Joint modelling of multivariate longitudinal and time-to-event data

Graeme L. Hickey[1], Pete Philipson[2], Andrea Jorgensen[1], Ruwanthi-Kolamunnage-Dona[1]

[1]Department of Biostatistics, University of Liverpool, UK
[2]Mathematics, Physics and Electrical Engineering, University of Northumbria, UK

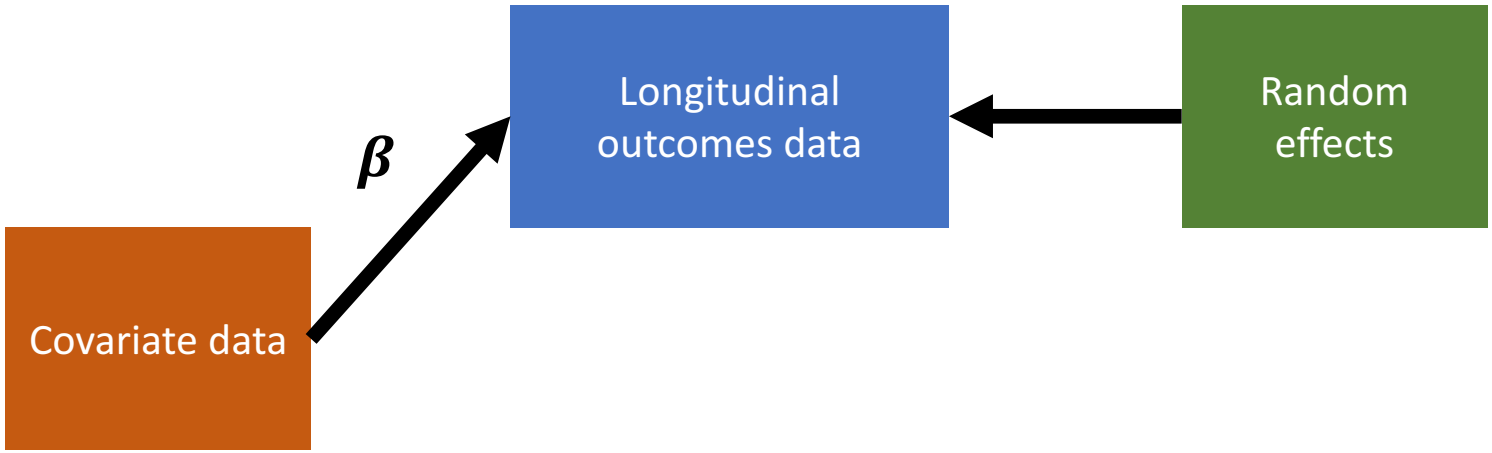
graeme.hickey@liverpool.ac.uk

Funded by 

Grant MR/M013227/1

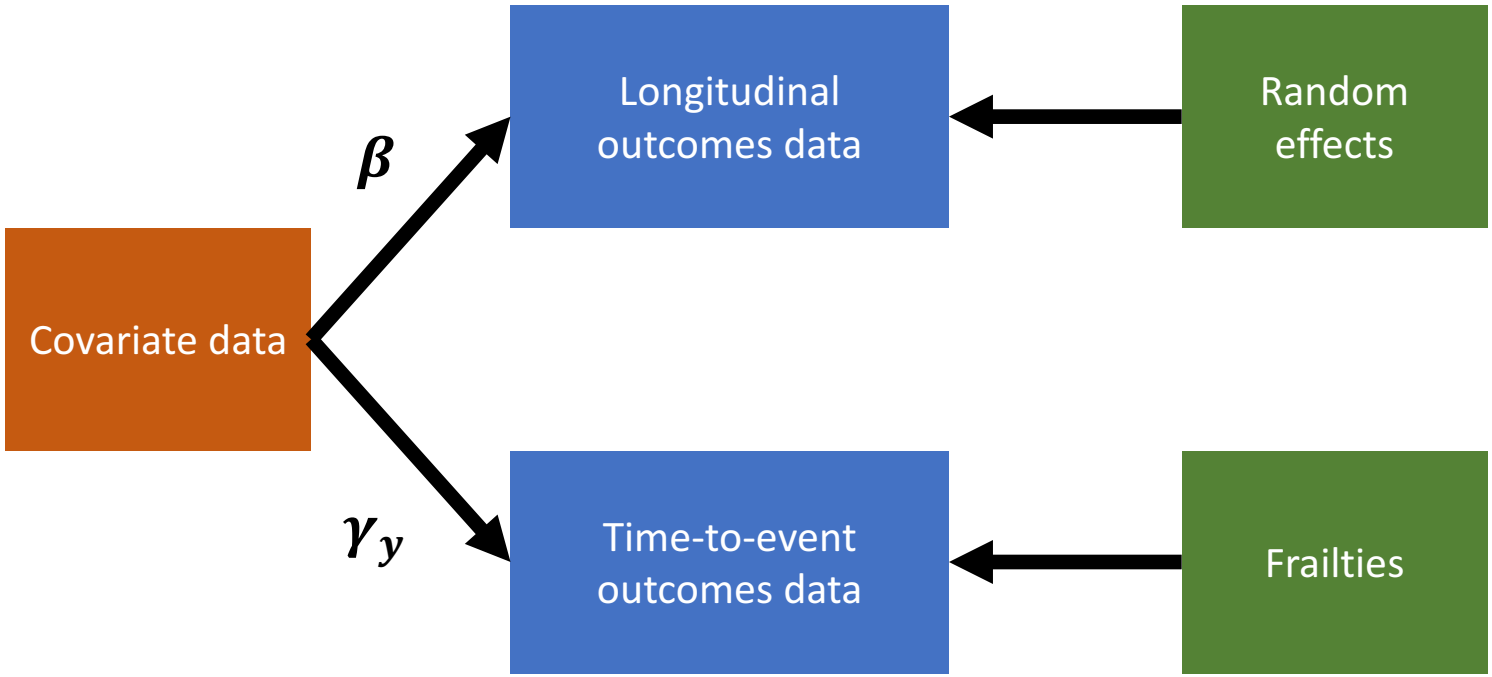
13th November 2017

Joint modelling

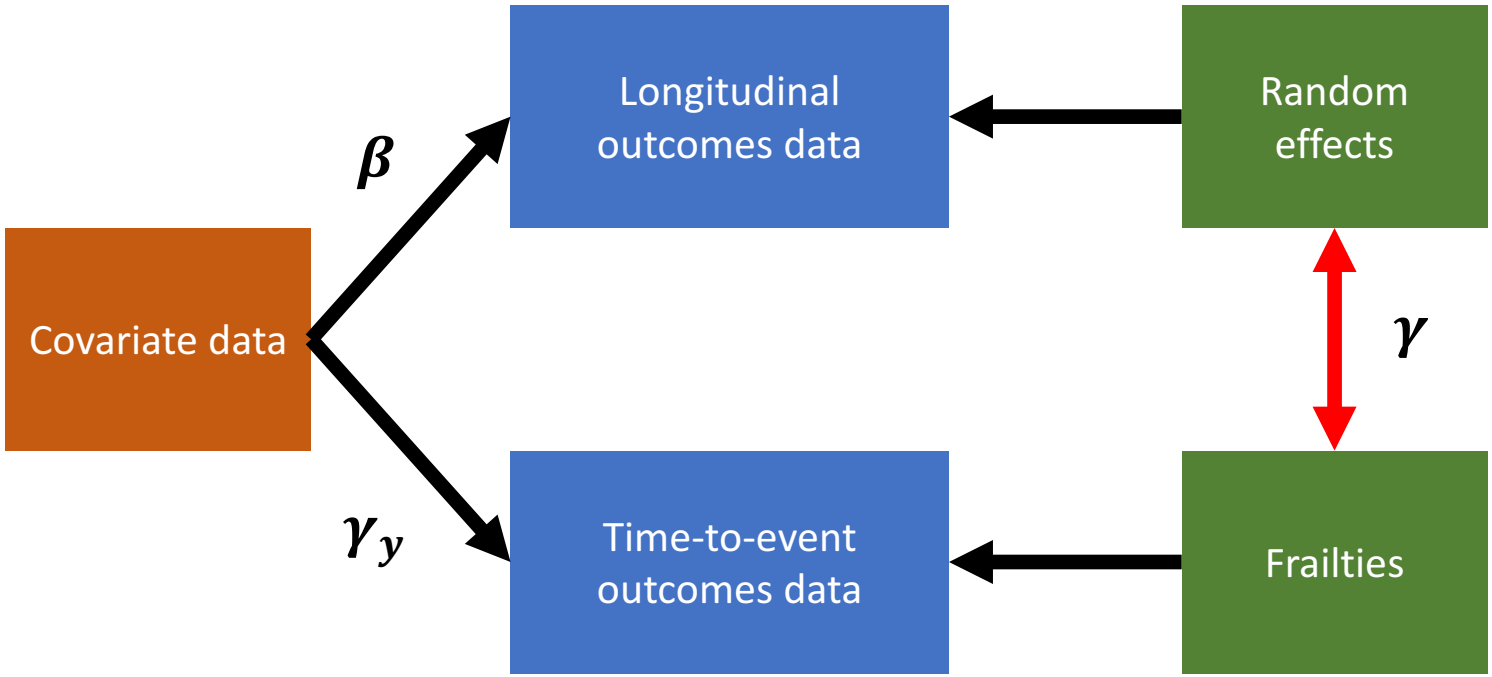


Joint modelling

Joint modelling



Joint modelling



Why use a joint model?

