

1. BayesSurv_HReg : independent, univariate time-to-event data fit to a Cox PH model with Weibull baseline hazard
2. BayesSurv_HReg : independent, univariate time-to-event data fit to a Cox PH model with PEM baseline hazard
3. BayesSurv_AFT : independent, univariate time-to-event data fit to an AFT model with LN baseline survival distribution
4. BayesSurv_AFT : independent, univariate time-to-event data fit to an AFT model with DPM baseline survival distribution
5. BayesSurv_HReg : cluster-correlated, univariate time-to-event data fit to a Cox PH model with Weibull baseline hazard
6. BayesSurv_HReg : cluster-correlated, univariate time-to-event data fit to a Cox PH model with PEM baseline hazard
7. BayesID_HReg : independent semi-competing risks data using an illness-death model with Weibull baseline hazards
8. BayesID_HReg : independent semi-competing risks data using an illness-death model with PEM baseline hazards
9. BayesID_AFT : independent semi-competing risks data using an AFT illness-death model with LN baseline survival distribution
10. BayesID_AFT : independent semi-competing risks data using an AFT illness-death model with DPM baseline survival distribution
11. BayesID_HReg : cluster-correlated semi-competing risks data using an illness-death model with Weibull baseline hazards
12. BayesID_HReg : cluster-correlated semi-competing risks data using an illness-death model with PEM baseline hazards

Definition of the survival model

Let t_i denote the time-to-event of interest for individuals $i = 1, \dots, n$, subject to right censoring at time c_i . Let (y_i, δ_i, x_i) denote independent observations, where $y_i = \min(t_i, c_i)$, $\delta_i = \mathbb{1}(y_i \leq c_i)$, and x_i is a vector of covariates for individual i . The following Cox proportional hazards model is assumed

$$h(t_i|x_i) = h_0(t_i) \exp(x_i^\top \beta), \quad t_i > 0,$$

where the baseline hazard h_0 is defined parametrically by a Weibull hazard, $h_0(t) = \alpha \kappa t^{\alpha-1}$.

In the Bayesian framework, priors must be specified for the regression parameter, β , and the shape and scale parameters of baseline hazard function, α and κ , respectively. The following specifications are made

$$\begin{aligned} \pi(\beta) &\propto 1, \\ \pi(\alpha) &\sim \text{Gamma}(a, b), \\ \pi(\kappa) &\sim \text{Gamma}(c, d). \end{aligned}$$

Hyperparameters

The hyperparameters a and b must be specified for the prior distribution of α which is a Gamma distribution with mean ab and variance ab^2 . Similarly, the hyperparameters c and d must be specified for the Gamma prior of κ .

Arguments to specify

Model-related

<code>Y</code>	an $(n \times 2)$ -dimensional data.frame with columns y and δ , where $y = (y_1, \dots, y_n)^\top$ and $\delta = (\delta_1, \dots, \delta_n)^\top$.
<code>data</code>	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code>
<code>lin.pred</code>	a formula object that corresponds to the hazard $h(t_i x_i)$. Example: <code>lin.pred <- as.formula(~ x1 + x2)</code> , where <code>x1</code> and <code>x2</code> are columns of <code>data</code> .

Hyperparameters

<code>WB.ab</code>	a 2-vector of positive hyperparameters a and b of the prior distribution for the shape parameter α of the Weibull baseline hazard. Example: <code>WB.ab <- c(0.5, 0.01)</code> .
<code>WB.cd</code>	a 2-vector of positive hyperparameters c and d of the prior distribution for the scale parameter κ of the Weibull baseline hazard. Example: <code>WB.cd <- c(0.5, 0.05)</code> .

MCMC Settings

<code>numReps</code>	total number of scans
<code>thin</code>	extent of thinning, e.g. if <code>thin=10</code> retain every 10 th sample.
<code>burninPerc</code>	the proportion of burn-in (samples to be discarded before analyzing the data).
<code>mhProp_alpha_var</code>	the shape parameter α is updated using a Metropolis-Hastings random walk step generating proposals from a Gamma distribution with variance <code>mhProp_alpha_var</code> .

Starting Values

<code>startValues</code>	use <code>initiate.startValues_HReg(Y, lin.pred, data, model, beta = NULL, WB.alpha = NULL, WB.kappa = NULL)</code> which initiates starting values for β , α and κ in the Metropolis-Hastings algorithm if left unspecified. Users may set non-null starting values for any of these parameters.
--------------------------	---

Storage

<code>path</code>	name of the directory where results are stored. Can leave unspecified.
-------------------	--

Implementation

```
data(survData)
Y <- survData[,c(1,2)]
lin.pred <- as.formula(~ cov1 + cov2)
##
WB.ab <- c(0.5, 0.01) # prior parameters for alpha
WB.cd <- c(0.5, 0.05) # prior parameters for kappa
hyperParams <- list(WB=list(WB.ab=WB.ab, WB.cd=WB.cd))
##
numReps <- 2000
burninPerc <- 0.5
thin <- 10
mhProp_alpha_var <- 0.01
mcmc <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
             tuning=list(mhProp_alpha_var=mhProp_alpha_var))
##
myModel <- "Weibull"
myPath <- "Output/01-Results-WB/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_HReg(Y, lin.pred, survData, model=myModel)
startValues[[2]] <- initiate.startValues_HReg(Y, lin.pred, survData, model=myModel, WB.alpha=1.12)
##
fit_WB <- BayesSurv_HReg(Y, lin.pred, survData, cluster=NULL, model=myModel, hyperParams, startValues, mcmc, myPath)
summary(fit_WB)
plot(fit_WB, tseq=seq(from=0, to=30, by=5))
plot(fit_WB, tseq=seq(from=0, to=30, by=5), plot.est="BH")
```

Definition of the survival model

Let t_i denote the time-to-event of interest for individuals $i = 1, \dots, n$, subject to right censoring at time c_i . Let (y_i, δ_i, x_i) denote independent observations, where $y_i = \min(t_i, c_i)$, $\delta_i = \mathbb{1}(y_i \leq c_i)$, and x_i is a vector of covariates for individual i . The following Cox proportional hazards model is assumed

$$h(t_i|x_i) = h_0(t_i) \exp(x_i^\top \beta), \quad t_i > 0.$$

The baseline hazard h_0 is defined non-parametrically by a mixture of piecewise exponential functions as follows

$$\lambda_0(t) = \log h_0(t) = \sum_{k=1}^{K+1} \lambda_k \mathbb{1}\{t \in (s_{k-1}, s_k]\},$$

where λ_k is constant and the time interval between 0 and the largest observed failure time, denoted s_K , is partitioned into $K + 1$ disjoint intervals: $0 < s_1 < \dots < s_{K+1}$.

In the Bayesian framework, priors must be specified for the regression parameter, β , the number of intervals, K , and the partition points (s_1, \dots, s_{K+1}) , respectively. The following specifications are made

$$\begin{aligned} \pi(\beta) &\propto 1, \\ \lambda|K, \mu_\lambda, \sigma_\lambda^2 &\sim MVN_{K+1}(\mu_\lambda \mathbb{1}, \sigma_\lambda^2 \Sigma_\lambda) \\ K &\sim \text{Poisson}(\alpha), \\ \pi(s|K) &\propto \frac{(2K+1)! \prod_{k=1}^{K+1} (s_k - s_{k-1})}{(s_{K+1})^{(2K+1)}}, \\ \pi(\mu_\lambda) &\propto 1, \\ \sigma_\lambda^{-2} &\sim \text{Gamma}(a, b). \end{aligned}$$

The prior specification for λ follows a MVN-ICAR (see Supplemental Material to Lee, Haneuse, Schrag and Dominici, 2015). Note that K and s jointly form a time-homogeneous Poisson process prior for the partition.

Hyperparameters

The hyperparameter α must be specified for the prior distribution of K , as well as a and b , the rate and shape of the Gamma distributed hyperprior for σ_λ^{-2} .

Arguments to specify

Model-related	
<code>Y</code>	an $(n \times 2)$ -dimensional data.frame with columns y and δ , where $y = (y_1, \dots, y_n)^\top$ and $\delta = (\delta_1, \dots, \delta_n)^\top$.
<code>data</code>	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code>
<code>lin.pred</code>	a formula object that corresponds to the hazard $h(t_i x_i)$. Example: <code>lin.pred <- as.formula(~ x1 + x2)</code> , where <code>x1</code> and <code>x2</code> are columns of <code>data</code> .
Hyperparameters	
<code>PEM.ab</code>	a 2-vector of positive hyperparameters a and b of the prior distribution for σ_λ^{-2} . Example: <code>PEM.ab <- c(0.7, 0.7)</code> .
<code>PEM.alpha</code>	hyperparameter α of the prior distribution for K , which is one less than the number of partition points. Example: <code>PEM.alpha <- 10</code> .
MCMC Settings	
<code>numReps</code>	total number of scans
<code>thin</code>	extent of thinning, e.g. if <code>thin=10</code> retain every 10^{th} sample.
<code>burninPerc</code>	the proportion of burn-in (samples to be discarded before analyzing the data).
<code>C</code>	a numeric value for the proportion that determines the sum of probabilities choosing the birth and death moves. ¹
<code>delPert</code>	the perturbation parameter in the birth updates; values must be between 0 and 0.5. ¹
<code>rj.scheme</code>	<code>rj.scheme=1</code> : the birth update will draw the proposal time split from $1 : s_{max}$; <code>rj.scheme=2</code> : the birth update will draw the proposal time split from uniquely ordered failure times in the data.
<code>K_max</code>	the number of splits allowed in each iteration of the Metropolis-Hastings-Green algorithm.
<code>s_max</code>	the largest observed failure time, given by <code>s_max <- max(Y\$time[Y\$event==1])</code>
<code>time_lambda</code>	time points at which the λ is monitored for convergence. Example: <code>time_lambda <- seq(1, s_max, 1)</code> . The chains for these monitoring points can be found in <code>lambda.fin</code> in the chains of the <code>BayesSurv_HReg</code> object.
Starting Values	
<code>startValues</code>	use <code>initiate.startValues_HReg(Y, lin.pred, data, model, beta = NULL)</code> which initiates all necessary starting values in the Metropolis-Hastings-Green algorithm. Users may set non-null starting values for <code>beta</code> .
Storage	
<code>path</code>	name of the directory where results are stored. Can leave unspecified.

¹See Section A in Supplemental Material to Lee et al. (2015)

Implementation

```
data(survData)
Y <- survData[,c(1,2)]
lin.pred <- as.formula( ~ cov1 + cov2)
##
PEM.ab <- c(0.7, 0.7) # prior parameters for 1/sigma^2
PEM.alpha <- 10 # prior parameters for K
hyperParams <- list(PEM=list(PEM.ab, PEM.alpha))
##
numReps <- 2000
burninPerc <- 0.5
thin <- 10
C <- 0.2
delPert <- 0.5
rj.scheme <- 2
K_max <- 50
s_max <- max(Y$time[Y$event == 1])
time_lambda <- seq(1, s_max, 0.5)
mcmc <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
            tuning=list(C=C, delPert=delPert, rj.scheme=rj.scheme,
                       K_max=K_max, s_max=s_max, time_lambda=time_lambda) )
##
myModel <- "PEM"
myPath <- "Output/02-Results-PEM/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_HReg(Y, lin.pred, survData, model=myModel)
startValues[[2]] <- initiate.startValues_HReg(Y, lin.pred, survData, model=myModel,
beta=rep(0.1,2))
##
fit_PEM <- BayesSurv_HReg(Y, lin.pred, survData, cluster=NULL, model=myModel,
                        hyperParams, startValues, mcmc, path=myPath)
summary(fit_PEM)
plot(fit_PEM, tseq=seq(from=0, to=30, by=5))
plot(fit_PEM, tseq=seq(from=0, to=30, by=5), plot.est="BH")
```

Definition of the survival model

Let t_i denote the time-to-event of interest for individuals $i = 1, \dots, n$. In the presence of interval censoring, the time-to-event for the i^{th} subject satisfies $c_{ij} \leq t_i < c_{ij+1}$. Let $(c_{ij}, c_{ij+1}, L_i, x_i)$ denote independent observations, where L_i is the left-truncation time and x_i is a vector of covariates for individual i . The following AFT model is assumed

$$\log(t_i) = x_i^\top \beta + \epsilon_i, \quad t_i > 0.$$

We take ϵ_i to follow the Normal(μ, σ^2) distribution for ϵ_i for the parametric AFT model. In the Bayesian framework, priors must be specified for β , μ , and σ^2 . The following specifications are made

$$\begin{aligned} \pi(\beta, \mu) &\propto 1, \\ \sigma^2 &\sim \text{Inverse-Gamma}(a_\sigma, b_\sigma). \end{aligned}$$

Hyperparameters

The hyperparameters, a_σ and b_σ , must be specified for the prior distribution of σ^2 .

Arguments to specify

Model-related	
<code>Y</code>	an $(n \times 3)$ -dimensional data.frame with columns c_j , c_{j+1} , and L , where $c_j = (c_{ij}, \dots, c_{nj})^\top$, $c_{j+1} = (c_{ij+1}, \dots, c_{nj+1})^\top$ and $L = (L_1, \dots, L_n)^\top$.
<code>data</code>	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code>
<code>lin.pred</code>	a formula object that corresponds to the hazard $\log(t_i)$. Example: <code>lin.pred <- as.formula(~ x1 + x2)</code> , where <code>x1</code> and <code>x2</code> are columns of <code>data</code> .
Hyperparameters	
<code>LN.ab</code>	a 2-vector of positive hyperparameters a and b of the prior distribution for σ^2 . Example: <code>LN.ab <- c(0.7, 0.7)</code> .
MCMC Settings	
<code>numReps</code>	total number of scans
<code>thin</code>	extent of thinning, e.g. if <code>thin=10</code> retain every 10^{th} sample.
<code>burninPerc</code>	the proportion of burn-in (samples to be discarded before analyzing the data).
<code>beta.prop.var</code>	the parameter β is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance <code>beta.prop.var</code> .
<code>mu.prop.var</code>	the parameter μ is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance <code>mu.prop.var</code> .
<code>zeta.prop.var</code>	the parameter $\zeta = 1/\sigma^2$ is updated using a Metropolis-Hastings random walk step generating proposals from a log-Normal distribution with variance <code>zeta.prop.var</code> .
Starting Values	
<code>startValues</code>	use <code>initiate.startValues_AFT(Y, lin.pred, data, model, beta = NULL, y = NULL, LN.mu = NULL, LN.sigSq = NULL)</code> which initiates all necessary starting values in the Metropolis-Hastings algorithm. Users may set non-null starting values for <code>beta</code> , <code>y</code> , <code>LN.mu</code> , <code>LN.sigSq</code> .
Storage	
<code>path</code>	name of the directory where results are stored. Can leave unspecified.

