

Bayesian survival analysis with INLA

Danilo Alvares¹ | Janet van Niekerk² | Elias Teixeira Krainski² | Håvard Rue² | Denis Rustand²

¹MRC Biostatistics Unit, University of Cambridge, Cambridge, UK

²Computer, Electrical and Mathematical Sciences and Engineering Division, King Abdullah University of Science and Technology, Thuwal, Kingdom of Saudi Arabia

Correspondence

Danilo Alvares, MRC Biostatistics Unit, University of Cambridge, CB20SR, UK.
Email: danilo.alvares@mrc-bsu.cam.ac.uk

Funding information

Medical Research Council, Grant/Award Number: MC_UU_00002/5

This tutorial shows how various Bayesian survival models can be fitted using the integrated nested Laplace approximation in a clear, legible, and comprehensible manner using the INLA and INLAjoint R-packages. Such models include accelerated failure time, proportional hazards, mixture cure, competing risks, multi-state, frailty, and joint models of longitudinal and survival data, originally presented in the article “*Bayesian survival analysis with BUGS*.” In addition, we illustrate the implementation of a new joint model for a longitudinal semi-continuous marker, recurrent events, and a terminal event. Our proposal aims to provide the reader with syntax examples for implementing survival models using a fast and accurate approximate Bayesian inferential approach.

KEYWORDS

Bayesian inference, INLA, joint modeling, R-packages, time-to-event analysis

1 | INTRODUCTION

Time until an event of interest is a very common data type in many fields such as medicine (eg, time to death), biology (eg, resistance time of a bacterium), political science (time until a protest occurs), engineering (eg, failure time of a compress), and so on. To model this type of problem, some features must be considered: (i) non-negative variable of interest (and potentially positive skewness), (ii) censoring, and (iii) truncation.^{1,2}

Censoring is a type of incomplete/partial information as the exact time of the event of interest is not recorded during the study time window. For instance, an individual has already experienced the event of interest before starting the study (left-censoring), or the study ends without the subject having experienced such an event (right-censoring), or the event occurred within an interval instead of being observed exactly (interval-censoring). Truncation occurs when only those individuals whose event time lies within a certain interval are observed. Specifically, truncation is due to sampling bias in the study design and can be classified as left, right, or interval truncation. For instance, there are studies that define a minimum age (left-truncation) to consider individuals at risk (eg, 65 years of age for Alzheimer’s patients), so any record of individuals younger than the minimum age is not observed/considered and all survival/risk analyses are conditioned on the minimum age. See Kleinbaum and Klein³ for more examples of censoring and truncation.

The distinct features of time-to-event data are what make the use of standard statistical techniques unfeasible. However, this problem is circumvented with survival (or reliability) theory, which proposes mechanisms that incorporate such features into the inferential process.⁴ To introduce these mechanisms, let T be a non-negative random variable representing the time-to-event with hazard and survival functions expressed as $h(t|\theta)$ and $S(t|\theta)$, respectively, where θ is a parameter vector. The contribution of a certain individual to the likelihood function depends on their type of censoring and whether the study presents truncation. In order to explore all scenarios, let’s define some elements: t is the

This is an open access article under the terms of the [Creative Commons Attribution](https://creativecommons.org/licenses/by/4.0/) License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2024 The Author(s). *Statistics in Medicine* published by John Wiley & Sons Ltd.

TABLE 1 Types of censoring and truncation and their influences on the likelihood function.

	Likelihood = numerator/denominator	
	Censoring (numerator)	Truncation (denominator)
No	$h(t \theta)S(t \theta)$	1
Left	$1 - S(C_l \theta)$	$S(L \theta)$
Right	$S(C_r \theta)$	$1 - S(R \theta)$
Interval	$S(C_l \theta) - S(C_r \theta)$	$S(L \theta) - S(R \theta)$

exact/observed survival time; C_l is the left-censoring time, indicating that the event occurred before this time (eg, before the data is collected or study has started); C_r is the last time of registration of an individual (loss-to-follow-up censoring) or end of the study period (administrative censoring) and the event of interest did not occur (right-censoring); L is the lower limit of time for individuals to be at risk (left-truncation); and R is the upper limit of time for individuals to be at risk (right-truncation). Table 1 summarizes how the peculiarities of survival data influence an individual's contribution to the likelihood function.

A well-known approach to describing the hazard function, $h(t|\theta)$, is assuming *proportional hazards*.⁵ This means that $h(t|\theta)$ can be factorized into a baseline hazard function, $h_0(t)$, which is common to all individuals, and an exponential term with linear predictors. Interpretatively, the role of the baseline hazard function is to characterize the shape of $h(t|\theta)$. The most popular specifications of $h_0(t)$ includes the exponential distribution, which depends on a single scale parameter and assume the hazard is constant over time. A direct extension is the Weibull distribution, where the additional shape parameter allows for increasing/decreasing hazard over time when its value is above/below 1, respectively, and simplifies to the exponential when equal to 1. To avoid the assumption of parametric distributions for the baseline hazard, the piecewise constant specification allows for a decomposition of the follow-up into intervals, each associated to a different baseline hazard value. Finally, to avoid overfitting while being able to capture various trajectories, the use of smooth splines is a popular option.⁶ Some of these examples are explored in more detail throughout this work and a class of models that does not assume proportional hazards, the so-called *accelerated failure time models*,⁷ is also presented.

In the last decades, many books, articles, and packages have proposed methodologies for the survival framework. In particular, the Bayesian approach has been widely used to measure uncertainties, incorporate prior knowledge, and make dynamic predictions.^{8–12} However, Markov chain Monte Carlo (MCMC) methods can be quite time-consuming for complex survival models and/or massive data. Integrated nested Laplace approximation (INLA) is a convenient alternative to MCMC for Bayesian inference due to its execution speed and accuracy.¹³ It can be used to fit most survival models¹⁰ but there is limited literature highlighting it, mostly because INLA was further considered for survival analysis recently^{14,15} and its use through the INLA R-package becomes tricky when fitting survival models more complex than a simple parametric proportional hazards model.¹⁶

Inspired by “*Bayesian survival analysis with BUGS*”,¹⁷ this tutorial aims to implement the same survival models presented in the original work, and more complex models, using INLA.

The rest of the work is organized as follows. Section 2 introduces the INLA methodology and its R-packages that will support the implementation of our Bayesian survival models. Sections 3 to 9 describe accelerated failure time, proportional hazards, mixture cure, competing risks, multi-state, frailty, and joint models of longitudinal and survival data, respectively, as in Alvares et al.¹⁷ Section 10 exemplifies the implementation of joint models of longitudinal semicontinuous, recurrent events and terminal event data using INLA. Section 11 gives some insights on model check and illustrates how to perform prior sensitivity analysis. Section 12 compares computational time and posterior distributions between INLA and MCMC for the models from Sections 3 to 9. Finally, the article is concluded by a brief discussion in Section 13. R codes are available at <https://github.com/DenisRustand/Bayes-surv-INLA>.

2 | THE INTEGRATED NESTED LAPLACE APPROXIMATION (INLA)

The integrated nested Laplace approximation (INLA) is a deterministic alternative to sampling-based methods.¹³ It performs fast and accurate Bayesian inference for latent Gaussian models (LGM) using Laplace approximations and efficient

numerical integrations.¹⁸ Recent developments within the INLA methodology, using variational Bayes with a Laplace approximation,¹⁹ further improve accuracy and efficiency, making INLA an attractive approximate Bayesian inference method for fast and accurate results while maintaining the flexibility of the models being considered.

Consider the data \mathbf{y} , latent field \mathbf{x} and a set of hyperparameters $\boldsymbol{\theta}$. A statistical model can be formulated as an LGM if it is a Bayesian hierarchical model where each data point is connected to one linear predictor, such that the data is conditionally independent given the latent field and the set of hyperparameters (ie, $y_i \perp y_j | \mathbf{x}, \boldsymbol{\theta}$). An LGM can thus be defined as follows:

$$\begin{aligned} y_i &\sim f(\eta_i, \boldsymbol{\theta}), & \boldsymbol{\eta} &= \mathbf{A}\mathbf{x} \quad \text{with likelihood function} & \prod_{i=1}^n f(y_i | \mathbf{x}, \boldsymbol{\theta}), \\ \mathbf{x} | \boldsymbol{\theta} &\sim \text{Normal}(\mathbf{0}, \mathbf{Q}_{\text{prior}}^{-1}), \\ \boldsymbol{\theta} &\sim g(\cdot). \end{aligned}$$

Hence, we have three levels in the model, the likelihood from the data (this can be constructed by many different models like Binomial, Poisson, Gamma, Beta, Student's t, etc), the latent field (containing all the fixed effects and random effects) is assumed to be a Gaussian Markov random field (GMRF), $p(\mathbf{x} | \boldsymbol{\theta})$, and the set of hyperparameters (containing hyperparameters from both the likelihood model and the GMRF) can assume any prior, $g(\boldsymbol{\theta})$. The Gaussian assumption is thus only for the latent field prior.

The INLA methodology can be summarized in mainly two stages, in the first stage the posterior density of the hyperparameters, $\pi(\boldsymbol{\theta} | \mathbf{y})$, is approximated while the posterior density of the latent field, $\pi(\mathbf{x} | \mathbf{y})$, is approximated in the second stage. Various numerical and computational techniques are used in both stages, including the use of a smart gradient²⁰ in the optimization step of stage 1, an optimization-based low-rank implicit variational Bayes correction²¹ in stage 2, and others—more details are available in works dedicated to the INLA methodology.^{13,19,22} The various stages can be summarized as follows:

- Stage 1: Define

$$\pi(\boldsymbol{\theta} | \mathbf{y}) = \frac{p(\mathbf{x} | \boldsymbol{\theta}) g(\boldsymbol{\theta}) \prod_{i=1}^n f(y_i | \mathbf{x}, \boldsymbol{\theta})}{\pi(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta})}.$$

Now we approximate $\pi(\boldsymbol{\theta} | \mathbf{y})$ with $\tilde{\pi}(\boldsymbol{\theta} | \mathbf{y})$ using a Gaussian approximation of $\pi(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta})$ around its mode $\boldsymbol{\mu}$ such that

$$\tilde{\pi}(\boldsymbol{\theta} | \mathbf{y}) = \left. \frac{p(\mathbf{x} | \boldsymbol{\theta}) g(\boldsymbol{\theta}) \prod_{i=1}^n f(y_i | \mathbf{x}, \boldsymbol{\theta})}{\pi_G(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta})} \right|_{\mathbf{x}=\boldsymbol{\mu}},$$

with posterior marginals

$$\tilde{\pi}(\theta_j | \mathbf{y}) = \int \tilde{\pi}(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}_{-j}.$$

- Stage 2: Using variational Bayes, correct the mean of the Gaussian approximation $\pi(\mathbf{x} | \mathbf{y}, \boldsymbol{\theta})$ to obtain a new improved joint mean $\boldsymbol{\mu}^*$,²¹ from which the conditional posteriors $\pi(x_j | \mathbf{y}, \boldsymbol{\theta})$ can be extracted. Then the posterior marginals of the latent field can be calculated as

$$\tilde{\pi}(x_j | \mathbf{y}) = \int \pi(x_j | \mathbf{y}, \boldsymbol{\theta}) \tilde{\pi}(\boldsymbol{\theta} | \mathbf{y}) d\boldsymbol{\theta}.$$

Another strategy (originally proposed in Rue et al¹³) uses subsequent Gaussian approximations (equal to the size of the latent field, which can be large) to find the element-wise conditional posteriors, and is named the Laplace strategy. The use of the low-rank Variational Bayes correction achieves the same accuracy with much less computational cost as the Laplace strategy.

The integration over the hyperparameter space is done using a central composite design (CCD) approach where integration points and corresponding weights are used for the numerical integration. If the dimension of the hyperparameter space is large (roughly more than 20), this integration can be slow, and either an empirical Bayes approach can be used in INLA, or parallelization.²³ The empirical Bayes approach uses only the mode of the hyperparameter posterior and circumvents the integration in totality. This method presents excellent frequentist properties for multivariate joint longitudinal-survival models although it does not fully account for uncertainty.²⁴

2.1 | Survival analysis with INLA

At first glance, it might seem that survival models are not LGMs, but previous works show how various survival models can be formulated as LGMs and thus INLA can be used for approximate Bayesian inference of these models.^{10,15,25,26} The most common method for analyzing survival data assumes proportional hazards, initially proposed by Sir David Cox.⁵ It has been shown that the Cox model can be approximated using a Poisson regression framework.^{27–29} The approximation involves transforming the continuous-time hazard function into a discrete-time event count intensity function. By considering sufficiently small time intervals, the hazard can be approximated as relatively constant within each interval. Consequently, the number of events in each interval follows a Poisson distribution, with the mean proportional to the hazard in that specific interval. This approximation becomes more accurate as the time intervals become narrower.

The hazard function of the Cox model is given by $h(t|\theta)$ and the Poisson approximation in discrete time is based on the assumption that the number of events N_i in a small time interval of length Δt_i follows a Poisson distribution:

$$N_i \sim \text{Poisson}(\lambda_i),$$

where the mean λ_i is proportional to the hazard within the interval such that:

$$\lambda_i = h(t_i|\theta)\Delta t_i.$$

The Poisson regression model aligns with the LGM framework which implies that INLA can be used as a unified framework for approximate Bayesian inference of various complex survival models, some of which are considered here.

Since this approach involves a decomposition of the follow-up into many small intervals, we can also take advantage of this data augmentation to account for time-dependent components in the hazard function, thus allowing to fit joint models for longitudinal and survival outcomes with time-dependent association structures. Parametric proportional hazards models and accelerated failure time models can also be integrated into this framework by specifying a specific distribution for the baseline hazard approximation (eg, exponential and Weibull). Various censoring schemes, such as right, left, or interval censoring, are also accommodated.

For a semi-parametric approach, the default option for baseline hazards in `INLAjoint` is a random walk model. In most cases, one does not know much about the shape of the baseline hazard and thus could let the data determine it. However, it is important to avoid overfitting, this is why we use Bayesian smooth splines (ie, random walk of order 1 or 2) to avoid abrupt changes in the baseline hazard which is usually unrealistic. The random walks are defined by hyperparameters that are part of the model parameters estimated by `INLAjoint`. The baseline hazard in `INLAjoint` is specified with the argument “`basRisk`” with the default value being “`rw1`” (ie, random walk of order 1) and alternative options being “`rw2`” (ie, random walk of order 2), “`exponentialsurv`” and “`weibullsurv`” (exponential and Weibull parametric distributions, respectively).

2.2 | The INLA and INLAjoint R-packages

The INLA approach is available for R users in the `INLA` R-package, sometimes termed R-INLA, and can be installed from its website (<https://www.r-inla.org/download-install>). The package is ever-evolving with new functionality being added continuously. The core method is implemented in C language considering the state of the art of the necessary numerical algorithms. It uses two levels of parallelization to run simultaneously multiple tasks on shared memory with `OPENMP` and each one can also run in parallel. This parallel computation approach enables the use of R-INLA for complex models and big data applications.^{23,25}

Because the INLA R-package is able to fit a wide variety of statistical models (eg, generalized linear and additive mixed effects models, spatial and spatio-temporal models, temporal models, stochastic volatility models, disease mapping models, survival models),²² it is required to formulate the survival model as an LGM, which becomes cumbersome for complex models. In this context, the INLAjoint R-package is an interface that facilitates the use of INLA for the more specific sub-class of models that involves longitudinal and/or survival outcomes. It makes the use of INLA for those models much simpler while being flexible enough to fit most survival models that INLA can fit. In this article, we will use this package for all the survival models fitted as it keeps the syntax simple, especially when considering the default priors.³⁰

The R package INLAjoint is available on CRAN (<https://cran.r-project.org/web/packages/INLAjoint/index.html>) and Github (<https://github.com/DenisRustand/INLAjoint>). It revolves around the main function `joint()` used to fit the models:

```
joint(
  formLong = NULL, dataLong = NULL, id = NULL, timeVar = NULL,
  family = "gaussian", link = "default", corLong = FALSE,
  corRE = TRUE, formSurv = NULL, dataSurv = NULL,
  basRisk = "rw1", NbasRisk = 15, cutpoints = NULL,
  assocSurv = NULL, assoc = NULL, control = list(), ...
)
```

The arguments `formLong` and `formSurv` corresponds to the formula for longitudinal and survival models, respectively. Both can be specified as lists in case of multivariate outcomes and can be used simultaneously for joint models of longitudinal and survival data. The arguments `dataLong` and `dataSurv` give the datasets corresponding to each formula, although it is possible to specify only one dataset that contains information for all formulas.

