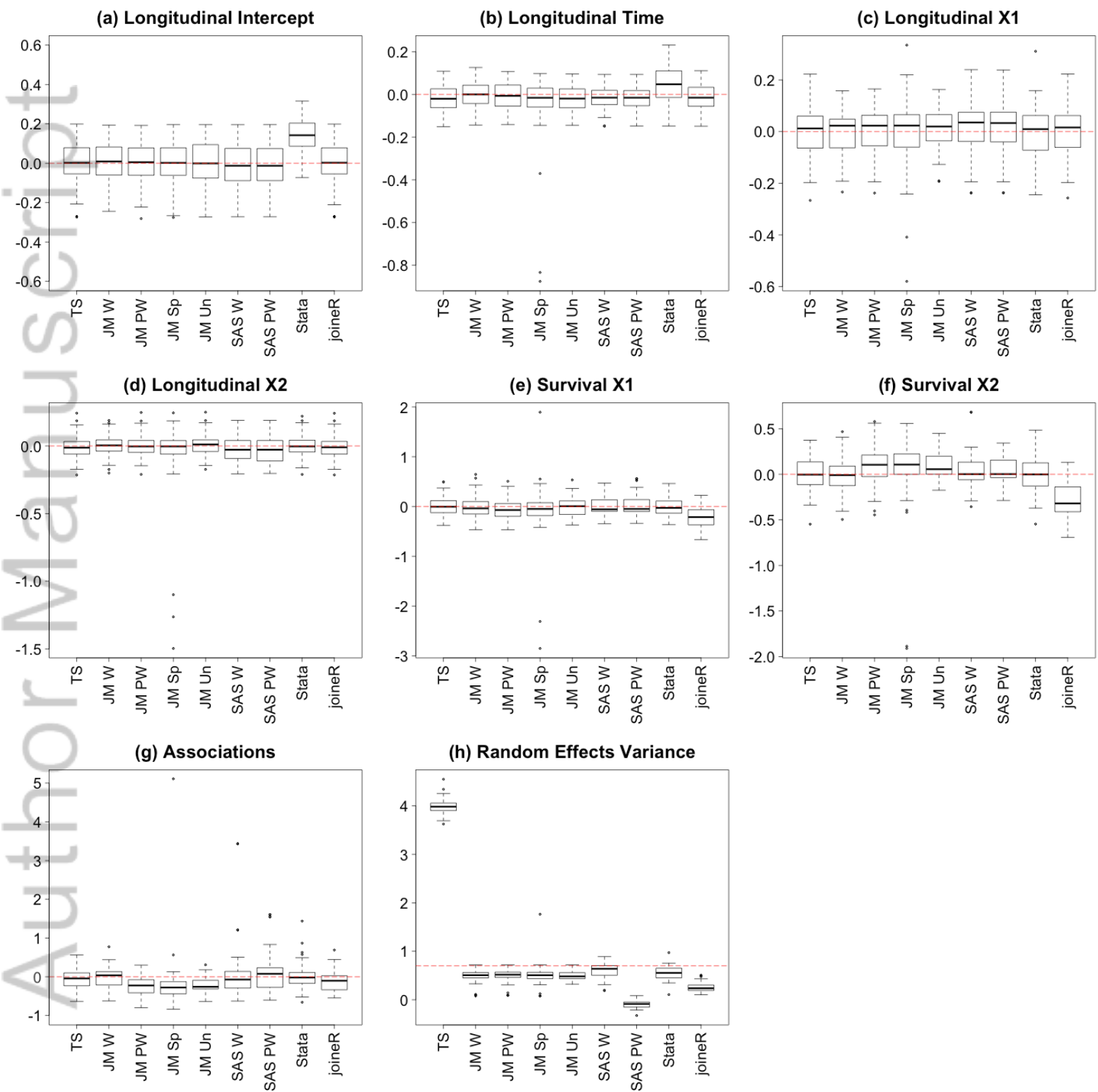


Figure3\_Scen3Boxplots.png