Interest lies with

- adjustment of inferences about longitudinal measurements for possibly outcome-dependent drop-out
- adjustment of inferences about the time-to-event distribution conditional on intermediate and/or error prone longitudinal measurements
- the joint evolution of the measurement and event time processes
- biomarker surrogacy
- dynamic prediction

Motivation for *multivariate* joint models

UNIVERSITY OF LIVERPOOL

- Clinical studies often repeatedly measure *multiple* biomarkers or other measurements **and** an event time
- Research has predominantly focused on a single event time and single measurement outcome
- Ignoring correlation leads to bias and reduced efficiency in estimation
- Harnessing all available information in a single model is advantageous and should lead to improved model predictions

Clinical example

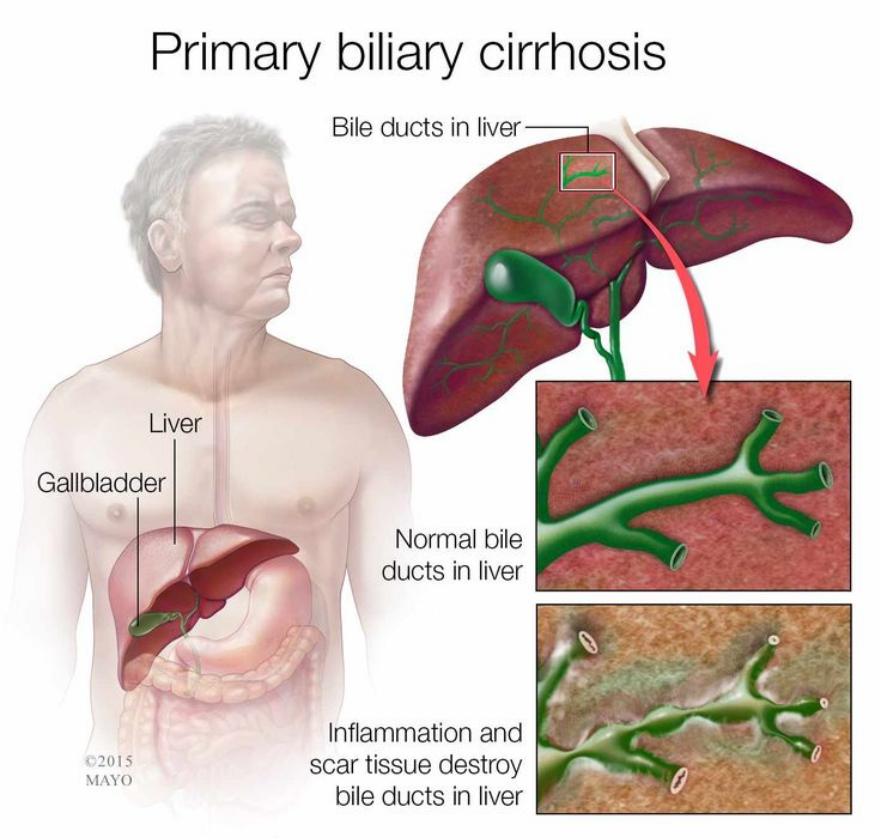


Figure source: `https://www.medgadget.com`

Primary biliary cirrhosis (PBC) is a chronic liver disease characterized by inflammatory destruction of the small bile ducts, which eventually leads to cirrhosis of the liver and death

Clinical example

- Consider a subset of 154 patients randomized to placebo treatment from Mayo Clinic trial (Murtaugh et al. 1994)
- Multiple biomarkers repeatedly measured at intermittent times, of which we consider 3 clinically relevant ones:
  1. serum bilirunbin (mg/dl)
  2. serum albumin (mg/dl)
  3. prothrombin time (seconds)

Objective 1

1. Determine if longitudinal biomarker trajectories are associated with death

Objective 1

1. Determine if longitudinal biomarker trajectories are associated with death

Objective 2

1. Dynamically predict the biomarker trajectories and time to death for a new patient

Objective 1

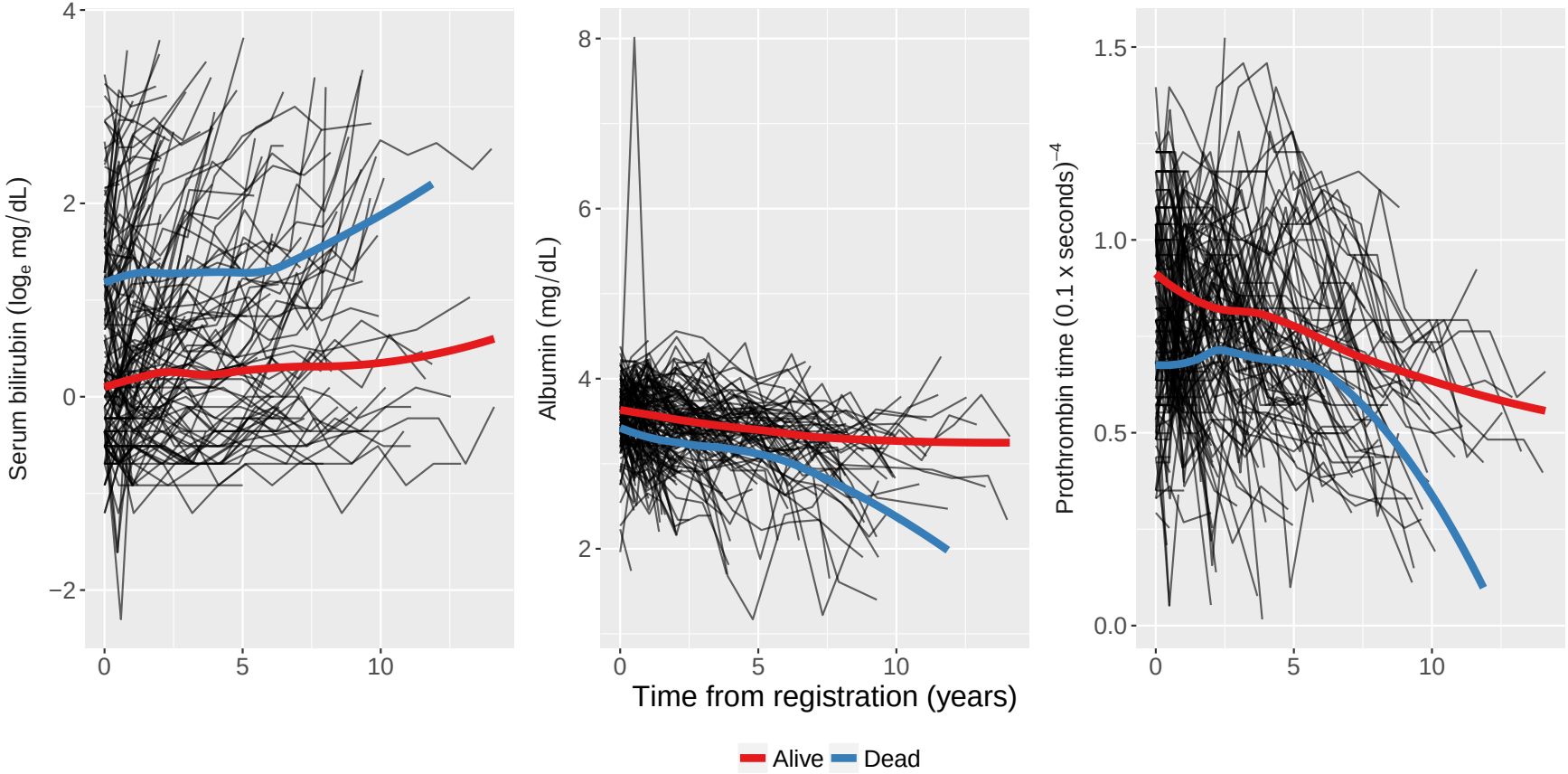
1. Determine if longitudinal biomarker trajectories are associated with death