When fitting longitudinal models, the following arguments can be specified: `id` (name of the grouping variable for repeated measurements), `timeVar` (name of the time variable), `family` (family of the likelihood, which is specified as a vector in case of multivariate longitudinal outcomes), `link` (link between the linear predictor and the outcome for each family), `corLong` (Boolean to indicate whether longitudinal outcomes should have their random effects correlated), `corRE` (Boolean indicating if random effects of a given longitudinal model should be correlated or independent, this should be a vector if there are multiple groups of random effects). On the other hand, for survival submodels, the following arguments can be specified: `basRisk` gives the baseline risk function, with default random walk but it can also be specified as a parametric distribution as explained in Section 2.1, `NbasRisk` gives the number of cutpoints to decompose the follow-up into small intervals as described in Section 2.1, this allows to approximate the Cox model with a Poisson distribution and it is also used for the evolution over time of baseline hazards and time-dependent components in the baseline hazards function. Alternatively, the argument `cutpoints` allows to manually specify the cutpoints instead of the default equidistant cutpoints between baseline and the maximum follow-up time, which should be provided as a vector.

Finally, parameters related to the association between submodels in the context of joint modeling are: `assocSurv` which allows to share a frailty term (ie, random effect in a survival submodel) into another survival submodel, as illustrated in our example described in Section 10 and `assoc` which allows to specify the association between longitudinal submodels and survival submodels, with the possibility to share the entire linear predictor (“CV” for current value association), the individual deviation from the population mean (“SRE” for shared random effects), the rate of change of the linear predictor (“CS” for current slope) and the individual deviation defined by each random effect shared and scaled separately (“SRE_ind” for shared random effects independently).

The last argument is `control`, which is a list of options that can be specified. Details about these available control options are listed in the help page of the `joint` function that can be accessed with the `?joint` call in R after loading INLAjoint.

2.3 | Priors in the INLAjoint R-package

The default prior distributions in INLAjoint are specified in a weakly informative framework. All regression coefficients and the Weibull log-scale parameter follow a Normal($\mu = 0$, $\sigma^2 = 100$) prior. For hyperparameters, some default priors

are penalizing complexity priors as defined by Simpson et al.³¹ PC priors are derived based on an exponential prior on the distance between a model and its simpler counterpart, obtained by a specific value or limit of the hyperparameter. The prior contracts toward the simpler model at a fixed exponential rate. These priors are computationally simple and allow for adjustment of the prior informativeness with a single user-defined contraction parameter. Particularly, the Weibull shape parameter follows a penalized complexity prior, PC(5) (see `inla.doc("pcalphaw")` for more details). This type of prior is an informative shrinking alternative to existing prior choices such as Gamma or Uniform distributions, as described in van Niekerk et al.¹⁴ The precision parameter for the random walks also follows a penalized complexity prior such that the probability of the marginal standard deviation of the constrained random walk specified at the knots being above a given threshold $P(\sigma > 0.5) = 0.01$ is fixed (see `inla.doc("pc.prec")` for more details). A log-Gamma($a = 0.01$, $b = 0.01$) prior is specified to the frailty term variance (see `inla.doc("loggamma")` for more details). For joint model specifications, the prior on the multivariate random effects precision matrix is a Wishart(r , Ω^{-1}) where $r = 10$ and Ω is an identity matrix (see `inla.doc("iidkd")` for more details).

Obviously, all default prior distributions can be modified according to user preferences. Indeed, in the call of the `joint` function, priors can be specified through control options, for example, priors for fixed effects can be specified with the `priorFixed` control argument, which is a list with entries: `mean`, `prec`, `mean.intercept` and `prec.intercept` where `mean` and `prec` are the mean and precision (ie, inverse of the variance) of the fixed effects, respectively and `mean.intercept` and `prec.intercept` are the corresponding parameters for the intercept. Default values are mean 0 and precision 0.01 (ie, variance 100). Additionally, the control option `priorRandom` is a list with prior distribution for the multivariate random effects (inverse-Wishart). Default is `list(r=10, R=1)`, see `inla.doc("iidkd")` for more details. Prior distributions for baseline hazards hyperparameters (eg, parameters related to parametric distributions or random walks) can be specified through the `baselineHyper` control argument. Details about the default and available prior distributions can be displayed with the `inla.doc` command, for example `inla.doc("weibullsurv")` for the Weibull baseline hazards priors. While multiple options for each priors are available, there is no limitation for the specification of alternative priors and more distributions can be added upon request. Finally, the function `priors.used` applied to an `INLAjoint` object returns the priors that were used to fit the model.

3 | ACCELERATED FAILURE TIME MODELS

3.1 | Motivation: larynx dataset (from the KMSurv R-package)

The *larynx* dataset contains observations of 90 male larynx-cancer patients, diagnosed and treated in the period 1970 to 1978.³² Each patient has the following variables:

- `stage`: disease stage (1-4).
- `time`: time (in months) from first treatment until death, or end of study.
- `age`: age (in years) at diagnosis of larynx cancer.
- `diagyr`: year of diagnosis of larynx cancer.
- `delta`: death indicator (1: if patient died; 0: otherwise).

The distribution of disease stages was 33 for stage 1, 17 for stage 2, 27 for stage 3, and 13 for stage 4; the observed survival times had 0.1 years as minimum, 10.7 as maximum, and 4 as median, with 44.4% right-censored; ages at diagnosis ranged between 41 and 86 years old with mean and median at 64.6 and 65, respectively; and the years of diagnosis fluctuated between 1970 and 1978, where the highest number of cases (19) was observed in 1976.

3.2 | Model specification

Time to death, T , can be modeled by an accelerated failure time (AFT) model as follows:

$$\log(T) = \beta_1 + \beta_2 I_{(\text{stage}=2)} + \beta_3 I_{(\text{stage}=3)} + \beta_4 I_{(\text{stage}=4)} + \beta_5 \text{age} + \beta_6 \text{diagyr} + \sigma \epsilon, \quad (1)$$

where β_1 is an intercept; $I_{(\text{stage}=\cdot)}$ is an indicator variable for $\text{stage}=2, 3, 4$ with regression coefficients β_2, β_3 , and β_4 , respectively ($\text{stage}=1$ is considered as the reference category); and β_5 and β_6 are regression coefficients for age and *diagyr* covariates, respectively. The errors ϵ 's are independent and identically distributed (i.i.d.) random variables that follow a standard Gumbel distribution, $f(\epsilon) = \exp\{-[\epsilon + \exp(-\epsilon)]\}$, and σ is a scale parameter. Prior distributions are set according to Section 2.3.

3.3 | Model implementation

We start by loading the data, rescaling the continuous variables age and *diagyr*, and defining the categorical variable *stage* as a factor:

```
R> library(KMsurv)
R> data(larynx)
R> larynx$age <- as.numeric(scale(larynx$age))
R> larynx$diagyr <- as.numeric(scale(larynx$diagyr))
R> larynx$stage <- as.factor(larynx$stage)
```

So, we fit a Weibull model with an INLA survival object as the outcome:

```
R> m1.weib <- joint(formSurv = inla.surv(time = time, event = delta) ~ stage + age
+ diagyr, + basRisk = "weibullsurv", dataSurv = larynx, control = list(config=TRUE))
```

However, specification (1) is a Gumbel AFT model. The relationship between the regression coefficients of the Weibull and Gumbel AFT models is just a sign inversion, so we can do this transformation with the *inla.tmarginal* function that allows applying any function to transform the marginals and *inla.zmarginal* that computes summary statistics of marginal posteriors:

```
R> m1.gumb <- sapply(m1.weib$marginals.fixed, function(m)
+ inla.zmarginal(inla.tmarginal(function(x) -x, m)))
```

Rearranging some terms, the posterior distribution summary is given by:

```
R> gumb.scale <- inla.tmarginal(function(x) 1/x, m1.weib$marginals.hyperpar[[1]])
R> rbind("Gumbel scale"=inla.zmarginal(gumb.scale), t(m1.gumb))
```

	mean	sd	quant0.025	quant0.5	quant0.975
Gumbel scale	0.9794	0.0266	0.9198	0.9829	1.0210
Intercept_S1	2.5846	0.2718	2.0506	2.5841	3.1180
stage2_S1	-0.1294	0.4594	-1.0323	-0.1302	0.7719
stage3_S1	-0.6580	0.3523	-1.3505	-0.6586	0.0333
stage4_S1	-1.6866	0.4183	-2.5086	-1.6873	-0.8659
age_S1	-0.2125	0.1532	-0.5135	-0.2127	0.0881
diagyr_S1	0.0738	0.1550	-0.2308	0.0736	0.3779

The scale parameter can be obtained by $\sigma = 1/\alpha$, where α is the Weibull shape parameter.

The marginal posterior distributions are available in the output of the joint function. It is easy to convert those marginals from the Weibull formulation to the Gumbel AFT model by using the *inla.tmarginal* function:

```
R> m1.marg <- append(list("Gumbel scale" = gumb.scale),
+ lapply(m1.weib$marginals.fixed, function(m) inla.tmarginal(function(x) -x, m)))
R> sapply(1:7, function(x) plot(m1.marg[[x]], type = "l", xlab = "", ylab = "",
+ main = paste0("Density of ", names(m1.marg)[x])))
```

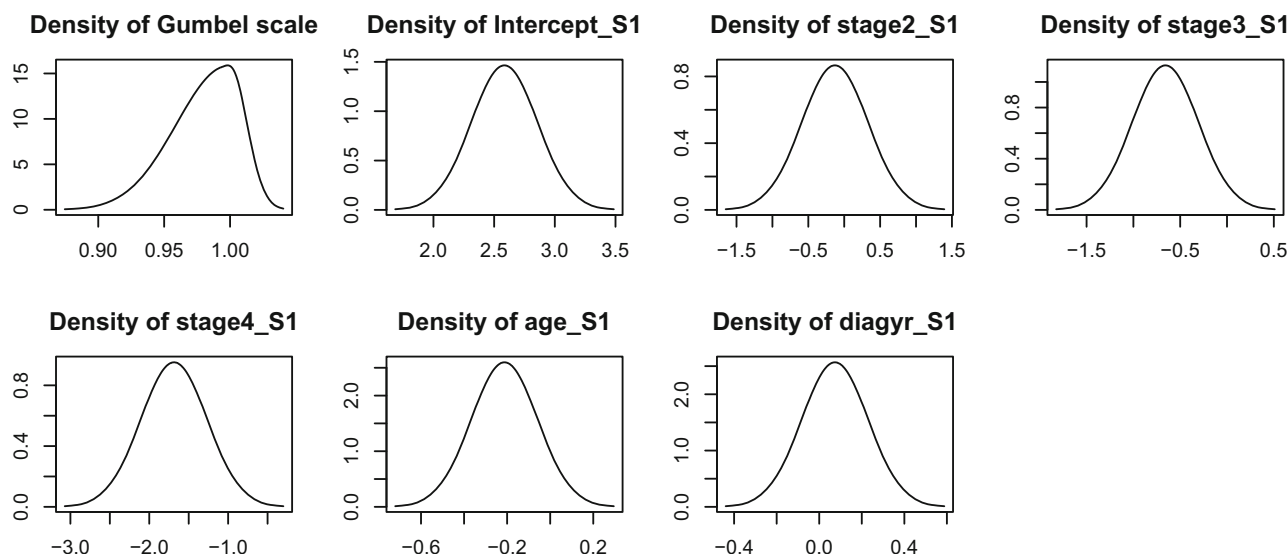


FIGURE 1 Marginal posterior density plots for AFT model parameters.

Figure 1 shows the marginal posterior distributions of the model parameters, which can provide additional insight about the distribution of model parameters compared to the summary that focuses on mean, standard deviation, and quantiles (such as visualization of possible skewness of the distribution). When no transformation is required, one can plot the marginal posterior distributions directly with the `plot()` function, as illustrated for a joint model in Section 9.

Posterior samples of the hyperparameters and the latent field can be generated using the following command:

```
R> m1.sample <- inla.posterior.sample(n, m1.weib)
```

where n indicates the number of samples required. Note that the model needs to save some additional information in order to be able to sample from the posterior, therefore the option “`config = TRUE`” must be added to control options of the `joint()` function when fitting the model.

3.4 | Model interpretation

In the AFT model, the fixed effects β represent the log-linear effect of covariates on the survival time. When positive, it suggests that an increase in a given covariate value is associated with a decreasing risk of event (ie, increases the time to failure) while a negative β value suggests that an increase in a given covariate decreases the time to failure. It is although usually preferred to interpret the multiplicative effect of $\exp(\beta)$ on survival. Because we apply an exponential transformation, which is non-linear, it is important to consider the mean of the exponentiated marginal posterior distribution instead of just exponentiating the mean posterior value. For example, for the parameter β_4 associated with stage 4 of the disease:

```
R> inla.zmarginal(inla.tmarginal(function(x) exp(-x),
                                m1.weib$marginals.fixed[["stage4_S1"]]))
      mean      sd quant0.025 quant0.25 quant0.5 quant0.75 quant0.975
0.2020    0.0878    0.0813    0.1393    0.1850    0.2457    0.4208
```

Having cancer at stage 4 accelerates the time to event by a factor 0.20 [0.08, 0.43] compared to the reference stage 1 (ie, 0.20 times shorter survival time compared to the baseline survival). Note how this is different from simply exponentiating β as $\exp(\hat{\beta}_4) = \exp(-1.6866) = 0.1851$. The location parameter $\exp(\beta_1)$ represents the center of the Gumbel distribution of the survival time for the reference individual. A positive value is associated with longer survival times while a negative value results in shorter survival times. Finally, the interpretation of the Gumbel scale parameter is related to the spread of the baseline survival time distribution, when greater than 1, the survival times are more dispersed. It suggests that

there is heterogeneity in the data with extreme or long-surviving cases. On the other hand, the survival times are more concentrated when this parameter is less than 1, meaning that survival times are more similar or clustered around a central value.

4 | PROPORTIONAL HAZARDS MODELS

4.1 | Model specification

We will again use the *larynx* dataset (see Section 3.1). In a proportional hazard (PH) approach, the term to be modeled is the hazard function at time t :

$$h(t|\lambda, \beta) = h_0(t|\lambda) \exp \{ \beta_2 1_{(\text{stage}=2)} + \beta_3 1_{(\text{stage}=3)} + \beta_4 1_{(\text{stage}=4)} + \beta_5 \text{age} + \beta_6 \text{diagyr} \}, \quad t > 0, \quad (2)$$

where $1_{(\text{stage}=)}$ is an indicator variable for $\text{stage}=2, 3, 4$ with regression coefficients β_2, β_3 , and β_4 , respectively ($\text{stage}=1$ is considered as the reference category); and β_5 and β_6 are regression coefficients for age and diagyr covariates, respectively. Here, the baseline hazard function is set as a mixture of piecewise constant functions, $h_0(t|\lambda) = \sum_{k=1}^K \lambda_k I_{(a_{k-1}, a_k]}(t)$, where $I_{(a_{k-1}, a_k]}(t)$ is the indicator function defined as 1 when $t \in (a_{k-1}, a_k]$ and 0 otherwise. In this application, we use $K = 3$ knots and an equally-spaced partition of the time axis, $a_0 = 0, a_1 = 3.57, a_2 = 7.13$, and $a_3 = 10.70$. Prior distributions are set according to Section 2.3.