Implementation

```
data(survData)
Y <- matrix(NA, dim(survData)[1], 3)
Y[,1] <- Y[,2] <- survData[,1]
Y[which(scrData[,2] == 0),2] <- Inf
Y[,3] <- rep(0, dim(survData)[1])
lin.pred <- as.formula( ~ cov1 + cov2)
##
LN.ab <- c(0.3, 0.3)
hyperParams <- list(LN=list(LN.ab=LN.ab))
##
numReps <- 1000
thin <- 10
burninPerc <- 0.5
beta.prop.var <- 0.01
mu.prop.var <- 0.1
zeta.prop.var <- 0.1
mcmcParams <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
tuning=list(beta.prop.var=beta.prop.var, mu.prop.var=mu.prop.var,
zeta.prop.var=zeta.prop.var))
##
myModel <- "LN"
myPath <- "Output/01-Results-LN/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_AFT(Y, lin.pred, survData, model=myModel)
startValues[[2]] <- initiate.startValues_AFT(Y, lin.pred, survData, model=myModel,
beta=c(0.05, -0.05))
##
```

```
fit_LN <- BayesSurv_AFT(Y, lin.pred, survData, model=myModel, hyperParams,  
startValues, mcmcParams, path=myPath)
```

```
summary(fit_LN)
```

```
plot(fit_LN, time = seq(0, 35, 1), tseq=seq(from=0, to=30, by=5))
```

```
plot(fit_LN, time = seq(0, 35, 1), tseq=seq(from=0, to=30, by=5), plot.est = "BH")
```

Definition of the survival model

Let t_i denote the time-to-event of interest for individuals $i = 1, \dots, n$. Considering interval censoring, the time-to-event for the i^{th} subject satisfies $c_{ij} \leq t_i < c_{ij+1}$. Let $(c_{ij}, c_{ij+1}, L_i, x_i)$ denote independent observations, where L_i is the left-truncation time and x_i is a vector of covariates for individual i . The following AFT model is assumed

$$\log(t_i) = x_i^\top \beta + \epsilon_i, \quad t_i > 0,$$

where ϵ_i is assumed to be taken as draws from the DPM of normal distributions:

$$\begin{aligned} \epsilon_i | r_i &\sim \text{Normal}(\mu_{r_i}, \sigma_{r_i}^2), \\ (\mu_r, \sigma_r^2) &\sim G_0, \quad \text{for } r = 1, \dots, M, \\ r_i | p &\sim \text{Discrete}(r_i | p_1, \dots, p_M), \\ p &\sim \text{Dirichlet}(\tau/M, \dots, \tau/M). \end{aligned}$$

In the Bayesian framework, priors must be specified for the unknown parameters. We take the G_0 as a normal distribution centered at μ_0 with a variance σ_0^2 for μ_r and an inverse-Gamma(a_σ, b_σ) for σ_r^2 . For β , we adopt non-informative flat priors on the real line. Finally, we specify a Gamma(a_τ, b_τ) hyperprior for the precision parameter τ .

Hyperparameters

The hyperparameter $(\mu_0, \sigma_0^2, a_\sigma, b_\sigma)$ must be specified for the centering distribution G_0 , as well as a_τ and b_τ , the rate and shape of the Gamma distributed hyperprior for τ .

Arguments to specify

Model-related	
<code>Y</code>	an $(n \times 3)$ -dimensional data.frame with columns c_j, c_{j+1} , and L , where $c_j = (c_{ij}, \dots, c_{nj})^\top$, $c_{j+1} = (c_{ij+1}, \dots, c_{nj+1})^\top$ and $L = (L_1, \dots, L_n)^\top$.
<code>data</code>	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code>
<code>lin.pred</code>	a formula object that corresponds to the hazard $\log(t_i)$. Example: <code>lin.pred <- as.formula(~ x1 + x2)</code> , where <code>x1</code> and <code>x2</code> are columns of <code>data</code> .
Hyperparameters	
<code>DPM.mu</code>	a hyperparameter μ_0 of the centering distribution G_0 .
<code>DPM.sigSq</code>	a positive-valued hyperparameter σ_0^2 of the centering distribution G_0 .
<code>DPM.ab</code>	a 2-vector of positive hyperparameters a_σ and b_σ of the centering distribution G_0 .
<code>Tau.ab</code>	a 2-vector of positive hyperparameters a_τ and b_τ of the hyperprior distribution for τ . Example: <code>Tau.ab <- c(1.5, 0.0125)</code> .
MCMC Settings	
<code>numReps</code>	total number of scans
<code>thin</code>	extent of thinning, e.g. if <code>thin=10</code> retain every 10 th sample.
<code>burninPerc</code>	the proportion of burn-in (samples to be discarded before analyzing the data).
<code>beta.prop.var</code>	the parameter β is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance <code>beta.prop.var</code> .
<code>mu.prop.var</code>	the parameter μ_r is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance <code>mu.prop.var</code> .
<code>zeta.prop.var</code>	the parameter $\zeta_r = 1/\sigma_r^2$ is updated using a Metropolis-Hastings random walk step generating proposals from a log-Normal distribution with variance <code>zeta.prop.var</code> .
Starting Values	
<code>startValues</code>	use <code>initiate.startValues_AFT(Y, lin.pred, data, model, beta = NULL, y = NULL, DPM.class = NULL, DPM.mu = NULL, DPM.zeta=NULL, DPM.tau=NULL)</code> which initiates all necessary starting values in the Metropolis-Hastings algorithm. Users may set non-null starting values for <code>beta, y, DPM.class, DPM.mu, DPM.zeta, DPM.tau</code> .
Storage	
<code>path</code>	name of the directory where results are stored. Can leave unspecified.

Implementation

```
data(survData)
Y <- matrix(NA, dim(survData)[1], 3)
Y[,1] <- Y[,2] <- survData[,1]
Y[which(scrData[,2] == 0),2] <- Inf
Y[,3] <- rep(0, dim(survData)[1])
lin.pred <- as.formula( ~ cov1 + cov2)
##
DPM.mu <- log(12)
DPM.sigSq <- 100
DPM.ab <- c(2, 1)
Tau.ab <- c(1.5, 0.0125)
hyperParams <- list(DPM=list(DPM.mu=DPM.mu, DPM.sigSq=DPM.sigSq, DPM.ab=DPM.ab, Tau.ab=Tau.ab))
##
numReps <- 1000
thin <- 10
burninPerc <- 0.5
beta.prop.var <- 0.01
mu.prop.var <- 0.1
zeta.prop.var <- 0.1
mcmcParams <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
tuning=list(beta.prop.var=beta.prop.var, mu.prop.var=mu.prop.var,
zeta.prop.var=zeta.prop.var))
##
myModel <- "DPM"
myPath <- "Output/O2-Results-DPM/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_AFT(Y, lin.pred, survData, model=myModel)
startValues[[2]] <- initiate.startValues_AFT(Y, lin.pred, survData, model=myModel,
beta=c(0.05, -0.05))
##
fit_DPM <- BayesSurv_AFT(Y, lin.pred, survData, model=myModel, hyperParams,
startValues, mcmcParams, path=myPath)

summary(fit_DPM)
plot(fit_DPM, time = seq(0, 35, 1), tseq=seq(from=0, to=30, by=5))
plot(fit_DPM, time = seq(0, 35, 1), tseq=seq(from=0, to=30, by=5), plot.est = "BH")
```


Definition of the survival model

Let t_{ji} denote the time-to-event of interest for individuals $i = 1, \dots, n_j$ in cluster $j = 1, \dots, J$, subject to right censoring at time c_{ji} . Let $(y_{ji}, \delta_{ji}, x_{ji})$ denote independent observations, where $y_{ji} = \min(t_{ji}, c_{ji})$, $\delta_{ji} = \mathbb{1}(y_{ji} \leq c_{ji})$, and x_{ji} is a vector of covariates for individual i . The following Cox proportional hazards model is assumed

$$h(t_{ji}|x_{ji}) = h_0(t_{ji}) \exp(x_{ji}^\top \beta + V_j), \quad t_{ji} > 0,$$

where the V_j 's are cluster-specific random effects and the baseline hazard h_0 is defined parametrically by a Weibull hazard, $h_0(t) = \alpha \kappa t^{\alpha-1}$.

In the Bayesian framework, priors must be specified for the regression parameter, β , the cluster-specific random effects, V_j , and the shape and scale parameters of baseline hazard function, α and κ , respectively. The prior distributions for β , α and κ are given below.

$$\begin{aligned} \pi(\beta) &\propto 1, \\ \pi(\alpha) &\sim \text{Gamma}(a, b), \\ \pi(\kappa) &\sim \text{Gamma}(c, d). \end{aligned}$$

We provide two possible prior specifications for the cluster-specific random effects below.

$$\begin{aligned} V_j &\sim \text{Normal}(0, \sigma^2), & V_j|m_j &\sim \text{Normal}(\mu_m, \sigma_m^2), \\ \zeta = \frac{1}{\sigma^2} &\sim \text{Gamma}(a_N, b_N), & (\mu_m, \sigma_m^2) &\sim G_0, \text{ for } m = 1, \dots, M, \\ & & m_j|p &\sim \text{Discrete}(m_j|p_1, \dots, p_M), \\ & & p &\sim \text{Dirichlet}(\tau/M, \dots, \tau/M), \\ & & \tau &\sim \text{Gamma}(a_\tau, b_\tau). \end{aligned}$$

In the first column, the individual specific-random effects are assumed to be $iid N(0, \sigma^2)$. In the second column, the cluster-specific random effects are drawn from a mixture of M normal distributions each with mean and variance (μ_m, σ_m^2) which are distributed as a multivariate Normal/Inverse-Gamma (NIG), denoted by G_0 ; we refer to this as the Dirichlet process mixture (DPM) prior. The probability density of G_0 is defined by the product

$$f_{\text{NIG}}(\mu, \sigma^2 | \mu_0, \zeta_0, a_0, b_0) = f_{\text{Normal}}(\mu | \mu_0, 1/\zeta_0^2) \times f_{\text{Gamma}}(\zeta = 1/\sigma^2 | a_0, b_0).$$

We assume $\mu_0 = 0$ and $\zeta_0 = 1$.

Hyperparameters

a, b	: shape and rate of Gamma prior for α
c, d	: shape and rate of Gamma prior for κ
a_N, b_N	: mean and variance of normal prior for V_j
a_0, b_0	: shape and rate of Gamma component of the prior distribution, G_0 , of (μ_m, σ_m^2) (DPM prior)
a_τ, b_τ	: shape and rate of Gamma hyperprior for τ (DPM prior)

Arguments to specify

Model-related

<code>Y</code>	an $(n \times 2)$ -dimensional data.frame with columns y and δ , where $y = (y_1, \dots, y_n)^\top$ and $\delta = (\delta_1, \dots, \delta_n)^\top$.
<code>data</code>	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code>
<code>lin.pred</code>	a formula object that corresponds to the hazard $h(t_{ji} x_{ji})$. Example: <code>lin.pred <- as.formula(~ x1 + x2)</code> , where <code>x1</code> and <code>x2</code> are columns of <code>data</code> .
<code>model</code>	a character vector that specifies the type of components in the model. Use <code>model <- c("Weibull", "Normal")</code> for Normal prior for V_j and use <code>model <- c("Weibull", "DPM")</code> for DPM prior.
<code>cluster</code>	an n -vector of cluster information where cluster membership corresponds to one of the positive integers $1, \dots, J$.

Hyperparameters

<code>WB.ab</code>	a 2-vector of positive hyperparameters a and b of the prior distribution for the shape parameter α of the Weibull baseline hazard. Example: <code>WB.ab <- c(0.5, 0.01)</code> .
<code>WB.cd</code>	a 2-vector of positive hyperparameters c and d of the prior distribution for the scale parameter κ of the Weibull baseline hazard. Example: <code>WB.cd <- c(0.5, 0.05)</code> .

Normal prior for V_j

<code>Normal.ab</code>	a 2-vector of positive hyperparameters a_N and b_N of the prior for $1/\sigma^2$, the precision of the normally distributed cluster-specific random effects. Example: <code>Normal.ab <- c(0.5, 0.01)</code> .
------------------------	--

DPM prior for V_j

<code>DPM.ab</code>	a 2-vector of positive hyperparameters a_0 and b_0 of the prior for (μ_m, σ_m^2) , the parameters of the normally distributed cluster-specific random effects. Example: <code>DPM.ab <- c(0.5, 0.01)</code> .
<code>aTau</code>	a positive-valued hyperparameter corresponding to the shape parameter, a_τ , of the Gamma prior of τ .
<code>bTau</code>	a positive-valued hyperparameter corresponding to the rate parameter, b_τ , of the Gamma prior of τ .

