Week 5: Dynamic predictions with Joint Modeling

Laurent Briollais (laurent@lunenfeld.ca)

Lunenfeld-Tanenbaum Research Institute, Mount Sinai Hospital
Dalla Lana School of Public Health, UofT

June 3, 2020

Outline of this lecture

- General concepts related to dynamic predictions
- Dynamic predictions of survival probabilities
- Dynamic predictions for the longitudinal outcome
- Effect of the biomarker parametrization on predictions
- Accuracy measures for JMs

1 General concepts

- Often the motivation for building a statistical model is to provide predictions for an outcome of interest.

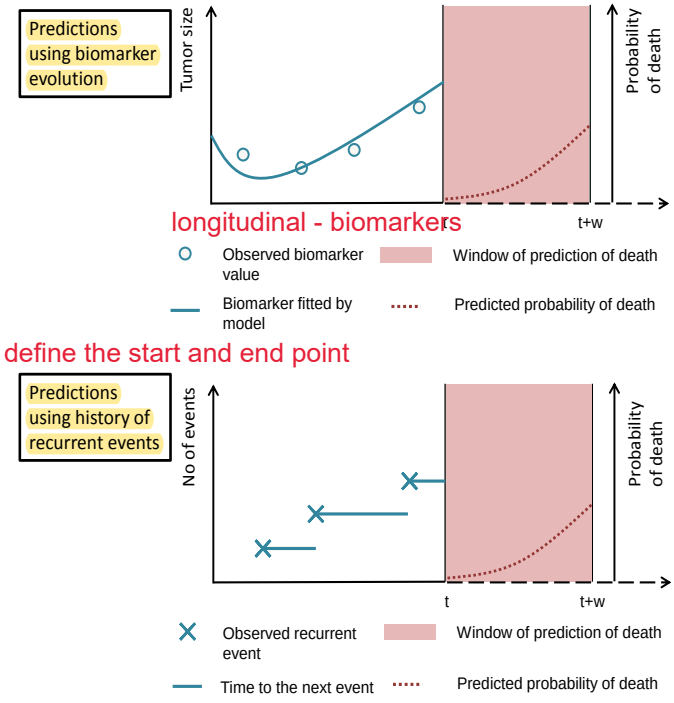
- Specific predictions can be obtained in the framework of JM either for the terminal event, recurrent event or longitudinal outcome.

- Prediction of an event has a great importance in clinical practice with the ultimate goal to provide personalized medicine.

- The predictions have a dynamic nature: as time progresses, additional information is recorded for the patient and thus the predictions can be updated using this new information.

- To assess the quality of the predictions, some measures of predictive accuracy can be computed, e.g. AUC, Brier score, Epoce, etc.

- Conceptual framework:



- $\mathcal{H}_i(t)$ - history of recurrences of individual i until t
  $\mathcal{Y}_i(t)$ - history of the biomarker of individual i until t
- Predicted probability of the terminal event T_i^* in a horizon $[t, t+w]$

$$\mathbb{P}(T_i^* \leq t+w | T_i^* > t, \mathcal{F}_i(t), \boldsymbol{X}_i; \boldsymbol{\xi})$$

$\mathcal{F}_i(t) = \mathcal{H}_i(t)$,
$\mathcal{F}_i(t) = \mathcal{Y}_i(t)$
$\mathcal{F}_i(t) = \{\mathcal{H}_i(t), \mathcal{Y}_i(t)\}$ (see Part 4)

Figure 1: Dynamic predictions: conceptual framework

2 Dynamic predictions of survival probabilities

- We are interested in predicting the survival probability for a new patient i at a time point t during follow-up given all information accumulated up to t, including baseline information and the biomarker longitudinal levels.

- Example: We consider patients 2 and 25 from the PBC dataset that have provided us with 9 and 12 serum bilirubin measurements, respectively.

  $\Rightarrow$ Dynamic Prediction survival probabilities are dynamically updated as additional longitudinal information is recorded.
  biomarkers

- We need to account for the endogenous nature of the marker:

  Providing measurements up to time point t $\Rightarrow$ the patient was still alive at time t

  Predict the terminal event given information include biomarkers and terminal event

- More formally, for a new subject i we have available measurements up to time point t

$$\mathcal{Y}_i(t) = \{y_i(s), 0 \leq s < t\}.$$

- As we have seen in previous lectures, an important characteristic of the endogenous nature of $y_i(t)$ is that it is directly related to the failure mechanism.
  terminal event

- That is, providing longitudinal measurements up to time t, implies survival up to this time point. Hence, it is more relevant to focus on the

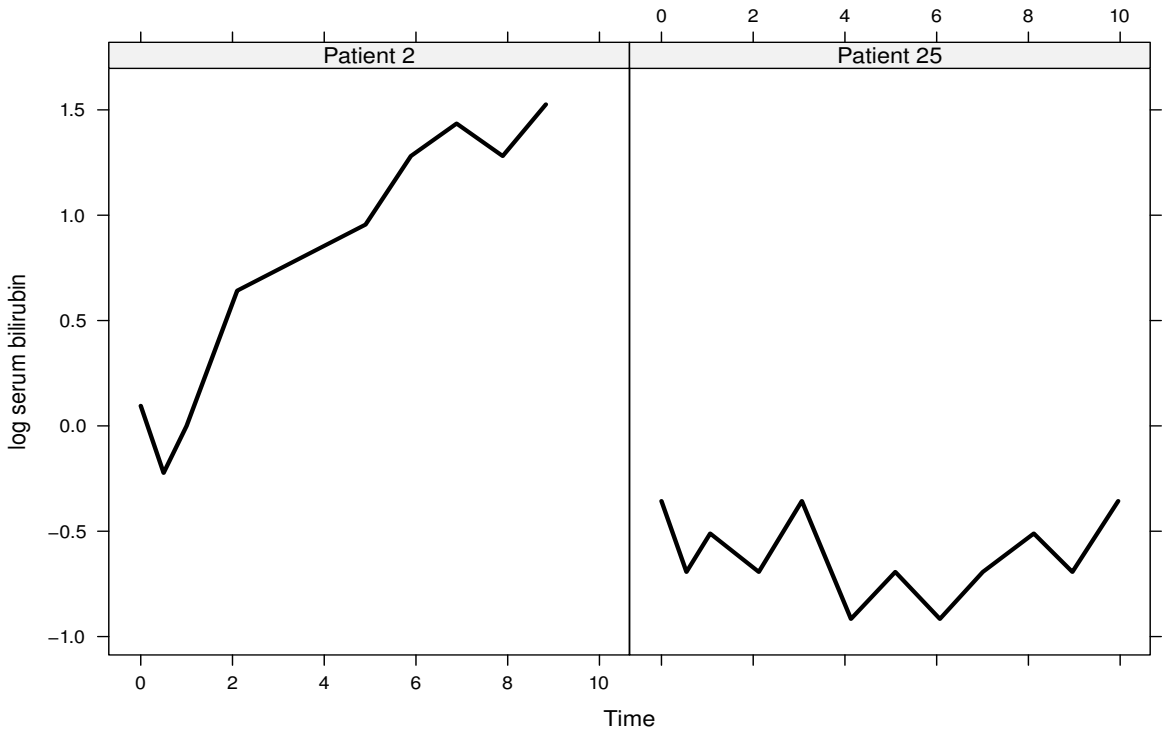


Figure 2: Longitudinal PBC profiles of patients 2 and 25

conditional probability of surviving time $u > t$ given survival up to t, i.e.

u is time point that we want the prediction

$$\pi_i(u|t) = P(T_i^* \geq u | T_i^* > t, \mathcal{Y}_i(t), \mathcal{D}_n), \quad t > 0,$$

what's the prob that this patient's true event time past or equal at time u given that his true event time past time t

where $u > t$ and $\mathcal{D}_n$ denotes the sample on which the joint model was fitted, $\mathcal{D}_n = \{T_i, \delta_i, y_i; \quad i = 1, \cdots, n\}$.

dynamic = nature

- From its definition, note that $\pi_i(u|t)$ has a dynamic nature $\Rightarrow$ when new information is recorded for the patient at time $t' > t$, we can update the prediction to get $\pi_i(u|t')$ and proceed in a time dynamic manner.