4.2 | Model implementation

After loading and transforming the data as in Section 3.3, we fit the PH model with an INLA survival object outcome:

```
R> m2.pwc <- joint(formSurv = inla.surv(time = time, event = delta) ~ stage
+ age + diagyr, + basRisk = "rw1", NbasRisk = 3, dataSurv = larynx)
```

Note that the baseline hazard function (“basRisk”) is specified as a random walk of order one prior (“rw1”),¹⁰ which is based on first-order differences, that is, a mixture of piecewise constant functions, where “NbasRisk” represents the number of knots. The posterior distribution summary for regression coefficients and baseline risk function variance is given by:

```
R> summary(m2.pwc)
Survival outcome
```

	mean	sd	0.025quant	0.5quant	0.975quant
Baseline risk (variance)	0.0228	0.0539	0.0001	0.0058	0.1576
stage2	0.1269	0.4613	-0.7777	0.1269	1.0314
stage3	0.6530	0.3535	-0.0401	0.6530	1.3461
stage4	1.6954	0.4222	0.8677	1.6953	2.5236
age	0.2133	0.1541	-0.0888	0.2133	0.5154
diagyr	-0.0660	0.1571	-0.3740	-0.0661	0.2423

```
log marginal-likelihood (integration)      log marginal-likelihood (Gaussian)
                                -161.5613                                -161.5613

Deviance Information Criterion: 297.5805
Widely applicable Bayesian information criterion: 299.4084
Computation time: 0.66 seconds
```

One can also get the posterior distribution summary for λ 's:

```
R> summary(m2.pwc)$BaselineValues
```

```

      time lower median upper
[1,]  0.0000 0.0436 0.0750 0.1286
[2,]  3.5667 0.0462 0.0809 0.1434
[3,]  7.1333 0.0422 0.0799 0.1560
[4,] 10.7000 0.0383 0.0798 0.1747

```

4.3 | Model interpretation

While in an AFT model the covariates act multiplicatively on time, in a PH model the covariates act multiplicatively on the hazard. Therefore, β parameters represent the log hazard ratio associated with a one-unit change in the associated covariate. As for the AFT model, it is more common to exponentiate the β parameters to obtain hazard ratios. This can easily be done with `INLAjoint` by adding the argument “`hr = TRUE`” to the call of the summary or plot function, so the summary statistics are given on the exponential scale (note that baseline parameters are not transformed, as it is not meaningful):

```

R> summary(m2.pwc, hr = TRUE)
Survival outcome

              exp(mean)      sd 0.025quant 0.5quant 0.975quant
Baseline risk (variance)  0.0228 0.0539      0.0001  0.0058      0.1576
stage2                   1.2582 0.5946      0.4638  1.1342      2.7708
stage3                   2.0410 0.7272      0.9681  1.9201      3.8065
stage4                   5.9393 2.5537      2.4024  5.4439     12.3344
age                     1.2521 0.1908      0.9183  1.2374      1.6674
diagyr                   0.9474 0.1473      0.6905  0.9358      1.2688

```

An individual with stage 4 cancer has a hazard 5.94 [2.40, 12.34] times higher than an individual with stage 1, meaning they are nearly six times more likely to die, at any given point in time. The baseline risk represents the risk for the reference individual, here it is assumed piecewise constant with 3 intervals (to match the example previously proposed in Alvares et al¹⁷) but the default number of intervals in `INLAjoint` is 15. Having more intervals leads to more flexibility but can overfit the data. An alternative approach consists in using random walks of order 2, which can be seen as Bayesian smooth splines, to have flexible curves while avoiding parametric assumptions and overfitting, see Section 10 for more details.

4.4 | Stratified proportional hazards model

In some situations, the proportional hazards assumption may not hold for some covariates, meaning that the hazard ratio may not be constant in time. In this context, it is possible to stratify the model so that instead of having a proportional effect of a covariate on the risk, we can have a distinct baseline hazard function for each group defined by this covariate. To illustrate this feature in `INLAjoint`, let's assume the hazards are not proportional conditional on the stage of cancer in the model described in Section 4.1. The model is then defined as:

$$h(t|\lambda, \beta) = h_0^{stage_k}(t|\lambda) \exp\{\beta_1 \text{age} + \beta_2 \text{diagyr}\}, \quad t > 0,$$

where $h_0^{stage_k}(t|\lambda)$ corresponds to the baseline hazard function for stage k . The corresponding R code to fit this model then simply includes the covariate “stage” as a strata in control options instead of having it directly in the formula:

```

R> m2b.pwc <- joint(formSurv = inla.surv(time = time, event = delta) ~ age + diagyr,
+                   dataSurv = larynx, control = list(strata=list("stage")))

```

The plot function allows to display the baseline hazard function for each strata:

```

R> plot(m2b.pwc)

```

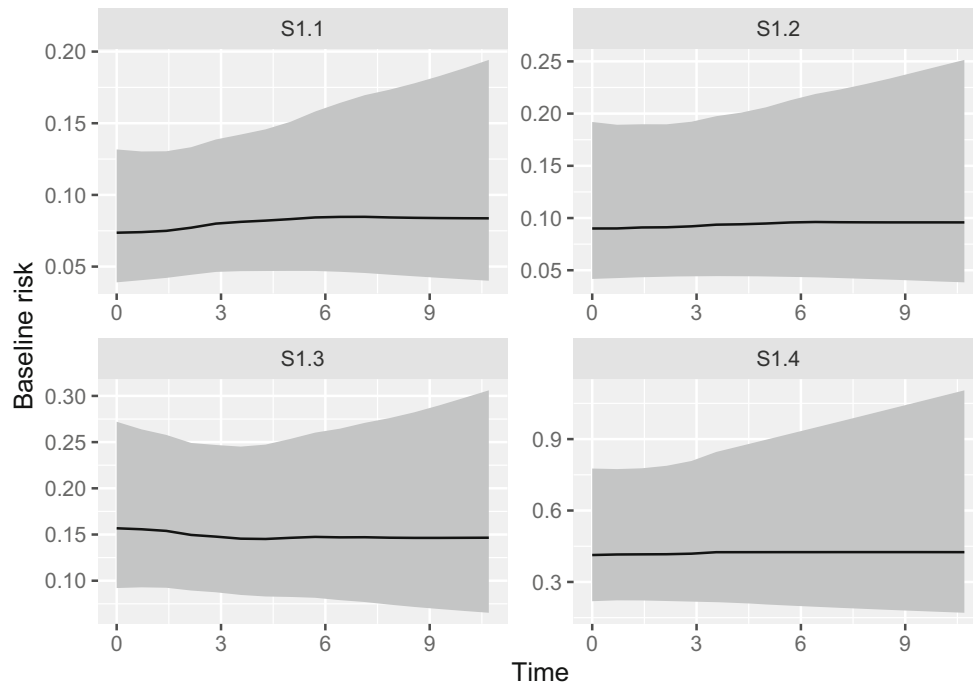


FIGURE 2 Baseline hazard function of each of the 4 stages of cancer for the stratified proportional hazards model.

Figure 2 shows the stata-specific baseline hazard functions. Here, we can see that although the Y-axis scale is different for each baseline hazard function (ie, higher stage of cancer corresponds to more advanced disease and thus higher risk of death), there is no evidence for non-proportional hazards conditional on the stage covariate and the results are in line with results from the non-stratified proportional hazards model. Allowing for non-proportional hazards conditional on the cancer stage does not seem to provide benefit here but when baseline hazard curves are not parallel, it would indicate non-proportionality conditional on the strata covariate.

5 | MIXTURE CURE MODELS

5.1 | Motivation: bmt dataset (from the `smcure` R-package)

The *bmt* dataset contains 91 patients with refractory acute lymphoblastic leukemia who participated in a bone marrow transplant study.³³ Each patient has the following variables:

- Time: time to death (in days).
- Status: censoring indicator (1: if patient is uncensored; 0: otherwise).
- TRT: treatment group indicator (1: autologous; 0: allogeneic).

Of the 91 patients, 46 received allogeneic transplant (treatment); and the observed survival times had 11 days as minimum, 1845 as maximum, and 179 as median, with 24.2% right-censored.

5.2 | Model specification

Let Z be a binary random variable defined as 0 for susceptible (ie, $Z = 1 - \text{Status}$) and 1 for cured or immune individuals. So, the incidence submodel is given by:

$$Z \sim \text{Bernoulli}(\eta), \quad (3)$$

$$\text{logit}(\eta) = \beta_{C1} + \beta_{C2}\text{TRT}. \quad (4)$$

where β_{C1} represents an intercept and β_{C2} is the regression coefficient for the TRT covariate. The latency submodel is expressed through a PH specification:

$$h(t|\lambda, \alpha, \beta_U) = h_0(t|\lambda, \alpha) \exp\{\beta_U \text{TRT}\}, \quad t > 0, \quad (5)$$

with $h_0(t|\lambda, \alpha) = \lambda \alpha t^{\alpha-1}$ specified as a Weibull baseline hazard function, where λ and α are the scale and shape parameters, respectively; and β_U is the regression coefficient for the TRT covariate. Prior distributions are set according to Section 2.3.

5.3 | Model implementation

We start by loading the data:

```
R> library(smcure)
R> data(bmt)
```

So, we setup an INLA survival object indicating that the incidence submodel (“cure”) contains an intercept and the TRT covariate and fit the mixture cure model:

```
R> m3.cure <- joint(formSurv = inla.surv(time = Time, event = Status,
+                                     cure = cbind("Int"=1, "TRT"=TRT)) ~ TRT,
+                  basRisk = "weibullsurv", dataSurv = bmt)
```

The posterior distribution summary is given by:

```
R> summary(m3.cure)
Survival outcome
```

	mean	sd	0.025quant	0.5quant	0.975quant
Int(cure)	-1.0019	0.3220	-1.6355	-1.0020	-0.3677
TRT(cure)	-0.3966	0.4555	-1.2996	-0.3945	0.4938
Weibull (shape)	1.0921	0.0883	0.9414	1.0847	1.2877
Weibull (scale)	0.0027	0.0009	0.0013	0.0026	0.0046
TRT	0.7028	0.2556	0.2014	0.7030	1.2038

```
log marginal-likelihood (integration)      log marginal-likelihood (Gaussian)
                                -487.1486                                -487.1486
```

Deviance Information Criterion: 954.4586
 Widely applicable Bayesian information criterion: 955.6154
 Computation time: 0.67 seconds

5.4 | Model interpretation

In order to interpret the model, we first need to compute the cure fraction of the model, with `INLAjoint` the parameters of the incidence submodel are considered as hyperparameters and we can easily sample them with the function `inla.hyperpar.sample`:

```
R> smp.cure <- inla.hyperpar.sample(500, m3.cure)[, 2:3]
R> quantile(inv.logit(smp.cure[,1]), c(0.025, 0.5, 0.975)) # allogeneic
 2.5%    50%   97.5%
0.1595 0.2641 0.4075
R> quantile(inv.logit(rowSums(smp.cure)), c(0.025, 0.5, 0.975)) # autologous
 2.5%    50%   97.5%
0.1045 0.1970 0.3272
```

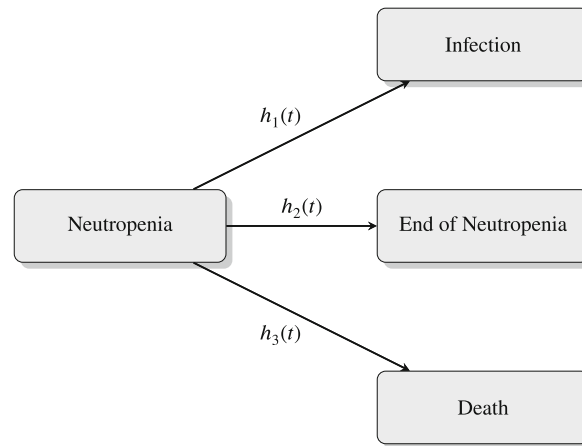


FIGURE 3 Graphical representation of competing risks for *okiss* dataset. All the patients enter the follow-up with neutropenia and are at risk to have an infection, the end of neutropenia or death if they are not censored before observing one of these three competing events.

Therefore the proportion of the population that is “cured” and will not experience the event is 26% [15%, 41%] for the treatment group allogeneic and 20% [10%, 33%] for the treatment group autologous. For the remaining patients who are not considered cured, we can estimate the hazard ratio of treatment:

```

R> summary(m3.cure, hr=T)$SurvEff[[1]]["TRT_S1",]
      exp(mean)      sd 0.025quant 0.5quant 0.975quant
TRT_S1      2.084 0.5310      1.2302  2.0189  3.3099
  
```

In the non-cured fraction of patients, those who received autologous treatment have a hazard of experiencing the event 2.08 [1.23, 3.31] times higher than those who received allogeneic treatment, indicating a significantly greater risk associated with the autologous treatment option.

6 | COMPETING RISKS MODELS

6.1 | Motivation: okiss dataset (from the *compeir* R-package)

The *okiss* dataset contains a random subsample of 1000 patients from ONKO-KISS, which is part of the surveillance program of the German National Reference Centre for Surveillance of Hospital-Acquired Infections.³⁴ After a peripheral blood stem-cell transplantation, these patients are neutropenic (low number of white blood cells). Each patient has the following variables:

- **time**: time (in days) of neutropenia until first event.
- **status**: event status indicator (1: infection; 2: end of neutropenia; 7: death; 11: censored observation).
- **allo**: transplant type indicator (1: allogeneic; 0: autologous).
- **sex**: sex of each patient (m: if patient is male; f: if patient is female).

Of the 1000 patients, 564 received allogeneic transplant; 61.9% were male; and the median survival times per event were 6 (infection), 12 (end of neutropenia), and 15 (death), with 13% right-censored. Figure 3 illustrates the competitive risks transition structure for *okiss* dataset.

6.2 | Model specification

Cause-specific hazard functions for infection ($k = 1$), end of neutropenia ($k = 2$), and death ($k = 3$) are modeled from a PH specification:

$$h_k(t|\lambda_1, \alpha_2, \beta_1) = h_{0k}(t|\lambda_1, \alpha_2) \exp\{\beta_{11}\text{allo} + \beta_{21}\text{sex}\}, \quad t > 0, \quad (6)$$

$$h_2(t|\lambda_2, \alpha_2, \beta_2) = h_{02}(t|\lambda_2, \alpha_2) \exp\{\beta_{12}\text{allo} + \beta_{22}\text{sex}\}, \quad t > 0, \quad (7)$$

$$h_3(t|\lambda_3, \alpha_3, \beta_3) = h_{03}(t|\lambda_3, \alpha_3) \exp\{\beta_{13}\text{allo} + \beta_{23}\text{sex}\}, \quad t > 0, \quad (8)$$

with $h_{0k}(t|\lambda_k, \alpha_k) = \lambda_k \alpha_k t^{\alpha_k-1}$ for event k specified as a Weibull baseline hazard function, where λ_k and α_k are the scale and shape parameters, respectively; and $\beta_k = (\beta_{1k}, \beta_{2k})^\top$ are regression coefficients for the `allo` and `sex` covariates, respectively, for $k = 1, 2, 3$. Prior distributions are set according to Section 2.3.

6.3 | Model implementation

We start by loading the data and creating an auxiliary variable (`delta`) from `status`:

```
R> library(compeir)
R> data(okiss)
R> delta <- matrix(c(as.integer(okiss$status == 1),
+                   as.integer(okiss$status == 2),
+                   as.integer(okiss$status == 7)), ncol = 3)
R> head(delta)
      [,1] [,2] [,3]
[1,]    0    1    0
[2,]    1    0    0
[3,]    0    1    0
[4,]    0    1    0
[5,]    0    1    0
[6,]    0    1    0
```

In this auxiliary variable, the events 1 (infection), 2 (end of neutropenia), or 7 (death) are indicated with a value of 1 in columns 1, 2, or 3, respectively, and a row with only 0's represents a censored observation.