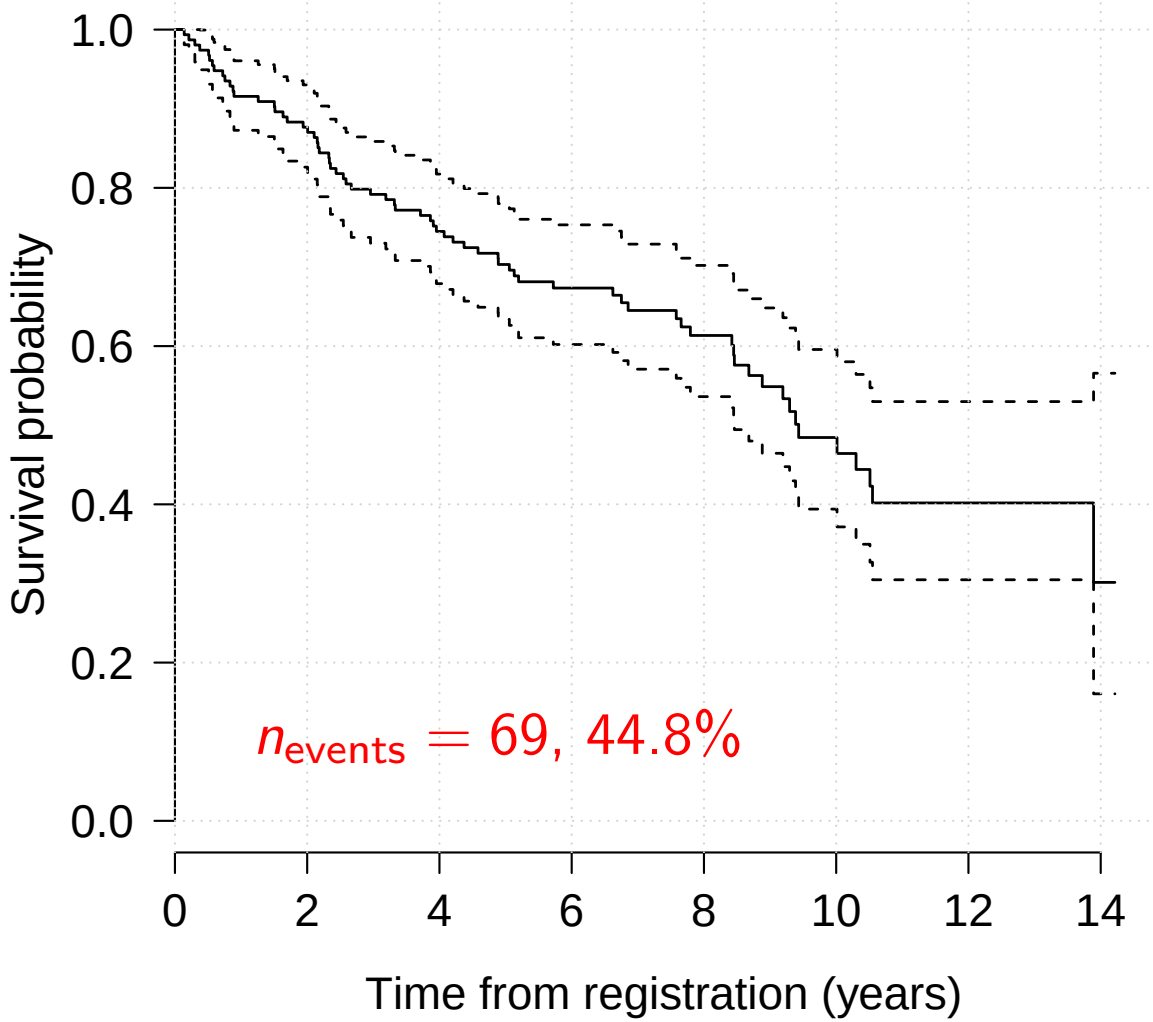
Objective 2

1. Dynamically predict the biomarker trajectories and time to death for a new patient

Objective 3

1. Wrap it all up into a freely available software package


Serum bilirubin (log$_e$ mg/dL)
4
2
0
−2
0
5
10
Albumin (mg/dL)
8
6
4
2
0
5
10
Time from registration (years)
Prothrombin time (0.1 x seconds)$^{-4}$
1.5
1.0
0.5
0.0
0
5
10
Alive
Dead


1.0
0.8
0.6
0.4
0.2
0.0
Survival probability
$n_{\text{events}} = 69,\ 44.8\%$
0
2
4
6
8
10
12
14
Time from registration (years)

Data

For each subject $i = 1, \ldots, n$, we observe

- $y_i = (y_{i1}^\top, \ldots, y_{iK}^\top)$ is a K-variate continuous outcome vector, where each y_{ik} denotes an $(n_{ik} \times 1)$-vector of observed longitudinal measurements for the k-th outcome type: $y_{ik} = (y_{i1k}, \ldots, y_{in_{ik}k})^\top$
- Observation times t_{ijk} for $j = 1, \ldots, n_{ik}$, which can differ between subjects and outcomes
- (T_i, δ_i), where $T_i = \min(T_i^*, C_i)$, where T_i^* is the true event time, C_i corresponds to a potential right-censoring time, and δ_i is the failure indicator equal to 1 if the failure is observed $(T_i^* \leq C_i)$ and 0 otherwise

Longitudinal sub-model

Following Henderson et al. (2000) for the univariate case

$$y_i(t) = \mu_i(t) + W_{1i}(t) + \varepsilon_i(t),$$

where

- $\varepsilon_i(t)$ is the model error term, which is i.i.d. $N(0, \sigma^2)$ and independent of $W_{1i}(t)$
- $\mu_i(t) = x_i^\top(t)\beta$ is the mean response
- $x_i(t)$ is a p-vector of (possibly) time-varying covariates with corresponding fixed effect terms β
- $W_{1i}(t)$ is a zero-mean *latent* Gaussian process

Longitudinal sub-model

We can extend it to K-separate sub-models (with $k = 1, \ldots, K$)

$$y_{ik}(t) = \mu_{ik}(t) + W_{1i}^{(k)}(t) + \varepsilon_{ik}(t),$$

where

- $\varepsilon_{ik}(t)$ is the model error term, which is i.i.d. $N(0, \sigma_k^2)$ and independent of $W_{1i}^{(k)}(t)$
- $\mu_{ik}(t) = x_{ik}^{\top}(t)\beta_k$ is the mean response
- $x_{ik}(t)$ is a p_k-vector of (possibly) time-varying covariates with corresponding fixed effect terms β_k
- $W_{1i}^{(k)}(t)$ is a zero-mean *latent* Gaussian process

Time-to-event sub-model

$$\begin{aligned}\lambda_i(t) &= \lim_{dt \to 0} \frac{P(t \leq T_i < t + dt \mid T_i \geq t)}{dt} \\ &= \lambda_0(t) \exp\left\{ v_i^\top(t)\gamma_v + W_{2i}(t) \right\},\end{aligned}$$

where

- $\lambda_0(\cdot)$ is an unspecified baseline hazard function
- $v_i(t)$ is a q-vector of (possibly) time-varying covariates with corresponding fixed effect terms γ_v
- $W_{2i}(t)$ is a zero-mean *latent* Gaussian process, independent of the censoring process

Correlation

Following Laird and Ware (1982):

$$W_{1i}^{(k)}(t) = z_{ik}^{\top}(t) b_{ik} \text{ for } k = 1, \ldots, K$$

Correlation

Following Laird and Ware (1982):

$$W_{1i}^{(k)}(t) = z_{ik}^{\top}(t) b_{ik} \text{ for } k = 1, \ldots, K$$

Three sources of correlation:

Correlation

Following Laird and Ware (1982):

$$W_{1i}^{(k)}(t) = z_{ik}^{\top}(t) b_{ik} \text{ for } k = 1, \ldots, K$$

Three sources of correlation:

1. Within-subject correlation between longitudinal measurements: $b_{ik} \sim N(0, D_{kk})$

Correlation

Following Laird and Ware (1982):

$$W_{1i}^{(k)}(t) = z_{ik}^{\top}(t) b_{ik} \text{ for } k = 1, \ldots, K$$

Three sources of correlation:

1. Within-subject correlation between longitudinal measurements: $b_{ik} \sim N(0, D_{kk})$
2. Between longitudinal outcomes correlation: $\text{cov}(b_{ik}, b_{il}) = D_{kl}$ for $k \neq l$

Correlation

Following Laird and Ware (1982):

$$W_{1i}^{(k)}(t) = z_{ik}^{\top}(t) b_{ik} \text{ for } k = 1, \ldots, K$$

Three sources of correlation:

1. Within-subject correlation between longitudinal measurements: $b_{ik} \sim N(0, D_{kk})$
2. Between longitudinal outcomes correlation: $\text{cov}(b_{ik}, b_{il}) = D_{kl}$ for $k \neq l$
3. Correlation between sub-models[1]: $W_{2i}(t) = \sum_{k=1}^{K} \gamma_{yk} W_{1i}^{(k)}(t)$

[1] Extends model proposed Henderson et al. (2000)

Association structure: alternatives

Many other proposals for association structures in the literature:

- Current value parameterisation: $W_{2i}(t) = \gamma_y \{\mu_i(t) + W_{1i}(t)\}$
- Random effects parameterisation: $W_{2i}(t) = \gamma_{y1}^\top b_i$
- Bivariate distribution: $(W_{1i}, W_{2i}) \sim N(0, \Omega)$
- Random-slopes parameterisation:
  $W_{2i}(t) = \gamma_{y1} \{\mu_i(t) + W_{1i}(t)\} + \gamma_{y2} \frac{\partial}{\partial t} \{\mu_i(t) + W_{1i}(t)\}$
- ...

