Joint Modelling of Longitudinal and Survival Data: Tools to Evaluate Exposures and Predict Outcome Across the Lifespan

Eleni-Rosalina Andrinopoulou and Rhonda Szczesniak
Erasmus University Medical Center and Cincinnati Childrens Hospital

material based on work by Dimitris Rizopoulos

ACE 2018

September 23, 2018, Cincinnati

Practicals

Practical 1

- We will fit a simple joint model to the PBC dataset
- Start R and load package **JMbayes**, using `library(JMbayes)`
- The longitudinal (long format) and survival information for the PBC patients can be found in data frames `pbc2` and `pbc2.id`. The variables that we will need are:

Practical 1 (cont'd)

- `pbc2`
  - `id`: patient id number
  - `serBilir`: serum bilirubin
  - `year`: follow-up times in years

- `pbc2.id`
  - `years`: observed event times in years
  - `status`: 'alive', 'transplanted', 'dead'
  - `drug`: treatment indicator

Practical 1 (cont'd)

- T1: Fit the linear mixed effects model for log serum bilirubin using function `lme()`, assuming simple linear evolutions over time for each subject, i.e., a simple random-intercepts and random-slopes structure and different average evolutions per treatment group

$$y_i(t) = \beta_0 + \beta_1 t + \beta_2 \{\texttt{D-penic}_i \times t\} + b_{i0} + b_{i1} t + \varepsilon_i(t)$$

- T2: Create the indicator for the composite event (i.e., 'alive' = 0, 'transplanted' or 'dead' = 1) using the code

```
pbc2.id$status2 <- as.numeric(pbc2.id$status != "alive")
```

Practical 1 (cont'd)

- T3: Fit the Cox PH model using `coxph()` that includes only treatment as baseline covariate, remember to set `x = TRUE`

- We want to fit the joint model

$$\begin{cases} y_i(t) &= m_i(t) + \varepsilon_i(t) \\ &= \beta_0 + \beta_1 t + \beta_2 \{\texttt{D-penic}_i \times t\} + b_{i0} + b_{i1} t + \varepsilon_i(t), \quad \varepsilon_i(t) \sim \mathcal{N}(0, \sigma^2), \\ \\ h_i(t) &= h_0(t) \exp\{\gamma \texttt{D-penic}_i + \alpha m_i(t)\}, \end{cases}$$

Practical 1 (cont'd)

- T4: Fit this joint model based on the fitted linear mixed and Cox models using function `jointModelBayes()`
  - ▷ with P-splines baseline hazard

- T5: Use the `summary()` method to obtain a detailed output of the fitted joint model – interpret the results

- T6: Produce 95% confidence intervals for the parameters in the longitudinal submodel, and for the hazard ratios in the survival submodel using function `confint()` (the `parm` argument of `confint()` can take as values `"all"` (default), `"Longitudinal"` and `"Event"`)

Practical 1 (cont'd)

- This model assumes that the strength of the association between the level of serum bilirubin and the risk of the composite event is the same in the the two treatment groups

- To relax this additivity assumption we will add the interaction effect between serum bilirubin and treatment

$$\begin{cases} y_i(t) &= m_i(t) + \varepsilon_i(t) \\ &= \beta_0 + \beta_1 t + \beta_2 \{\texttt{D-penic}_i \times t\} + b_{i0} + b_{i1} t + \varepsilon_i(t), \quad \varepsilon_i(t) \sim \mathcal{N}(0, \sigma^2), \\ \\ h_i(t) &= h_0(t) \exp\big[\gamma \texttt{D-penic}_i + \alpha_1 m_i(t) + \alpha_2 \{\texttt{D-penic}_i \times m_i(t)\}\big], \end{cases}$$

Practical 1 (cont'd)

- To fit this model with package **JMbayes** we need to define the `transFun` argument of `jointModelBayes()`. This should be a function with:
  - ▷ `x`: denoting the term from the longitudinal model
  - ▷ `data`: a data frame that contains variables that potentially should be included in the calculation

- T7: Define this list and fit the corresponding joint model. Use the `summary()` method to obtained a detailed output and interpret the results

Practical 1 (cont'd)

- Based on the fitted joint model we can test for three treatment effects, namely
  - ▷ in the longitudinal process:

$$H_0 : \beta_2 = 0$$

  - ▷ in the survival process:

$$H_0 : \gamma = \alpha_2 = 0$$

  - ▷ in the joint process:

$$H_0 : \beta_2 = \gamma = \alpha_2 = 0$$

Practical 1 (cont'd)

- We would like test these hypotheses using likelihood ratio tests

- T8: Fit the three joint models under the corresponding H_0, and use function `anova()` to perform the LRTs (this function accepts as a first argument the joint model under the null, and as second the joint model under the alternative)

Practical 1 (cont'd)

- We are interested in producing predictions of survival probabilities for Patient 5 using the joint model with the best fitting