The next step is to setup an INLA survival object for each hazard function and fit the competing risks model:

```
R> m4.cr <- joint(formSurv = list(inla.surv(time = time, event = delta[,1]) ~ allo
+                               sex, + nla.surv(time = time, event = delta[,2]) ~ allo + sex,
+                               inla.surv(time = time, event = delta[,3]) ~ allo + sex),
+               basRisk = rep("weibullsurv", 3), dataSurv = okiss)
```

Note that the regressors and the baseline hazard of each hazard function are set independently. Finally, the posterior distribution summary is given by:

```
R> summary(m4.cr)
Survival outcome (S1)
              mean      sd 0.025quant 0.5quant 0.975quant
Weibull (shape)_{S}1  1.1030 0.0638    0.9829   1.1011    1.2339
Weibull (scale)_{S}1  0.0156 0.0026    0.0111   0.0154    0.0213
allo_{S}1            -0.5032 0.1443   -0.7861  -0.5032   -0.2202
sexm_{S}1            0.1547 0.1477   -0.1349   0.1547    0.4443

Survival outcome (S2)
              mean      sd 0.025quant 0.5quant 0.975quant
Weibull (shape)_{S}2  2.0278 0.0445    1.9414   2.0273    2.1166
Weibull (scale)_{S}2  0.0077 0.0007    0.0064   0.0077    0.0092
allo_{S}2            -1.1919 0.0734   -1.3357  -1.1919   -1.0480
sexm_{S}2            -0.1009 0.0737   -0.2453  -0.1009    0.0436

Survival outcome (S3)
              mean      sd 0.025quant 0.5quant 0.975quant
```


Weibull (shape)_{S}3	2.2053	0.2848	1.6759	2.1946	2.7939
Weibull (scale)_{S}3	0.0000	0.0000	0.0000	0.0000	0.0001
allo_{S}3	-0.4756	0.6729	-1.7950	-0.4756	0.8440
sexm_{S}3	0.4677	0.6672	-0.8407	0.4677	1.7760

log marginal-likelihood (integration)	log marginal-likelihood (Gaussian)
-3735.786	-3735.786

Deviance Information Criterion: 7409.673

Widely applicable Bayesian information criterion: 7421.033

Computation time: 1.1 seconds

6.4 | Model interpretation

For each submodel, the Weibull distribution describes how the hazard of the event of interest changes with time. For example the Weibull shape parameter for the death event, $\hat{\alpha}_3 = 2.21$ [1.67, 2.80], suggests that the hazard of death increases over time while the hazard of infection is more stable ($\hat{\alpha}_1 = 1.10$ [0.98, 1.24]). The effect of covariates is straightforward, for example, the hazard ratio of allogeneic versus autologous transplant type for the risk of infection is obtained as follows:

```
R> summary(m4.cr, hr=T)$SurvEff[[1]]["allo_S1",]
      exp(mean)      sd 0.025quant 0.5quant 0.975quant
allo_S1      0.6107 0.0871      0.4571  0.6045      0.7992
```

This suggests that patients receiving allogeneic transplants have a 39% [20%, 55%] lower risk of infection compared to patients receiving autologous transplants, accounting for other competing risks. We can also plot the *cumulative incidence function* (CIF), which describes the cumulative probability of experiencing a specific event of interest over time while accounting for the presence of other competing events. For example for the reference individual (ie, female with autologous transplant type):

```
R> riskW <- function(t, lambda, alpha) lambda*alpha*t^(alpha-1)
R> t <- seq(0, 100, len=100)
R> risk1 <- riskW(t, exp(m4.cr$summary.fixed["Intercept_{S}1", "mean"]),
R>           m4.cr$summary.hyperpar$mean[1])
R> risk2 <- riskW(t, exp(m4.cr$summary.fixed["Intercept_{S}2", "mean"]),
R>           m4.cr$summary.hyperpar$mean[2])
R> risk3 <- riskW(t, exp(m4.cr$summary.fixed["Intercept_{S}3", "mean"]),
R>           m4.cr$summary.hyperpar$mean[3])
R> surv <- exp(-cumsum(risk1) - cumsum(risk2) - cumsum(risk3))
R> CIF <- cbind(cumsum(risk1*surv), cumsum(risk2*surv), cumsum(risk3*surv))
```

CIFs are displayed in Figure 4, showing a higher probability of experiencing the end of neutropenia compared to other competing events over time. Note that it is possible to add uncertainty by sampling parameters as described in Section 3.3. Note that CIFs are preferred over survival curves in the context of competing risks as survival curves would represent the probability of having an event in a hypothetical world where it is not possible to have any of the competing events.

7 | MULTI-STATE MODELS

7.1 | Motivation: heart2 dataset (from the p3state.msm R-package)

The *heart2* dataset contains a sample of 103 patients of the Stanford Heart Transplant Program.³⁵ The patients are on the waiting list (initial state) and can either be transplanted (non-terminal event) and then die (terminal event), or just one or none of them because they continue to be on the waiting list. Each patient has the following variables:

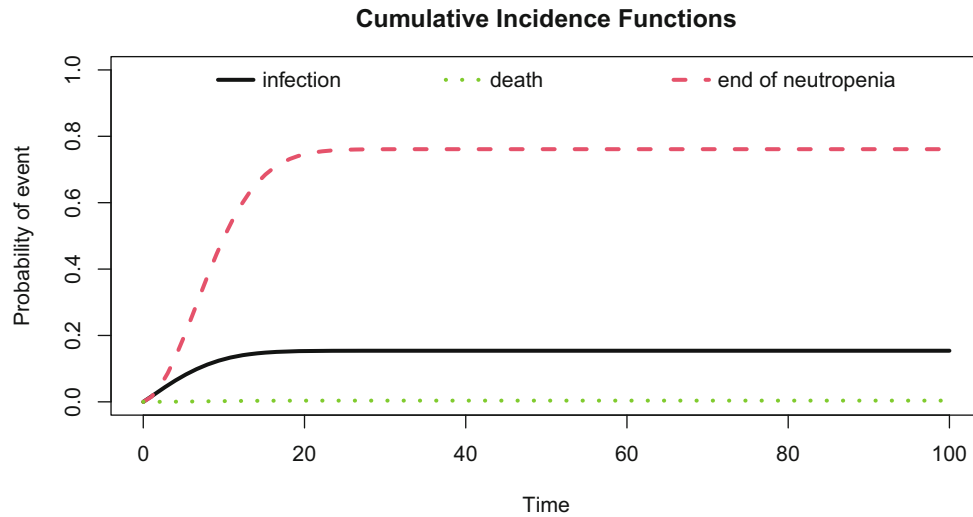


FIGURE 4 Cumulative incidence functions of the reference individual (ie, female with autologous transplant type) for the competing risks model.

- times1: time of transplant/censoring time.
- delta: transplant indicator (1: yes; 0: no).
- times2: time to death since the transplant/censoring time.
- time: times1 + times2.
- status: censoring indicator (1: dead; 0: alive).
- age: age-48 years.
- year: year of acceptance (in years after 1 Nov 1967).
- surgery: prior bypass surgery (1: yes; 0: no).

The patients' ages (minus 48) were distributed between -39.2 and 16.4 with a mean of -2.8 and a standard deviation of 9.8 ; the year of acceptance ranged approximately uniformly from -0.05 to 6.47 ; and 84.5% of patients had not undergone prior bypass surgeries. In addition, 30 patients died before transplantation, 69 were transplanted, of which 45 died after transplantation, and only 4 patients did not experience either event (censored). Figure 5 illustrates the multi-state transition structure for *heart2* dataset.

7.2 | Model specification

Hazard functions from waiting list to transplant (h_1), from waiting list to death (h_2), and from waiting list to death after transplant (h_3) are modeled from a PH specification:

$$h_1(t|\lambda_1, \alpha_1, \beta_1) = h_{01}(t|\lambda_1, \alpha_1) \exp \{ \beta_{12} \text{age} + \beta_{13} \text{year} + \beta_{14} \text{surgery} \}, \quad t > 0, \quad (9)$$

$$h_2(t|\lambda_2, \alpha_2, \beta_2) = h_{02}(t|\lambda_2, \alpha_2) \exp \{ \beta_{22} \text{age} + \beta_{23} \text{year} + \beta_{24} \text{surgery} \}, \quad t > 0, \quad (10)$$

$$h_3(t|\lambda_3, \alpha_3, \beta_3, T_1 = t_1) = h_{03}(t - t_1|\lambda_3, \alpha_3) \exp \{ \beta_{32} \text{age} + \beta_{33} \text{year} + \beta_{34} \text{surgery} \}, \quad t > t_1, \quad (11)$$

with $h_{0k}(t|\lambda_k, \alpha_k) = \lambda_k \alpha_k t^{\alpha_k - 1}$ specified as a Weibull baseline hazard function, where λ_k and α_k are the scale and shape parameters, respectively; and $\beta_k = (\beta_{1k}, \beta_{2k}, \beta_{3k})^\top$ are regression coefficients for the age, year and surgery covariates, respectively, for $k = 1, 2, 3$. Note that we adopted a semi-Markovian specification in (11).³⁶ In this case, the model can also

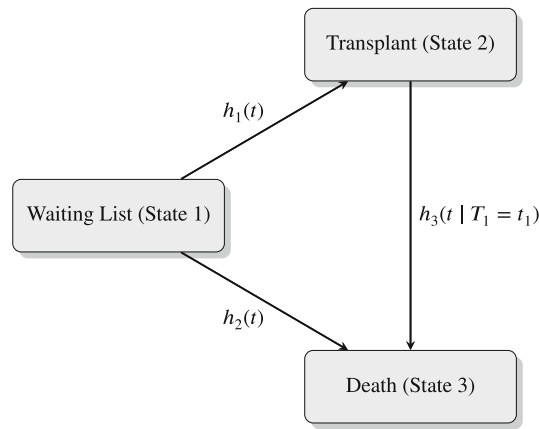


FIGURE 5 Graphical representation of (illness-death) multi-state for *heart2* dataset. Patients enter the follow-up while being on the waiting list for transplantation. They can either receive transplantation and die after their transplantation or directly die while still being on the waiting list. Additionally, censoring can occur at anytime during the follow-up.

be seen as a clock-reset specification,³ in which time starts at zero again after each transition. Prior distributions are set according to Section 2.3.

7.3 | Model implementation

We start by loading the data, creating an auxiliary variable (*event*) from *status*, and setting the time from transplant to death or censoring:

```

R> library(p3state.msm)
R> data(heart2)
R> event <- matrix(c(heart2$delta, heart2$status * (1 - heart2$delta),
+                   heart2$delta * heart2$status), ncol = 3)
R> head(event)
      [,1] [,2] [,3]
[1,]    0    1    0
[2,]    0    1    0
[3,]    1    0    1
[4,]    1    0    1
[5,]    0    1    0
[6,]    0    1    0
R> heart2$times3 <- ifelse(heart2$times2 == 0, heart2$times1, heart2$times2)

```

Each column in this auxiliary variable represents the transition of a patient. For example, the first row indicates that the patient has passed from waiting list to death, and the third row indicates that the patient has passed from waiting list to transplant and after from transplant to death. A row with only 0's represents a censored observation from waiting list.

Similar to the competing risks model, the next step is to set an INLA survival object for each hazard function and fit the multi-state model:

```

R> m5.ms <- joint(formSurv = list(inla.surv(times1, event[,1]) ~ age + year
+                               surgery, + inla.surv(times3, event[,2]) ~ age + year + surgery,
+                               inla.surv(times2, event[,3]) ~ age + year + surgery),
+               basRisk = rep("weibullsurv", 3), dataSurv = heart2)

```

Finally, the posterior distribution summary is given by:

```

R> summary(m5.ms)

```

Survival outcome (S1)

	mean	sd	0.025quant	0.5quant	0.975quant
Weibull (shape)_S1	0.8174	0.0592	0.7062	0.8156	0.9393
Weibull (scale)_S1	0.0349	0.0105	0.0186	0.0334	0.0596
age_S1	0.0508	0.0140	0.0235	0.0508	0.0782
year_S1	-0.0001	0.0686	-0.1346	-0.0001	0.1345
surgery_S1	0.2301	0.3141	-0.3857	0.2301	0.8460

Survival outcome (S2)

	mean	sd	0.025quant	0.5quant	0.975quant
Weibull (shape)_S2	0.4947	0.0399	0.4202	0.4933	0.5774
Weibull (scale)_S2	0.0535	0.0201	0.0243	0.0500	0.1026
age_S2	0.0038	0.0179	-0.0313	0.0038	0.0390
year_S2	-0.2226	0.1144	-0.4469	-0.2226	0.0017
surgery_S2	-0.9450	0.6205	-2.1617	-0.9450	0.2718

Survival outcome (S3)

	mean	sd	0.025quant	0.5quant	0.975quant
Weibull (shape)_S3	0.6699	0.0630	0.5522	0.6677	0.7998
Weibull (scale)_S3	0.0193	0.0077	0.0083	0.0179	0.0382
age_S3	0.0590	0.0215	0.0170	0.0590	0.1012
year_S3	0.0085	0.0913	-0.1705	0.0085	0.1876
surgery_S3	-1.0341	0.4447	-1.9060	-1.0341	-0.1621

log marginal-likelihood (integration)	log marginal-likelihood (Gaussian)
-925.9973	-925.9973

Deviance Information Criterion: 1763.492

Widely applicable Bayesian information criterion: 1769.848

Computation time: 0.87 seconds

Alternatively, one can change the transition specification $2 \rightarrow 3$ (h_3) in order to enter the at-risk group after the transition $1 \rightarrow 2$ (no clock-reset) as follows:

```
R> surv.obj.ms3 <- inla.surv(time = heart2$time, truncation = heart2$times1,
  event = event[,3])
```

Note that now the INLA survival object for hazard function h_3 includes the "truncation" argument, which indicates the transition time from waiting list to transplant.

7.4 | Model interpretation

In the context of multi-state models with time-dependent transition probabilities, it is often preferred to employ graphical representations to illustrate the evolution of these transition probabilities over time. It is possible to compute the transition probabilities from the coefficients of the survival submodels fitted with `INLAjoint`. For example, we compute the probability of each transition conditional on covariate "surgery" as follows (we consider the reference individual and assume a transition to state 2 occurred at a time 0 for the intensities p_{22} and p_{23} for simplicity):

```
R> t <- seq(0.1, 1000, by = 1)
R> riskW <- function(t, lambda, alpha) lambda*alpha*t^(alpha-1)
R> risk1 <- riskW(t, exp(m5.ms$summary.fixed["Intercept_{S}1", "mean"]),
R>           m5.ms$summary.hyperpar$mean[1])
R> risk2 <- riskW(t, exp(m5.ms$summary.fixed["Intercept_S2", "mean"]),
```

```

R> m5.ms$summary.hyperpar$mean[2])
R> risk3 <- riskW(t, exp(m5.ms$summary.fixed["Intercept_{S}3", "mean"]),
R> m5.ms$summary.hyperpar$mean[3])
R> risk1s <- riskW(t, exp(m5.ms$summary.fixed["Intercept_S1", "mean"] +
R> m5.ms$summary.fixed["surgery_{S}1", "mean"]),
R> m5.ms$summary.hyperpar$mean[1])
R> risk2s <- riskW(t, exp(m5.ms$summary.fixed["Intercept_S2", "mean"] +
R> m5.ms$summary.fixed["surgery_{S}2", "mean"]),
R> m5.ms$summary.hyperpar$mean[2])
R> risk3s <- riskW(t, exp(m5.ms$summary.fixed["Intercept_S3", "mean"] +
R> m5.ms$summary.fixed["surgery_{S}3", "mean"]),
R> m5.ms$summary.hyperpar$mean[3])
R> p11 <- exp(-cumsum(risk1)-cumsum(risk2))
R> p11s <- exp(-cumsum(risk1s)-cumsum(risk2s))
R> p22 <- exp(-cumsum(risk3))
R> p22s <- exp(-cumsum(risk3s))
R> p12 <- cumsum(p11*risk1*p22)
R> p12s <- cumsum(p11s*risk1s*p22s)
R> p13 <- 1-p11-p12
R> p13s <- 1-p11s-p12s
R> p23 <- 1-p22
R> p23s <- 1-p22s

```

We plotted each transition intensity conditional on prior surgery in Figure 6. We can read from this plot that for reference values of age (48 years old) and year of acceptance (1967), patients who received prior bypass surgery are more likely to get transplantation and less likely to experience death before transplantation compared to patients without prior surgery. For patients who did receive transplantation, those who received prior bypass surgery are also less likely to die over time compared to those who did not.

8 | FRAILTY MODELS

8.1 | Motivation: Kidney dataset (from the frailtyHL R-package)

The kidney dataset contains 38 kidney patients using a portable dialysis machine.³⁷ Times up to the first and second infections are recorded (ie, each patient has two times-to-infection). Infections can occur at the location of insertion of the catheter. When this is the case, then the catheter is later removed, but it can also be removed for other reasons (censored observation). Each patient has the following variables:

- `id`: patient number.
- `time`: time (in days) from insertion of the catheter to infection in kidney patients using portable dialysis machine.
- `status`: censoring indicator (1: if patient is uncensored; 0: otherwise).
- `age`: age (in years) of each patient.
- `sex`: sex of each patient (1: if patient is male; 2: if patient is female).
- `disease`: disease type (GN; AN; PKD; Other).
- `frail`: frailty estimate from original paper.
- `GN`: indicator for disease type GN.
- `AN`: indicator for disease type AN.
- `PKD`: indicator for disease type PKD.