- The estimation of the subject-specific conditional survival probabilities is derived from the following equations:

available longitudinal measurements up to time t

$$\pi_i(u|t) = \int \frac{S_i\{u|\mathcal{M}(u, b_i, \theta); \theta\}}{S_i\{t|\mathcal{M}(t, b_i, \theta); \theta\}} P(b_i|T_i^* > t, \mathcal{Y}_i(t); \theta) db_i,$$

where the longitudinal history $\mathcal{M}(.)$ is approximated by the linear mixed-effects model and is a function of both the random effects and the parameters.

- A naive estimator can be constructed by plugging-in the MLEs and the Empirical Bayes estimates

$$\tilde{\pi}_i(u|t) = \int \frac{S_i\{u|\mathcal{M}(u, \hat{b}_i, \hat{\theta}); \hat{\theta}\}}{S_i\{t|\mathcal{M}(t, \hat{b}_i, \hat{\theta}); \hat{\theta}\}}$$

this works relatively well in practice, but the standard errors are difficult to compute.

- It is convenient to proceed using a Bayesian formulation of the problem $\Rightarrow \pi_i(u|t)$ can be written as

$$P(T_i^* \geq u|T_i^* > t, \mathcal{Y}_i(t), \mathcal{D}_n) = \int P(T_i^* \geq u|T_i^* > t, \mathcal{Y}_i(t); \theta) P(\theta|\mathcal{D}_n)$$

- We have already seen the first part of the integrand

- Provided that the sample size is sufficiently large, we can approximate the posterior of the parameters by

introduce the Monte Carlto method
sampling according to the theta and random effects

$$\{\theta|\mathcal{D}_n\} \sim \mathcal{N}(\hat{\theta}; \hat{\mathcal{H}}),$$

where $\hat{\theta}$ are the MLEs of the parameter estimates and $\hat{\mathcal{H}}$ their covariance matrix.

- A Monte Carlo estimate of $\pi_i(u|t)$ can be obtained using the following simulation scheme:
  - Step 1. Draw $\theta^{(l)} \sim \mathcal{N}(\hat{\theta}; \hat{\mathcal{H}})$
  - Step 2. Draw $b_i^{(l)} \sim \{b_i|T_i^* > t, \mathcal{Y}_i(t), \theta^{(l)}\}$
  - Step 3. Compute $\pi_i^{(l)}(u|t) = \frac{S_i\{u|\mathcal{M}(u,b_i^{(l)},\theta^{(l)});\theta^{(l)}\}}{S_i\{t|\mathcal{M}(t,b_i^{(l)},\theta^{(l)});\theta^{(l)}\}}$

- Repeat Steps 1 to 3, $l = 1, \cdots, L$ times, where L denotes the number of Monte Carlo samples.

- Example: Dynamic predictions of survival probabilities for patients 2 & 25 from the PBC dataset: We fit the joint model

- We use 500 Monte Carlo samples, and we took as estimate

$$\hat{\pi}_i(u|t) = \text{median}\{\hat{\pi}_i^{(l)}(u|t), l = 1, \cdots, L\}.$$

and calculated a corresponding 95% point-wise CIs.

- Longitudinal submodel

  For the log serum bilirubin measurements $y_i(t)$ we assume the longitudinal model

$$\begin{aligned} y_i(t) &= m_i(t) + \epsilon_i(t) \\ &= (\beta_0 + b_{i0}) + (\beta_k + b_{ik})^T B(t, 4, 4) + \epsilon_i(t), \end{aligned}$$

??

  where $B(t, 4, 4)$ denotes a B-spline basis matrix for $q - 1$ degree splines with $df - q + 1$ internal knots placed at the percentiles of the follow-up times, β_k and b_{ik} denote the vectors of fixed and random effects corresponding to the B-splines matrix.

  The number of basis function = q + number of internal nodes = q + 1 = 4 + 1 = 5

  two boundary knots eg maximum time in the data and one internal knot

- Survival submodel

  For the survival model, we assume that the risk of the composite event (death or transplantation) depends on treatment, abnormal prothrombin time and the true level of log serum bilirubin

  time for blood clotting

$$h_i(t) = h_0(t) \exp\{\gamma_1 \text{D-pnc}_i + \gamma_2 \text{ProtTime}_i + \alpha m_i(t)\},$$

  drug

  1: outside normal range: [10, 13]

  0: inside normal range

  where ProtTime denotes a dummy variable taking the value 1 when the prothrombin time at baseline was outside the normal range of [10 sec, 13 sec].

- The dynamic survival probability in the package JM is provided by the function survfitJM().

```
library(JM)
data("pbc2.id")
data("pbc2")

pbc2.id$Pro <- with(pbc2.id, factor(prothrombin >=10 & prothrombin <=13,
labels=c("Abnormal", "Normal")))
```
create prothrombin binary variable

```
pbc2$Pro <-rep(pbc2.id$Pro, tapply(pbc2$id, pbc2$id, length))
```
repeat the variable depending the number of observations??????????????????