Likelihood

We can re-write the longitudinal sub-model as

$$y_i \mid b_i, \beta, \Sigma_i \sim N(X_i\beta + Z_i b_i, \Sigma_i), \text{ with } b_i \mid D \sim N(0, D),$$

where $\beta = (\beta_1^\top, \ldots, \beta_K^\top)$, $b_i = (b_{i1}^\top, \ldots, b_{iK}^\top)^\top$, and

$$X_i = \begin{pmatrix} X_{i1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & X_{iK} \end{pmatrix}, \quad D = \begin{pmatrix} D_{11} & \cdots & D_{1K} \\ \vdots & \ddots & \vdots \\ D_{1K}^\top & \cdots & D_{KK} \end{pmatrix}$$

$$Z_i = \begin{pmatrix} Z_{i1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & Z_{iK} \end{pmatrix}, \quad \Sigma_i = \begin{pmatrix} \sigma_1^2 I_{n_{i1}} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_K^2 I_{n_{iK}} \end{pmatrix}$$

Likelihood

The *observed* data likelihood is given by

$$\prod_{i=1}^{n}\left(\int_{-\infty}^{\infty} f(y_i \mid b_i,\theta) f(T_i,\delta_i \mid b_i,\theta) f(b_i \mid \theta) db_i\right)$$

where $\theta = (\beta^\top, \text{vech}(D), \sigma_1^2, \ldots, \sigma_K^2, \lambda_0(t), \gamma_v^\top, \gamma_y^\top)$

Likelihood

The *observed* data likelihood is given by

$$\prod_{i=1}^{n}\left(\int_{-\infty}^{\infty} f(y_i \mid b_i,\theta) f(T_i,\delta_i \mid b_i,\theta) f(b_i \mid \theta) db_i\right)$$

where $\theta = (\beta^\top, \text{vech}(D), \sigma_1^2, \ldots, \sigma_K^2, \lambda_0(t), \gamma_v^\top, \gamma_y^\top)$, and

$$f(y_i \mid b_i,\theta) = \left(\prod_{k=1}^{K}(2\pi)^{-\frac{n_{ik}}{2}}\right)|\Sigma_i|^{-\frac{1}{2}}$$
$$\exp\left\{-\frac{1}{2}(y_i - X_i\beta - Z_i b_i)^\top \Sigma_i^{-1}(y_i - X_i\beta - Z_i b_i)\right\}$$

Likelihood

The *observed* data likelihood is given by

$$\prod_{i=1}^{n}\left(\int_{-\infty}^{\infty} f(y_i \mid b_i,\theta) f(T_i,\delta_i \mid b_i,\theta) f(b_i \mid \theta) db_i\right)$$

where $\theta = (\beta^\top, \text{vech}(D), \sigma_1^2, \ldots, \sigma_K^2, \lambda_0(t), \gamma_v^\top, \gamma_y^\top)$, and

$$f(T_i,\delta_i \mid b_i;\theta) = \left[\lambda_0(T_i)\exp\left\{v_i^\top\gamma_v + W_{2i}(T_i,b_i)\right\}\right]^{\delta_i} \exp\left\{-\int_0^{T_i}\lambda_0(u)\exp\left\{v_i^\top\gamma_v + W_{2i}(u,b_i)\right\}du\right\}$$

Likelihood

The *observed* data likelihood is given by

$$\prod_{i=1}^{n}\left(\int_{-\infty}^{\infty} f(y_i \mid b_i,\theta) f(T_i,\delta_i \mid b_i,\theta) f(b_i \mid \theta) db_i\right)$$

where $\theta = (\beta^\top, \text{vech}(D), \sigma_1^2, \ldots, \sigma_K^2, \lambda_0(t), \gamma_v^\top, \gamma_y^\top)$, and

$$f(b_i \mid \theta) = (2\pi)^{-\frac{r}{2}} |D|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2} b_i^\top D^{-1} b_i\right\},$$

with $r = \dim(b_i)$

Estimation

Multiple approaches have been considered over the years:

- Markov chain Monte Carlo (MCMC)
- Direct likelihood maximisation (e.g. Newton-methods)
- Generalised estimating equations
- EM algorithm (treating the random effects as missing data)
- ...

EM algorithm (Dempster et al. 1977)

E-step. At the m-th iteration, we compute the expected log-likelihood of the *complete* data conditional on the *observed* data and the current estimate of the parameters.

$$\begin{aligned} Q(\theta \mid \hat{\theta}^{(m)}) &= \sum_{i=1}^{n} \mathbb{E}\Big\{ \log f(y_i, T_i, \delta_i, b_i \mid \theta) \Big\}, \\ &= \sum_{i=1}^{n} \int_{-\infty}^{\infty} \Big\{ \log f(y_i, T_i, \delta_i, b_i \mid \theta) \Big\} f(b_i \mid T_i, \delta_i, y_i; \hat{\theta}^{(m)}) db_i \end{aligned}$$

EM algorithm (Dempster et al. 1977)

E-step. At the m-th iteration, we compute the expected log-likelihood of the *complete* data conditional on the *observed* data and the current estimate of the parameters.

$$\begin{aligned} Q(\theta \mid \hat{\theta}^{(m)}) &= \sum_{i=1}^{n} \mathbb{E}\Big\{ \log f(y_i, T_i, \delta_i, b_i \mid \theta) \Big\}, \\ &= \sum_{i=1}^{n} \int_{-\infty}^{\infty} \Big\{ \log f(y_i, T_i, \delta_i, b_i \mid \theta) \Big\} f(b_i \mid T_i, \delta_i, y_i; \hat{\theta}^{(m)}) db_i \end{aligned}$$

M-step. We maximise $Q(\theta \mid \hat{\theta}^{(m)})$ with respect to θ. namely,

$$\hat{\theta}^{(m+1)} = \arg \max_{\theta} Q(\theta \mid \hat{\theta}^{(m)})$$

M-step: closed form estimators

$$
\begin{aligned}
\hat{\lambda}_0(t) &= \frac{\sum_{i=1}^{n} \delta_i I(T_i = t)}{\sum_{i=1}^{n} \mathbb{E}\left[\exp\left\{v_i^\top \gamma_v + W_{2i}(t, b_i)\right\}\right] I(T_i \geq t)} \\
\hat{\beta} &= \left(\sum_{i=1}^{n} X_i^\top X_i\right)^{-1} \left(\sum_{i=1}^{n} X_i^\top (y_i - Z_i \mathbb{E}[b_i])\right) \\
\hat{\sigma}_k^2 &= \frac{1}{\sum_{i=1}^{n} n_{ik}} \sum_{i=1}^{n} \Big\{ (y_{ik} - X_{ik}\beta_k)^\top (y_{ik} - X_{ik}\beta_k - 2Z_{ik}\mathbb{E}[b_{ik}]) \\
&\qquad + \text{trace}\left(Z_{ik}^\top Z_{ik} \mathbb{E}[b_{ik} b_{ik}^\top]\right) \Big\} \\
\hat{D} &= \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}\left[b_i b_i^\top\right]
\end{aligned}
$$

M-step: non-closed form estimators

There is no closed form update for $\gamma = (\gamma_v^\top, \gamma_y^\top)$, so use a one-step Newton-Raphson iteration

$$\hat{\gamma}^{(m+1)} = \hat{\gamma}^{(m)} + I\left(\hat{\gamma}^{(m)}\right)^{-1} S\left(\hat{\gamma}^{(m)}\right),$$

where

$$\begin{aligned} S(\gamma) &= \sum_{i=1}^{n} \left[\delta_i \mathbb{E}\left[\tilde{v}_i(T_i)\right] - \int_0^{T_i} \lambda_0(u) \mathbb{E}\left[\tilde{v}_i(u) \exp\{\tilde{v}_i^\top(u)\gamma\}\right] du \right] \\ I(\gamma) &= -\frac{\partial}{\partial \gamma} S(\gamma) \end{aligned}$$

with $\tilde{v}_i(t) = \left(v_i^\top, z_{i1}^\top(t) b_{i1}, \ldots, z_{iK}^\top(t) b_{iK}\right)$ a $(q+K)$–vector