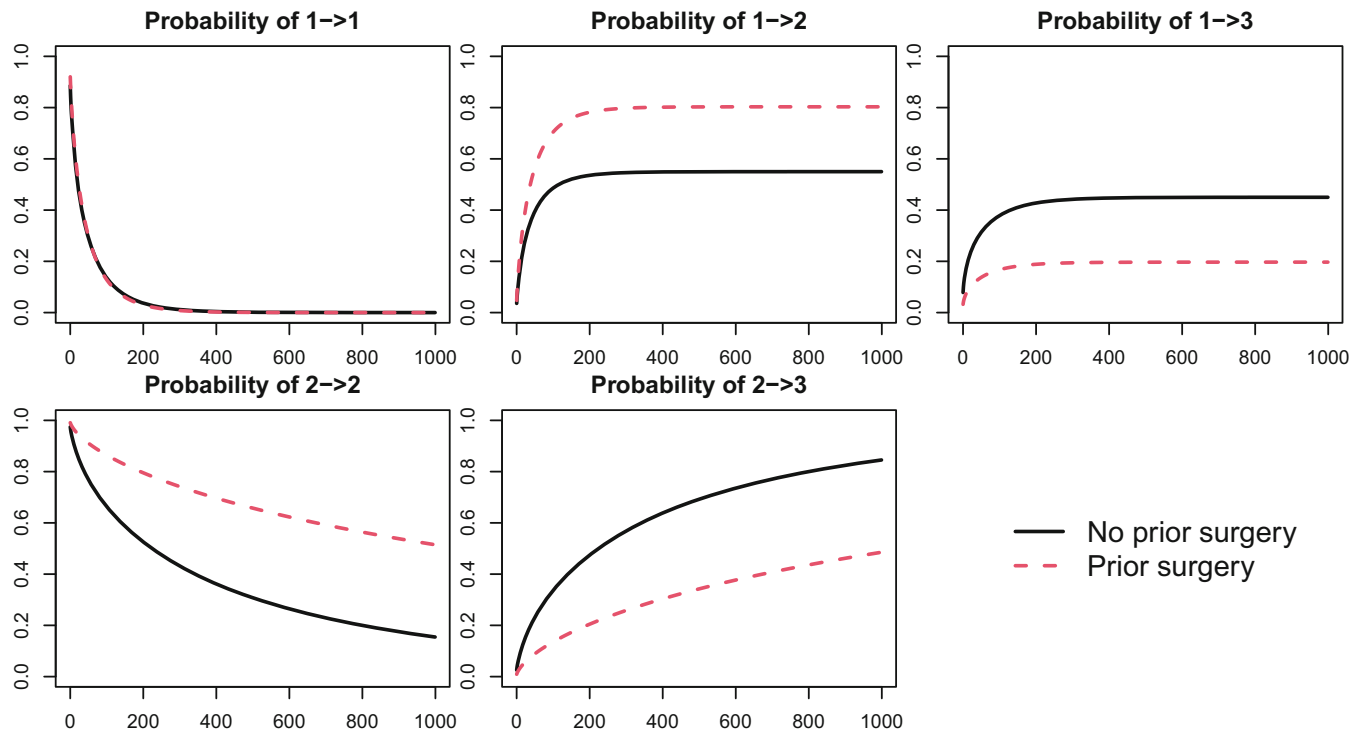


FIGURE 6 Posterior mean of the probability transitions of the multi-state model conditional on prior bypass surgery (assuming the reference individual and transition to state 2 occurred a time 0 for the intensities p_{22} and p_{23} for simplicity). The transition probabilities evolve over time but always sum to 1 for a given initial state at a given time point. Uncertainty over these curves can be obtained through sampling.

Of the 38 patients, 28 were female; and the median times until the first and second infection were 46 and 39 days, respectively, with 9 right-censored observations. The description of the other variables was omitted because they will not be used in the modeling.

8.2 | Model specification

Similar to the multi-state model, we will use the clock-reset specification, in which time starts at zero again after the first infection and the term frailty is what connects the two times-to-infection of the same patient. From a PH approach, the hazard function for the frailty model is described as follows:

$$h(t|\lambda, \alpha, \beta, b_i) = h_0(t|\lambda, \alpha) \exp\{\beta \text{sex} + b_i\}, \quad t > 0, \quad (12)$$

with $h_0(t|\lambda, \alpha) = \lambda \alpha t^{\alpha-1}$ specified as a Weibull baseline hazard function, where λ and α are the scale and shape parameters, respectively; and β is the regression coefficient for the `sex` covariate. Note that we used an additive frailty specification, where b_i 's are assumed as normally distributed with zero mean and unknown variance. For simplicity, we do not use the other covariates. Prior distributions are set according to Section 2.3.

8.3 | Model implementation

We start by loading the data and resetting the binary variable `sex` to a 0-1 format:

```
R> library(frailtyHL)
R> data(kidney)
R> kidney$sex <- kidney$sex - 1 # 0: male (reference)
```


So, we setup an INLA survival object and fit the frailty model:

```
R> m6.frlt <- joint(formSurv = inla.surv(time = time, event = status) ~ sex
+ (1 | id), + basRisk = "weibullsurv", id = "id", dataSurv = kidney)
```

Note that the term $(1 | id)$ specifies that frailty is an intercept random effect by *id* (ie, patient-specific). Finally, the posterior distribution summary is given by:

```
R> summary(m6.frlt)
Survival outcome
              mean      sd 0.025quant 0.5quant 0.975quant
Weibull (shape)  1.0720 0.0644      0.9591  1.0675      1.2119
Weibull (scale)  0.0205 0.0076      0.0093  0.0192      0.0391
sex              -1.4442 0.3980     -2.2321 -1.4418     -0.6695

Frailty term variance
              mean      sd 0.025quant 0.5quant 0.975quant
IDIntercept  0.4531 0.1932      0.1793  0.4191      0.9295

log marginal-likelihood (integration)    log marginal-likelihood (Gaussian)
                                   -343.211                                -343.211

Deviance Information Criterion: 665.3281
Widely applicable Bayesian information criterion: 666.2153
Computation time: 0.65 seconds
```

8.4 | Model interpretation

The covariate *sex* was found to have a statistically significant effect on the time to infections, the hazard ratio can be computed as follows:

```
R> summary(m6.frlt, hr=T)$SurvEff[[1]][ "sex_S1", ]
              exp(mean)      sd 0.025quant 0.5quant 0.975quant
sex_S1         0.2547 0.1022      0.1082  0.2363      0.5066
```

This suggests that females are associated with a 75% [49%, 90%] reduced susceptibility to infections when accounting for unobserved or latent individual-specific factors captured by the frailty term.

9 | JOINT MODELS OF LONGITUDINAL AND SURVIVAL DATA

9.1 | Motivation: prothro and prothros datasets (from the JMbayes R-package)

The *prothro* (longitudinal format) and *prothros* (survival format) datasets contain 488 liver cirrhosis patients from a placebo-controlled randomized trial, where the longitudinal observations of a biomarker (prothrombin) are recorded.³⁸ Each patient has the following variables:

- *id*: patient number.
- *pro*: prothrombin measurements.
- *time*: time points at which the prothrombin measurements were taken.
- *treat*: randomized treatment (placebo or prednisone).
- *Time*: time (in years) from the start of treatment until death or censoring.
- *death*: censoring indicator (1: if patient is died; 0: otherwise).

41% of patients received placebo; the number of longitudinal observations (prothrombin measurements) per patient varied between 1 and 7, with 5 being the highest frequency; and the observed survival times had 0.1 years as minimum, 13.4 as maximum, and 2.6 as median, with 40.2% right-censored.

9.2 | Model specification

The longitudinal submodel is expressed through a linear mixed-effects specification:

$$y_i(t|\boldsymbol{\phi}, \mathbf{b}_i, \tau) = \phi_1 + b_{1i} + (\phi_2 + b_{2i})t + \phi_3 \text{treat} + \epsilon_i(t|\tau), \quad i = 1, \dots, n, \quad (13)$$

where $y_i(t|\cdot)$ represents the log-prothrombin value for patient i at time t ; ϕ_1 and ϕ_2 are fixed effects for intercept and slope, respectively, with b_{1i} and b_{2i} being the respective individual random effects; ϕ_3 is the regression coefficient for `treat` covariate; and $\epsilon_i(t|\tau)$ is a measurement error for patient i at time t . We assume that the individual random effects, $\mathbf{b}_i = (b_{1i}, b_{2i})^\top$, given Σ follow a joint bivariate normal distribution with zero mean vector and variance-covariance matrix Σ ; and that the errors are conditionally i.i.d. as $\epsilon_i(t|\tau) \sim \text{Normal}(0, \tau)$, where τ represents the error precision (defined as one divided by the variance). Random effects and error terms are assumed mutually independent.

The survival submodel is expressed through a PH specification:

$$h(t|\lambda, \alpha, \beta, \gamma, \mathbf{b}_i) = h_0(t|\lambda, \alpha) \exp \{ \beta \text{treat} + \gamma(b_{1i} + b_{2i}t) \}, \quad t > 0, \quad (14)$$

with $h_0(t|\lambda, \alpha) = \lambda \alpha t^{\alpha-1}$ specified as a Weibull baseline hazard function, where λ and α are the scale and shape parameters, respectively; β is the regression coefficient for the `treat` covariate; and γ is an association parameter that measures the strength of the link between both submodels sharing random effects.

Prior distributions for the parameters of both submodels are set according to Section 2.3.

9.3 | Model implementation

We start by loading the data:

```
R> library(JMbayes)
R> data(prothro)
R> data(prothros)
```

So, we setup an INLA survival object and fit the joint model:

```
R> m7.jml <- joint(formSurv = inla.surv(time = Time, event = death) ~ treat,
  dataSurv = prothros, + formLong = pro ~ time + treat + (1 + time | id),
  + dataLong = prothro, basRisk = "weibullsurv", family = "lognormal",
  + id = "id", timeVar = "time", assoc = "SRE")
```

Note that the survival formula specification in the `joint` function is similar to a PH model. On the other hand, the longitudinal formula specification includes the outcome (`pro`), the regression structure indicating intercept and slope patient-specific random effects (`(1 + time | id)`), the family name for sampling distribution (`family = "lognormal"`, i.e. we fit the outcome on a logarithmic scale), the time variable (`timeVar = "time"`), and the type of association between the longitudinal and survival submodels (`assoc = "SRE"`). Specifically, SRE shares the individual deviation from the mean at time t as defined by the random effects, that is, as in (14). See `help(joint)` for other types of association. Finally, the posterior distribution summary is given by:

```
R> summary(m7.jml, sdcor = TRUE)
Longitudinal outcome (lognormal)

              mean      sd 0.025quant 0.5quant 0.975quant
Intercept_L1    4.2757 0.0215    4.2335    4.2757    4.3178
time_L1        -0.0014 0.0066   -0.0143   -0.0014    0.0115
```

treatprednisone_L1	-0.0973	0.0305	-0.1572	-0.0973	-0.0374
Res. err. (sd)	0.2577	0.0039	0.2500	0.2577	0.2652

Random effects standard deviation / correlation (L1)

	mean	sd	0.025quant	0.5quant	0.975quant
Intercept_L1	0.3092	0.0126	0.2888	0.3079	0.3369
time_L1	0.1118	0.0068	0.0982	0.1124	0.1240
Intercept_L1:time_L1	-0.0410	0.0237	-0.0906	-0.0396	0.0013

Survival outcome

	mean	sd	0.025quant	0.5quant	0.975quant
Weibull (shape)_S1	0.8808	0.0175	0.8455	0.8810	0.9144
Weibull (scale)_S1	0.1990	0.0199	0.1627	0.1980	0.2408
treatprednisone_S1	0.0753	0.1347	-0.1889	0.0753	0.3394

Association longitudinal - survival

	mean	sd	0.025quant	0.5quant	0.975quant
SRE_L1_S1	-2.1562	0.1323	-2.4014	-2.161	-1.8817

log marginal-likelihood (integration)	log marginal-likelihood (Gaussian)
-23339.31	-23339.31

Deviance Information Criterion: 9461.311

Widely applicable Bayesian information criterion: 8888.578

Computation time: 8.64 seconds

In order for the variability parameters to be expressed in terms of variance instead of standard deviation, set “sdcor” as FALSE or remove it from the summary function call. The `plot()` function can be applied to any model to return several plots, such as:

```
R> plot(m7.jml, sdcor = FALSE) # Variance instead of standard deviation
```

Figure 7 gives the posterior distributions for the the fixed effects (ϕ) and variance of the residual error term of the longitudinal submodel while Figure 8 shows the posterior distribution for variance and covariance terms of the random effects and Figure 9 gives the posterior distributions of parameters related to the survival submodel. Visualizing posterior distributions can help understanding the uncertainty over parameters compared to standard deviations and quantiles provided by the summary functions as it can clearly display departure from Gaussianity (eg, heavy tails and skewness).

9.4 | Model interpretation

There are multiple aspects of a joint model that can be interpreted. For example, we can start with the effect of treatment on prothrombin measurements ϕ_3 . In the context of the lognormal distribution, we can obtain a multiplicative effect of covariates by exponentiating the fixed effects value as follows:

```
R> inla.zmarginal(inla.tmarginal(function(x) exp(x),
                                m7.jml$marginals.fixed[["treatprednisone_{L}1"]]))
      mean      sd quant0.025 quant0.25 quant0.5 quant0.75 quant0.975
0.9077    0.0275    0.8547    0.8887    0.9072    0.9261    0.9629
```

The treatment prednisone is associated with a statistically significant average decrease of 9% [3%, 15%] in prothrombin level when accounting for individual heterogeneity and informative censoring by death. The association parameter

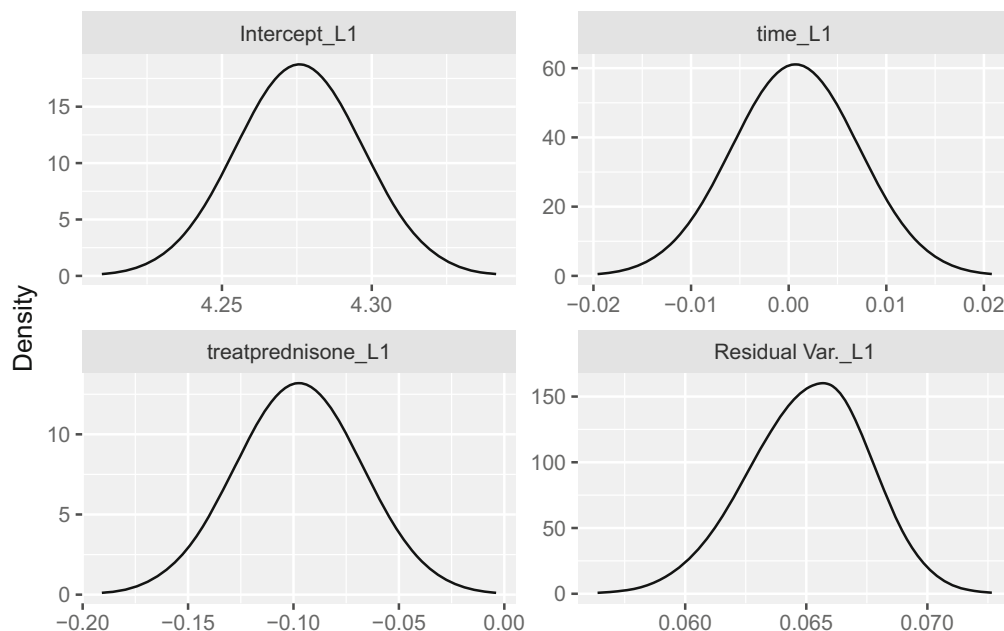


FIGURE 7 Posterior marginal distributions for the fixed effects (ϕ) and variance of the residual error term of the longitudinal submodel (13).