4 degree of freedoms, 1 internal knot

```
lmeFitBsp.pbc <- lme(fixed=log(serBilir)~bs(year, 4, Boundary.knots = c(0,15)),
                     random=list(id=pdDiag(form=~bs(year, 4, Boundary.knots = c(0,15)))),
                     data=pbc2)
```
B-splines is introduced through fixed effects and random effects

```
coxFit.pbc <- coxph(Surv(years, status2)~drug + Pro, data=pbc2.id, x=TRUE)

JointFitBsp.pbc <- jointModel(lmeFitBsp.pbc, coxFit.pbc, timeVar="year",
                              method="piecewise-PH-aGH")
```
piecewise baseline hazard function

```
summary(JointFitBsp.pbc)

> summary(JointFitBsp.pbc)

Call:
jointModel(lmeObject = lmeFitBsp.pbc, survObject = coxFit.pbc,
    timeVar = "year", method = "piecewise-PH-aGH")

Data Descriptives:
Longitudinal Process Event Process
Number of Observations: 1945 Number of Events: 140 (44.9%)
Number of Groups: 312

Joint Model Summary:
Longitudinal Process: Linear mixed-effects model
Event Process: Relative risk model with piecewise-constant
baseline risk function
Parameterization: Time-dependent

   log.Lik      AIC      BIC
 -1787.739 3617.478 3696.081

Variance Components:
                                        StdDev
(Intercept)                             1.0058
bs(year, 4, Boundary.knots = c(0, 15))1 0.4304
bs(year, 4, Boundary.knots = c(0, 15))2 2.0195
bs(year, 4, Boundary.knots = c(0, 15))3 1.7084
```

```
bs(year, 4, Boundary.knots = c(0, 15))4 2.6357
Residual                                0.2730

Coefficients:
Longitudinal Process
                                          Value Std.Err z-value p-value
(Intercept)                              0.5605  0.0546 10.2713 <0.0001
bs(year, 4, Boundary.knots = c(0, 15))1 -0.0491  0.0400 -1.2281  0.2194
bs(year, 4, Boundary.knots = c(0, 15))2  1.2939  0.1519  8.5164 <0.0001
bs(year, 4, Boundary.knots = c(0, 15))3  0.9211  0.2329  3.9546  0.0001
bs(year, 4, Boundary.knots = c(0, 15))4  2.2875  0.4637  4.9327 <0.0001
Event Process
                Value Std.Err  z-value p-value
drugD-penicil  0.0468  0.1794   0.2606  0.7944
ProNormal      0.0524  0.2485   0.2109  0.8330
Assoct         1.3194  0.1014  13.0155 <0.0001
log(xi.1)     -4.6360  0.3147 -14.7330
log(xi.2)     -4.5554  0.3321 -13.7163
log(xi.3)     -4.8170  0.3694 -13.0412
log(xi.4)     -4.7308  0.4151 -11.3959
log(xi.5)     -4.4028  0.3851 -11.4323
log(xi.6)     -4.0321  0.3981 -10.1277
log(xi.7)     -4.6786  0.5253  -8.9057

Integration:
method: (pseudo) adaptive Gauss-Hermite
quadrature points: 3

Optimization:
Convergence: 0
```

fixed parameters in the longitudinal model

hard to interpret, need to plot to see if the functions are smooth

After adjusting all other covariates in the model, xxx HR xxx confidence interval and the effect is not significant

- We then compute the conditional survival probabilities of patient 2. In the function survfitJM(), the argument newdata specifies the patient for whom the probabilities are computed.

- The function assumes that the patient has survived up to the last point t in newdata for which a serum bilirubin measurement was recorded and will produce survival probabilities for a set of predefined $u > t$ values.

will use all available biomarkers information

```
survPrbs <- survfitJM(JointFitBsp.pbc, newdata=pbc2[pbc2$id==2,])
```

only for patient 2

```
survPrbs

> survPrbs

Prediction of Conditional Probabilities of Event
based on 200 Monte Carlo samples

$`2`
     times   Mean Median  Lower  Upper
1   8.8325 1.0000 1.0000 1.0000 1.0000
1   8.9405 0.9840 0.9855 0.9665 0.9932
2   9.3609 0.9224 0.9287 0.8362 0.9683
3   9.7813 0.8617 0.8744 0.7012 0.9454
4  10.2017 0.8016 0.8241 0.5786 0.9243
5  10.6221 0.7620 0.7924 0.5088 0.9102
6  11.0425 0.7279 0.7623 0.4132 0.8965
7  11.4629 0.6923 0.7284 0.3192 0.8825
8  11.8833 0.6554 0.6975 0.1957 0.8697
9  12.3037 0.6176 0.6622 0.1112 0.8591
10 12.7241 0.5794 0.6266 0.0454 0.8534
11 13.1445 0.5410 0.5915 0.0103 0.8513
12 13.5649 0.5031 0.5529 0.0001 0.8444
13 13.9853 0.4663 0.5215 0.0000 0.8400
14 14.4057 0.4320 0.4850 0.0000 0.8298
```

mean value of the prediction
median value of the prediction
lower and upper bound of the 95% CI

last available measurement time in the dataset

- For each different subject in newdata and a series of time points u, we obtain the median and mean Monte Carlo estimates of $\pi_i(u|t)$ along with the 95% confidence intervals.

prob that u given t

- The first row corresponds to the last point we know the subject was still event free and thus the mean/median and CI are all 1.

- If we are only interested in point estimates, we can use the option simulate = F.

```
survPrbsEB <- survfitJM(JointFitBsp.pbc, newdata=pbc2[pbc2$id==2,], simulate=F)
survPrbsEB

Prediction of Conditional Probabilities for Events
```

```
$`2`
     times predSurv
1   8.8325   1.0000
1   8.9405   0.9855
2   9.3609   0.9296
3   9.7813   0.8748
4  10.2017   0.8210
5  10.6221   0.7867
6  11.0425   0.7582
7  11.4629   0.7291
8  11.8833   0.6992
9  12.3037   0.6681
10 12.7241   0.6357
11 13.1445   0.6015
12 13.5649   0.5653
13 13.9853   0.5266
14 14.4057   0.4850
```

same times point as previous one

- The two types of estimates are very similar.

- In some occasions, we also have information on the failure status of a patient at a specific time point after the last longitudinal measurement. For example,

```
pbc2[pbc2$id==2, c("id", "years", "status", "serBilir", "year")]
   id    years status serBilir      year
3   2 14.15234  alive      1.1 0.0000000
4   2 14.15234  alive      0.8 0.4983025
5   2 14.15234  alive      1.0 0.9993429
6   2 14.15234  alive      1.9 2.1027270
7   2 14.15234  alive      2.6 4.9008871
8   2 14.15234  alive      3.6 5.8892783
9   2 14.15234  alive      4.2 6.8858833
10  2 14.15234  alive      3.6 7.8907020
11  2 14.15234  alive      4.6 8.8325485
```

- We know that the patient was still in the study and alive up to year 14.2 years whereas the last serum bilirubin measurement was collected

at years 8.8 years.

- This information can be specified by the argument last.time in the function survfit(), which should be either a character string with the variable name in newdata or the last time point as a numeric vector.

```
survPrbs2 <- survfitJM(JointFitBsp.pbc, newdata=pbc2[pbc2$id==2,], last.time="years")
survPrbs2

Prediction of Conditional Probabilities of Event
based on 200 Monte Carlo samples

$`2`
    times  Mean Median  Lower  Upper
1 14.1523 1.000 1.0000 1.0000 1.0000
1 14.4057 0.948 0.9899 0.6315 0.9998
```

- Alternatively, to estimate $\pi_i(u|t)$ for specific us, this can be achieved by the argument survTimes. For example, the 14.5 and 15 years of Monte-Carlo estimates of $\pi_i(u|t)$ are obtained from

```
set.seed(123) # we set the seed for reproducibility

survfitJM(JointFitBsp.pbc, newdata=pbc2[pbc2$id==2,], survTimes = c(14.5, 15),
          last.time="years")

          Prediction of Conditional Probabilities of Event
based on 200 Monte Carlo samples

$`2`
    times   Mean Median  Lower  Upper
1 14.1523 1.0000 1.0000 1.0000 1.0000
1 14.5000 0.9140 0.9721 0.5004 0.9995
2 15.0000 0.8081 0.9339 0.0802 0.9991
```

- The estimates $\pi_i(u|t)$ can be graphically represented by the plot() function.

- The plot is based on the fact that the last time point the patient was still alive was at year 8.8, when he/she provided the last serum bilirubin measurement.

- Therefore, for all previous time points $u < t, \pi_i(u|t) = 1$.