MCEM algorithm

- E-step requires calculating several multidimensional integrals of form $\mathbb{E}\left[h(b_i) \mid T_i, \delta_i, y_i; \hat{\theta}\right]$
- Gauss-quadrature can be slow if $\dim(b_i)$ is large $\Rightarrow$ might not scale well as K increases
- Instead, we use the Monte Carlo Expectation-Maximization (MCEM; Wei and Tanner 1990)
- M-step updates remain the same

Monte Carlo E-step

Conventional EM algorithm: use quadrature to compute

$$\mathbb{E}\left[h(b_i) \mid T_i, \delta_i, y_i; \hat{\theta}\right] = \frac{\int_{-\infty}^{\infty} h(b_i) f(b_i \mid y_i; \hat{\theta}) f(T_i, \delta_i \mid b_i; \hat{\theta}) db_i}{\int_{-\infty}^{\infty} f(b_i \mid y_i; \hat{\theta}) f(T_i, \delta_i \mid b_i; \hat{\theta}) db_i},$$

where

$$\begin{aligned} h(\cdot) &= \text{any known fuction,} \\ b_i \mid y_i, \theta &\sim N\left(A_i \left\{Z_i^\top \Sigma_i^{-1}(y_i - X_i\beta)\right\}, A_i\right), \text{ and} \\ A_i &= \left(Z_i^\top \Sigma_i^{-1} Z_i + D^{-1}\right)^{-1} \end{aligned}$$

Monte Carlo E-step

MCEM algorithm E-step: use Monte Carlo integration to compute

$$\mathbb{E}\left[h(b_i) \mid T_i, \delta_i, y_i; \hat{\theta}\right] \approx \frac{\frac{1}{N}\sum_{d=1}^{N} h\left(b_i^{(d)}\right) f\left(T_i, \delta_i \mid b_i^{(d)}; \hat{\theta}\right)}{\frac{1}{N}\sum_{d=1}^{N} f\left(T_i, \delta_i \mid b_i^{(d)}; \hat{\theta}\right)}$$

where

$$\begin{aligned} h(\cdot) &= \text{any known fuction,} \\ b_i \mid y_i, \theta &\sim N\left(A_i\left\{Z_i^\top \Sigma_i^{-1}(y_i - X_i\beta)\right\}, A_i\right), \text{ and} \\ A_i &= \left(Z_i^\top \Sigma_i^{-1} Z_i + D^{-1}\right)^{-1} \\ b_i^{(1)}, b_i^{(2)}, \ldots, b_i^{(N)} &\sim b_i \mid y_i, \theta \text{ a Monte Carlo draw} \end{aligned}$$

Speeding up convergence

- Monte Carlo integration converges at a rate of $O(N^{-1/2})$, which is independent of K and $r = \dim(\boldsymbol{b}_i)$
- EM algorithm convergences linearly
- Can we speed this up?

Speeding up convergence

- Monte Carlo integration converges at a rate of $O(N^{-1/2})$, which is independent of K and $r = \dim(\boldsymbol{b}_i)$
- EM algorithm convergences linearly
- Can we speed this up?
  1. Antithetic variates
  2. Quasi-Monte Carlo

Variance reduction

Instead of directly sampling from the MVN distribution for $b_i \mid y_i; \theta$, we apply a variance reduction technique

Antithetic simulation

Sample $\Omega \sim N(0, I_r)$ and obtain the *pairs*

$$A_i \left\{ Z_i^\top \Sigma_i^{-1} (y_i - X_i \beta) \right\} \pm C_i \Omega,$$

where C_i is the Cholesky decomposition of A_i such that $C_i C_i^\top = A_i$

Negative correlation between the $N/2$ pairs $\Rightarrow$ smaller variance in the sample means than would be obtained from N independent simulations

Convergence

In standard EM, convergence usually declared at $(m+1)$-th iteration if one of the following criteria satisfied

- Relative change: $\Delta_{\text{rel}}^{(m+1)} = \max\left\{\frac{|\hat{\theta}^{(m+1)} - \hat{\theta}^{(m)}|}{|\hat{\theta}^{(m)}| + \epsilon_1}\right\} < \epsilon_0$
- Absolute change: $\Delta_{\text{abs}}^{(m+1)} = \max\left\{|\hat{\theta}^{(m+1)} - \hat{\theta}^{(m)}|\right\} < \epsilon_2$

for some choice of ϵ_0, ϵ_1, and ϵ_2

Convergence

In MCEM framework, there are 2 complications to account for

Convergence

In MCEM framework, there are 2 complications to account for

1. spurious convergence declared due to random chance

Convergence

In MCEM framework, there are 2 complications to account for

1. spurious convergence declared due to random chance

   $\Rightarrow$ **Solution**: require convergence for 3 iterations in succession

Convergence

In MCEM framework, there are 2 complications to account for

1. spurious convergence declared due to random chance
   $\Rightarrow$ **Solution**: require convergence for 3 iterations in succession
2. estimators swamped by Monte Carlo error, thus precluding convergence

Convergence

In MCEM framework, there are 2 complications to account for

1. spurious convergence declared due to random chance
   $\Rightarrow$ **Solution**: require convergence for 3 iterations in succession
2. estimators swamped by Monte Carlo error, thus precluding convergence
   $\Rightarrow$ **Solution**: increase Monte Carlo size N as algorithm moves closer towards maximizer

Dynamic MC size

- Using large N when far from maximizer = computationally inefficient
- Using small N when close to maximizer = unlikely to detect convergence

Solution (proposed by Ripatti et al. 2002): after a 'burn-in' phase, calculate the *coefficient of variation* statistic

$$\text{cv}(\Delta_{\text{rel}}^{(m+1)}) = \frac{\text{sd}(\Delta_{\text{rel}}^{(m-1)}, \Delta_{\text{rel}}^{(m)}, \Delta_{\text{rel}}^{(m+1)})}{\text{mean}(\Delta_{\text{rel}}^{(m-1)}, \Delta_{\text{rel}}^{(m)}, \Delta_{\text{rel}}^{(m+1)})},$$

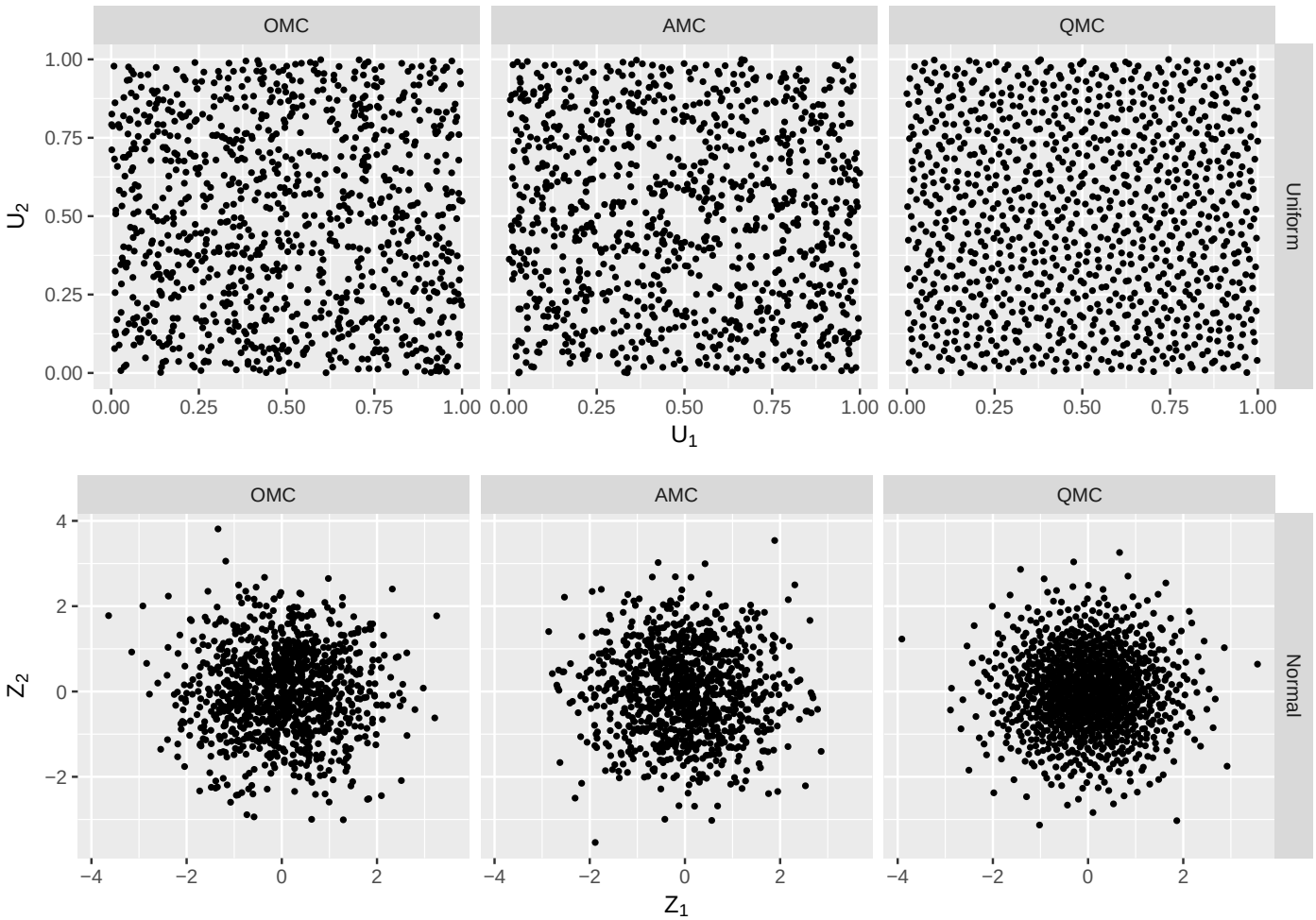
and increase N to $N + \lfloor N/\delta \rfloor$ if $\text{cv}(\Delta_{\text{rel}}^{(m+1)}) > \text{cv}(\Delta_{\text{rel}}^{(m)})$ for some small positive integer δ