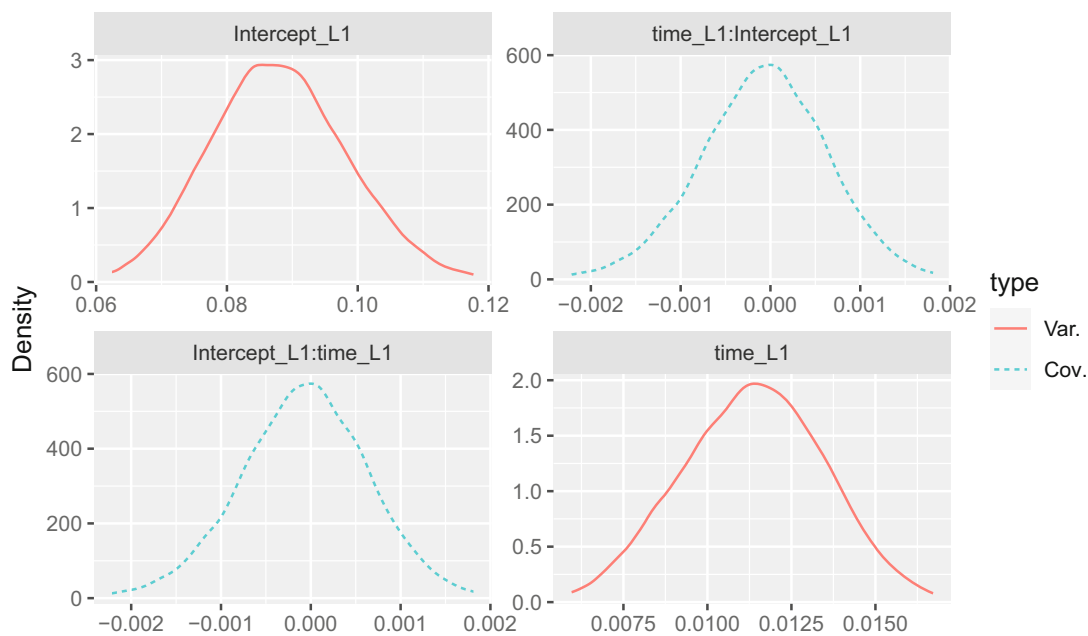


FIGURE 8 Posterior marginal distributions for the variance-covariance matrix (Σ) of the random effects.

$\hat{\gamma} = -2.16[-2.41, -1.88]$ indicates a statistically significant effect of the heterogeneity of the population captured by the random effects in the longitudinal part on the risk of death, meaning that patients with higher/lower prothrombin levels compared to the population average are associated to a reduced/increased risk of death at any given time point, respectively. Finally, the effect of treatment on the risk of death $\hat{\beta} = 0.08[-0.19, 0.34]$ is not statistically significant, meaning that there is no evidence for different risk profiles for each treatment line when taking into account individual heterogeneity of the population due to factors not included in the model.

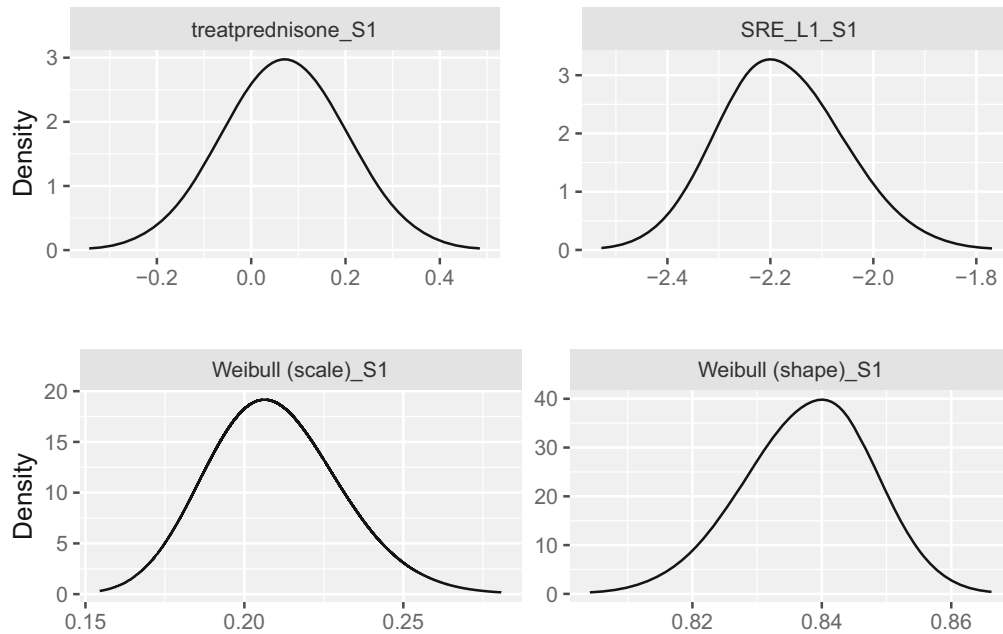


FIGURE 9 Posterior marginal distributions for the fixed effects (β and γ) and Weibull baseline hazard parameters of the survival submodel (14).

10 | JOINT MODELS OF LONGITUDINAL SEMICONTINUOUS, RECURRENT EVENTS AND TERMINAL EVENT DATA

10.1 | Motivation: colorectalLongi and colorectal datasets (from the frailtypack R-package)

The `colorectalLongi` (longitudinal format) and `colorectal` (survival format) datasets contain a random subsample of 150 patients from the follow-up of the FFCO 2000-05 multicenter phase III clinical trial with metastatic colorectal cancer randomized into two therapeutic strategies: combination and sequential.³⁹ Each patient has the following variables:

Only longitudinal format:

- `year`: time of visit (in years) counted from baseline.
- `tumor.size`: individual longitudinal measurement of transformed (Box-Cox with parameter 0.3) sums of the longest diameters.

Only survival format:

- `time0`: start of interval (0 or previous recurrence time).
- `time1`: recurrence or censoring time.
- `new.lesions`: appearance of new lesions status (1: new lesions; 0: censored or no event).
- `state`: death indicator (1: dead; 0: alive).
- `gap.time`: interrecurrence time or censoring time.

Both formats:

- `id`: patient number.
- `treatment`: to which treatment arm a patient was allocated? (1: sequential (S); 2: combination (C)).

- `age`: age at baseline (1: < 50 years, 2: 50 – 69 years, 3: > 69 years).
- `who.ps`: WHO performance status at baseline (1: status 0; 2: status 1; 3: status 2).
- `prev.resection`: previous resection of the prostate tumor? (1: yes; 0: no).

51% of patients received sequential treatment; the number of longitudinal observations (tumor size measurements) per patient varied between 1 and 18, with 5 being the highest frequency; the proportion of zeros in the longitudinal outcome is 3.8%. The number of recurrent events (ie, new lesions) ranges between 0 and 4 per individual with a median of 1, these recurrent events are observed between 0.12 and 2.70 years of follow-up with a median of 0.75. The observed death times had 0.04 years as minimum, 3.84 as maximum, and 0.96 as median, with 19.3% right-censored.

10.2 | Model specification

The proposed model is a new model that extends the two-part joint model for a longitudinal semicontinuous biomarker and a terminal event introduced by Rustand et al⁴⁰ and successfully implemented within the INLA framework to overcome the limitations of standard estimation strategies such as Newton-like algorithms or MCMC (ie, long computation time and convergence issues).⁴¹ Here we include an additional submodel that accounts for recurrent events, linked with all the other submodels through shared random effects. In the context of the application, this new model is able to handle the appearance of new lesions over the follow-up of the clinical trial in addition to the longitudinal semicontinuous measurements of the target tumors measurements and the time to death, which is an important additional piece of information.

The longitudinal submodel is expressed through a two-part mixed-effects specification:

$$Z_i(t|\boldsymbol{\phi}_1, b_{1i}) \sim \text{Bernoulli}(\eta_i(t|\boldsymbol{\phi}_1, b_{1i})), \quad i = 1, \dots, n, \quad (15)$$

$$\text{logit}(\eta_i(t|\boldsymbol{\phi}_1, b_{1i})) = \phi_{11} + b_{1i} + \phi_{12}t + \phi_{13}\text{treatment} + \phi_{14} \times t \times \text{treatment}, \quad (16)$$

$$\log(y_i(t|\boldsymbol{\phi}_2, \mathbf{b}_{2i}, \tau, y_i(t) > 0)) = \phi_{21} + b_{21i} + (\phi_{22} + b_{22i})t + \phi_{23}\text{treatment} + \phi_{24} \times t \times \text{treatment} + \epsilon_i(t|\tau), \quad (17)$$

where $Z_i(t|\cdot)$ is a binary variable defined as 0 if $y_i(t|\cdot) = 0$ and 1 otherwise, with $y_i(t|\cdot)$ being the tumor size for patient i at time t ; $\boldsymbol{\phi}_1$ and $\boldsymbol{\phi}_2$ are fixed effects parameters of binary (15) and semicontinuous (17) parts, including `treatment`, `time (year)` and their interaction; and $\epsilon_i(t|\tau)$ is a normally distributed error term for patient i at time t . For the binary submodel, we include an intercept random effect, b_{1i} while for the semicontinuous submodel, we consider intercept and slope random effects, $\mathbf{b}_{2i} = (b_{21i}, b_{22i})^\top$. The random effects follow a joint multivariate normal distribution with zero mean vector and variance-covariance matrix Σ , thus accounting for correlation between repeated measurements of an individual as well as the correlation between the binary and continuous parts. Random effects and error terms are assumed mutually independent.

The recurrent events corresponding to the appearance of new lesions are modeled with a frailty model (18) and the survival times are modeled with a PH model (19). The frailty term is shared in the PH submodel, it is normally distributed and assumed independent from random effects in the longitudinal parts. In addition, the random effects from the two-part mixed-effects model are shared in both the frailty model and the PH model:

$$h_1(t|\lambda_1, \beta_1, \gamma_1, b_{1i}, \mathbf{b}_{2i}, b_{3i}) = h_{01}(t|\lambda_1) \exp \left\{ \beta_1 \text{treatment}_i + \gamma_{11}b_{1i} + \gamma_{12}b_{21i} + \gamma_{13}b_{22i} + b_{3i} \right\}, \quad t > 0, \quad (18)$$

$$h_2(t|\lambda_2, \beta_2, \gamma_2, b_{1i}, \mathbf{b}_{2i}, b_{3i}) = h_{02}(t|\lambda_2) \exp \left\{ \beta_2 \text{treatment}_i + \gamma_{21}b_{1i} + \gamma_{22}b_{21i} + \gamma_{23}b_{22i} + \gamma_{33}b_{3i} \right\}, \quad t > 0. \quad (19)$$

The baseline hazard function of both survival submodels is approximated with smooth splines corresponding to a second-order random walk prior with precision parameters λ_1 and λ_2 ; β_1 and β_2 are the regression coefficient for the `treatment` covariate in the two survival submodels (18) and (19), respectively; and γ 's are association parameters that scale the shared random effects and measure the strength of the link between the submodels.

Prior distributions for the parameters of both submodels are set according to Section 2.3.

10.3 | Model implementation

We start by loading the data:

```
R> library(frailtypack)
R> data(colorectal)
R> data(colorectalLongi)
```

Then, we reverse the Box-Cox transformation to have tumor size on the original scale, select only positive sizes of the tumor, and create the binary outcome which indicates the presence of tumor:

```
R> colorectalLongi$y <- round((colorectalLongi$tumor.size*0.3+1)^(1/0.3), 5)
R> colorectalLongiPositive <- colorectalLongi[colorectalLongi$y > 0,]
R> colorectalLongi$z <- ifelse(colorectalLongi$y == 0, 0, 1)
```

So, we extract terminal event data for submodel (19), set INLA survival objects for recurrent and terminal events, and fit the joint model:

```
R> colorectalSurv <- subset(colorectal, new.lesions == 0)
R> m8.jm2 <- joint(formSurv = list(inla.surv(time = time1, truncation = time0,
                                         event = new.lesions) ~ treat-
ment + (1 | id),
+                               inla.surv(time = time1, event = state) ~ treatment),
+                               formLong = list(z ~ year * treatment + (1 | id),
+                                               y ~ year * treatment + (1 + year | id)),
+                               dataSurv = list(colorectal, colorectalSurv),
+                               dataLong = list(colorectalLongi, colorectalLongiPositive),
+                               basRisk = c("rw2", "rw2"), assocSurv = TRUE,
+                               timeVar = "year", family = c("binomial", "lognormal"), id = "id",
+                               corLong = TRUE, assoc = list(c("SRE_ind", "SRE_ind"),
+                                                             c("SRE_ind", "SRE_ind")))
```

The association between the longitudinal and survival submodels (`assoc = "SRE_ind"`) indicates that each random effect is shared independently in the survival submodels, as described in (18) and (19) (as opposed to the individual deviation at time t defined by the random effects, as illustrated in Section 9). This association is decomposed in a list of two vectors with two association parameters in each that corresponds to the two longitudinal submodels (15) and (17) sharing into the two survival submodels (18) and (19), respectively (see `help(joint)` for other types of association). The argument `corLong = TRUE` indicates that the random effects across the two longitudinal submodels are correlated (setting this parameter to `FALSE` would make the random intercept in (15) independent from the random intercept and slope from (17)). The argument `assocSurv = TRUE` indicates that the frailty term from the recurrent events submodel is shared in the terminal event submodel. Finally, the posterior distribution summary is given by:

```
R> summary(m8.jm2)
Longitudinal outcome (L1, binomial)
              mean      sd 0.025quant 0.5quant 0.975quant
Intercept_L1    6.9695 1.0946      4.8241    6.9695    9.1148
year_L1          0.2716 1.0366     -1.7600    0.2716    2.3033
treatmentC_L1   -2.0468 1.1744     -4.3485   -2.0468    0.2550
year:treatmentC_L1 -1.0604 1.0659     -3.1495   -1.0604    1.0287

Longitudinal outcome (L2, lognormal)
              mean      sd 0.025quant 0.5quant 0.975quant
```

Intercept_L2	2.1274	0.0845	1.9617	2.1274	2.2931
year_L2	-0.1432	0.0780	-0.2961	-0.1432	0.0098
treatmentC_L2	0.0539	0.1212	-0.1838	0.0539	0.2915
year:treatmentC_L2	-0.3157	0.1116	-0.5345	-0.3157	-0.0969
Res. err. (variance)	0.1647	0.0093	0.1472	0.1644	0.1838

Random effects variance-covariance

	mean	sd	0.025quant	0.5quant	0.975quant
Intercept_L1	2.6064	1.5135	0.7542	2.2540	6.6371
Intercept_L2	0.4888	0.0664	0.3738	0.4836	0.6300
year_L2	0.1854	0.0503	0.1025	0.1798	0.2978
Intercept_L1:Intercept_L2	0.7259	0.2691	0.2605	0.6980	1.3347
Intercept_L1:year_L2	-0.1882	0.1793	-0.6226	-0.1617	0.0908
Intercept_L2:year_L2	-0.0914	0.0498	-0.2103	-0.0848	-0.0131

Survival outcome (S1)

	mean	sd	0.025quant	0.5quant	0.975quant
Baseline risk (variance)_S1	0.1343	0.0896	0.0315	0.1117	0.3730
treatmentC_S1	-0.2842	0.2129	-0.7015	-0.2842	0.1332

Frailty term variance (S1)

	mean	sd	0.025quant	0.5quant	0.975quant
IDIntercept_S1	0.5228	0.2901	0.1699	0.4534	1.2857

Survival outcome (S2)

	mean	sd	0.025quant	0.5quant	0.975quant
Baseline risk (variance)_S2	0.0590	0.0554	0.0071	0.0425	0.2122
treatmentC_S2	-0.1457	0.3179	-0.7688	-0.1457	0.4774

Association longitudinal - survival

	mean	sd	0.025quant	0.5quant	0.975quant
SRE_Intercept_L1_S1	0.2230	0.2487	-0.2812	0.2279	0.6977
SRE_Intercept_L1_S2	0.5471	0.4352	-0.3324	0.5547	1.3808
SRE_Intercept_L2_S1	-0.1158	0.4017	-0.8841	-0.1233	0.6972
SRE_year_L2_S1	-0.0877	0.3444	-0.7570	-0.0907	0.5991
SRE_Intercept_L2_S2	0.3320	0.6159	-0.8565	0.3239	1.5685
SRE_year_L2_S2	-0.4435	0.5100	-1.4632	-0.4382	0.5448

Association survival - survival

	mean	sd	0.025quant	0.5quant	0.975quant
IDIntercept_S1_S2	1.5642	0.3673	0.8232	1.5702	2.2694

log marginal-likelihood (integration)	log marginal-likelihood (Gaussian)
-3534.745	-3518.836

Deviance Information Criterion: 6436.598

Widely applicable Bayesian information criterion: 6458.738

Computation time: 18.08 seconds

The recurrent events are set to use calendar times, that is, the beginning of the “at-risk” period for the second recurrent event starts after the time of occurrence of the first event. However, it is possible to use gap times in order to have a clock-reset after each recurrent event:

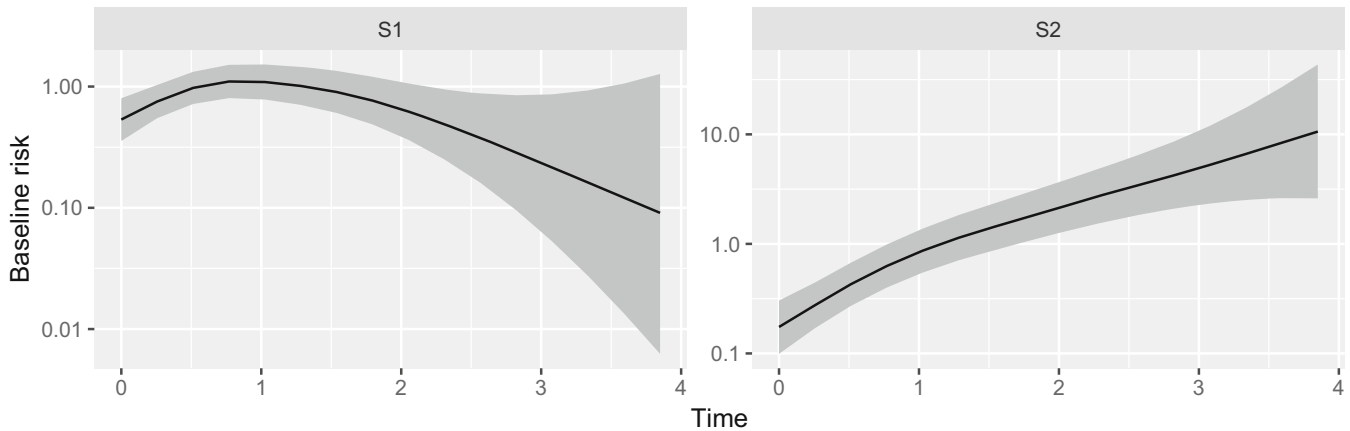


FIGURE 10 Baseline hazard curves from joint model (15) fitted by smooth splines with 15 equidistant knots.

```
R> surv.obj.jm2.recu <- inla.surv(time = colorectal$gap.time,
                                   event = colorectal$new.lesions)
```

Baseline hazard curves can be plotted (in log10 scale) using ggplot2 R-package. Figure 10 displays the baseline hazards curves for each of the two competing risks. The baseline hazard for the recurrent events submodel shows an initial increase of the hazard followed by a decrease, clearly illustrating the usefulness of using Bayesian smooth splines instead of parametric distributions while the second curve corresponding to the baseline hazard of death shows a constant increase that could alternatively be captured with some parametric distribution such as Weibull.

```
R> plot(m8.jm2)$Baseline + scale_{y}_log10()
```

10.4 | Model interpretation

The first part of the model displayed in the summary is the longitudinal binary part that evaluates the effect of covariates on the probability of positive tumor size. In the dataset, only a few zeros are observed, thus leading to a high probability of positive value. For example, we can compute the probability of positive tumor size 3 years after randomization conditional on treatment (credible intervals follow trivially from sampling):

```
R> inv.logit(m8.jm2$summary.fixed["Intercept_L1", 1]
+ 3*m8.jm2$summary.fixed["year_{L}1", 1])
0.9996
R> inv.logit(m8.jm2$summary.fixed["Intercept_L1", 1]
+ 3*m8.jm2$summary.fixed["year_{L}1", 1] +
+ m8.jm2$summary.fixed["treatmentC_L1", 1] +
+ 3*m8.jm2$summary.fixed["year.X.treatmentC_{L}1", 1])
0.9280
```

This suggests that the probability of positive tumor size 3 years after randomization for an average patient is 0% for sequential treatment while it is 7% for the combination treatment, accounting for informative censoring related to the appearance of new lesions and death. Note that due to the non-linear link function, this interpretation is subject-specific and corresponds to individuals with random effects equal to zero (ie, average patients). From the second part of the model, we can compute the multiplicative effect of treatment on the average tumor size evolution over time, conditional on observing a positive tumor size:

```
R> inla.zmarginal(inla.tmarginal(function(x) exp(x), m8.jm2$marginals.fixed[[10]]))
      mean      sd quant0.025 quant0.25 quant0.5 quant0.75 quant0.975
0.7338    0.0816    0.5864    0.6763    0.7292    0.7861    0.8067
```

Patients with positive tumor size receiving the combination treatment are associated with a statistically significant average decrease of 27% [19%, 42%] in tumor size per year of follow-up compared to patients receiving sequential treatment, accounting for informative censoring related to the appearance of new lesions and death. The association parameters between the longitudinal markers and the recurrent and terminal event submodels give the effect of the heterogeneity captured by each random effect independently shared in the survival submodels, this can be useful to account for individual heterogeneity due to unobserved confounders in survival models. As opposed to the previous joint model (see Section 9) where the linear combination of the random effects was shared, here we share and scale each random effect separately, thus offering more flexibility in the association. The frailty term in the recurrent event submodel captures some additional individual heterogeneity due to unobserved factors in the risk of the appearance of new lesions. This frailty term is shared and scaled in the survival model, the scaling parameter is $\hat{\gamma}_{33} = 1.56$ [0.82, 2.27]. While the association of the random effects from the longitudinal submodels with the survival submodels is not significantly different from zero, this frailty term effect on the risk of death is statistically significant, indicating that the individual heterogeneity in the risk of appearance of new lesions is predictive of the risk of death (ie, higher risk of new lesions lead to higher risk of death).

11 | MODEL CHECKS AND PRIOR SENSITIVITY ANALYSIS

In Bayesian inference, a model that is ill-defined or non-identifiable due to a lack of data or a flat likelihood will not fail as it would in the frequentist inference framework because when there is no information to estimate a parameter, the posterior distribution should be driven by the priors. With MCMC, such models are often associated to slow convergence properties while with INLAjoint, the computation time is not affected. Therefore it is important to be able to identify these models. Several warnings are included in INLAjoint to inform the user about potentially ill-defined models, these warnings are triggered under specific circumstances (highly correlated hyperparameters, abnormal hyperparameters skewness correction, number of random effects larger than number of data points) and can help the user to identify potential issues with the model. Additionally, while MCMC-like convergence diagnostics does not apply to the INLA methodology (because INLA is based on numerical approximations to the posterior densities instead of MCMC samples), all the traditional Bayesian model checks holds. We implemented a method to visualize the model sensitivity to priors specification. We illustrate in this section how to perform prior sensitivity analysis with INLAjoint by looking at a simple example. Consider the logistic submodel (ie, binary part) from the last example defined by Equation (15) of Section 10. We can fit this model with a specific prior distribution by specifying priors in the control options as follows:

```
R> m8.sen <- joint(formLong = z ~ year * treatment + (1 | id),
+                 dataLong = colorectalLongi, timeVar="year",
+                 family = c("binomial"), id = "id",
+                 control=list(priorFixed=list(
+                 mean=0, prec=0.001, mean.intercept=0, prec.intercept=0.001)))
R> print(summary(m8.sen)$FixedEff[[1]], digits=2)
```

	mean	sd	0.025quant	0.5quant	0.975quant
Intercept_{L}1	5.944	1.08	3.8	5.944	8.063
year_{L}1	-0.065	0.99	-2.0	-0.065	1.876
treatmentC_{L}1	-2.127	1.13	-4.3	-2.127	0.094
year:treatmentC_{L}1	-1.165	1.01	-3.2	-1.165	0.823

Here, the fixed effects are assigned a Gaussian prior with mean 0 and variance 1000 (instead of the default 100), note that the prior for the intercept is defined separately through the mean.intercept and prec.intercept control arguments. We can display the prior versus posterior distributions for these fixed effects by calling the plot function with the argument priors=TRUE:

```
R> plot(m8.sen, priors=TRUE)$Outcomes$L1
```

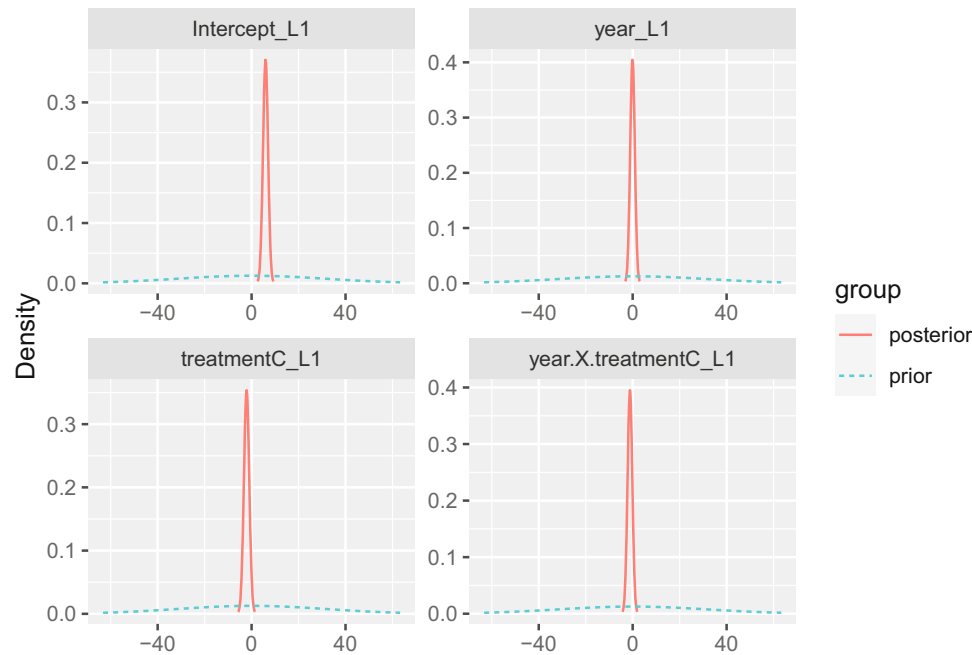


FIGURE 11 Prior and posterior distributions for the fixed effects of the identifiable model described in Section 11 with Gaussian priors with mean 0 and variance 1000 for all the fixed effects.

In Figure 11, the priors are flat while the posteriors are peaked, indicating that posteriors are driven by the information from the data. Since the model is identifiable, changing the prior to another reasonable distribution will have a limited impact on the posterior distributions:

```
R> m8.sen2 <- joint(formLong = z ~ year * treatment + (1 | id),
+                   dataLong = colorectalLongi, timeVar="year",
+                   family = c("binomial"), id = "id",
+                   control=list(priorFixed=list(
+                     mean=10, prec=0.01, mean.intercept=10, prec.intercept=1)))
R> print(summary(m8.sen2)$FixedEff[[1]], digits=2)
```

	mean	sd	0.025quant	0.5quant	0.975quant
Intercept_L1	5.848	1.04	3.8	5.848	7.89
year_L1	-0.011	0.98	-1.9	-0.011	1.91
treatmentC_L1	-2.024	1.09	-4.2	-2.024	0.12
year:treatmentC_L1	-1.221	1.00	-3.2	-1.221	0.74

Figure 12 shows how the posterior distributions are almost unchanged despite a very different prior specification as the posteriors are driven by the data information. Now we can change the data so that the model is not identifiable, for example we can only keep longitudinal measurements at time 0 so that there is no information to identify slope-related parameters, then we can visualize the prior vs. posterior distributions again:

```
R> colorectalLongi2 <- colorectalLongi[colorectalLongi$year==0,]
R> m8.sen3 <- joint(formLong = z ~ year * treatment + (1 | id),
+                   dataLong = colorectalLongi2, timeVar="year",
+                   family = c("binomial"), id = "id")
R> plot(m8.sen3, priors=TRUE)$Outcomes$L1
R> print(summary(m8.sen3)$FixedEff[[1]], digits=2)
```

	mean	sd	0.025quant	0.5quant	0.975quant
Intercept_L1	1.3e+01	3.7	6.0	1.3e+01	20
year_L1	8.8e-17	10.0	-19.6	1.7e-14	20

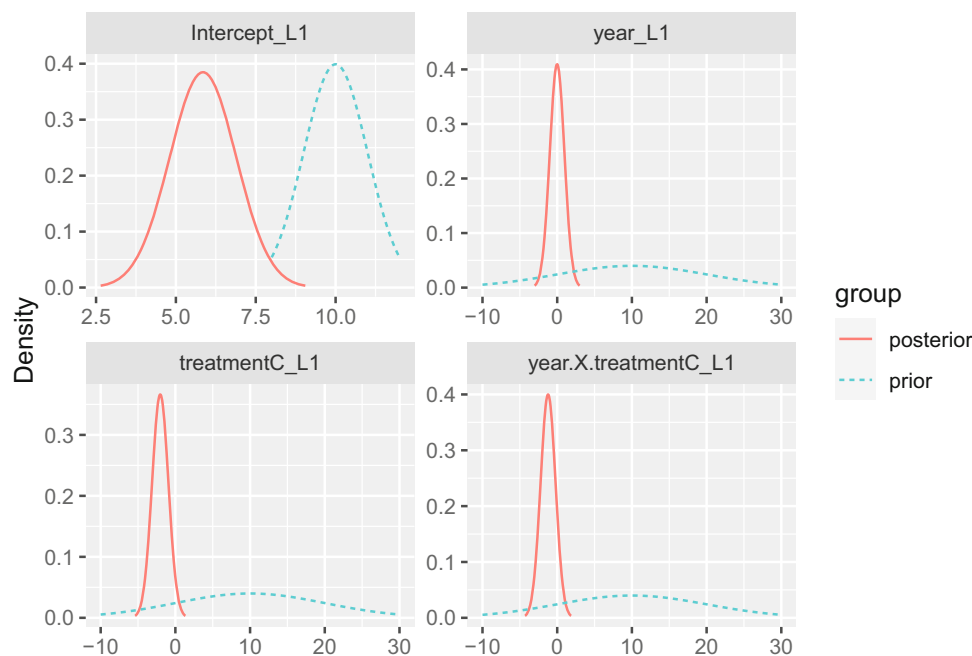


FIGURE 12 Prior and posterior distributions for the fixed effects of the identifiable model described in Section 11 with Gaussian priors with mean 10 and variance 1 for the fixed intercept, mean 10 and variance 100 for the other fixed effects.

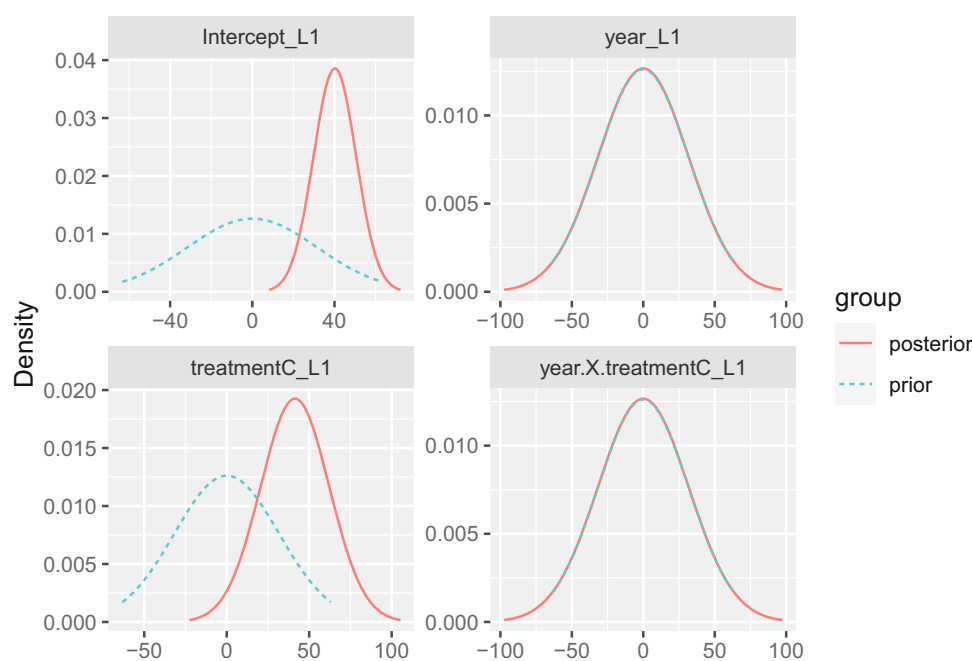


FIGURE 13 Prior and posterior distributions for the fixed effects of the non-identifiable model described in Section 11.

treatmentC_L1	6.9e+00	6.8	-6.5	6.9e+00	20
year:treatmentC_L1	8.8e-17	10.0	-19.6	1.7e-14	20

It is clear from Figure 13 that posteriors are now matching priors for the two slope-related parameters, showing a lack of information in the data to estimate the slope-related parameters. Here we used the default prior (ie, Gaussian with mean 0 and variance 100) but regardless of the prior choice, the posterior will match them as the data is not informative

TABLE 2 Computation time (in seconds) for the first 7 models with INLA, JAGS, and STAN.