- T9: Extract the data of Patient 5 using the code

```
dataP5 <- pbc2[pbc2$id == 5, ]
```

Practical 1 (cont'd)

- T10: Using the first measurement of Patient 5, and the fitted joint model calculate his conditional survival probabilities using function `survfitJM()` and plot it using the `plot` method

- T11: Repeat the same procedure by including each time the next measurement of Patient 155 and see how his survival probabilities evolve dynamically over time as extra prothrobin measurements are recorded
  - ▷ check arguments `conf.int` and `fill.area` of the `plot()` method for including the 95% confidence intervals

Practical 2 - extra

- We will fit a multivariate joint model to the PBC dataset
- Start R and load package **JMbayes**, using `library(JMbayes)`
- The longitudinal (long format) and survival information for the PBC patients can be found in data frames `pbc2` and `pbc2.id`. The variables that we will need are:

Practical 2 - extra (cont'd)

- `pbc2`
  - `id`: patient id number
  - `serBilir`: serum bilirubin
  - `hepatomegaly`: hepatomegaly
  - `year`: follow-up times in years
  - `drug`: treatment indicator

- `pbc2.id`
  - `years`: observed event times in years
  - `status2`: 'alive', 'event'
  - `drug`: treatment indicator
  - `sex`: gender

Practical 2 - extra (cont'd)

- T1: Fit a multivariate mixed effects model for log serum bilirubin, hepatomegaly and prothrombin using function `mvglmer()`, assuming simple linear evolutions over time for each subject, i.e., a simple random-intercepts and random-slopes structure and different average evolutions per treatment group

$$\begin{aligned} y_{i1}(t) &= \beta_{01} + \beta_{11} t + \beta_{21}\{\texttt{D-penic}_i \times t\} + b_{i01} + b_{i11} t + \varepsilon_{i1}(t) \\ logit\{y_{i2}(t)\} &= \beta_{02} + \beta_{12} t + \beta_{22}\{\texttt{D-penic}_i \times t\} + b_{i02} + b_{i12} t + \varepsilon_{i2}(t) \end{aligned}$$

- T2: Create the indicator for the composite event (i.e., 'alive' = 0, 'transplanted' or 'dead' = 1) using the code

```
pbc2.id$status2 <- as.numeric(pbc2.id$status != "alive")
```

Practical 2 - extra (cont'd)

- T3: Fit the Cox PH model using `coxph()` that includes treatment and gender as baseline covariate, remember to set `x = TRUE`

- We want to fit the joint model

$$\begin{cases} y_{i1}(t) &= \beta_{01} + \beta_{11}t + \beta_{21}\{\texttt{D-penic}_i \times t\} + b_{i01} + b_{i11}t + \varepsilon_{i1}(t), \\ logit\{y_{i2}(t)\} &= \beta_{02} + \beta_{12}t + \beta_{22}\{\texttt{D-penic}_i \times t\} + b_{i02} + b_{i12}t + \varepsilon_{i2}(t), \\ \\ h_i(t) &= h_0(t)\exp\{\gamma_1\texttt{D-penic}_i + \gamma_1\texttt{Gender}_i + \\ & \quad \alpha_1 m_{i1}(t) + \alpha_2 m_{i2}(t)\}, \end{cases}$$

Practical 2 - extra (cont'd)

- T4: Fit this joint model based on the fitted multivariate mixed and Cox models using function `mvJointModelBayes()`
  - ▷ with P-splines baseline hazard

- T5: Use the `summary()` method to obtain a detailed output of the fitted joint model – interpret the results

Practical 2 - extra (cont'd)

- The fitted joint model has the value of the longitudinal outcomes. Include also the slope for all the longitudinal outcomes and apply penalties. To fit this model with package **JMbayes** we need to define the `Formulas` argument and `priors` of `mvJointModelBayes()`.

- T6: Define this and fit the corresponding joint model. Use the `summary()` method to obtain a detailed output of the fitted joint model – interpret the results

Practical 2 - extra (cont'd)

- Include also the time-varying effects for all the longitudinal outcomes. To fit this model with package **JMbayes** we need to define the `Interactions` argument of `mvJointModelBayes()`.

- T7: Define this and fit the corresponding joint model. Use the `summary()` method to obtain a detailed output of the fitted joint model – interpret the results

Practical 2 - extra (cont'd)

- We are interested in producing predictions of survival probabilities for Patient 155

- T8: Extract the data of Patient 81 using the code

Practical 2 - extra (cont'd)

- T9: Using the first measurement of Patient 81 and the fitted joint model of T4, calculate his conditional survival probabilities using function `survfitJM()` and plot it using the `plot` method

- T10: Repeat the same procedure by including each time the next measurement of Patient 81 and see how his survival probabilities evolve dynamically over time as extra prothrobin measurements are recorded