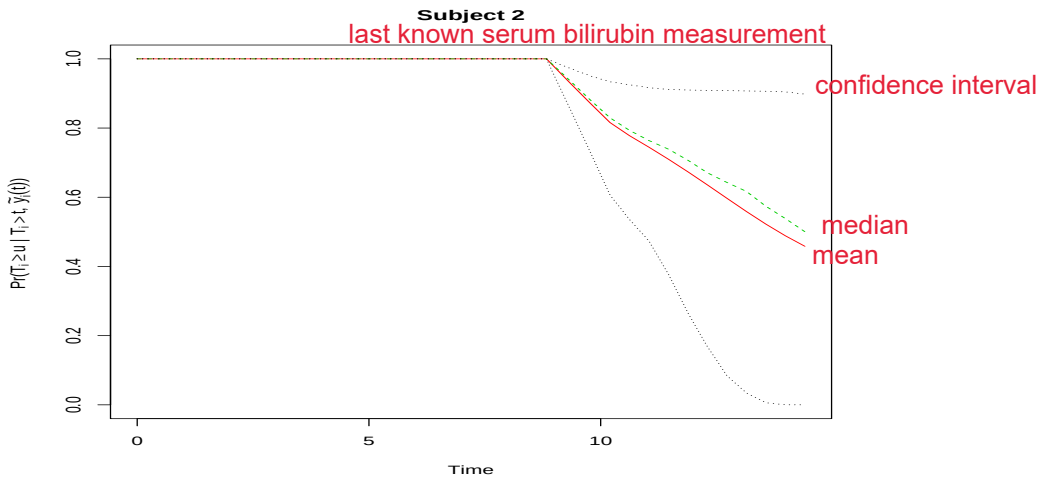


Figure 3: Conditional survival prediction for individual 2 from the PBC dataset. The dashed and solid lines correspond to the median and mean estimators.

- We can dynamically update $\pi_i(u|t)$ for patient 2 after each longitudinal measurement has been recorded. We can use this small R program for this

for each of the follow up time, fit the joint model and do dynamic prediction
updating the information and prediction for individual 2

```
ND<-pbc2[pbc2$id==2,]
survPreds <- vector("list", nrow(ND))
for(i in nrow(ND)){
  set.seed(123)
  survPreds[[i]]<-survfitJM(JointFitBsp.pbc, newdata=ND[1:i,])
}
```

At each time, recall a ....
updating the information using the new dataset
include the information of individual i

- The idea with the loop is that it provides the updated data frame that contains the extra measurements in the newdata argument of the surv-

fitJM() function. The results are saved in the list survPreds. The following plot shows the updated survival curves after the baseline, third, fifth and seventh measurements.

- The include.y argument provides the fitted longitudinal profile for patient 2 up to the time point of the last available bilirubin measurement and the estimated survival probability after this time point.

```
ND<-pbc2[pbc2$id==2,]
survPreds <- vector("list", nrow(ND))
for(i in 1:nrow(ND)){
  set.seed(123)
  survPreds[[i]]<-survfitJM(JointFitBsp.pbc, newdata=ND[1:i,])
}

pdf("Pred2_pbc.pdf")
par(mfrow=c(2,2), oma=c(0,2,0,2))
for(i in c(1,3,5,7)){
  plot(survPreds[[i]], estimator="median", conf.int=T,
       include.y=T, main=paste("Follow-up time:",
      round(survPreds[[i]]$last.time,1)))
}
mtext("log serum bilirubin", side=2, line=-1, outer=T)
mtext("Survival probability", side=4, line=-1, outer=T)
dev.off()
```

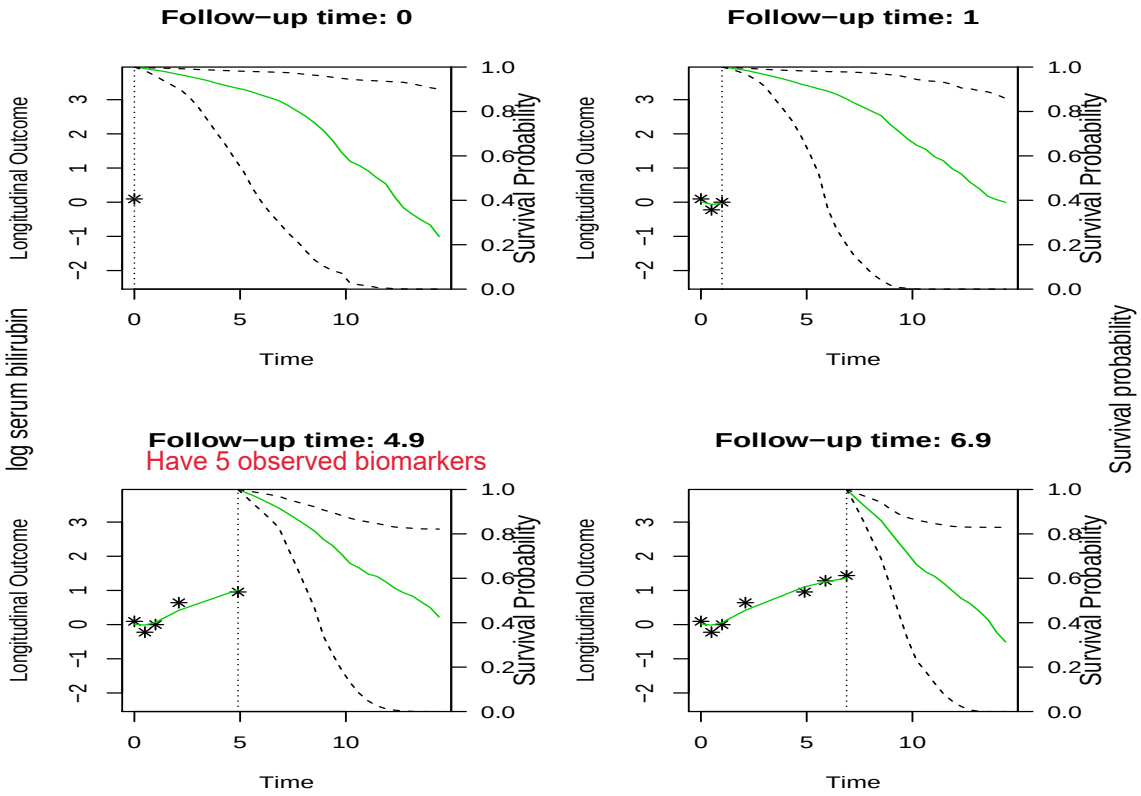
- We observe that after the 3rd measurement, where a clear increase in the bilirubin is observed, the rate decrease of the conditional survival function becomes steeper.

- We then made comparisons between the estimates of $\pi_i(u|t)$ between patients 2 and 25.

starting at 0, not plotting everything, but some discrete time.

At time 0, start to do the prediction. Only one measurement at time 0

3 observed biomarkers - updating the prediction



Have 5 observed biomarkers

have more biomarkers - prediction is more accurate

Figure 4: Dynamic survival probabilities for patient 2 from the PBC data. The vertical dotted lines represent the time point of the last serum bilirubin measurement. Left of this vertical line, the fitted longitudinal trajectory is depicted. Right of the vertical line, the solid line represents the median estimator for $\pi_i(u|t)$, and the dashed lines the corresponding 95% CI.