MCMC Settings

<code>numReps</code>	total number of scans
<code>thin</code>	extent of thinning, e.g. if <code>thin=10</code> retain every 10 th sample.
<code>burninPerc</code>	the proportion of burn-in (samples to be discarded before analyzing the data).
<code>mhProp_alpha_var</code>	the shape parameter α is updated using a Metropolis-Hastings random walk algorithm which generates proposals from a Gamma distribution with variance <code>mhProp_alpha_var</code> .
<code>mhProp_V_var</code>	the cluster-specific random effects, V_{ji} , are updated using a Metropolis-Hastings random walk algorithm which generates proposals from a Normal distribution with variance <code>mhProp_V_var</code>

Starting Values

<code>startValues</code>	use <code>initiate.startValues_HReg(Y, lin.pred, data, model, cluster, beta = NULL, WB.alpha = NULL, WB.kappa = NULL, V.j = NULL, Normal.zeta = NULL, DPM.class = NULL, DPM.tau = NULL)</code> which initiates starting values for β , α , κ , V_j , ζ (in the DPM model for V_j) and τ in the Metropolis-Hastings-Green algorithm if left unspecified; <code>DPM.class</code> sets the starting value for class membership in the DPM model. Users may set non-null starting values for any of these parameters.
--------------------------	---

Storage

path name of the directory where results are stored. Can leave unspecified.
storeV a TRUE/FALSE logical constant indicating storage of V_j values.

Implementation

```
data(survData)
Y <- survData[,c(1,2)]
cluster <- survData[,3]
lin.pred <- as.formula( ~ cov1 + cov2)
##
WB.ab <- c(0.5, 0.01) # prior parameters for alpha
WB.cd <- c(0.5, 0.05) # prior parameters for kappa
Normal.ab <- c(0.5, 0.01) # for Normal random effects
DPM.ab <- c(0.5, 0.01) # For DPM
  aTau <- 1.5
  bTau <- 0.0125
hyperParams.WB.Normal <- list(WB=list(WB.ab=WB.ab, WB.cd=WB.cd),
  Normal=list(Normal.ab=Normal.ab))
hyperParams.WB.DPM <- list(WB=list(WB.ab=WB.ab, WB.cd=WB.cd),
  DPM=list(DPM.ab=DPM.ab, aTau=aTau, bTau=bTau))
##
numReps <- 2000
burninPerc <- 0.5
thin <- 10
mhProp_alpha_var <- 0.01
mhProp_V_var <- 0.05
storeV <- TRUE
mcmc.WB <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
  storage=list(storeV=storeV),
  tuning=list(mhProp_alpha_var=mhProp_alpha_var, mhProp_V_var=mhProp_V_var))
##
myModel.WB.Normal <- c("Weibull","Normal")
myPath.WB.Normal <- "Output/03-Results-WB_Normal/"
startValues.WB.Normal <- vector("list", 2)
startValues.WB.Normal[[1]] <- initiate.startValues_HReg(Y, lin.pred, survData, cluster, model=myModel.WB.Normal)
startValues.WB.Normal[[2]] <- initiate.startValues_HReg(Y, lin.pred, survData, cluster, model=myModel.WB.Normal, WB.alpha=1.12)
##
fit_WB_Normal <- BayesSurv_HReg(Y, lin.pred, survData, cluster, model=myModel.WB.Normal, hyperParams.WB.Normal,
  startValues.WB.Normal, mcmc.WB, myPath.WB.Normal)
summary(fit_WB_Normal)
plot(fit_WB_Normal, tseq=seq(from=0, to=30, by=5))
plot(fit_WB_Normal, tseq=seq(from=0, to=30, by=5), plot.est="BH")
##
myModel.WB.DPM <- c("Weibull","DPM")
myPath.WB.DPM <- "Output/04-Results-WB_DPM/"
startValues.WB.DPM <- vector("list", 2)
startValues.WB.DPM[[1]] <- initiate.startValues_HReg(Y, lin.pred, survData, cluster, model=myModel.WB.DPM)
startValues.WB.DPM[[2]] <- initiate.startValues_HReg(Y, lin.pred, survData, cluster, model=myModel.WB.DPM, Normal.zeta=0.95)
##
fit_WB_DPM <- BayesSurv_HReg(Y, lin.pred, survData, cluster, model=myModel.WB.DPM, hyperParams.WB.DPM,
  startValues.WB.DPM, mcmc.WB, myPath.WB.DPM)
summary(fit_WB_DPM)
plot(fit_WB_DPM, tseq=seq(from=0, to=30, by=5))
plot(fit_WB_DPM, tseq=seq(from=0, to=30, by=5), plot.est="BH")
```

Definition of the survival model

Let t_{ji} denote the time-to-event of interest for individuals $i = 1, \dots, n_j$ in cluster $j = 1, \dots, J$, subject to right censoring at time c_{ji} . Let $(y_{ji}, \delta_{ji}, x_{ji})$ denote independent observations, where $y_{ji} = \min(t_{ji}, c_{ji})$, $\delta_{ji} = \mathbb{1}(y_{ji} \leq c_{ji})$, and x_{ji} is a vector of covariates for individual i . The following Cox proportional hazards model is assumed

$$h(t_{ji}|x_{ji}) = h_0(t_{ji}) \exp(x_{ji}^\top \beta + V_j), \quad t_{ji} > 0,$$

The baseline hazard h_0 is defined non-parametrically by a mixture of piecewise exponential functions as follows

$$\lambda_0(t) = \log h_0(t) = \sum_{k=1}^{K+1} \lambda_k \mathbb{1}\{t \in (s_{k-1}, s_k]\},$$

where λ_k is constant and the time interval between 0 and the largest observed failure time, denoted s_k , is partitioned into $K + 1$ disjoint intervals: $0 < s_1 < \dots < s_{K+1}$.

In the Bayesian framework, priors must be specified for the regression parameter, β , the number of intervals, K , and the partition points (s_1, \dots, s_{K+1}) , respectively. The following specifications are made

$$\begin{aligned} \pi(\beta) &\propto 1, \\ \lambda|K, \mu_\lambda, \sigma_\lambda^2 &\sim MVN_{K+1}(\mu_\lambda \mathbb{1}, \sigma_\lambda^2 \Sigma_\lambda) \\ K &\sim \text{Poisson}(\alpha), \\ \pi(s|K) &\propto \frac{(2K+1)! \prod_{k=1}^{K+1} (s_k - s_{k-1})}{(s_{K+1})^{(2K+1)}}, \\ \pi(\mu_\lambda) &\propto 1, \\ \sigma_\lambda^{-2} &\sim \text{Gamma}(a, b). \end{aligned}$$

The prior specification for λ follows a MVN-ICAR (see Supplemental Material to Lee, Haneuse, Schrag and Dominici, 2015). Note that K and s jointly form a time-homogeneous Poisson process prior for the partition.

We provide two possible prior specifications for the cluster-specific random effects below.

$$\begin{array}{ll} V_j & \sim \text{Normal}(0, \sigma^2), \\ \zeta = \frac{1}{\sigma^2} & \sim \text{Gamma}(a_N, b_N). \end{array} \qquad \begin{array}{ll} V_j|m_j & \sim \text{Normal}(\mu_m, \sigma_m^2), \\ (\mu_m, \sigma_m^2) & \sim G_0, \text{ for } m = 1, \dots, M, \\ m_j|p & \sim \text{Discrete}(m_j|p_1, \dots, p_M), \\ p & \sim \text{Dirichlet}(\tau/M, \dots, \tau/M), \\ \tau & \sim \text{Gamma}(a_\tau, b_\tau). \end{array}$$

In the first column, the individual specific-random effects are assumed to be iid $N(0, \sigma^2)$. In the second column, the cluster-specific random effects are drawn from a mixture of M normal distributions each with mean and variance (μ_m, σ_m^2) which are distributed as a multivariate Normal/Inverse-Gamma (NIG), denoted by G_0 ; we refer to this as the Dirichlet process mixture (DPM) prior. The probability density of G_0 is defined by the product

$$f_{\text{NIG}}(\mu, \sigma^2 | \mu_0, \zeta_0, a_0, b_0) = f_{\text{Normal}}(\mu | \mu_0, 1/\zeta_0^2) \times f_{\text{Gamma}}(\zeta = 1/\sigma^2 | a_0, b_0).$$

We assume $\mu_0 = 0$ and $\zeta_0 = 1$.

Hyperparameters

α	: hyperparameter of K
a, b	: shape and rate of Gamma prior for σ_λ^{-2}
a_N, b_N	: mean and variance of normal prior for V_j
a_0, b_0	: shape and rate of Gamma component of the prior distribution, G_0 , of (μ_m, σ_m^2)
a_τ, b_τ	: shape and rate of Gamma hyperprior for τ

Arguments to specify**Model-related**

<code>Y</code>	an $(n \times 2)$ -dimensional data.frame with columns y and δ , where $y = (y_1, \dots, y_n)^\top$ and $\delta = (\delta_1, \dots, \delta_n)^\top$.
<code>data</code>	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code>
<code>lin.pred</code>	a formula object that corresponds to the hazard $h(t_{ji} x_{ji})$. Example: <code>lin.pred <- as.formula(~ x1 + x2)</code> , where <code>x1</code> and <code>x2</code> are columns of <code>data</code> .
<code>model</code>	a character vector that specifies the type of components in the model. Use <code>model <- c("PEM", "DPM")</code> .
<code>cluster</code>	an n -vector of cluster information where cluster membership corresponds to one of the positive integers $1, \dots, J$.

Hyperparameters

<code>PEM.ab</code>	a 2-vector of positive hyperparameters a and b of the prior distribution for σ_λ^{-2} . Example: <code>PEM.ab <- c(0.7, 0.7)</code> .
<code>PEM.alpha</code>	hyperparameter α of the prior distribution for K , which is one less than the number of partition points. Example: <code>PEM.alpha <- 10</code> .

Normal prior for V_j

<code>Normal.ab</code>	a 2-vector of positive hyperparameters a_N and b_N of the prior for $1/\sigma^2$, the precision of the normally distributed cluster-specific random effects. Example: <code>Normal.ab <- c(0.5, 0.01)</code> .
------------------------	--

DPM prior for V_j

<code>DPM.ab</code>	a 2-vector of positive hyperparameters a_0 and b_0 of the prior for (μ_m, σ_m^2) , the parameters of the normally distributed cluster-specific random effects. Example: <code>DPM.ab <- c(0.5, 0.01)</code> .
<code>aTau</code>	a positive-valued hyperparameter corresponding to the shape parameter, a_τ , of the Gamma prior of τ .
<code>bTau</code>	a positive-valued hyperparameter corresponding to the rate parameter, b_τ , of the Gamma prior of τ .

MCMC Settings

numReps	total number of scans
thin	extent of thinning, e.g. if <code>thin=10</code> retain every 10 th sample.
burninPerc	the proportion of burn-in (samples to be discarded before analyzing the data).
mhProp_V_var	the cluster-specific random effects, V_{ji} , are updated using a Metropolis-Hastings random walk algorithm which generates proposals from a Normal distribution with variance <code>mhProp_V_var</code>
C	a numeric value for the proportion that determines the sum of probabilities choosing the birth and death moves. ²
delPert	the perturbation parameter in the birth updates; values must be between 0 and 0.5. ²
rj.scheme	<code>rj.scheme=1</code> : the birth update will draw the proposal time split from 1 : s_{max} ; <code>rj.scheme=2</code> : the birth update will draw the proposal time split from uniquely ordered failure times in the data.
K_max	the number of splits allowed in each iteration of the Metropolis-Hastings-Green algorithm.
s_max	the largest observed failure time, given by <code>s_max <- max(Y\$time[Y\$event==1])</code>
time_lambda	time points at which the λ is monitored for convergence. Example: <code>time_lambda <- seq(1, s_max, 1)</code> . The chains for these monitoring points can be found in <code>lambda.fin</code> in the chains of the <code>BayesSurv</code> object.

Starting Values

startValues	use <code>initiate.startValues_HReg(Y, lin.pred, data, model, beta = NULL, V.j=NULL, Normal.zeta=NULL, DPM.class=NULL, DPM.tau=NULL)</code> which initiates starting values for β , V_j , ζ (in the DP model for V_j) and τ in the Metropolis-Hastings-Green algorithm if left unspecified; <code>DPM.class</code> sets the starting value for class membership in the DP model. Users may set non-null starting values for any of these parameters.
-------------	--

Storage

path	name of the directory where results are stored. Can leave unspecified.
storeV	a TRUE/FALSE logical constant indicating storage of V_j values.

Implementation

```
data(survData)
Y <- survData[,c(1,2)]
cluster <- survData[,3]
lin.pred <- as.formula( ~ cov1 + cov2)
##
PEM.ab <- c(0.7, 0.7) # prior parameters for 1/sigma^2
PEM.alpha <- 10 # prior parameters for K
Normal.ab <- c(0.5, 0.01) # for Normal random effects
DPM.ab <- c(0.5, 0.01) # For DPM
  aTau <- 1.5
  bTau <- 0.0125
hyperParams.PEM.Normal <- list(PEM=list(PEM.ab=PEM.ab, PEM.alpha=PEM.alpha),
  Normal=list(Normal.ab=Normal.ab))
hyperParams.PEM.DPM <- list(PEM=list(PEM.ab=PEM.ab, PEM.alpha=PEM.alpha),
  DPM=list(DPM.ab=DPM.ab, aTau=aTau, bTau=bTau))
##
numReps <- 2000
burninPerc <- 0.5
thin <- 10
mhProp_V_var <- 0.05
storeV <- TRUE
C <- 0.2
delPert <- 0.5
rj.scheme <- 2
K_max <- 50
s_max <- max(Y$time[Y$event == 1])
time_lambda <- seq(1, s_max, 0.5)
mcmc.PEM <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
  storage=list(storeV=storeV),
  tuning=list(mhProp_V_var=mhProp_V_var, C=C, delPert=delPert, rj.scheme=rj.scheme,
    K_max=K_max, s_max=s_max, time_lambda=time_lambda) )
##
myModel.PEM.Normal <- c("PEM", "Normal")
myPath.PEM.Normal <- "Output/05-Results-PEM_Normal/"
startValues.PEM.Normal <- vector("list", 2)
startValues.PEM.Normal[[1]] <- initiate.startValues_HReg(Y, lin.pred, survData, cluster, model=myModel.PEM.Normal)
startValues.PEM.Normal[[2]] <- initiate.startValues_HReg(Y, lin.pred, survData, cluster, model=myModel.PEM.Normal, Normal.zeta=0.95)
##
fit_PEM_Normal <- BayesSurv_HReg(Y, lin.pred, survData, cluster, model=myModel.PEM.Normal, hyperParams.PEM.Normal,
  startValues.PEM.Normal, mcmc.PEM, path=myPath.PEM.Normal)
summary(fit_PEM_Normal)
plot(fit_PEM_Normal, tseq=seq(from=0, to=30, by=5))
plot(fit_PEM_Normal, tseq=seq(from=0, to=30, by=5), plot.est="BH")
##
myModel.PEM.DPM <- c("PEM", "DPM")
myPath.PEM.DPM <- "Output/06-Results-PEM_DPM/"
startValues.PEM.DPM <- vector("list", 2)
startValues.PEM.DPM[[1]] <- initiate.startValues_HReg(Y, lin.pred, survData, cluster, model=myModel.PEM.DPM)
startValues.PEM.DPM[[2]] <- initiate.startValues_HReg(Y, lin.pred, survData, cluster, model=myModel.PEM.DPM, Normal.zeta=0.95)
##
```