Quasi-Monte Carlo

UNIVERSITY OF LIVERPOOL

- Replaces the (pseudo-)random sequence by a deterministic one
- Quasi-random sequences yield smaller errors than standard Monte Carlo integration methods
- Convergence is $O\left(\frac{(logN)^r}{N}\right)$
- Research on-going...

Quasi-Monte Carlo



Key: OMC = ordinary Monte Carlo; AMC = antithetic Monte Carlo; QMC = quasi-Monte Carlo

Standard error estimation

Method 1: Bootstrap

Conceptually simple + theoretically superior (Hsieh et al. 2006)...

but computationally slow!

Standard error estimation

Method 1: Bootstrap

Conceptually simple + theoretically superior (Hsieh et al. 2006)...

but computationally slow!

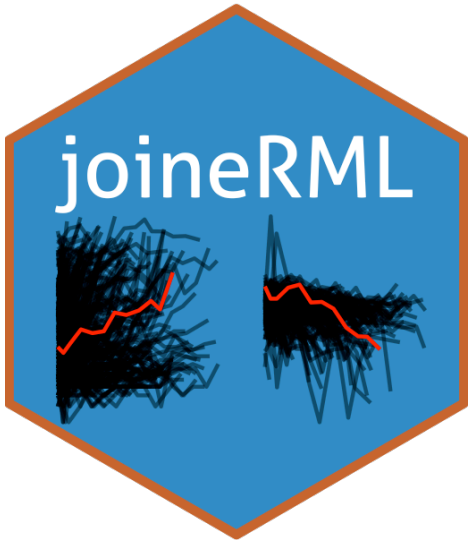
Method 2: Empirical information matrix approximation

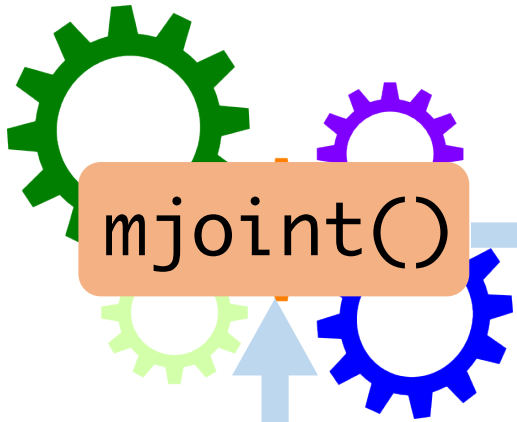
Following McLachlan and Krishnan (2008), $\text{SE}(\theta) \approx I_e^{-1/2}(\hat{\theta})$, where

$$I_e(\theta) = \sum_{i=1}^{n} s_i(\theta) s_i^\top(\theta) - \frac{1}{n} S(\theta) S^\top(\theta),$$

$S(\theta) = \sum_{i=1}^{n} s_i(\theta)$ is the score vector for $\theta_{-\lambda_0(t)}$ (baseline hazards a profiled out of the likelihood)

joineRML





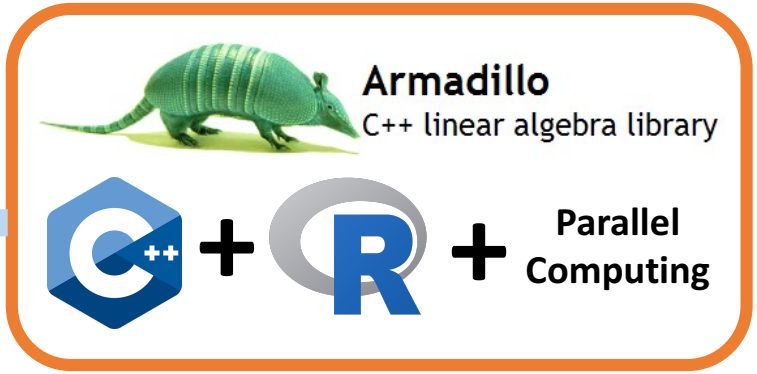
Rich collection of associated methods

* associated with additional plot methods

getVarCov()	print()
vcov()	summary()
fixef()	plot()
ranef()*	sigma()
AIC()	coef()
BIC()	update()
confint()	baseHaz()
formula()	residuals()
sampleData()	fitted()
dynSurv()*	logLik()
dynLong()*	bootSE()

Version 0.3.0 available on CRAN
https://cran.r-project.org/web/packages/joineRML/

Developmental version available on GitHub
https://github.com/graemeleehickey/joineRML



Alternative options

- Pre-2017: none!
- 2017-onwards:
  - `joineRML`: discussed today
  - `stjm`: a new extension to the Stata package[2] written by Michael Crowther
  - `megenreg`: similar to `stjm`, but can handle other models
  - `rstanarm`: development branch that absorbs package written by Sam Brilleman[3]
  - `JMbayes`: a new extension[4] to the R package written by Dimitris Rizopoulos

[2] Crowther MJ. *Joint Statistical Meeting.* Seattle; 2015.

[3] `github.com/sambrilleman/rstanjm`

[4] `github.com/drizopoulos/JMbayes`

Proposed model for PBC data

UNIVERSITY OF LIVERPOOL

Longitudinal sub-model

$$
\begin{aligned}
\log(\texttt{serBilir}) &= (\beta_{0,1} + b_{0i,1}) + (\beta_{1,1} + b_{1i,1})\texttt{year} + \varepsilon_{ij1}, \\
\texttt{albumin} &= (\beta_{0,2} + b_{0i,2}) + (\beta_{1,2} + b_{1i,2})\texttt{year} + \varepsilon_{ij2}, \\
(0.1 \times \texttt{prothrombin})^{-4} &= (\beta_{0,3} + b_{0i,3}) + (\beta_{1,3} + b_{1i,3})\texttt{year} + \varepsilon_{ij3}, \\
b_i &\sim N_6(0, D), \text{ and } \varepsilon_{ijk} \sim N(0, \sigma_k^2) \text{ for } k = 1, 2, 3;
\end{aligned}
$$

Time-to-event sub-model

$$
\begin{aligned}
\lambda_i(t) &= \lambda_0(t) \exp\left\{\gamma_v \text{age} + W_{2i}(t)\right\}, \\
W_{2i}(t) &= \gamma_{\texttt{bil}}(b_{0i,1} + b_{1i,1}t) + \gamma_{\texttt{alb}}(b_{0i,2} + b_{1i,2}t) + \gamma_{\texttt{pro}}(b_{0i,3} + b_{1i,3}t).
\end{aligned}
$$

Example code

```
data(pbc2)
placebo <- subset(pbc2, drug == "placebo")
fit.pbc <- mjoint(
    formLongFixed = list(
        "bil" = log(serBilir) ~ year,
        "alb" = albumin ~ year,
        "pro" = (0.1 * prothrombin)^-4 ~ year),
    formLongRandom = list(
        "bil" = ~ year | id,
        "alb" = ~ year | id,
        "pro" = ~ year | id),
    formSurv = Surv(years, status2) ~ age,
    data = placebo,
    timeVar = "year",
    control = list(tol0 = 0.001, burin = 400))
```