	AFT	PH	Mixture cure	Competing risk	Multi-state	Frailty	Joint model 1
INLA	0.63	0.62	0.60	1.35	0.80	0.59	12.03
JAGS	24.25	14.95	61.55	1271.93	154.81	4.70	1753.70
STAN	67.36	66.00	74.24	349.50	94.32	72.89	948.50

and the model is not identifiable. Evaluating the sensitivity of posterior distributions assuming different priors can help to assess the model stability and reliability. Another example of prior specification and prior sensitivity analysis has been proposed in the supplementary materials of Rustand et al,²⁴ where the illustration is given for the correlation parameter between two random effects of a longitudinal mixed effects model.

12 | COMPARISON OF THE FITTED MODELS WITH MCMC

Comparing the model fit between INLA and MCMC is an important step in assessing the computational efficiency and reliability of our approach. In this paper, we have the opportunity to make such a comparison, as a previous work¹⁷ fitted all the models (with the exception of the last joint model) using the JAGS software⁴² using the R package `rjags`.⁴³ Additionally, we fitted these models with STAN⁴⁴ using the R package `rstan`,⁴⁵ another popular implementation of MCMC available in R that relies on the Hamiltonian Monte Carlo method. The last joint model (described in Section 10) is not included in the comparison because writing the code for this model with MCMC becomes more complex as there is no readily available user-friendly package or prior implementation of this model in the literature. This complex joint model aims at illustrating how `INLAjoint` allows to build more complex models with simple code syntax. We used INLA version 24.05.10, `INLAjoint` version 24.6.5, `rjags` version 4-15 with 4 chains, 5000 iterations per chain including 1000 burn-in iterations and thinning of 1 for all models and `rstan` version 2.32.5 with default options, that is, 4 chains, 2000 iterations per chain including 1000 burn-in iterations and thinning of 1. Figure 14 displays the posterior marginals for all the fixed effects in each model in order to assess the agreement between the INLA and MCMC results. As expected, these posterior distributions exhibit a high degree of concordance, indicating that INLA and both MCMC methods yield consistent results. This further underscores the reliability of INLA as an alternative to MCMC, particularly when computational efficiency is of utmost importance. Computation time is an essential consideration in Bayesian modeling, especially when dealing with extensive datasets or complex models. To illustrate the computational efficiency of INLA, we detailed in Table 2 the time required for fitting all the models using both INLA and MCMC. This comparison highlights the substantial time savings achievable with INLA. For example, for the first joint model, INLA delivered results within 12 seconds, whereas MCMC required more than 15 minutes with STAN and 29 minutes with JAGS to converge to the same posterior distributions (see Figure 14).

13 | CONCLUSIONS

We have revisited several of the main survival models using integrated nested Laplace approximations (INLA) to perform approximate Bayesian inference. Specifically, we have implemented accelerated failure time, proportional hazard, mixture cure, competing risks, multi-state, frailty, and two joint model specifications for longitudinal and time-to-event data. Of particular note is the last joint model, a novel addition to the literature, which combines a two-part mixed-effects model for a longitudinal semicontinuous outcome, a frailty model for recurrent events, and a proportional hazards model for a terminal event, all linked through shared random effects. This illustrates the flexibility of `INLAjoint`, where each model component is considered as a building block and thus allows the combination of different components to construct various models that satisfy clinicians' and health researchers' needs.

In conclusion, our work demonstrates that Bayesian inference with INLA, and specifically the `INLAjoint` R-package, offers a user-friendly and flexible approach for fitting various survival models. It simplifies the implementation of complex models, requiring considerably fewer lines of code compared to traditional methods such as JAGS or STAN. Its efficiency, both in terms of coding simplicity and computation time, makes it a compelling alternative to MCMC methods. Recently,¹⁹ INLA was shown to perform very well for survival analysis in the case of big data (a Cox model with

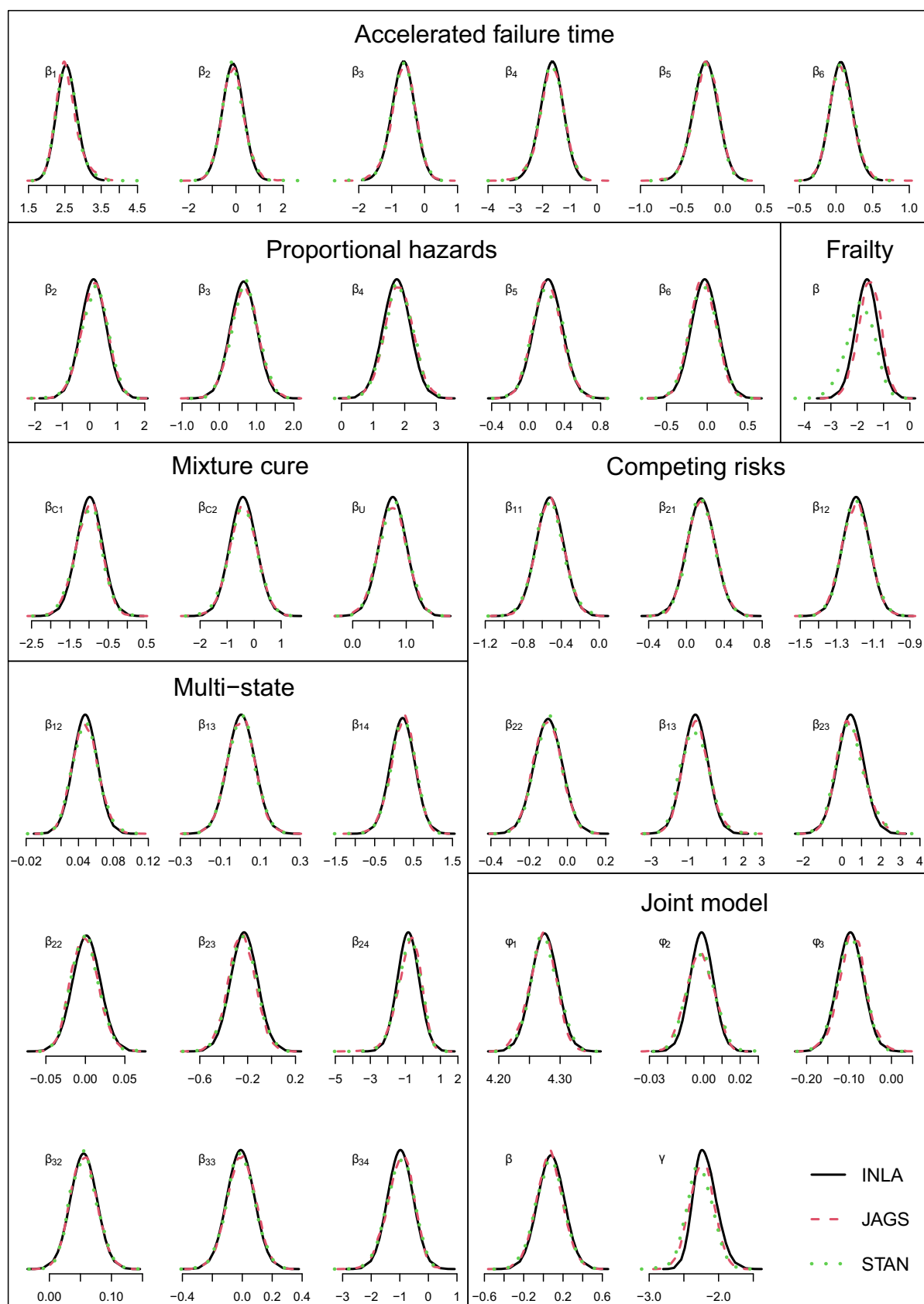


FIGURE 14 Marginal posterior densities for all the fixed effects of the first 7 models fitted with INLA, JAGS, and STAN. Corresponding computation times are provided in Table 2.

100 000 patients was fitted in 22.5 seconds). Future research for big data and joint models using INLA is ongoing. We hope this tutorial has convinced the reader that Bayesian inference with INLA presents an efficient approach to survival analysis.

ACKNOWLEDGEMENTS

We are grateful to the two anonymous referees and the Associate Editor, whose comments and feedback on our work help us improve the manuscript substantially. For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) license to any Author Accepted Manuscript version arising from this submission.

CONFLICT OF INTEREST STATEMENT

The authors declare no potential conflict of interest.

FUNDING INFORMATION

This work was supported by the UKRI Medical Research Council, Grant/Award Number: MC_UU_00002/5.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Bayes-surv-INLA at <https://github.com/DenisRustand>.

ORCID

Danilo Alvares  <https://orcid.org/0000-0003-3764-0397>

Janet van Niekerk  <https://orcid.org/0000-0002-4334-2057>

REFERENCES

- Klein JP, Moeschberger ML. *Survival Analysis: Techniques for Censored and Truncated Data*. 2nd ed. New York, US: Springer; 2003.
- Collett D. *Modelling survival data in medical research*. 4th ed. New York, US: Chapman & Hall/CRC; 2023.
- Kleinbaum DG, Klein M. *Survival Analysis: A Self-Learning Text*. 3rd ed. New York, US: Springer; 2012.
- Aalen OO, Borgan Ø, Gjessing HK. *Survival and event History Analysis: A Process Point of View*. 1st ed. New York, US: Springer; 2008.
- Cox DR. Regression models and life-tables. *J Royal Stat Soc B (Methodol)*. 1972;34(2):187-202.
- Lázaro E, Armero C, Alvares D. Bayesian regularization for flexible baseline hazard functions in Cox survival models. *Biom J*. 2021;63(1):7-26.
- Wei LJ. The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis. *Stat Med*. 1992;11(14-15):1871-1879.
- Sinha D, Dey DK. Semiparametric Bayesian analysis of survival data. *J Am Stat Assoc*. 1997;92(439):1195-1212.
- Ibrahim JG, Chen MH, Sinha D. *Bayesian Survival Analysis*. 1st ed. New York, US: Springer; 2001.
- Martino S, Akerkar R, Rue H. Approximate Bayesian inference for survival models. *Scand J Stat*. 2011;38(3):514-528.
- Rizopoulos D, Hatfield LA, Carlin BP, Takkenberg JJM. Combining dynamic predictions from joint models for longitudinal and time-to-event data using Bayesian model averaging. *J Am Stat Assoc*. 2014;109(508):1385-1397.
- Bartoš F, Aust F, Haaf JM. Informed Bayesian survival analysis. *BMC Med Res Methodol*. 2022;22(238):1-22.
- Rue H, Martino S, Chopin N. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *J Royal Stat Soc B (Stat Methodol)*. 2009;71(2):319-392.
- van Niekerk J, Bakka H, Rue H. A principled distance-based prior for the shape of the Weibull model. *Stat Probab Lett*. 2021;174:1-5.
- van Niekerk J, Bakka H, Rue H. Competing risks joint models using R-INLA. *Stat Model*. 2021;21:56-71.
- Gómez-Rubio V. *Bayesian Inference with INLA*. 1st ed. Boca Raton, US: Chapman & Hall/CRC; 2020.
- Alvares D, Lázaro E, Gómez-Rubio V, Armero C. Bayesian survival analysis with BUGS. *Stat Med*. 2021;40(12):2975-3020.
- Wang X, Yue YR, Faraway JJ. *Bayesian Regression Modeling with INLA*. 1st ed. New York, US: Chapman & Hall/CRC; 2018.
- van Niekerk J, Krainski E, Rustand D, Rue H. A new avenue for Bayesian inference with INLA. *Computat Stat Data Anal*. 2023;181:107692.
- Fattah EA, van Niekerk J, Rue H. Smart gradient—an adaptive technique for improving gradient estimation. *Found Data Sci*. 2022;4(1):123-136.
- van Niekerk J, Rue H. Correcting the Laplace method with variational Bayes. *J Mach Learn Res*. 2024;25(62):1-25.
- Rue H, Riebler A, Sørbye SH, Illian JB, Simpson DP, Lindgren FK. Bayesian computing with INLA: a review. *Annu Rev Stat Appl*. 2017;4:395-421.
- Gaedke-Merzhäuser L, van Niekerk J, Schenk O, Rue H. Parallelized integrated nested Laplace approximations for fast Bayesian inference. *Stat Comput*. 2023;33(25):1-16.

24. Rustand D, van Niekerk J, Krainski ET, Rue H, Proust-Lima C. Fast and flexible inference for joint models of multivariate longitudinal and survival data using integrated nested Laplace approximations. *Biostatistics*. 2024;25:429-448.
25. van Niekerk J, Bakka H, Rue H, Schenk O. New frontiers in Bayesian modeling using the INLA package in R. *J Stat Softw*. 2021;100:1-28.
26. van Niekerk J, Haakon B, Rue H. Stable non-linear generalized Bayesian joint models for survival-longitudinal data. *Sankhya A Indian J Stat*. 2023;85(1):102-128.
27. Holford TR. Life tables with concomitant information. *Biometrics*. 1976;32(3):587-597.
28. Whitehead J. Fitting Cox's regression model to survival data using GLIM. *J Royal Stat Soc C (Appl Stat)*. 1980;29(3):268-275.
29. Johansen S. An extension of Cox's regression model. *Int Stat Rev*. 1983;51(2):165-174.
30. Rustand D, van Niekerk J, Krainski ET, Rue H. Joint modeling of multivariate longitudinal and survival outcomes with the R package INLAjoint. arXiv preprint, arXiv:2402.08335 2024.
31. Simpson D, Rue H, Riebler A, Martins TG, Sørbye SH. Penalising model component complexity: a principled, practical approach to constructing priors. *Stat Sci*. 2017;32(1):1-28.
32. Kardaun O. Statistical survival analysis of male larynx-cancer patients—a case study. *Statistica Neerlandica*. 1983;37(3):103-125.
33. Kersey JH, Weisdorf D, Nesbit ME, et al. Comparison of autologous and allogeneic bone marrow transplantation for treatment of high-risk refractory acute lymphoblastic leukemia. *N Engl J Med*. 1987;317(8):461-467.
34. Dettenkofer M, Wenzler-Rottele S, Babikir R, et al. Surveillance of nosocomial sepsis and pneumonia in patients with a bone marrow or peripheral blood stem cell transplant: a multicenter project. *Clin Infect Dis*. 2005;40(7):926-931.
35. Crowley J, Hu M. Covariance analysis of heart transplant survival data. *J Am Stat Assoc*. 1977;72(357):27-36.
36. Alvares D, Haneuse S, Lee C, Lee KH. SemiCompRisks: an R package for the analysis of independent and cluster-correlated semi-competing risks data. *R J*. 2019;11(1):376-400.
37. McGilchrist CA, Aisbett CW. Regression with frailty in survival analysis. *Biometrics*. 1991;47(2):461-466.
38. Andersen PK, Borgan Ø, Gill RD, Keiding N. *Statistical Models Based on Counting Processes*. 1st ed. New York, US: Springer; 1993.
39. Ducreux M, Malka D, Mendiboure J, et al. Sequential versus combination chemotherapy for the treatment of advanced colorectal cancer (FFCD 2000-05): an open-label, randomised, phase 3 trial. *Lancet Oncol*. 2011;12(11):1032-1044.
40. Rustand D, Briollais L, Tournigand C, Rondeau V. Two-part joint model for a longitudinal semicontinuous marker and a terminal event with application to metastatic colorectal cancer data. *Biostatistics*. 2020;23(1):50-68.
41. Rustand D, van Niekerk J, Rue H, Tournigand C, Rondeau V, Briollais L. Bayesian estimation of two-part joint models for a longitudinal semicontinuous biomarker and a terminal event with INLA: interests for cancer clinical trial evaluation. *Biom J*. 2023;65(4):1-22.
42. Plummer M. JAGS: a program for analysis of Bayesian graphical models using Gibbs sampling. *Proceedings of the 3rd International Workshop on Distributed Statistical Computing (DSC 2003)*. Vienna, Austria: Technische Universität Wien; 2003:1-10.
43. Plummer M. RJAGS: Bayesian graphical models using MCMC. 2023 R Package Version 4-15.
44. Carpenter B, Gelman A, Hoffman MD, et al. Stan: a probabilistic programming language. *J Stat Softw*. 2017;76(1):1-32.
45. Stan Development Team. RStan: the R interface to Stan. 2024 R Package Version 2.32.5.

How to cite this article: Alvares D, van Niekerk J, Krainski ET, Rue H, Rustand D. Bayesian survival analysis with INLA. *Statistics in Medicine*. 2024;43(20):3975-4010. doi: 10.1002/sim.10160