²See Section A in Supplemental Material to Lee et al. (2015)

```
fit_PEM_DPM <- BayesSurv_HReg(Y, lin.pred, survData, cluster, model=myModel.PEM.DPM, hyperParams.PEM.DPM,  
  startValues.PEM.DPM, mcmc.PEM, path=myPath.PEM.DPM)  
summary(fit_PEM_DPM)  
plot(fit_PEM_DPM, tseq=seq(from=0, to=30, by=5))  
plot(fit_PEM_DPM, tseq=seq(from=0, to=30, by=5), plot.est="BH")
```

Definition of the survival model

Let t_{i1} and t_{i2} denote the time to nonterminal event and terminal event from subject $i = 1, \dots, n$, subject to right censoring at time c_i . Let $(y_{i1}, y_{i2}, \delta_{i1}, \delta_{i2}, x_i)$ denote independent observations, where $y_{i1} = \min(t_{i1}, t_{i2}, c_i)$, $\delta_{i1} = \mathbb{1}\{t_{i1} \leq \min(t_{i2}, c_i)\}$, $y_{i2} = \min(t_{i2}, c_i)$, $\delta_{i2} = \mathbb{1}\{t_{i2} \leq c_i\}$, and x_i is a vector of covariates for individual i . The independent semi-competing risks data are assumed to arise from an illness-death model system with transitions that are modeled through the following three hazard functions:

$$h_1(t_{i1} | \gamma_{ji}, x_{i1}) = \gamma_{ji} h_{01}(t_{i1}) \exp(x_{i1}^\top \beta_1), \quad t_{i1} > 0, \quad (1)$$

$$h_2(t_{i2} | \gamma_{ji}, x_{i2}) = \gamma_{ji} h_{02}(t_{i2}) \exp(x_{i2}^\top \beta_2), \quad t_{i2} > 0, \quad (2)$$

$$h_3(t_{i2} | t_{i1}, \gamma_{ji}, x_{i3}) = \gamma_{ji} h_{03}(t_{i2}) \exp(x_{i3}^\top \beta_3), \quad t_{i2} > 0, \quad (3)$$

where γ_{ji} is a subject-specific frailty with vectors of covariates x_{i1} , x_{i2} and x_{i3} which are subsets of x_i . The baseline hazard functions are defined parametrically by Weibull hazards of the form $h_{0g}(t) = \alpha_g \kappa_g t^{\alpha_g - 1}$, for $g \in \{1, 2, 3\}$. The baseline hazard function h_{03} is assumed to be Markov with respect to t_{i1} ; we will refer to the set of conditional hazard functions in (13)-(15) as the Markov model. Alternatively, we consider modeling h_3 as follows:

$$h_3(t_{i2} | t_{i1}, \gamma_{ji}, x_{i3}) = \gamma_{ji} h_{03}(t_{i2} - t_{i1}) \exp(x_{i3}^\top \beta_3), \quad 0 < t_{i1} < t_{i2}. \quad (4)$$

We will refer to the set of conditional hazard functions in (13), (14) and (16) as the semi-Markov model.

In the Bayesian framework, priors must be specified for the regression parameter, β_g , the shape and scale parameters of baseline hazard function, α_g and κ_g , and the frailty parameter, γ_{ji} , respectively, for $g \in \{1, 2, 3\}$. The following specifications are made

$$\begin{aligned} \pi(\beta_g) &\propto 1, \\ \alpha_g &\sim \text{Gamma}(a_g, b_g), \\ \kappa_g &\sim \text{Gamma}(c_g, d_g), \\ \gamma_{ji} | \theta &\sim \text{Gamma}(\theta^{-1}, \theta^{-1}), \\ \theta^{-1} &\sim \text{Gamma}(\psi, \omega). \end{aligned}$$

Hyperparameters

a_g, b_g	: shape and rate of Gamma prior for α_g for $g \in \{1, 2, 3\}$
c_g, d_g	: shape and rate of Gamma prior for κ_g for $g \in \{1, 2, 3\}$
ψ	: the shape of Gamma prior for θ^{-1}
ω	: the rate of Gamma prior for θ^{-1}

Arguments to specify**Model-related**

Y	an $(n \times 4)$ -dimensional data.frame with columns $y_1, \delta_1, y_2, \delta_2$.
model	c("Markov", "Weibull") for Markov definition of h_3 in (15); c("semi-Markov", "Weibull") for semi-Markov definition of h_3 in (16).
data	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code> below.
lin.pred	a list of three formula objects that correspond to h_g , for $g \in \{1, 2, 3\}$. Example: <code>form1 <- as.formula(~ x1 + x2 + x3)</code> , <code>form2 <- as.formula(~ x1)</code> , <code>form3 <- as.formula(~ x1 + x4)</code> , <code>lin.pred <- list(form1, form2, form3)</code> .

Hyperparameters

WB.ab1	a 2-vector of positive hyperparameters a_1 and b_1 of the prior distribution for the shape parameter α_1 of the Weibull baseline hazard. Example: <code>WB.ab1 <- c(0.5, 0.01)</code> .
WB.ab2	a 2-vector of positive hyperparameters a_2 and b_2 of the prior for α_2 .
WB.ab3	a 2-vector of positive hyperparameters a_3 and b_3 of the prior for α_3 .
WB.cd1	a 2-vector of positive hyperparameters c_1 and d_1 of the prior distribution for the scale parameter κ_1 of the Weibull baseline hazard. Example: <code>WB.cd1 <- c(0.5, 0.05)</code> .
WB.cd2	a 2-vector of positive hyperparameters c_2 and d_2 of the prior for κ_2 .
WB.cd3	a 2-vector of positive hyperparameters c_3 and d_3 of the prior for κ_3 .
theta	a 2-vector of positive hyperparameters ψ and ω for the hyperprior θ .

MCMC Settings

numReps	total number of scans
thin	extent of thinning, e.g. if <code>thin=10</code> retain every 10^{th} sample.
burninPerc	the proportion of burn-in (samples to be discarded before analyzing the data).
mhProp_theta_var	the parameter θ is updated using a Metropolis-Hastings random walk step generating proposals from a Gamma distribution with variance <code>mhProp_theta_var</code> .
mhProp_alphag_var	a 3-vector which specifies the variances of the three random walk Metropolis-Hastings Gamma proposal distributions corresponding to $\alpha_1, \alpha_2, \alpha_3$.

Starting Values

startValues	use <code>initiate.startValues_HReg(Y, lin.pred, data, model, beta1 = NULL, beta2 = NULL, beta3 = NULL, gamma.ji=NULL, theta = NULL, WB.alpha = NULL, WB.kappa = NULL)</code> which initiates starting values for $\beta_g, \gamma_{ji}, \theta, \alpha_g$, and κ_g in the Metropolis-Hastings algorithm if left unspecified. Users may set non-null starting values for any of these parameters.
-------------	---

Storage

path	name of the directory where results are stored. Can leave unspecified.
nGam_save	the number of γ to be stored.

Implementation

```
data(scrData)
Y <- scrData[,c(1,2,3,4)]
form1 <- as.formula( ~ x1 + x2 + x3)
form2 <- as.formula( ~ x1 + x2)
form3 <- as.formula( ~ x1 + x2)
lin.pred <- list(form1, form2, form3)
##
WB.ab1 <- c(0.5, 0.01)
WB.ab2 <- c(0.5, 0.01)
WB.ab3 <- c(0.5, 0.01)
WB.cd1 <- c(0.5, 0.05)
WB.cd2 <- c(0.5, 0.05)
WB.cd3 <- c(0.5, 0.05)
theta <- c(0.7, 0.7) # prior params for 1/theta
hyperParams <- list(theta=theta,
                    WB=list(WB.ab1=WB.ab1, WB.ab2=WB.ab2, WB.ab3=WB.ab3,
                           WB.cd1=WB.cd1, WB.cd2=WB.cd2, WB.cd3=WB.cd3))
##
numReps <- 2000
thin <- 10
burninPerc <- 0.25
mhProp_theta_var <- 0.05
mhProp_alphag_var <- c(0.01, 0.01, 0.01)
nGam_save <- 0
mcmc.WB <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
               storage=list(nGam_save=nGam_save),
               tuning=list(mhProp_theta_var=mhProp_theta_var, mhProp_alphag_var=mhProp_alphag_var))
##
myModel <- c("semi-Markov", "Weibull")
myPath <- "Output/01-Results-WB/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel)
startValues[[2]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, theta = 0.23)
##
fit_WB <- BayesID_HRegY, lin.pred, scrData, cluster=NULL, model=myModel,
        hyperParams, startValues, mcmc.WB, path=myPath)
fit_WB
summary(fit_WB)
plot(fit_WB, tseq=seq(from=0, to=30, by=5))
plot(fit_WB, tseq=seq(from=0, to=30, by=5), plot.est = "BH")
```

Definition of the survival model

Let t_{i1} and t_{i2} denote the time to nonterminal event and terminal event from subject $i = 1, \dots, n$, subject to right censoring at time c_i . Let $(y_{i1}, y_{i2}, \delta_{i1}, \delta_{i2}, x_i)$ denote independent observations, where $y_{i1} = \min(t_{i1}, t_{i2}, c_i)$, $\delta_{i1} = \mathbb{1}\{t_{i1} \leq \min(t_{i2}, c_i)\}$, $y_{i2} = \min(t_{i2}, c_i)$, $\delta_{i2} = \mathbb{1}\{t_{i2} \leq c_i\}$, and x_i is a vector of covariates for individual i . The independent semi-competing risks data are assumed to arise from an illness-death model system with transitions that are modeled through the following three hazard functions:

$$h_1(t_{i1} | \gamma_{ji}, x_{i1}) = \gamma_{ji} h_{01}(t_{i1}) \exp(x_{i1}^\top \beta_1), \quad t_{i1} > 0, \quad (5)$$

$$h_2(t_{i2} | \gamma_{ji}, x_{i2}) = \gamma_{ji} h_{02}(t_{i2}) \exp(x_{i2}^\top \beta_2), \quad t_{i2} > 0, \quad (6)$$

$$h_3(t_{i2} | t_{i1}, \gamma_{ji}, x_{i3}) = \gamma_{ji} h_{03}(t_{i2}) \exp(x_{i3}^\top \beta_3), \quad t_{i2} > 0, \quad (7)$$

where γ_{ji} is a subject-specific frailty with vectors of covariates x_{i1} , x_{i2} and x_{i3} which are subsets of x_i . The baseline hazard h_0 is defined non-parametrically by a mixture of piecewise exponential functions as follows

$$\lambda_0(t) = \log h_0(t) = \sum_{k=1}^{K+1} \lambda_k \mathbb{1}\{t \in (s_{k-1}, s_k]\},$$

where λ_k is constant and the time interval between 0 and the largest observed failure time, denoted s_K , is partitioned into $K + 1$ disjoint intervals: $0 < s_1 < \dots < s_{K+1}$. The baseline hazard function h_{03} is assumed to be Markov with respect to t_{i1} ; we will refer to the set of conditional hazard functions in (13)-(15) as the Markov model. Alternatively, we consider modeling h_3 as follows:

$$h_3(t_{i2} | t_{i1}, \gamma_{ji}, x_{i3}) = \gamma_{ji} h_{03}(t_{i2} - t_{i1}) \exp(x_{i3}^\top \beta_3), \quad 0 < t_{i1} < t_{i2}. \quad (8)$$

We will refer to the set of conditional hazard functions in (13), (14) and (16) as the semi-Markov model.

In the Bayesian framework, priors must be specified for the regression parameter, β , the number of intervals, K , the partition points (s_1, \dots, s_{K+1}) , and the frailty, γ_{ji} , respectively. The following specifications are made

$$\begin{aligned} \pi(\beta) &\propto 1, \\ \lambda | K, \mu_\lambda, \sigma_\lambda^2 &\sim MVN_{K+1}(\mu_\lambda \mathbb{1}, \sigma_\lambda^2 \Sigma_\lambda) \\ K &\sim \text{Poisson}(\alpha), \\ \pi(s | K) &\propto \frac{(2K+1)! \prod_{k=1}^{K+1} (s_k - s_{k-1})}{(s_{K+1})^{(2K+1)}}, \\ \pi(\mu_\lambda) &\propto 1, \\ \sigma_\lambda^{-2} &\sim \text{Gamma}(a, b), \\ \gamma_{ji} | \theta &\sim \text{Gamma}(\theta^{-1}, \theta^{-1}), \\ \theta^{-1} &\sim \text{Gamma}(\psi, \omega). \end{aligned}$$

The prior specification for λ follows a MVN-ICAR (see Supplemental Material to Lee, Haneuse, Schrag and Dominici, 2015). Note that K and s jointly form a time-homogeneous Poisson process prior for the partition.