3 Dynamic predictions for the longitudinal outcome

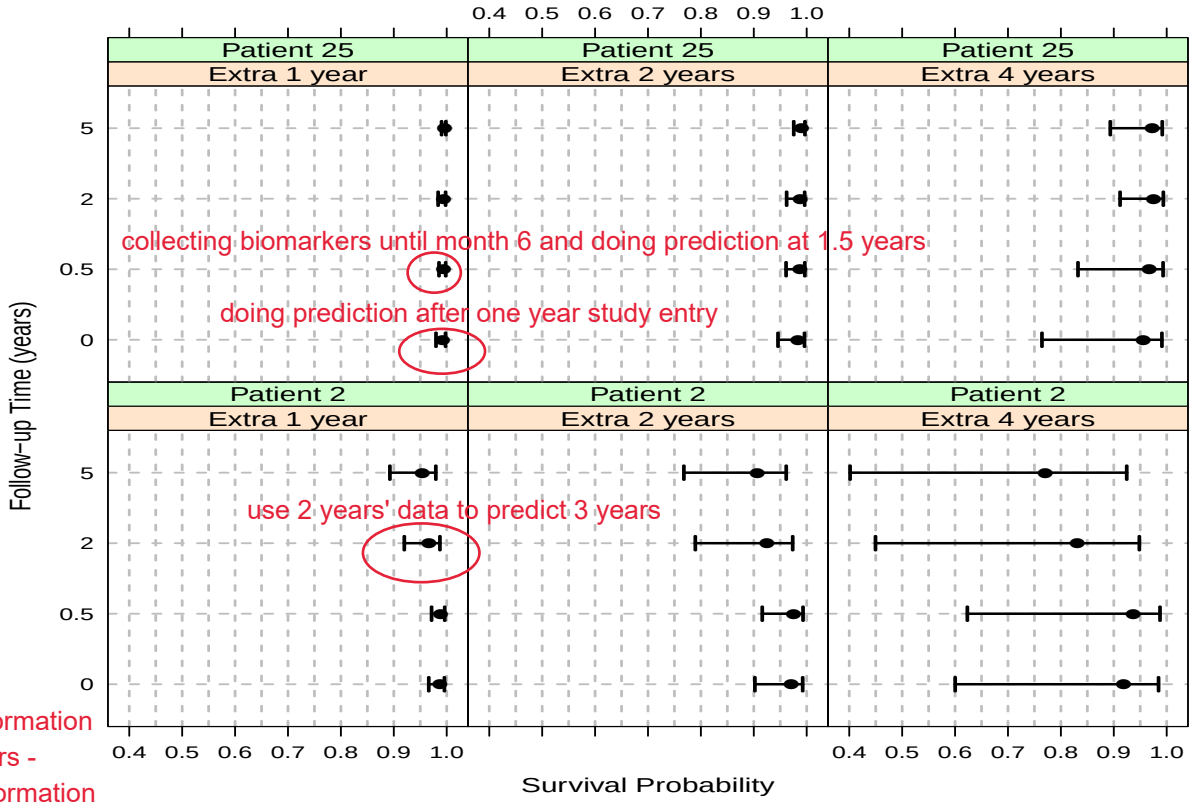
- In man occasions, the interest is to predict the longitudinal outcome, e.g., the CD4 cell count in the HIV data.

- The predicted longitudinal profile can help determine the best treatment for the patient at a given time point.

$\Rightarrow$ for a patient i who is still alive by follow-up time t, we are interested in the expected value of his/her longitudinal outcome at time $u > t$ given his/her observed responses up to that time point $\mathcal{Y}_i(t) = \{y_i(s), 0 \leq s <$

comparison between patient 2 and 25.
start at baseline (time 0).
Patient 2 has increased bilirubin level and 25 has stable level. Predictions don't change that much.

For patient 2, you see more variation.



updating the information on the biomarkers - use baseline information about biomarkers

Figure 5: A comparison of dynamic survival probabilities for patients 2 and 25 from the PBC data. In each panel, 4 estimates of $\pi_i(u|t)$ with 95% CI are presented for t set to the time point the most recent serum bilirubin measurements was collected. The top panels corresponds to patient 25, the bottom panels to patient 2, and in the left-hand side panels $u = t + 1$, in the middle panels $u = t + 2$, and in the right-hand panel side panels $u = t + 4$.

$t\}$, i.e.,

$$w_i(u|t) = E(y_i(u)|T_i^* > t, \mathcal{Y}_i(t), \mathcal{D}_n; \theta*\}, u > t.$$

- These predictions are also **dynamically** updated, e.g., using $w_i(u|t')$ that uses additional longitudinal information up to this latter time point.

- To compute these predictions, because we do no know the true param-

eter θ^*, a Bayesian formulation of the JM is applied to calculate the expectation of $w_i(u|t)$ with respect to the posterior distribution of the parameters $\{\theta|\mathcal{D}_n\}$ as

$$E\{y_i(u)|T_i^* > t, \mathcal{Y}_i(t), \mathcal{D}_n\} = \int E\{y_i(u)|T_i^* > t, \mathcal{Y}_i(t); \theta\} p(\theta|\mathcal{D}_n) d\theta.$$

The first part of the integrand can be obtained by

$$E\{y_i(u)|T_i^* > t, \mathcal{Y}_i(t); \theta\} = x_i^T(u)\beta + z_i^T(u)\bar{b}_i^{(t)},$$

where

$$\bar{b}_i^{(t)} = \int b_i p(b_i|T_i^* > t, \mathcal{Y}_i(t); \theta) db_i.$$

- A straightforward estimator of $w_i(u|t)$ is obtained by replacing θ with $\hat{\theta}$ and calculating the mean of the posterior distribution $p(b_i|T_i^* > t, \mathcal{Y}_i(t); \hat{\theta})$.

- Instead of the mean $\bar{b}_i^{(t)}$, we can use the mode of the posterior distribution

$$\hat{b}_i^{(t)} = \text{argmax}_b \log p(b|T_i^* > t, \mathcal{Y}_i(t); \hat{\theta}),$$

and get the estimate

$$\tilde{w}_i(u|t) = x_i^T(u)\hat{\beta} + z_i^T(u)\hat{b}_i^{(t)}.$$

- The 2 estimators are usually very close. Their standard error is however difficult to compute and can be obtained by Monte Carlo methods.

- A Monte Carlo estimate of $w_i(u|t)$ can be obtained using the following simulation scheme:

  - Step 1. Draw $\theta^{(l)} \sim \mathcal{N}(\hat{\theta}; \text{var}(\hat{\theta}))$
  - Step 2. Draw $b_j^{(l)} \sim \{b_j | T_j^* > t, \mathcal{Y}_j(t), \theta^{(l)}\}$
  - Step 3. Compute $w_i^{(l)}(u|t) = x_i^T(u)\beta^{(l)} + z_i^T(u)b_i^{(l)}$.

- Note that prediction intervals can be easily computed by replacing Step 3 with a draw from:

$$w_i^{(l)}(u|t) \sim \mathcal{N}(x_i^T(u)\beta^{(l)} + z_i^T(u)b_i^{(l)}, \sigma^{2^{(l)}}).$$

We can also estimate $w_i^{(l)}(u|t)$ by the Monte Carlo estimate

$$\hat{w}_i^{(l)}(u|t) = L^{-1}\sum_{l=1}^{L} w_i^{(l)}(u|t).$$

- The subject-specific predictions for the longitudinal outcome in R are obtained using the function predict().

- The argument newdata specifies the data on which to base the predictions. It should contain the baseline covariates and the longitudinal responses up to time t, which will be used to estimate $w_i(u|t), u > t$

- By default, population-based predictions are provided. For subject-specific predictions, we have to use the argument type="Subject".