Results

Parameter	Estimate	SE	95% CI
$\beta_{0,1}$	0.5541	0.0858	(0.3859, 0.7223)
$\beta_{1,1}$	0.2009	0.0201	(0.1616, 0.2402)
$\beta_{0,2}$	3.5549	0.0356	(3.4850, 3.6248)
$\beta_{1,2}$	-0.1245	0.0101	(-0.1444, -0.1047)
$\beta_{0,3}$	0.8304	0.0212	(0.7888, 0.8719)
$\beta_{1,3}$	-0.0577	0.0062	(-0.0699, -0.0456)
γ_v	0.0462	0.0151	(0.0166, 0.0759)
$\gamma_{\texttt{bil}}$	0.8181	0.2046	(0.4171, 1.2191)
$\gamma_{\texttt{alb}}$	-1.7060	0.6181	(-2.9173, -0.4946)
$\gamma_{\texttt{pro}}$	-2.2085	1.6070	(-5.3582, 0.9412)

Results

Parameter	Estimate	SE	95% CI
$\beta_{0,1}$	0.5541	0.0858	(0.3859, 0.7223)
$\beta_{1,1}$	0.2009	0.0201	(0.1616, 0.2402)
$\beta_{0,2}$	3.5549	0.0356	(3.4850, 3.6248)
$\beta_{1,2}$	-0.1245	0.0101	(-0.1444, -0.1047)
$\beta_{0,3}$	0.8304	0.0212	(0.7888, 0.8719)
$\beta_{1,3}$	-0.0577	0.0062	(-0.0699, -0.0456)
γ_v	0.0462	0.0151	(0.0166, 0.0759)
$\gamma_{\texttt{bil}}$	0.8181	0.2046	(0.4171, 1.2191)
$\gamma_{\texttt{alb}}$	-1.7060	0.6181	(-2.9173, -0.4946)
$\gamma_{\texttt{pro}}$	-2.2085	1.6070	(-5.3582, 0.9412)

Results

Effect of multivariate inference over univariate joint model:

Parameter	Model	Estimate	95% CI
$\gamma_{\texttt{bil}}$	UV	1.2182	(0.9789, 1.6130)
$\gamma_{\texttt{bil}}$	MV	0.8181	(0.4171, 1.2191)
$\gamma_{\texttt{alb}}$	UV	-3.0770	(-4.4865, -2.3466)
$\gamma_{\texttt{alb}}$	MV	-1.7060	(-2.9173, -0.4946)
$\gamma_{\texttt{pro}}$	UV	-7.2078	(-10.5410, -5.3917)
$\gamma_{\texttt{pro}}$	MV	-2.2085	(-5.3582, 0.9412)

UV = univariate joint model (fitted with `joineR` package); MV = multivariate joint model

Dynamic prediction

- So far we have only discussed inference from joint models
- How we can use them for prediction?
- Predict what?
  1. Failure probability at time $u > t$ given longitudinal data observed up until time t
  2. Longitudinal trajectories at time $u > t$ given longitudinal data observed up until time t

Dynamic prediction: example

UNIVERSITY OF LIVERPOOL

Bivariate joint model

We will consider the PBC data again (as above) with $K = 2$ biomarkers only: serurm bilirubin (log-transformed) and albumin (untransformed), since prothrombin time was non-significant in the trivariate model

Dynamic prediction: survival

For a new subject $i = n + 1$, we want to calculate

$$P[T^*_{n+1} \geq u \mid T^*_{n+1} > t, \boldsymbol{y}_{n+1}; \theta] = \mathbb{E}\left[\frac{S_{n+1}\left(u \mid \mathcal{W}_{2,n+1}(u, b_{n+1}; \theta); \theta\right)}{S_{n+1}\left(t \mid \mathcal{W}_{2,n+1}(t, b_{n+1}; \theta); \theta\right)}\right],$$

where $\mathcal{W}_{2i}(t, b_i; \theta) = \{W_{2i}(s, v_i; \theta); 0 \leq s < t\}$ and the expectation is taken with respect to the distribution

$$p(b_{n+1} \mid T^*_{n+1} > t, \boldsymbol{y}_{n+1}; \theta)$$

Dynamic prediction: survival

Rizopoulos (2011) proposed two estimators for this:

1. A first-order approximation

$$P[T^*_{n+1} \geq u \mid T^*_{n+1} > t, \boldsymbol{y}_{n+1}; \theta] \approx \frac{S_{n+1}\left(u \mid \mathcal{W}_{2,n+1}(u, \hat{b}_{n+1}; \hat{\theta}_{\text{mle}}); \hat{\theta}_{\text{mle}}\right)}{S_{n+1}\left(t \mid \mathcal{W}_{2,n+1}(t, \hat{b}_{n+1}; \hat{\theta}_{\text{mle}}); \hat{\theta}_{\text{mle}}\right)},$$

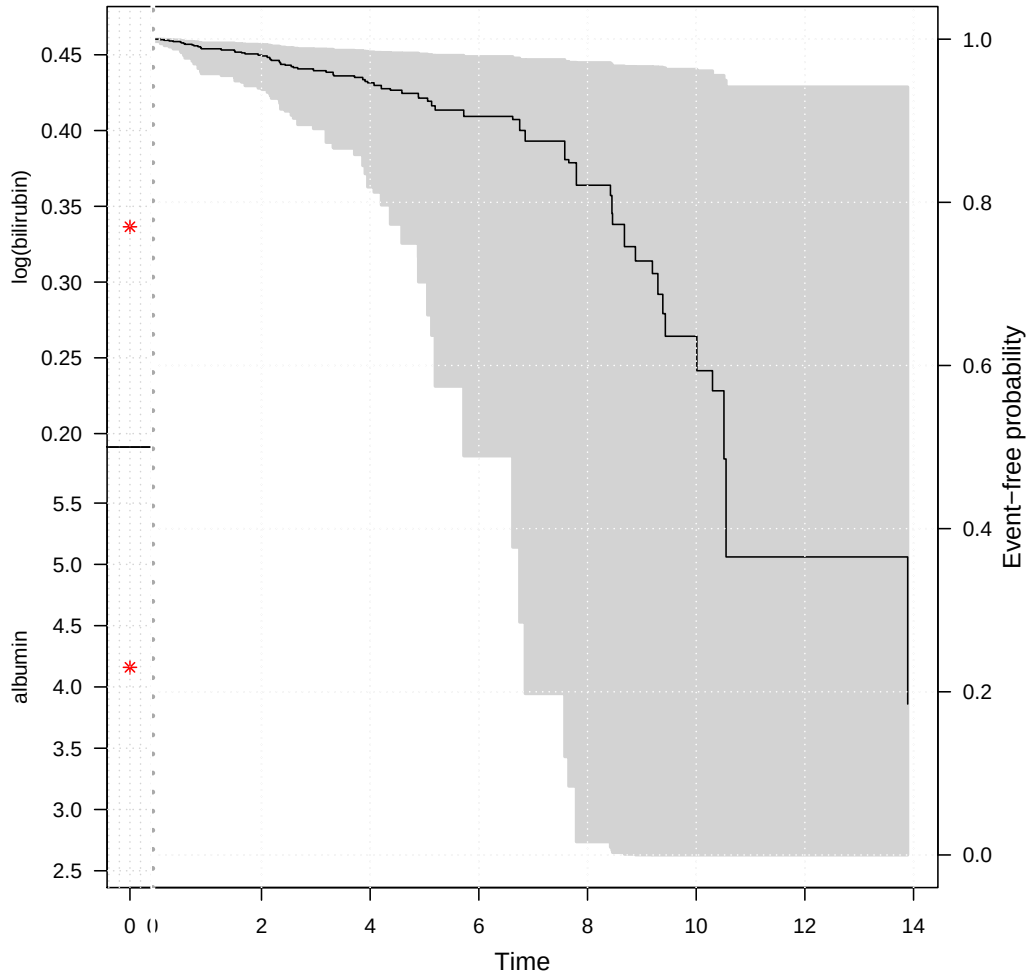
   where $\hat{b}_{n+1}$ is the mode of $p(b_{n+1} \mid T^*_{n+1} > t, \boldsymbol{y}_{n+1}; \theta)$

2. A simulated scheme
   1. Draw $\theta^{(l)} \sim N(\hat{\theta}_{\text{mle}}, V(\hat{\theta}_{\text{mle}}))$
   2. Draw $b^{(l)}_{n+1} \sim p(b_{n+1} \mid T^*_{n+1} > t, \boldsymbol{y}_{n+1}; \theta)$ [Metropolis-Hastings]
   3. Calculate $\frac{S_{n+1}\left(u \mid \mathcal{W}_{2,n+1}(u, b^{(l)}_{n+1}; \theta^{(l)}); \theta^{(l)}\right)}{S_{n+1}\left(t \mid \mathcal{W}_{2,n+1}(t, b^{(l)}_{n+1}; \theta^{(l)}); \theta^{(l)}\right)}$
   4. Repeat Steps 1–3 $l = 2, \ldots, L$ times