Hyperparameters

- α : parameter corresponding to the Poisson prior of K
- a, b : shape and rate of Gamma prior for σ_λ^{-2}
- ψ : the shape of Gamma prior for θ^{-1}
- ω : the rate of Gamma prior for θ^{-1}

Arguments to specify**Model-related**

<code>Y</code>	an $(n \times 4)$ -dimensional data.frame with columns $y_1, \delta_1, y_2, \delta_2$.
<code>model</code>	<code>c("Markov", "PEM")</code> for Markov definition of h_3 in (15); <code>c("semi-Markov", "PEM")</code> for semi-Markov definition of h_3 in (16).
<code>data</code>	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code> below.
<code>lin.pred</code>	a list of three formula objects that correspond to h_g , for $g \in \{1, 2, 3\}$. Example: <code>form1 <- as.formula(~ x1 + x2 + x3), form2 <- as.formula(~ x1), form3 <- as.formula(~ x1 + x4),</code> <code>lin.pred <- list(form1, form2, form3).</code>

Hyperparameters

<code>PEM.ab1</code>	a 2-vector of positive hyperparameters a_1 and b_1 which represent the shape and rate of the Gamma prior for $\sigma_{\lambda,1}^{-2}$. Example: <code>PEM.ab1 <- c(0.7, 0.7)</code> .
<code>PEM.ab2</code>	a 2-vector of positive hyperparameters a and b of the prior distribution for $\sigma_{\lambda,2}^{-2}$.
<code>PEM.ab3</code>	a 2-vector of positive hyperparameters a and b of the prior distribution for $\sigma_{\lambda,3}^{-2}$.
<code>PEM.alpha1</code>	hyperparameter α of the prior distribution for K_1 , which is one less than the number of partition points.
<code>PEM.alpha2</code>	hyperparameter α of the prior distribution for K_2 , which is one less than the number of partition points.
<code>PEM.alpha3</code>	hyperparameter α of the prior distribution for K_3 , which is one less than the number of partition points.
<code>theta</code>	a 2-vector of positive hyperparameters ψ and ω for the hyperprior θ .

MCMC Settings

<code>numReps</code>	total number of scans
<code>thin</code>	extent of thinning, e.g. if <code>thin=10</code> retain every 10^{th} sample.
<code>burninPerc</code>	the proportion of burn-in (samples to be discarded before analyzing the data).
<code>mhProp_theta_var</code>	the parameter θ is updated using a Metropolis-Hastings random walk step generating proposals from a Gamma distribution with variance <code>mhProp_theta_var</code> .

<code>Cg</code>	a 3-vector for the proportion that determines the sum of probabilities choosing the birth and death moves for each of the baseline hazards, h_{0g} , for $g \in \{1, 2, 3\}$. ³
<code>delPertg</code>	a 3-vector for the perturbation parameter in the birth updates for all three baseline hazard functions; values must be between 0 and 0.5. ³
<code>rj.scheme</code>	<code>rj.scheme=1</code> : the birth update will draw the proposal time split from $1 : s_{max}$; <code>rj.scheme=2</code> : the birth update will draw the proposal time split from uniquely ordered failure times in the data.
<code>Kg_max</code>	a 3-vector for the number of splits allowed in each iteration of the Metropolis-Hastings-Green algorithm for the three baseline hazard functions.
<code>sg_max</code>	the largest observed failure time, given by <code>sg_max <- c(max(Y\$time1[Y\$event1==1]), max(Y\$time2[Y\$event1==0 & Y\$event2==1]), max(Y\$time2[Y\$event1==1] & Y\$event2==1))</code>
<code>time_lambda1</code>	time points at which the λ_1 is monitored for convergence. Example: <code>time_lambda1 <- seq(1, sg_max[1], 1)</code> . The chains for these monitoring points can be found in <code>lambda.fin</code> in the chains of the BayesID object.
<code>time_lambda2</code>	time points at which the λ_2 is monitored for convergence. Example: <code>time_lambda2 <- seq(1, sg_max[2], 1)</code> .
<code>time_lambda3</code>	time points at which the λ_3 is monitored for convergence. Example: <code>time_lambda3 <- seq(1, sg_max[3], 1)</code> .
Starting Values	
<code>startValues</code>	use <code>initiate.startValues_HReg(Y, lin.pred, data, model)</code> which initiates all necessary starting values. Users may set non-null starting values for any of the following: <code>beta1</code> , <code>beta2</code> , <code>beta3</code> , <code>gamma.ji</code> , <code>theta</code> .
Storage	
<code>path</code>	name of the directory where results are stored. Can leave unspecified.
<code>nGam_save</code>	the number of γ to be stored.

Implementation

```

data(scrData)
Y <- scrData[,c(1,2,3,4)]
form1 <- as.formula( ~ x1 + x2 + x3)
form2 <- as.formula( ~ x1 + x2)
form3 <- as.formula( ~ x1 + x2)
lin.pred <- list(form1, form2, form3)
##
theta <- c(0.7, 0.7)
PEM.ab1 <- c(0.7, 0.7) # prior parameters for 1/sigma_1^2
PEM.ab2 <- c(0.7, 0.7) # prior parameters for 1/sigma_2^2
PEM.ab3 <- c(0.7, 0.7) # prior parameters for 1/sigma_3^2
PEM.alpha1 <- 10 # prior parameters for K1
PEM.alpha2 <- 10 # prior parameters for K2
PEM.alpha3 <- 10 # prior parameters for K3
hyperParams <- list(theta=theta,
  PEM=list(PEM.ab1=PEM.ab1, PEM.ab2=PEM.ab2, PEM.ab3=PEM.ab3,
    PEM.alpha1=PEM.alpha1, PEM.alpha2=PEM.alpha2, PEM.alpha3=PEM.alpha3))
##
numReps <- 2000
thin <- 10
burninPerc <- 0.25
mhProp_theta_var <- 0.05
Cg <- c(0.2, 0.2, 0.2)
delPertg <- c(0.5, 0.5, 0.5)
rj.scheme <- 1
Kg_max <- c(50, 50, 50)
sg_max <- c(max(Y$time1[Y$event1 == 1]),
  max(Y$time2[Y$event1 == 0 & Y$event2 == 1]),
  max(Y$time2[Y$event1 == 1 & Y$event2 == 1]))
time_lambda1 <- seq(1, sg_max[1], 1)
time_lambda2 <- seq(1, sg_max[2], 1)
time_lambda3 <- seq(1, sg_max[3], 1)
nGam_save <- 0
mcmc.PEM <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
  storage=list(nGam_save=nGam_save),
  tuning=list(mhProp_theta_var=mhProp_theta_var,
    Cg=Cg, delPertg=delPertg,
    rj.scheme=rj.scheme, Kg_max=Kg_max, sg_max=sg_max,
    time_lambda1=time_lambda1, time_lambda2=time_lambda2,
    time_lambda3=time_lambda3))
##
myModel <- c("semi-Markov", "PEM")
myPath <- "Output/O2-Results-PEM/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel)
startValues[[2]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, theta = 0.23)
##
fit_PEM <- BayesID_HReg(Y, lin.pred, scrData, cluster=NULL, model=myModel,
  hyperParams, startValues, mcmc.PEM, path=myPath)
fit_PEM

```

³See Section A in Supplemental Material to Lee et al. (2015)

```
summ.fit_PEM <- summary(fit_PEM); names(summ.fit_PEM)
summ.fit_PEM
plot(fit_PEM)
plot(fit_PEM, plot.est = "BH")
names(fit_PEM.plot <- plot(fit_PEM, plot=FALSE))
```

Definition of the survival model

Let t_{i1} , t_{i2} denote time to non-terminal and terminal event from subject $i = 1, \dots, n$. The independent semi-competing risks data are assumed to arise from an illness-death model system with transitions that are modeled through the following three hazard functions:

$$\begin{aligned}\log(t_{i1}) &= \mathbf{x}_{i1}^\top \boldsymbol{\beta}_1 + \gamma_i + \epsilon_{i1}, & t_{i1} > 0, \\ \log(t_{i2}) &= \mathbf{x}_{i2}^\top \boldsymbol{\beta}_2 + \gamma_i + \epsilon_{i2}, & t_{i2} > 0, \\ \log(t_{i2} - t_{i1}) &= \mathbf{x}_{i3}^\top \boldsymbol{\beta}_3 + \gamma_i + \epsilon_{i3}, & t_{i2} > t_{i1},\end{aligned}$$

where γ_i is a study participant-specific random effect, \mathbf{x}_{ig} is a vector of transition-specific covariates, $\boldsymbol{\beta}_g$ is a corresponding vector of transition-specific regression parameters, and ϵ_{ig} is a transition-specific random variable whose distribution determines that of the corresponding transition time, $g \in \{1, 2, 3\}$. In the presence of interval censoring, the times-to-event for the i^{th} subject satisfy $c_{ij} \leq t_{i1} < c_{ij+1}$ for some j and $c_{ik} \leq t_{i2} < c_{ik+1}$ for some k . Let $\{c_{ij}, c_{ij+1}, c_{ik}, c_{ik+1}, L_i, \mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{x}_{i3}\}$ denote independent observations, where L_i is the left-truncation time.

For the parametric AFT illness-death model, we build on the log-Normal formulation and take the ϵ_{ig} to follow independent $\text{Normal}(\mu_g, \sigma_g^2)$ distributions, $g=1,2,3$. In the Bayesian framework, priors must be specified for the unknown parameters. The following specifications are made

$$\begin{aligned}\pi(\boldsymbol{\beta}_g, \mu_g) &\propto 1, \\ \sigma_g^2 &\sim \text{inverse-Gamma}(a_{\sigma_g}, b_{\sigma_g}), \\ \gamma_i | \theta &\sim \text{Normal}(0, \theta), \\ \theta^{-1} &\sim \text{inverse-Gamma}(a_\theta, b_\theta).\end{aligned}$$

Hyperparameters

The hyperparameters a_{σ_g} and b_{σ_g} must be specified for the prior of σ_g^2 , as well as a_θ and b_θ , the rate and shape of the inverse-Gamma distributed hyperprior for θ .

Arguments to specify

Model-related

<code>Y</code>	an $(n \times 5)$ -dimensional data.frame with columns $c_{ij}, c_{ij+1}, c_{ik}, c_{ik+1}, L_i$.
<code>data</code>	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code> below.
<code>lin.pred</code>	a list of three formula objects that correspond to h_g , for $g \in \{1, 2, 3\}$. Example: <code>form1 <- as.formula(~ x1 + x2 + x3)</code> , <code>form2 <- as.formula(~ x1)</code> , <code>form3 <- as.formula(~ x1 + x4)</code> , <code>lin.pred <- list(form1, form2, form3)</code> .

Hyperparameters

<code>theta</code>	a 2-vector of positive hyperparameters a_θ and b_θ for the hyperprior θ .
<code>LN.ab1</code>	a 2-vector of positive hyperparameters a_{σ_1} and b_{σ_1} which represent the shape and rate of the inverse-Gamma prior for σ_1^2 . Example: <code>LN.ab1 <- c(0.3, 0.3)</code> .
<code>LN.ab2</code>	a 2-vector of positive hyperparameters a_{σ_2} and b_{σ_2} which represent the shape and rate of the inverse-Gamma prior for σ_2^2 . Example: <code>LN.ab2 <- c(0.3, 0.3)</code> .
<code>LN.ab3</code>	a 2-vector of positive hyperparameters a_{σ_3} and b_{σ_3} which represent the shape and rate of the inverse-Gamma prior for σ_3^2 . Example: <code>LN.ab3 <- c(0.3, 0.3)</code> .

MCMC Settings

<code>numReps</code>	total number of scans
<code>thin</code>	extent of thinning, e.g. if <code>thin=10</code> retain every 10^{th} sample.
<code>burninPerc</code>	the proportion of burn-in (samples to be discarded before analyzing the data).
<code>betag.prop.var</code>	the parameter β_g is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance <code>betag.prop.var</code> .
<code>gamma.prop.var</code>	the parameter γ is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance <code>gamma.prop.var</code> .
<code>mug.prop.var</code>	the parameter μ_g is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance <code>mug.prop.var</code> .
<code>zetag.prop.var</code>	the parameter $\zeta_g = 1/\sigma_g^2$ is updated using a Metropolis-Hastings random walk step generating proposals from a log-Normal distribution with variance <code>zetag.prop.var</code> .

Starting Values

<code>startValues</code>	use <code>initiate.startValues_AFT(Y, lin.pred, data, model)</code> which initiates all necessary starting values. Users may set non-null starting values for any of the following: <code>beta1</code> , <code>beta2</code> , <code>beta3</code> , <code>gamma</code> , <code>theta</code> , <code>y1</code> , <code>y2</code> , <code>LN.mu</code> , <code>LN.sigSq</code> .
--------------------------	---

Storage

<code>nGam_save</code>	the number of γ to be stored
<code>nY1_save</code>	the number of $\log(t_1)$ to be stored
<code>nY2_save</code>	the number of $\log(t_2)$ to be stored
<code>nY1.NA_save</code>	the number of $1\{t_1 > t_2\}$ to be stored
<code>path</code>	name of the directory where results are stored. Can leave unspecified.