- We illustrate the use of this function to produce estimate of $w_i^{(l)}(u|t)$ for patient 2 of the PBC study based on our previous fitted model.

- The predict() function returns the estimate $\tilde{w}_i^{(l)}(u|t)$ and by using the argument interval = "confidence", we obtain Monte Carlo scheme to compute the standard errors and CIs.

- Below, the for-loop is used to update patient 2's bilirubin measurements in newdata.

```
# dynamic prediction for serum bilirubin
ND <- pbc2[pbc2$id == 2, ]
longPreds <- vector("list", nrow(ND))
for (i in 1:nrow(ND)) {
  set.seed(123) # we set the seed for reproducibility
  longPreds[[i]] <- predict(JointFitBsp.pbc, newdata = ND[1:i, ],
                            type = "Subject", interval = "confidence", returnData = TRUE)
  longPreds[[i]]$FollowUp <- round(max(ND[1:i, "year"]), 1)
}
```

looping over something

subject-specific prediction

the follow up time is rounded up by 1 decimal place

- The default output of predict() is a vector of predicted values for each of the individuals included in newdata.

- The time points u at which the predictions are calculated are by default chosen at regular sequence of length.

- However, by using the argument returnData=T, the output is the data frame supplied in newdata augmented with extra lines for each subject with his/her predictions at time points u.

- The last line of the for-loop adds an extra column for the last follow-up time t at which a longitudinal response has been recorded.

- To visualize the resulting estimates, we first collect the results in a single data frame

```
# put all results in the same data frame
longPreds.all <- do.call(rbind, longPreds)
longPreds.all$FollowUp <- with(longPreds.all, factor(FollowUp,
        labels = paste("Follow-up time:", unique(FollowUp))))
```

adding follow up time to dataset

- The 2nd line transforms the column FollowUp to a factor with the corresponding levels. The dynamic predictions are then obtained with

```
# plot of dynamic predictions
xyplot(pred + low + upp ~ year | FollowUp, data = longPreds.all,
       panel = function (x, y) {
         xx <- x[seq_len(length(x) / 3)]
         yy <- matrix(y, ncol = 3)
         last.time <- max(x[is.na(y)])
         ind <- xx >= last.time
         ind[which(ind)[1]] <- FALSE
         lpolygon(c(xx[ind], rev(xx[ind])),
                  c(yy[ind, 2], rev(yy[ind, 3])), border = "transparent", col = "grey")
         panel.xyplot(xx[ind], yy[ind, 1], lty = 2, lwd = 2,
                      type = "l", col = 1)
         panel.xyplot(xx[!ind], yy[!ind, 1], lty = 1, lwd = 2,
                      type = "l", col = 1)
         panel.abline(v = last.time, lty = 3, lwd = 2)
```

```
}, as.table = TRUE, xlab = "Time",
ylab = "Predicted log serum bilirubin", layout = c(3,3))
```

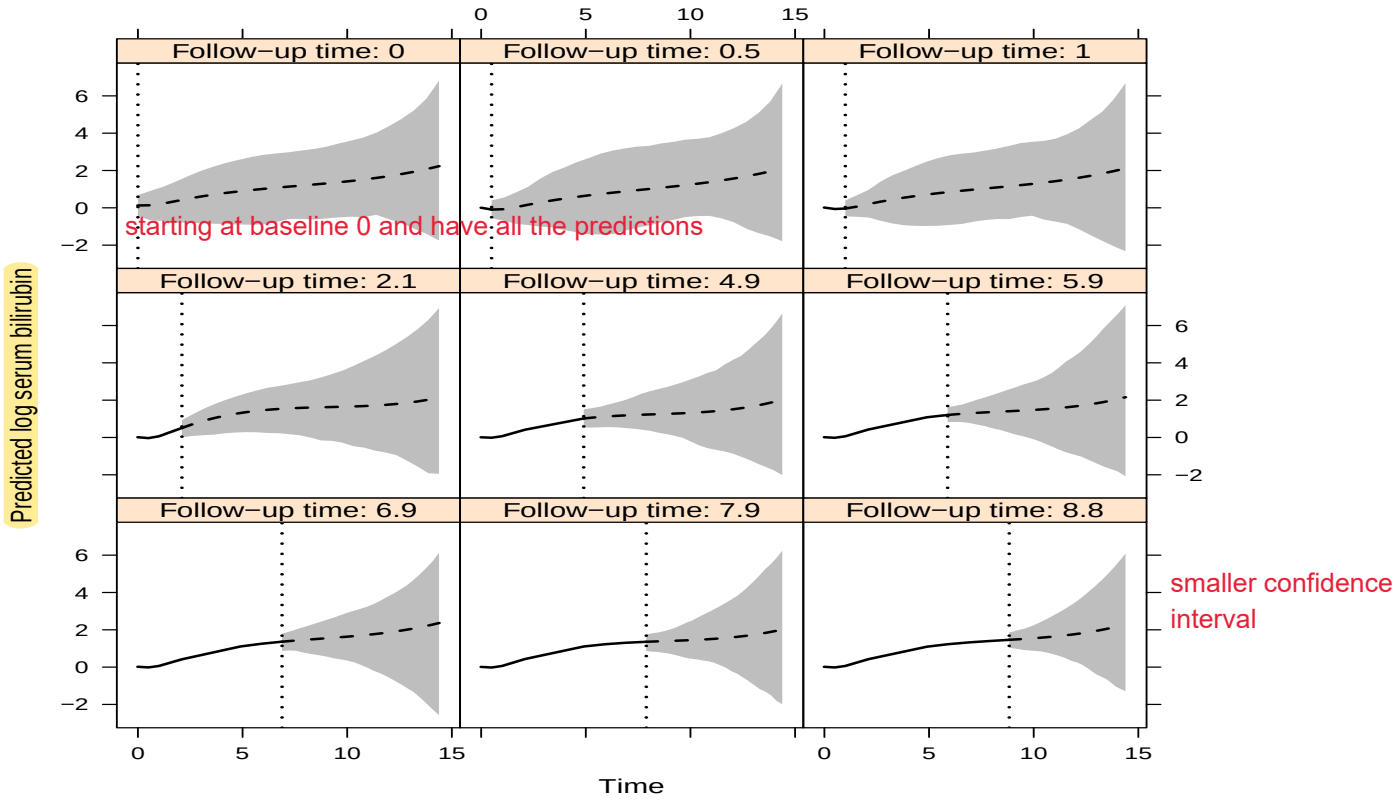


Figure 6: Dynamic predictions of the bilirubin longitudinal responses for patient 2 from the PBC dataset. In each panel, the dotted vertical line denotes the time point of the last observed longitudinal response. The solid line left to the dotted line denotes the fitted longitudinal trajectory prior to the last visit, and the dashed line right to the dotted line, denotes the predicted longitudinal trajectory. The grey areas denote the 95% pointwise conditional intervals.

- We observe that the width of the prediction intervals increases as time progresses, indicating we have more confidence on predictions shortly after the last available longitudinal measurement.

4 Effect of the biomarker parametrization on predictions

- All previous predictions were based on the standard joint model

$$\begin{cases} y_i(t) = m_i(t) + \epsilon_i(t), \\ \qquad\quad = x_i^T(t)\beta + z_i^T(t)b_i + \epsilon_i(t), \quad \epsilon_i(t) \sim \mathcal{N}(0, \sigma^2), \\ h_i(t|\mathcal{M}_i(t)) = h_0(t)\exp\{\gamma^T w_i + \alpha m_i(t)\}, \end{cases}$$

  where

  $\mathcal{M}_i(t) = \{m_i(s), 0 \leq s < t\}$ is the longitudinal history.