Example code

```
# New patient
nd <- subset(placebo, id == "11") # patient 11

# First-order prediction (default)
pred1 <- dynSurv(fit.pbc, nd[1:5, ])
pred1
plot(pred1)

# Simulated prediction
pred2 <- dynSurv(fit.pbc, nd[1:5, ], type = "simulated", scale = 2)
pred2
plot(pred2)
```

Dynamic predicton: survival



Dynamic prediction: longitudinal

For a new subject $i = n + 1$, we want to calculate

$$\mathbb{E}\left[y_{n+1}(u) \mid T^*_{n+1} > t, \boldsymbol{y}_{n+1}; \theta\right] = X^\top_{n+1}(u)\beta + Z^\top_{n+1}(u)\mathbb{E}[b_{n+1}],$$

Dynamic prediction: longitudinal

Again, we can use the same estimation proposals:

1. A first-order approximation

$$\mathbb{E}\left[y_{n+1}(u) \mid T^*_{n+1} > t, \boldsymbol{y}_{n+1}; \theta\right] \approx X^\top_{n+1}(u)\hat{\beta} + Z^\top_{n+1}(u)\hat{b}_{n+1},$$

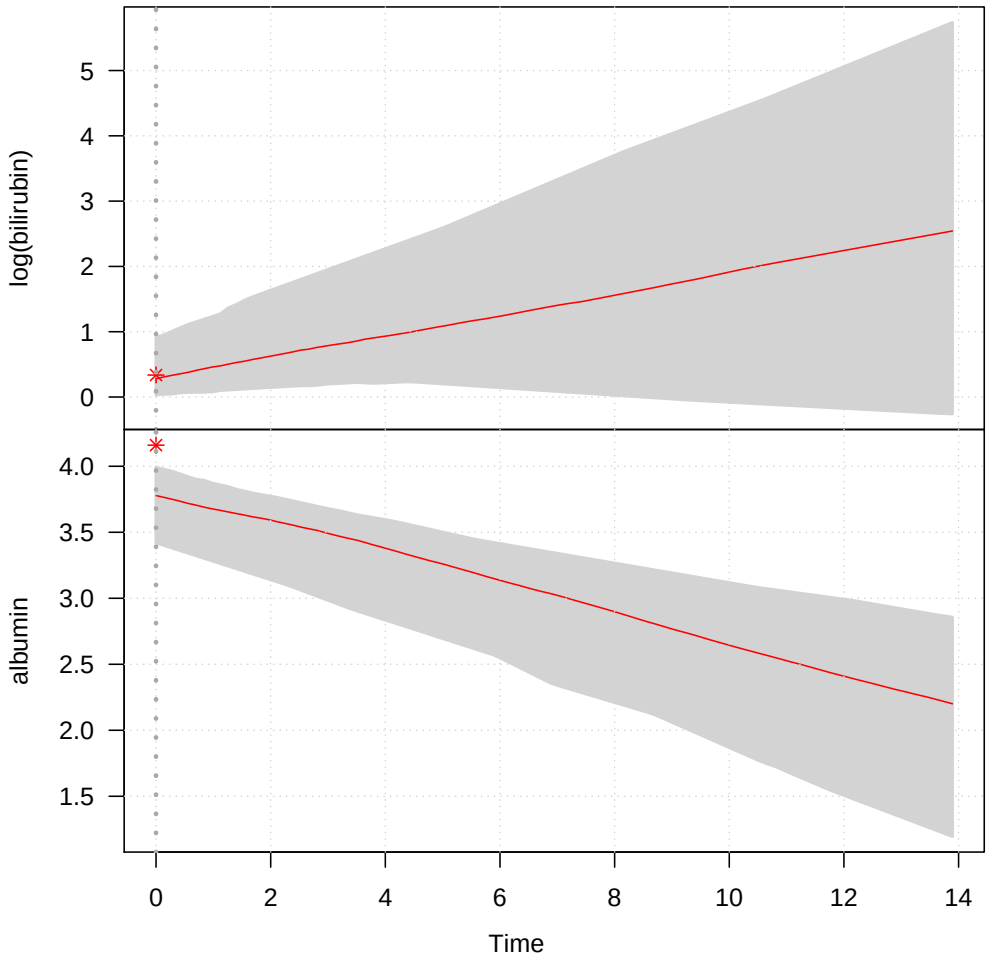
   where $\hat{b}_{n+1}$ is the mode of $p(b_{n+1} \mid T^*_{n+1} > t, \boldsymbol{y}_{n+1}; \theta)$

2. A simulated scheme
   1. Draw $\theta^{(l)} \sim N(\hat{\theta}_{\text{mle}}, V(\hat{\theta}_{\text{mle}}))$
   2. Draw $b^{(l)}_{n+1} \sim p(b_{n+1} \mid T^*_{n+1} > t, \boldsymbol{y}_{n+1}; \theta)$ [Metropolis-Hastings]
   3. Calculate $X^\top_{n+1}(u)\beta^{(l)} + Z^\top_{n+1}(u)b^{(l)}_{n+1}$
   4. Repeat Steps 1–3 $l = 2, \ldots, L$ times

Example code

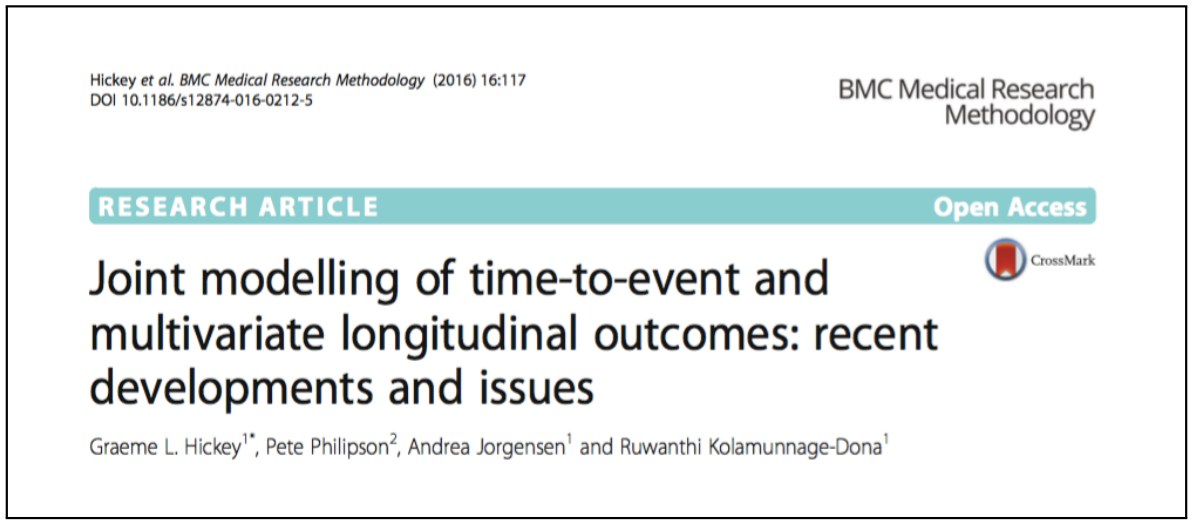
```
# First-order prediction (default)
pred1 <- dynLong(fit.pbc, nd[1:5, ])
pred1
plot(pred1)

# Simulated prediction
pred2 <- dynLong(fit.pbc, nd[1:5, ], type = "simulated", scale = 2)
pred2
plot(pred2)
```

Dynamic predicton: longitudinal



Open challenges

- How can we incorporate high-dimensional K? E.g. $K = 10$?
- Data reduction techniques: can we project high-dimensional K onto a lower order plane?
- Speed-up calculations using approximations (e.g. Laplace approximations)



References


Murtaugh, Paul A et al. (1994). Primary biliary cirrhosis: prediction of short-term survival based on repeated patient visits. *Hepatology* 20(1), pp. 126–134.


Henderson, R et al. (2000). Joint modelling of longitudinal measurements and event time data. *Biostatistics* 1(4), pp. 465–480.


Laird, NM and Ware, JH (1982). Random-effects models for longitudinal data. *Biometrics* 38(4), pp. 963–74.


Dempster, AP et al. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J Roy Stat Soc B* 39(1), pp. 1–38.


Wei, GC and Tanner, MA (1990). A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms. *J Am Stat Assoc* 85(411), pp. 699–704.


Ripatti, S et al. (2002). Maximum likelihood inference for multivariate frailty models using an automated Monte Carlo EM algorithm. *Life Dat Anal* 8(2002), pp. 349–360.


Hsieh, F et al. (2006). Joint modeling of survival and longitudinal data: Likelihood approach revisited. *Biometrics* 62(4), pp. 1037–1043.


McLachlan, GJ and Krishnan, T (2008). The EM Algorithm and Extensions. Second. Wiley-Interscience.


Rizopoulos, Dimitris (2011). Dynamic predictions and prospective accuracy in joint models for longitudinal and time-to-event data. *Biometrics* 67(3), pp. 819–829.