Implementation

```
data(scrData)
Y <- matrix(NA, dim(scrData)[1], 5)
Y[,1] <- Y[,2] <- scrData[,1]
Y[,3] <- Y[,4] <- scrData[,3]
Y[which(scrData[,2] == 0),2] <- Inf
Y[which(scrData[,4] == 0),4] <- Inf
Y[,5] <- rep(0, dim(scrData)[1])
```

```

form1 <- as.formula( ~ x1 + x2 + x3)
form2 <- as.formula( ~ x1 + x2)
form3 <- as.formula( ~ x1 + x2)
lin.pred <- list(form1, form2, form3)
##
theta.ab <- c(0.5, 0.05)
LN.ab1 <- c(0.3, 0.3)
LN.ab2 <- c(0.3, 0.3)
LN.ab3 <- c(0.3, 0.3)
hyperParams <- list(theta=theta.ab,
LN=list(LN.ab1=LN.ab1, LN.ab2=LN.ab2, LN.ab3=LN.ab3))
##
numReps <- 300
thin <- 3
burninPerc <- 0.5
nGam_save <- 10
nY1_save <- 10
nY2_save <- 10
nY1.NA_save <- 10
betag.prop.var <- c(0.01,0.01,0.01)
mug.prop.var <- c(0.1,0.1,0.1)
zetag.prop.var <- c(0.1,0.1,0.1)
gamma.prop.var <- 0.01
mcmcParams <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
storage=list(nGam_save=nGam_save, nY1_save=nY1_save, nY2_save=nY2_save, nY1.NA_save=nY1.NA_save),
tuning=list(betag.prop.var=betag.prop.var, mug.prop.var=mug.prop.var, zetag.prop.var=zetag.prop.var,
gamma.prop.var=gamma.prop.var))
##
myModel <- "LN"
myPath <- "Output/01-Results-LN/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_AFT(Y, lin.pred, scrData, model=myModel)
startValues[[2]] <- initiate.startValues_AFT(Y, lin.pred, scrData, model=myModel, theta = 0.20)
##
fit_LN <- BayesID_AFT(Y, lin.pred, scrData, model=myModel, hyperParams,
startValues, mcmcParams, path=myPath)
summary(fit_LN)
plot(fit_LN, time = seq(0, 35, 1), tseq=seq(from=0, to=30, by=5))
plot(fit_LN, time = seq(0, 35, 1), tseq=seq(from=0, to=30, by=5), plot.est = "BH")

```

Definition of the survival model

Let t_{i1} , t_{i2} denote time to non-terminal and terminal event from subject $i = 1, \dots, n$. The independent semi-competing risks data are assumed to arise from an illness-death model system with transitions that are modeled through the following three hazard functions:

$$\begin{aligned}\log(t_{i1}) &= \mathbf{x}_{i1}^\top \beta_1 + \gamma_i + \epsilon_{i1}, & t_{i1} > 0, \\ \log(t_{i2}) &= \mathbf{x}_{i2}^\top \beta_2 + \gamma_i + \epsilon_{i2}, & t_{i2} > 0, \\ \log(t_{i2} - t_{i1}) &= \mathbf{x}_{i3}^\top \beta_3 + \gamma_i + \epsilon_{i3}, & t_{i2} > t_{i1},\end{aligned}$$

where γ_i is a study participant-specific random effect, \mathbf{x}_{ig} is a vector of transition-specific covariates, β_g is a corresponding vector of transition-specific regression parameters, and ϵ_{ig} is a transition-specific random variable whose distribution determines that of the corresponding transition time, $g \in \{1, 2, 3\}$. In the presence of interval censoring, the times-to-event for the i^{th} subject satisfy $c_{ij} \leq t_{i1} < c_{ij+1}$ for some j and $c_{ik} \leq t_{i2} < c_{ik+1}$ for some k . Let $\{c_{ij}, c_{ij+1}, c_{ik}, c_{ik+1}, L_i, \mathbf{x}_{i1}, \mathbf{x}_{i2}, \mathbf{x}_{i3}\}$ denote independent observations, where L_i is the left-truncation time.

For our semi-parametric AFT illness-death model, ϵ_{ig} is assumed to be taken as draws from the independent DPM of normal distributions:

$$\begin{aligned}\epsilon_{ig} | r_i &\sim \text{Normal}(\mu_{r_i}, \sigma_{r_i}^2), \\ (\mu_{gr}, \sigma_{gr}^2) &\sim G_{g0}, \text{ for } r = 1, \dots, M_g, \\ r_i | p_g &\sim \text{Discrete}(r_i | p_{g1}, \dots, p_{gM_g}), \\ p_g &\sim \text{Dirichlet}(\tau_g / M_g, \dots, \tau_g / M_g).\end{aligned}$$

In the Bayesian framework, priors must be specified for the unknown parameters. We take the G_{g0} as a normal distribution centered at μ_{g0} with a variance σ_{g0}^2 for μ_{gr} and an $\text{IG}(a_{\sigma_g}, b_{\sigma_g})$ for σ_{gr}^2 . For regression parameters $\{\beta_1, \beta_2, \beta_3\}$, we adopt non-informative flat priors on the real line. For γ , we assume that each γ_i is an independent random draw from a $\text{Normal}(0, \theta)$ distribution. In the absence of prior knowledge on the variance component θ , we adopt a conjugate inverse-Gamma hyperprior, $\text{IG}(a_\theta, b_\theta)$. Finally, we specify a $\text{Gamma}(a_{\tau_g}, b_{\tau_g})$ hyperprior for the precision parameter τ_g .

Hyperparameters

- a_θ, b_θ : the shape and rate of inverse-Gamma prior for θ
- $\mu_{g0}, \sigma_{g0}^2, a_{\sigma_g}, b_{\sigma_g}$: hyperparameters for G_{g0}
- a_{τ_g}, b_{τ_g} : shape and rate of Gamma hyperprior for τ_g

Arguments to specify**Model-related**

- Y** an $(n \times 5)$ -dimensional data.frame with columns $c_{ij}, c_{ij+1}, c_{ik}, c_{ik+1}, L_i$.
- data** an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in `lin.pred` below.
- lin.pred** a list of three formula objects that correspond to h_g , for $g \in \{1, 2, 3\}$.
Example: `form1 <- as.formula(~ x1 + x2 + x3)`, `form2 <- as.formula(~ x1)`, `form3 <- as.formula(~ x1 + x4)`,
`lin.pred <- list(form1, form2, form3)`.

Hyperparameters

- theta** a 2-vector of positive hyperparameters a_θ and b_θ for the hyperprior θ .
- DPM.mu1** a hyperparameter μ_{10}
- DPM.mu2** a hyperparameter μ_{20}
- DPM.mu3** a hyperparameter μ_{30}
- DPM.sigSq1** a hyperparameter σ_{10}^2
- DPM.sigSq2** a hyperparameter σ_{20}^2
- DPM.sigSq3** a hyperparameter σ_{30}^2
- DPM.ab1** a 2-vector of positive hyperparameters $a_{\sigma_1}, b_{\sigma_1}$
- DPM.ab2** a 2-vector of positive hyperparameters $a_{\sigma_2}, b_{\sigma_2}$
- DPM.ab3** a 2-vector of positive hyperparameters $a_{\sigma_3}, b_{\sigma_3}$
- Tau.ab1** a 2-vector of positive hyperparameters a_{τ_1}, b_{τ_1}
- Tau.ab2** a 2-vector of positive hyperparameters a_{τ_2}, b_{τ_2}
- Tau.ab3** a 2-vector of positive hyperparameters a_{τ_3}, b_{τ_3}

MCMC Settings

- numReps** total number of scans
- thin** extent of thinning, e.g. if `thin=10` retain every 10^{th} sample.
- burninPerc** the proportion of burn-in (samples to be discarded before analyzing the data).
- betag.prop.var** the parameter β_g is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance `betag.prop.var`.
- gamma.prop.var** the parameter γ is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance `gamma.prop.var`.
- mug.prop.var** the parameter μ_{gr} is updated using a Metropolis-Hastings random walk step generating proposals from a Normal distribution with variance `mug.prop.var`.
- zetag.prop.var** the parameter $\zeta_{gr} = 1/\sigma_{gr}^2$ is updated using a Metropolis-Hastings random walk step generating proposals from a log-Normal distribution with variance `zetag.prop.var`.

Starting Values

- startValues** use `initiate.startValues_AFT(Y, lin.pred, data, model)` which initiates all necessary starting values. Users may set non-null starting values for any of the following:
`beta1, beta2, beta3, gamma, theta, y1, y2, DPM.class1, DPM.class2, DPM.class3, DPM.mu1, DPM.mu2, DPM.mu3, DPM.zeta1, DPM.zeta2, DPM.zeta3, DPM.tau.`

Storage

- nGam_save** the number of γ to be stored
- nY1_save** the number of $\log(t_1)$ to be stored
- nY2_save** the number of $\log(t_2)$ to be stored

nY1.NA_save the number of $\mathbb{1}\{t_1 > t_2\}$ to be stored
path name of the directory where results are stored. Can leave unspecified.

Implementation

```
data(scrData)
Y <- matrix(NA, dim(scrData)[1], 5)
Y[,1] <- Y[,2] <- scrData[,1]
Y[,3] <- Y[,4] <- scrData[,3]
Y[which(scrData[,2] == 0),2] <- Inf
Y[which(scrData[,4] == 0),4] <- Inf
Y[,5] <- rep(0, dim(scrData)[1])
form1 <- as.formula( ~ x1 + x2 + x3)
form2 <- as.formula( ~ x1 + x2)
form3 <- as.formula( ~ x1 + x2)
lin.pred <- list(form1, form2, form3)
##
theta.ab <- c(0.5, 0.05)
##
DPM.mu1 <- log(12)
DPM.mu2 <- log(12)
DPM.mu3 <- log(12)
DPM.sigSq1 <- 100
DPM.sigSq2 <- 100
DPM.sigSq3 <- 100
DPM.ab1 <- c(2, 1)
DPM.ab2 <- c(2, 1)
DPM.ab3 <- c(2, 1)
Tau.ab1 <- c(1.5, 0.0125)
Tau.ab2 <- c(1.5, 0.0125)
Tau.ab3 <- c(1.5, 0.0125)
hyperParams <- list(theta=theta.ab,
DPM=list(DPM.mu1=DPM.mu1, DPM.mu2=DPM.mu2, DPM.mu3=DPM.mu3, DPM.sigSq1=DPM.sigSq1,
DPM.sigSq2=DPM.sigSq2, DPM.sigSq3=DPM.sigSq3, DPM.ab1=DPM.ab1, DPM.ab2=DPM.ab2,
DPM.ab3=DPM.ab3, Tau.ab1=Tau.ab1, Tau.ab2=Tau.ab2, Tau.ab3=Tau.ab3))
##
numReps <- 300
thin <- 3
burninPerc <- 0.5
nGam_save <- 10
nY1_save <- 10
nY2_save <- 10
nY1.NA_save <- 10
betag.prop.var <- c(0.01,0.01,0.01)
mug.prop.var <- c(0.1,0.1,0.1)
zetag.prop.var <- c(0.1,0.1,0.1)
gamma.prop.var <- 0.01
mcmcParams <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
storage=list(nGam_save=nGam_save, nY1_save=nY1_save, nY2_save=nY2_save, nY1.NA_save=nY1.NA_save),
tuning=list(betag.prop.var=betag.prop.var, mug.prop.var=mug.prop.var,
zetag.prop.var=zetag.prop.var, gamma.prop.var=gamma.prop.var))
##
myModel <- "DPM"
myPath <- "Output/02-Results-DPM/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_AFT(Y, lin.pred, scrData, model=myModel)
startValues[[2]] <- initiate.startValues_AFT(Y, lin.pred, scrData, model=myModel, theta = 0.23)
##
fit_DPM <- BayesID_AFT(Y, lin.pred, scrData, model=myModel, hyperParams,
startValues, mcmcParams, path=myPath)
summary(fit_DPM);
plot(fit_DPM, time = seq(0, 35, 1), tseq=seq(from=0, to=30, by=5))
plot(fit_DPM, time = seq(0, 35, 1), tseq=seq(from=0, to=30, by=5), plot.est = "BH")
```

Definition of the survival model

Let t_{ji1} and t_{ji2} denote the time to nonterminal event and terminal event from subject $i = 1, \dots, n_j$ in cluster $j = 1, \dots, J$, subject to right censoring at time c_{ji} . Let $(y_{ji1}, y_{ji2}, \delta_{ji1}, \delta_{ji2}, x_{ji})$ denote independent observations, where $y_{ji1} = \min(t_{ji1}, t_{ji2}, c_{ji})$, $\delta_{ji1} = \mathbb{1}\{t_{ji1} \leq \min(t_{ji2}, c_{ji})\}$, $y_{ji2} = \min(t_{ji2}, c_{ji})$, $\delta_{ji2} = \mathbb{1}\{t_{ji2} \leq c_{ji}\}$, and x_{ji} is a vector of covariates for individual i . The independent semi-competing risks data are assumed to arise from an illness-death model system with transitions that are modeled through the following three hazard functions:

$$h_1(t_{ji1} | \gamma_{ji}, x_{ji1}, V_{j1}) = \gamma_{ji} h_{01}(t_{ji1}) \exp(x_{ji1}^\top \beta_1 + V_{j1}), \quad t_{ji1} > 0, \quad (9)$$

$$h_2(t_{ji2} | \gamma_{ji}, x_{ji2}, V_{j2}) = \gamma_{ji} h_{02}(t_{ji2}) \exp(x_{ji2}^\top \beta_2 + V_{j2}), \quad t_{ji2} > 0, \quad (10)$$

$$h_3(t_{ji2} | t_{ji1}, \gamma_{ji}, x_{ji3}, V_{j3}) = \gamma_{ji} h_{03}(t_{ji2}) \exp(x_{ji3}^\top \beta_3 + V_{j3}), \quad t_{ji2} > 0, \quad (11)$$

where γ_{ji} is a subject-specific frailty, $V_j = (V_{j1}, V_{j2}, V_{j3})$ is a vector of cluster-specific random effects, and x_{ji1} , x_{ji2} and x_{ji3} which are subsets of x_i are vectors of covariates. The baseline hazard functions are defined parametrically by Weibull hazards of the form $h_{0g}(t) = \alpha_g \kappa_g t^{\alpha_g - 1}$, for $g \in \{1, 2, 3\}$. The baseline hazard function h_{03} is assumed to be Markov with respect to t_{ji1} ; we will refer to the set of conditional hazard functions in (13)-(15) as the Markov model. Alternatively, we consider modeling h_3 as follows:

$$h_3(t_{ji2} | t_{ji1}, \gamma_{ji}, x_{ji3}, V_{j3}) = \gamma_{ji} h_{03}(t_{ji2} - t_{ji1}) \exp(x_{ji3}^\top \beta_3 + V_{j3}), \quad 0 < t_{ji1} < t_{ji2}. \quad (12)$$

We will refer to the set of conditional hazard functions in (13), (14) and (16) as the semi-Markov model.