- The quality of the predictions typically depends on 2 factors: a) the ability of the biomarker to predict future events and b) the correct formulation of the JM.

- As we have seen in previous lectures, the various patametrizations of the biomarker can have considerable effect on the shapes of the estimated subject-specific hazard functions.

- Relevant questions:

  - Does the assumed parameterization affect predictions?
  - Which parameterization is the most optimal?

- Sensitivity analyses can be performed by changing the parametrizations of the biomarker.

- Example: We compare predictions for the longitudinal and survival outcomes under different parameterizations for Patient 51 from the PBC

study.

- Patient 51's longitudinal profile:

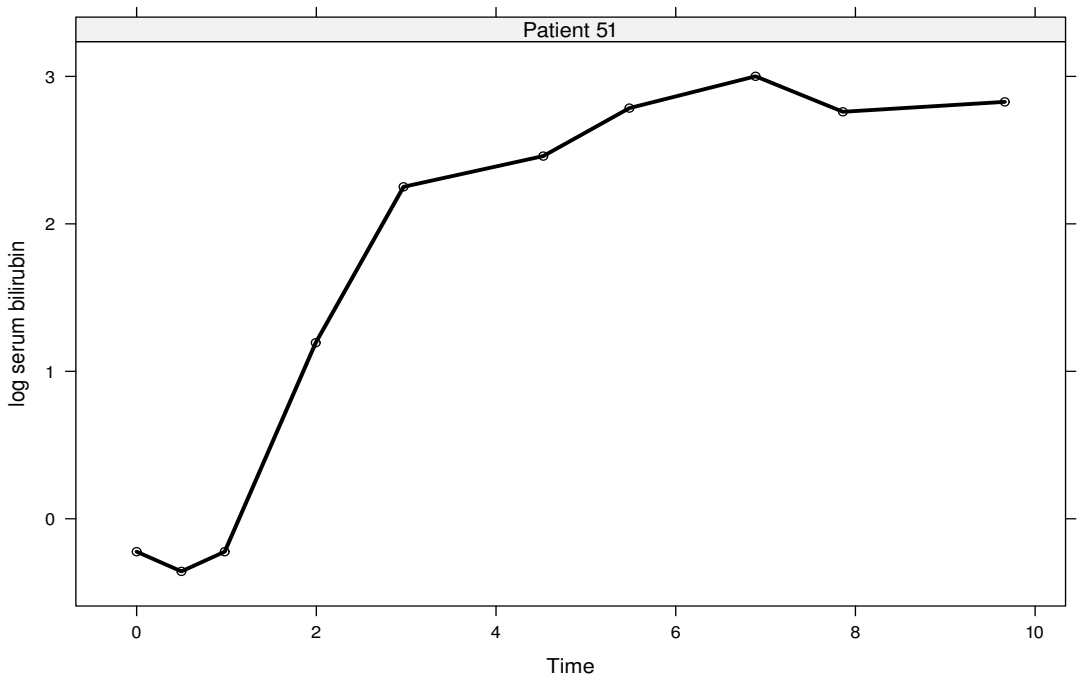


Figure 7: Observed longitudinal trajectory of patient 51 from the PBC dataset.

- Our sensitivity analyses will compare 6 models

(I) $h_i(t) = h_0(t) \exp\{\gamma_1 \text{D-pnc}_i + \gamma_2 \text{ProtTime}_i + \alpha_1 m_i(t)\}$,

(II) $h_i(t) = h_0(t) \exp\{\gamma_1 \text{D-pnc}_i + \gamma_2 \text{ProtTime}_i + \alpha_2 m_i'(t)\}$,

(III) $h_i(t) = h_0(t) \exp\{\gamma_1 \text{D-pnc}_i + \gamma_2 \text{ProtTime}_i + \alpha_1 m_i(t) + \alpha_2 m_i'(t)\}$,

$$\text{(IV)}\quad h_i(t) = h_0(t)\exp\{\gamma_1 \text{D-pnc}_i + \gamma_2 \text{ProtTime}_i + \alpha_1 m_i(t) + \alpha_1^{int}\{\text{ProtTime}_i \times m_i(t)\},$$

interaction

$$\text{(V)}\quad h_i(t) = h_0(t)\exp\{\gamma_1 \text{D-pnc}_i + \gamma_2 \text{ProtTime}_i + \alpha_2 m_i'(t) + \alpha_2^{int}\{\text{ProtTime}_i \times m_i'(t)\},$$

main effects

$$\text{(VI)}\quad h_i(t) = h_0(t)\exp\{\gamma_1 \text{D-pnc}_i + \gamma_2 \text{ProtTime}_i + \alpha_1 m_i(t) + \alpha_2 m_i'(t) + \alpha_1^{int}\{\text{ProtTime}_i \times m_i(t)\} + \alpha_2^{int}\{\text{ProtTime}_i \times m_i'(t)\},$$

- and the same longitudinal submodel as we used before:

$$\begin{aligned} y_i(t) &= m_i(t) + \epsilon_i(t) \\ &= (\beta_0 + b_{i0}) + (\beta_k + b_{ik})^T B(t, 4, 4) + \epsilon_i(t). \end{aligned}$$

- To use these models, we need to derive the expression for the derivative $m_i'(t)$ under the B-splines representation for $m_i(t)$. We can write $m_i(t)$ in the form

for longitudinal - have no covaraites but splines function

$$m_i(t) = (\beta_0 + b_{i0}) + \sum_k (\beta_k + b_{ik}) B_k(t, q),$$

where $B_k(t, q)$ denotes the B-spline basis of order q at knot λ_k. Its derivative is given by the expression

$$m_i'(t) = (q-1)\sum_k \frac{c_{i,k+1} - c_{i,k}}{\lambda_{k+q+1} - \lambda_k} B_k(t, q-1),$$

where $c_{i,k} = \beta_k + b_{ik}$.

- To specify the slope of the true trajectory using the formula argument in R, we used the mixed-effects model formulation:

$$m_i'(t) = [x_i^{sl}(t)]^T \beta^{sl} + [z_i^{sl}(t)]^T b_i^{sl}$$

$$= (q-1)B(t, \lambda, q-1)^T(\beta_{k,-1}/D\lambda) - qB(t, \lambda, q-1)^T(\beta_{k,-c}/D\lambda)$$

$$+qB(t, \lambda, q-1)^T(b_{ik,-1}/D\lambda) - qB(t, \lambda, q-1)^T(b_{ik,-c}/D\lambda),$$

where λ denotes the knots of the B-spline basis, $D\lambda$ the $q+1$ order differences of λ, and $\beta_{k,-l}$ and $\beta_{ik,-l}$ denote the vectors of fixed and random effects of β_k and b_{ik}, respectively, excluding their l element, with c denoting the last element.

- Since there is only one internal node, $D\lambda_k = \lambda_{k+q+1} - \lambda_k$.