In the Bayesian framework, priors must be specified for the regression parameter, β_g , the shape and scale parameters of baseline hazard function, α_g and κ_g , and the frailty parameter, γ_{ji} , respectively, for $g \in \{1, 2, 3\}$. The following specifications are made

$$\begin{aligned} \pi(\beta_g) &\propto 1, \\ \alpha_g &\sim \text{Gamma}(a_g, b_g), \\ \kappa_g &\sim \text{Gamma}(c_g, d_g), \\ \gamma_{ji} | \theta &\sim \text{Gamma}(\theta^{-1}, \theta^{-1}), \\ \theta^{-1} &\sim \text{Gamma}(\psi, \omega). \end{aligned}$$

We provide two possible prior specifications for the cluster-specific random effects below.

$$\begin{array}{ll} V_j \sim \text{MVN}(0, \Sigma_V), & V_j | m_j \sim \text{MVN}(\mu_{m_j}, \Sigma_{m_j}), \\ \Sigma_V \sim \text{Inverse-Wishart}(\Psi_V, \rho_V), & (\mu_m, \Sigma_m) \sim G_0, \text{ for } m = 1, \dots, M, \\ & m_j | p \sim \text{Discrete}(m_j | p_1, \dots, p_M), \\ & p \sim \text{Dirichlet}(\tau/M, \dots, \tau/M), \\ & \tau \sim \text{Gamma}(a_\tau, b_\tau). \end{array}$$

In the first column, the individual specific-random effects are assumed to be *iid* $\text{MVN}(0, \Sigma_V)$. In the second column, the cluster-specific random effects are drawn from a mixture of M multivariate normal distributions each with mean vector and covariance matrix (μ_m, Σ_m) which are distributed as a multivariate Normal/Inverse-Wishart (NIW), denoted by G_0 ; we refer to this as the Dirichlet process mixture (DPM) prior. The probability density of G_0 is defined by the product

$$f_{\text{NIW}}(\mu, \Sigma | \Psi_0, \rho_0) = f_{\text{MVN}}(\mu | 0, \Sigma) \times f_{\text{Inv-Wish}}(\Sigma | \Psi_0, \rho_0).$$

Hyperparameters

- a_g, b_g : shape and rate of Gamma prior for α_g for $g \in \{1, 2, 3\}$
- c_g, d_g : shape and rate of Gamma prior for κ_g for $g \in \{1, 2, 3\}$
- ψ : the shape of Gamma prior for θ^{-1}
- ω : the rate of Gamma prior for θ^{-1}
- Ψ_0, ρ_0 : shape and scale of Inverse-Wishart component of the prior distribution, G_0 , of (μ_m, Σ_m) (DPM prior)
- a_τ, b_τ : shape and rate of Gamma hyperprior for τ (DPM prior)

Arguments to specify

Model-related

Y	an $(n \times 4)$ -dimensional data.frame with columns $y_1, \delta_1, y_2, \delta_2$.
model	<code>c("Markov", "Weibull")</code> for Markov definition of h_3 in (15); <code>c("semi-Markov", "Weibull")</code> for semi-Markov definition of h_3 in (16).
data	an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in <code>lin.pred</code> below.
lin.pred	a list of three formula objects that correspond to h_g , for $g \in \{1, 2, 3\}$. Example: <code>form1 <- as.formula(~ x1 + x2 + x3)</code> , <code>form2 <- as.formula(~ x1)</code> , <code>form3 <- as.formula(~ x1 + x4)</code> , <code>lin.pred <- list(form1, form2, form3)</code> .
cluster	an n -vector of cluster information where cluster membership corresponds to one of the positive integers $1, \dots, J$.

Hyperparameters

WB.ab1	a 2-vector of positive hyperparameters a_1 and b_1 of the prior distribution for the shape parameter α_1 of the Weibull baseline hazard. Example: <code>WB.ab1 <- c(0.5, 0.01)</code> .
WB.ab2	a 2-vector of positive hyperparameters a_2 and b_2 of the prior for α_2 .
WB.ab3	a 2-vector of positive hyperparameters a_3 and b_3 of the prior for α_3 .
WB.cd1	a 2-vector of positive hyperparameters c_1 and d_1 of the prior distribution for the scale parameter κ_1 of the Weibull baseline hazard. Example: <code>WB.cd1 <- c(0.5, 0.05)</code> .
WB.cd2	a 2-vector of positive hyperparameters c_2 and d_2 of the prior for κ_2 .
WB.cd3	a 2-vector of positive hyperparameters c_3 and d_3 of the prior for κ_3 .
theta	a 2-vector of positive hyperparameters ψ and ω for the hyperprior θ .
MVN prior for V_{ji}	
Psi_v	a positive-definite scale matrix of the Inverse-Wishart prior for the cluster random effects, V_{ji} . Example:

<code>rho_v</code>	the degrees of freedom of the Inverse-Wishart prior for V_{ji} . Example: <code>rho_v <- 100</code> .
DPM prior for V_{ji}	
<code>Psi0</code>	a positive-definite scale matrix of the Inverse-Wishart component of G_0 . Example: <code>Psi0 <- diag(1,3)</code> .
<code>rho0</code>	the degrees of freedom of the Inverse-Wishart component of G_0 . Example: <code>rho0 <- 10</code> .
<code>aTau</code>	a positive-valued hyperparameter corresponding to the shape parameter, a_τ , of the Gamma prior of τ .
<code>bTau</code>	a positive-valued hyperparameter corresponding to the rate parameter, b_τ , of the Gamma prior of τ .
MCMC Settings	
<code>numReps</code>	total number of scans
<code>thin</code>	extent of thinning, e.g. if <code>thin=10</code> retain every 10^{th} sample.
<code>burninPerc</code>	the proportion of burn-in (samples to be discarded before analyzing the data).
<code>mhProp_theta_var</code>	the parameter θ is updated using a Metropolis-Hastings random walk step generating proposals from a Gamma distribution with variance <code>mhProp_theta_var</code> .
<code>mhProp_alphag_var</code>	a 3-vector which specifies the variances of the three random walk Metropolis-Hastings Gamma proposal distributions corresponding to α_1 , α_2 , α_3 .
<code>mhProp_Vg_var</code>	a 3-vector which specifies the variances of the three random walk Metropolis-Hastings proposals from normal distributions with the same variance <code>mhProp_Vg_var</code> .
Starting Values	
<code>startValues</code>	use <code>initiate.startValues_HReg(Y, lin.pred, data, model, cluster)</code> which initiates all necessary starting values. Users may set non-null starting values for: <code>beta1</code> , <code>beta2</code> , <code>beta3</code> , <code>theta</code> , <code>WB.alpha</code> , <code>WB.kappa</code> , <code>gamma.ji</code> , <code>V.j1</code> , <code>V.j2</code> , <code>V.j3</code> , <code>MVN.SigmaV</code> , <code>DPM.tau</code> , <code>DPM.class</code> .
Storage	
<code>path</code>	name of the directory where results are stored. Can leave unspecified.
<code>nGam_save</code>	the number of γ to be stored.
<code>storeV</code>	a 3-vector of TRUE/FALSE logical constants indicating storage of V_{ji} values for $g = 1, 2, 3$. Example: <code>storeV <- rep(TRUE, 3)</code> .

Implementation

```

data(scrData)
Y <- scrData[,c(1,2,3,4)]
cluster <- scrData[,5]
form1 <- as.formula( ~ x1 + x2 + x3)
form2 <- as.formula( ~ x1 + x2)
form3 <- as.formula( ~ x1 + x2)
lin.pred <- list(form1, form2, form3)
##
WB.ab1 <- c(0.5, 0.01)
WB.ab2 <- c(0.5, 0.01)
WB.ab3 <- c(0.5, 0.01)
WB.cd1 <- c(0.5, 0.05)
WB.cd2 <- c(0.5, 0.05)
WB.cd3 <- c(0.5, 0.05)
theta <- c(0.7, 0.7) # prior params for 1/theta
Psi_v <- diag(1, 3) # MVN cluster-specific random effects
rho_v <- 100
Psi0 <- diag(1, 3) # DPM cluster-specific random effects
rho0 <- 10
aTau <- 1.5
bTau <- 0.0125
hyperParams.WB.MVN <- list(theta=theta,
  WB=list(WB.ab1=WB.ab1, WB.ab2=WB.ab2, WB.ab3=WB.ab3,
    WB.cd1=WB.cd1, WB.cd2=WB.cd2, WB.cd3=WB.cd3),
  MVN=list(Psi_v=Psi_v, rho_v=rho_v))
hyperParams.WB.DPM <- list(theta=theta,
  WB=list(WB.ab1=WB.ab1, WB.ab2=WB.ab2, WB.ab3=WB.ab3,
    WB.cd1=WB.cd1, WB.cd2=WB.cd2, WB.cd3=WB.cd3),
  DPM=list(Psi0=Psi0, rho0=rho0, aTau=aTau, bTau=bTau))
##
numReps <- 2000
thin <- 10
burninPerc <- 0.25
mhProp_theta_var <- 0.05
mhProp_alphag_var <- c(0.01, 0.01, 0.01)
mhProp_Vg_var <- c(0.05, 0.05, 0.05)
nGam_save <- 0
storeV <- rep(TRUE, 3)
mcmc.WB <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
  storage=list(nGam_save=nGam_save, storeV=storeV),
  tuning=list(mhProp_theta_var=mhProp_theta_var, mhProp_alphag_var=mhProp_alphag_var,
    mhProp_Vg_var =mhProp_Vg_var))
##
Sigma_V <- diag(0.1, 3)
Sigma_V[1,2] <- Sigma_V[2,1] <- -0.05
Sigma_V[1,3] <- Sigma_V[3,1] <- -0.06
Sigma_V[2,3] <- Sigma_V[3,2] <- 0.07

```



```

##
myModel <- c("semi-Markov", "Weibull", "MVN")
myPath <- "Output/03-Results-WB_MVN/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, cluster)
startValues[[2]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, cluster,
MVN.SigmaV=Sigma_V)
##
fit_WB_MVN <- BayesID_HReg(Y, lin.pred, scrData, cluster, model=myModel,
hyperParams.WB.MVN, startValues, mcmc.WB, path=myPath)

fit_WB_MVN
summ.fit_WB_MVN <- summary(fit_WB_MVN); names(summ.fit_WB_MVN)
summ.fit_WB_MVN
plot(fit_WB_MVN, tseq=seq(from=0, to=30, by=5))
plot(fit_WB_MVN, tseq=seq(from=0, to=30, by=5), plot.est = "BH")
names(fit_WB_MVN.plot <- plot(fit_WB_MVN, tseq=seq(0, 30, 5), plot=FALSE))
##
myModel <- c("semi-Markov", "Weibull", "DPM")
myPath <- "Output/04-Results-WB_DPM/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, cluster)
startValues[[2]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, cluster,
MVN.SigmaV=Sigma_V)
fit_WB_DPM <- BayesID_HReg(Y, lin.pred, scrData, cluster, model=myModel,
hyperParams.WB.DPM, startValues, mcmc.WB, path=myPath)

fit_WB_DPM
summ.fit_WB_DPM <- summary(fit_WB_DPM); names(summ.fit_WB_DPM)
summ.fit_WB_DPM
plot(fit_WB_DPM, tseq=seq(from=0, to=30, by=5))
plot(fit_WB_DPM, tseq=seq(from=0, to=30, by=5), plot.est = "BH")
names(fit_WB_DPM.plot <- plot(fit_WB_DPM, tseq=seq(0, 30, 5), plot=FALSE))

```

Definition of the survival model

Let t_{ji1} and t_{ji2} denote the time to nonterminal event and terminal event from subject $i = 1, \dots, n_j$ in cluster $j = 1, \dots, J$, subject to right censoring at time c_{ji} . Let $(y_{ji1}, y_{ji2}, \delta_{ji1}, \delta_{ji2}, x_{ji})$ denote independent observations, where $y_{ji1} = \min(t_{ji1}, t_{ji2}, c_{ji})$, $\delta_{ji1} = \mathbb{1}\{t_{ji1} \leq \min(t_{ji2}, c_{ji})\}$, $y_{ji2} = \min(t_{ji2}, c_{ji})$, $\delta_{ji2} = \mathbb{1}\{t_{ji2} \leq c_{ji}\}$, and x_{ji} is a vector of covariates for individual i . The independent semi-competing risks data are assumed to arise from an illness-death model system with transitions that are modeled through the following three hazard functions:

$$h_1(t_{ji1} | \gamma_{ji}, x_{ji1}, V_{j1}) = \gamma_{ji} h_{01}(t_{ji1}) \exp(x_{ji1}^\top \beta_1 + V_{j1}), \quad t_{ji1} > 0, \quad (13)$$

$$h_2(t_{ji2} | \gamma_{ji}, x_{ji2}, V_{j2}) = \gamma_{ji} h_{02}(t_{ji2}) \exp(x_{ji2}^\top \beta_2 + V_{j2}), \quad t_{ji2} > 0, \quad (14)$$

$$h_3(t_{ji2} | t_{ji1}, \gamma_{ji}, x_{ji3}, V_{j3}) = \gamma_{ji} h_{03}(t_{ji2}) \exp(x_{ji3}^\top \beta_3 + V_{j3}), \quad t_{ji2} > 0, \quad (15)$$

where γ_{ji} is a subject-specific frailty, $V_j = (V_{j1}, V_{j2}, V_{j3})$ is a vector of cluster-specific random effects, and x_{ji1} , x_{ji2} and x_{ji3} which are subsets of x_i are vectors of covariates. The baseline hazard h_0 is defined non-parametrically by a mixture of piecewise exponential functions as follows

$$\lambda_0(t) = \log h_0(t) = \sum_{k=1}^{K+1} \lambda_k \mathbb{1}\{t \in (s_{k-1}, s_k]\},$$

where λ_k is constant and the time interval between 0 and the largest observed failure time, denoted s_k , is partitioned into $K + 1$ disjoint intervals: $0 < s_1 < \dots < s_{K+1}$. The baseline hazard function h_{03} is assumed to be Markov with respect to t_{i1} ; we will refer to the set of conditional hazard functions in (13)-(15) as the Markov model. Alternatively, we consider modeling h_3 as follows:

$$h_3(t_{ji2} | t_{ji1}, \gamma_{ji}, x_{ji3}, V_{j3}) = \gamma_{ji} h_{03}(t_{ji2} - t_{ji1}) \exp(x_{ji3}^\top \beta_3 + V_{j3}), \quad 0 < t_{ji1} < t_{ji2}. \quad (16)$$

We will refer to the set of conditional hazard functions in (13), (14) and (16) as the semi-Markov model.

In the Bayesian framework, priors must be specified for the regression parameter, β , the number of intervals, K , the partition points (s_1, \dots, s_{K+1}) , and the frailty, γ_{ji} , respectively. The following specifications are made

$$\begin{aligned} \pi(\beta) &\propto 1, \\ \lambda | K, \mu_\lambda, \sigma_\lambda^2 &\sim MVN_{K+1}(\mu_\lambda \mathbb{1}, \sigma_\lambda^2 \Sigma_\lambda) \\ K &\sim \text{Poisson}(\alpha), \\ \pi(s | K) &\propto \frac{(2K+1)! \prod_{k=1}^{K+1} (s_k - s_{k-1})}{(s_{K+1})^{(2K+1)}}, \\ \pi(\mu_\lambda) &\propto 1, \\ \sigma_\lambda^{-2} &\sim \text{Gamma}(a, b), \\ \gamma_{ji} | \theta &\sim \text{Gamma}(\theta^{-1}, \theta^{-1}), \\ \theta^{-1} &\sim \text{Gamma}(\psi, \omega). \end{aligned}$$

The prior specification for λ follows a MVN-ICAR (see Supplemental Material to Lee, Haneuse, Schrag and Dominici, 2015). Note that K and s jointly form a time-homogeneous Poisson process prior for the partition.

We provide two possible prior specifications for the cluster-specific random effects below.

$$\begin{array}{ll} V_j \sim MVN(0, \Sigma_V), & V_j | m_j \sim MVN(\mu_{m_j}, \Sigma_{m_j}), \\ \Sigma_V \sim \text{Inverse-Wishart}(\Psi_V, \rho_V). & (\mu_m, \Sigma_m) \sim G_0, \text{ for } m = 1, \dots, M, \\ & m_j | p \sim \text{Discrete}(m_j | p_1, \dots, p_M), \\ & p \sim \text{Dirichlet}(\tau/M, \dots, \tau/M), \\ & \tau \sim \text{Gamma}(a_\tau, b_\tau). \end{array}$$

In the first column, the individual specific-random effects are assumed to be iid $MVN(0, \Sigma_V)$. In the second column, the cluster-specific random effects are drawn from a mixture of M multivariate normal distributions each with mean vector and covariance matrix (μ_m, Σ_m) which are distributed as a multivariate Normal/Inverse-Wishart (NIW), denoted by G_0 ; we refer to this as the Dirichlet process mixture (DPM) prior. The probability density of G_0 is defined by the product

$$f_{NIW}(\mu, \Sigma | \Psi_0, \rho_0) = f_{MVN}(\mu | 0, \Sigma) \times f_{Inv-Wish}(\Sigma | \Psi_0, \rho_0).$$

Hyperparameters

- α : parameter corresponding to the Poisson prior of K
- a, b : shape and rate of Gamma prior for σ_λ^{-2}
- ψ : the shape of Gamma prior for θ^{-1}
- ω : the rate of Gamma prior for θ^{-1}
- Ψ_0, ρ_0 : shape and scale of Inverse-Wishart component of the prior distribution, G_0 , of (μ_m, Σ_m) (DPM prior)
- a_τ, b_τ : shape and rate of Gamma hyperprior for τ (DPM prior)

Arguments to specify

Model-related

- `Y` an $(n \times 4)$ -dimensional data.frame with columns $y_1, \delta_1, y_2, \delta_2$.
- `model` c("Markov", "PEM") for Markov definition of h_3 in (15); c("semi-Markov", "PEM") for semi-Markov definition of h_3 in (16).
- `data` an $(n \times q)$ -dimensional data.frame; the q -columns correspond to q covariate vectors named in the formula in `lin.pred` below.
- `lin.pred` a list of three formula objects that correspond to h_g , for $g \in \{1, 2, 3\}$.

Example: `form1 <- as.formula(~ x1 + x2 + x3)`, `form2 <- as.formula(~ x1)`, `form3 <- as.formula(~ x1 + x4)`,
`lin.pred <- list(form1, form2, form3)`.

`cluster` an n -vector of cluster information where cluster membership corresponds to one of the positive integers $1, \dots, J$.

Hyperparameters

`PEM.ab1` a 2-vector of positive hyperparameters a_1 and b_1 which represent the shape and rate of the Gamma prior for $\sigma_{\lambda_1}^{-2}$.
 Example: `PEM.ab1 <- c(0.7, 0.7)`.
`PEM.ab2` a 2-vector of positive hyperparameters a and b of the prior distribution for $\sigma_{\lambda_2}^{-2}$.
`PEM.ab3` a 2-vector of positive hyperparameters a and b of the prior distribution for $\sigma_{\lambda_3}^{-2}$.
`PEM.alpha1` hyperparameter α of the prior distribution for K_1 , which is one less than the number of partition points.
`PEM.alpha2` hyperparameter α of the prior distribution for K_2 , which is one less than the number of partition points.
`PEM.alpha3` hyperparameter α of the prior distribution for K_3 , which is one less than the number of partition points.
`theta` a 2-vector of positive hyperparameters ψ and ω for the hyperprior θ .

MVN prior for V_{ji}

`Psi_v` a positive-definite scale matrix of the Inverse-Wishart prior for the cluster random effects, V_{ji} .
 Example: `Psi_v <- diag(1,3)`.
`rho_v` the degrees of freedom of the Inverse-Wishart prior for V_{ji} . Example: `rho_v <- 100`.

DPM prior for V_{ji}

`Psi0` a positive-definite scale matrix of the Inverse-Wishart component of G_0 . Example: `Psi0 <- diag(1,3)`.
`rho0` the degrees of freedom of the Inverse-Wishart component of G_0 . Example: `rho0 <- 10`.
`aTau` a positive-valued hyperparameter corresponding to the shape parameter, a_τ , of the Gamma prior of τ .
`bTau` a positive-valued hyperparameter corresponding to the rate parameter, b_τ , of the Gamma prior of τ .

MCMC Settings

`numReps` total number of scans
`thin` extent of thinning, e.g. if `thin=10` retain every 10^{th} sample.
`burninPerc` the proportion of burn-in (samples to be discarded before analyzing the data).
`mhProp_theta_var` the parameter θ is updated using a Metropolis-Hastings random walk step generating proposals from a Gamma distribution with variance `mhProp_theta_var`.
`mhProp_Vg_var` 3-vector which specifies the variances of the three random walk Metropolis-Hastings proposals from normal distributions with the same variance `mhProp_Vg_var`.
`Cg` a 3-vector for the proportion that determines the sum of probabilities choosing the birth and death moves for each of the baseline hazards, h_{0g} , for $g \in \{1, 2, 3\}$.⁴
`delPertg` a 3-vector for the perturbation parameter in the birth updates for all three baseline hazard functions; values must be between 0 and 0.5.⁴
`rj.scheme` `rj.scheme=1`: the birth update will draw the proposal time split from $1 : s_{max}$; `rj.scheme=2`: the birth update will draw the proposal time split from uniquely ordered failure times in the data.
`Kg_max` a 3-vector for the number of splits allowed in each iteration of the Metropolis-Hastings-Green algorithm for the three baseline hazard functions.
`sg_max` the largest observed failure time, given by
`sg_max <- c(max(Y$time1[Y$event1==1]),
 max(Y$time2[Y$event1==0 & Y$event2==1]),
 max(Y$time2[Y$event1==1 & Y$event2==1]))`
`time_lambda1` time points at which the λ_1 is monitored for convergence. Example: `time_lambda1 <- seq(1, sg_max[1], 1)`. The chains for these monitoring points can be found in `lambda.fin` in the chains of the `BayesID_HReg` object.
`time_lambda2` time points at which the λ_2 is monitored for convergence. Example: `time_lambda2 <- seq(1, sg_max[2], 1)`.
`time_lambda3` time points at which the λ_3 is monitored for convergence. Example: `time_lambda3 <- seq(1, sg_max[3], 1)`.

Starting Values

`startValues` use `initiate.startValues_HReg(Y, lin.pred, data, model)` which initiates all necessary starting values. Users may set non-null starting values for any of the following: `beta1, beta2, beta3, gamma.ji, theta, V.j1, V.j2, V.j3, MVN.SigmaV, DPM.tau, DPM.class`.

Storage

`path` name of the directory where results are stored. Can leave unspecified.
`nGam_save` the number of γ to be stored.
`storeV` a 3-vector of TRUE/FALSE logical constants indicating storage of V_{ji} values for $g = 1, 2, 3$. Example: `storeV <- rep(TRUE, 3)`.

Implementation

```
data(scrData)
Y <- scrData[,c(1,2,3,4)]
form1 <- as.formula( ~ x1 + x2 + x3)
form2 <- as.formula( ~ x1 + x2)
form3 <- as.formula( ~ x1 + x2)
cluster <- scrData[,5]
lin.pred <- list(form1, form2, form3)
##
theta <- c(0.7, 0.7)
PEM.ab1 <- c(0.7, 0.7) # prior parameters for 1/sigma_1^2
PEM.ab2 <- c(0.7, 0.7) # prior parameters for 1/sigma_2^2
PEM.ab3 <- c(0.7, 0.7) # prior parameters for 1/sigma_3^2
PEM.alpha1 <- 10 # prior parameters for K1
PEM.alpha2 <- 10 # prior parameters for K2
PEM.alpha3 <- 10 # prior parameters for K3
Psi_v <- diag(1, 3) # MVN cluster-specific random effects
```

⁴See Section A in Supplemental Material to Lee et al. (2015)

```

rho_v <- 100
Psi0 <- diag(1, 3) # DPM cluster-specific random effects
rho0 <- 10
aTau <- 1.5
bTau <- 0.0125
hyperParams.PEM.MVN <- list(theta=theta,
  PEM=list(PEM.ab1=PEM.ab1, PEM.ab2=PEM.ab2, PEM.ab3=PEM.ab3,
    PEM.alpha1=PEM.alpha1, PEM.alpha2=PEM.alpha2, PEM.alpha3=PEM.alpha3),
  MVN=list(Psi_v=Psi_v, rho_v=rho_v))
hyperParams.PEM.DPM <- list(theta=theta,
  PEM=list(PEM.ab1=PEM.ab1, PEM.ab2=PEM.ab2, PEM.ab3=PEM.ab3,
    PEM.alpha1=PEM.alpha1, PEM.alpha2=PEM.alpha2, PEM.alpha3=PEM.alpha3),
  DPM=list(Psi0=Psi0, rho0=rho0, aTau=aTau, bTau=bTau))

##
numReps <- 2000
thin <- 10
burninPerc <- 0.25
mhProp_theta_var <- 0.05
mhProp_Vg_var <- c(0.05, 0.05, 0.05)
Cg <- c(0.2, 0.2, 0.2)
delPertg <- c(0.5, 0.5, 0.5)
rj.scheme <- 1
Kg_max <- c(50, 50, 50)
sg_max <- c(max(Y$time1[Y$event1 == 1]),
  max(Y$time2[Y$event1 == 0 & Y$event2 == 1]),
  max(Y$time2[Y$event1 == 1 & Y$event2 == 1]))
time_lambda1 <- seq(1, sg_max[1], 1)
time_lambda2 <- seq(1, sg_max[2], 1)
time_lambda3 <- seq(1, sg_max[3], 1)
nGam_save <- 0
storeV <- rep(TRUE, 3)
mcmc.PEM <- list(run=list(numReps=numReps, thin=thin, burninPerc=burninPerc),
  storage=list(nGam_save=nGam_save, storeV=storeV),
  tuning=list(mhProp_theta_var=mhProp_theta_var, mhProp_Vg_var=mhProp_Vg_var,
    Cg=Cg, delPertg=delPertg,
    rj.scheme=rj.scheme, Kg_max=Kg_max, sg_max=sg_max,
    time_lambda1=time_lambda1, time_lambda2=time_lambda2,
    time_lambda3=time_lambda3))

##
Sigma_V <- diag(0.1, 3)
Sigma_V[1,2] <- Sigma_V[2,1] <- -0.05
Sigma_V[1,3] <- Sigma_V[3,1] <- -0.06
Sigma_V[2,3] <- Sigma_V[3,2] <- 0.07
##
myModel <- c("semi-Markov", "PEM", "MVN")
myPath <- "Output/05-Results-PEM_MVN/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, cluster)
startValues[[2]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, cluster, MVN.SigmaV=Sigma_V)
##
fit_PEM_MVN <- BayesID_HReg(Y, lin.pred, scrData, cluster, model=myModel,
  hyperParams.PEM.MVN, startValues, mcmc.PEM, path=myPath)

fit_PEM_MVN
summ.fit_PEM_MVN <- summary(fit_PEM_MVN); names(summ.fit_PEM_MVN)
summ.fit_PEM_MVN
plot(fit_PEM_MVN)
plot(fit_PEM_MVN, plot.est = "BH")
names(fit_PEM_MVN.plot <- plot(fit_PEM_MVN, plot=FALSE))
##
myModel <- c("semi-Markov", "PEM", "DPM")
myPath <- "Output/06-Results-PEM_DPM/"
startValues <- vector("list", 2)
startValues[[1]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, cluster)
startValues[[2]] <- initiate.startValues_HReg(Y, lin.pred, scrData, model=myModel, cluster,
  MVN.SigmaV=Sigma_V)
##
fit_PEM_DPM <- BayesID_HReg(Y, lin.pred, scrData, cluster, model=myModel,
  hyperParams.PEM.DPM, startValues, mcmc.PEM, path=myPath)

fit_PEM_DPM
summ.fit_PEM_DPM <- summary(fit_PEM_DPM); names(summ.fit_PEM_DPM)
summ.fit_PEM_DPM
plot(fit_PEM_DPM)
plot(fit_PEM_DPM, plot.est = "BH")
names(fit_PEM_DPM.plot <- plot(fit_PEM_DPM, plot=FALSE))

```