- The code for the longitudinal and survival models are similar to the ones used previously

```
lmeFitBsp.pbc <- lme(fixed=log(serBilir)~bs(year, 4, Boundary.knots = c(0,15)),
                     random=list(id=pdDiag(form=~bs(year, 4, Boundary.knots = c(0,15)))),
                     data=pbc2)

coxFit.pbc <- coxph(Surv(years, status2)~drug + Pro, data=pbc2.id, x=TRUE)
```

- The term $m_i'(t)$ is specified through the derivForm argument of the JointModel() function.

```
# list to calculate the derivative of the B-spline
dform <- list(
    fixed = ~ -1
    + I(3 * bs(year, knots = 2.0534443,
        Boundary.knots = c(0, 15), degree = 2) / 15)
    + I(-3 * bs(year, knots = 2.0534443,
        Boundary.knots = c(0, 15), degree = 2) / 15),

    indFixed = c(3,4,5,2,3,4),

    random = ~ -1
    + I(3 * bs(year, knots = 2.0534443,
        Boundary.knots = c(0, 15), degree = 2) / 15)
    + I(-3 * bs(year, knots = 2.0534443,
        Boundary.knots = c(0, 15), degree = 2) / 15),

    indRandom = c(3,4,5,2,3,4))
```

- The B-splines basis in $m_i'(t)$ should be calculated at the exactly same knots as the B-spline basis for $m_i(t)$. To do this, we need to extract the value of the internal knot with the code

```
> attr(lmeFitBsp.pbc$terms, "predvars")
list(log(serBilir), bs(year, degree = 3L, knots = c(`50%` = 2.0534443105903),
    Boundary.knots = c(0, 15), intercept = FALSE))
```

and then we manually set the internal and boundary knots in the call to bs() using the knots and Boundary.knots arguments. The degree argument specifies the degree of the B-splines (default=3 $\Rightarrow$ the order $q = 4$).

The codes for the 6 models are

```
# model (I)
jointFitBsp.pbc <- jointModel(lmeFitBsp.pbc, coxFit.pbc,
    timeVar = "year", method = "piecewise-PH-aGH")

# model (II)
jointFitBsp2.pbc <- update(jointFitBsp.pbc,
```

```
    parameterization = "slope", derivForm = dform)

# model (III)
jointFitBsp3.pbc <- update(jointFitBsp.pbc,
    parameterization = "both", derivForm = dform)

# model (IV)
jointFitBsp4.pbc <- update(jointFitBsp.pbc,
    interFact = list(value = ~ Pro, data = pbc2.id))

# model (V)
jointFitBsp5.pbc <- update(jointFitBsp2.pbc,
    interFact = list(slope = ~ Pro, data = pbc2.id))

# model (VI)
jointFitBsp6.pbc <- update(jointFitBsp3.pbc,
    interFact = list(value = ~ Pro, slope = ~ Pro, data = pbc2.id))
```

- The parameter estimates and their SEs are represented in the following table 7.1 from Rizopulos' book.

TABLE 7.1: Parameter estimates and standard errors for the regression coefficients in relative risks models (I)–(VI) based on the corresponding joint models fitted to the PBC dataset

	Coefficient	Value	Std. Err.	z-value	p-value
`jointFitBsp.pbc`	γ_1	−0.02	0.16	−0.09	0.926
	γ_2	−0.08	0.22	−0.36	0.722
	α_1	1.30	0.09	14.15	< 0.001
`jointFitBsp2.pbc`	γ_1	−0.07	0.16	−0.45	0.649
	γ_2	0.31	0.21	1.50	0.135
	α_2	3.40	0.41	8.23	< 0.001
`jointFitBsp3.pbc`	γ_1	−0.02	0.17	−0.13	0.894
	γ_2	−0.05	0.22	−0.23	0.821
	α_1	1.29	0.10	13.51	< 0.001
	α_2	1.00	0.43	2.33	0.020
`jointFitBsp4.pbc`	γ_1	−0.00	0.17	−0.01	0.991
	γ_2	1.19	0.67	1.78	0.076
	α_1	1.89	0.31	6.08	< 0.001
	α_1^{int}	−0.69	0.33	−2.10	0.035
`jointFitBsp5.pbc`	γ_1	−0.07	0.16	−0.46	0.646
	γ_2	0.50	0.33	1.49	0.136
	α_2	3.86	0.75	5.17	< 0.001
	α_2^{int}	−0.59	0.82	−0.71	0.476
`jointFitBsp6.pbc`	γ_1	−0.01	0.17	−0.04	0.972
	γ_2	1.06	0.71	1.49	0.135
	α_1	1.89	0.32	5.88	< 0.001
	α_1^{int}	−0.68	0.34	−2.02	0.044
	α_2	0.41	0.89	0.47	0.642
	α_2^{int}	0.59	0.99	0.60	0.550

- The predictions for individual 51 can then be obtained by

```
# Data of Patient 51
ND2 <- pbc2[pbc2$id %in% 51, ]

# Dynamic predictions for the longitudinal outcome under the six joint models
longPreds1 <- longPreds2 <- longPreds3 <- vector("list", nrow(ND2))
longPreds4 <- longPreds5 <- longPreds6 <- vector("list", nrow(ND2))
for (i in 1:nrow(ND2)) {
    set.seed(123)
    nd2 <- ND2[1:i, ]
    time <- max(nd2$year) + 1
    longPreds1[[i]] <- predict(jointFitBsp.pbc, nd2, type = "Subject", FtTimes = time,
        interval = "confidence", M = 500)
    longPreds2[[i]] <- predict(jointFitBsp2.pbc, nd2, type = "Subject", FtTimes = time,
        interval = "confidence", M = 500)
    longPreds3[[i]] <- predict(jointFitBsp3.pbc, nd2, type = "Subject", FtTimes = time,
        interval = "confidence", M = 500)
    longPreds4[[i]] <- predict(jointFitBsp4.pbc, nd2, type = "Subject", FtTimes = time,
        interval = "confidence", M = 500)
    longPreds5[[i]] <- predict(jointFitBsp5.pbc, nd2, type = "Subject", FtTimes = time,
        interval = "confidence", M = 500)
    longPreds6[[i]] <- predict(jointFitBsp6.pbc, nd2, type = "Subject", FtTimes = time,
        interval = "confidence", M = 500)
}

# Dynamic predictions for the survival outcome under the six joint models
survPreds1 <- survPreds2 <- survPreds3 <- vector("list", nrow(ND2))
survPreds4 <- survPreds5 <- survPreds6 <- vector("list", nrow(ND2))
for (i in 1:nrow(ND2)) {
    set.seed(123)
    nd2 <- ND2[1:i, ]
    time <- max(nd2$year) + 1
    survPreds1[[i]] <- survfitJM(jointFitBsp.pbc, nd2, survTimes = time, M = 500)
    survPreds2[[i]] <- survfitJM(jointFitBsp2.pbc, nd2, survTimes = time, M = 500)
    survPreds3[[i]] <- survfitJM(jointFitBsp3.pbc, nd2, survTimes = time, M = 500)
    survPreds4[[i]] <- survfitJM(jointFitBsp4.pbc, nd2, survTimes = time, M = 500)
    survPreds5[[i]] <- survfitJM(jointFitBsp5.pbc, nd2, survTimes = time, M = 500)
    survPreds6[[i]] <- survfitJM(jointFitBsp6.pbc, nd2, survTimes = time, M = 500)
}
```

- The dynamic predictions of survival probabilities are given in Table 7.8 from Rizopoulos' book

- The chosen parameterization can influence the derived predictions, especially for the survival outcome

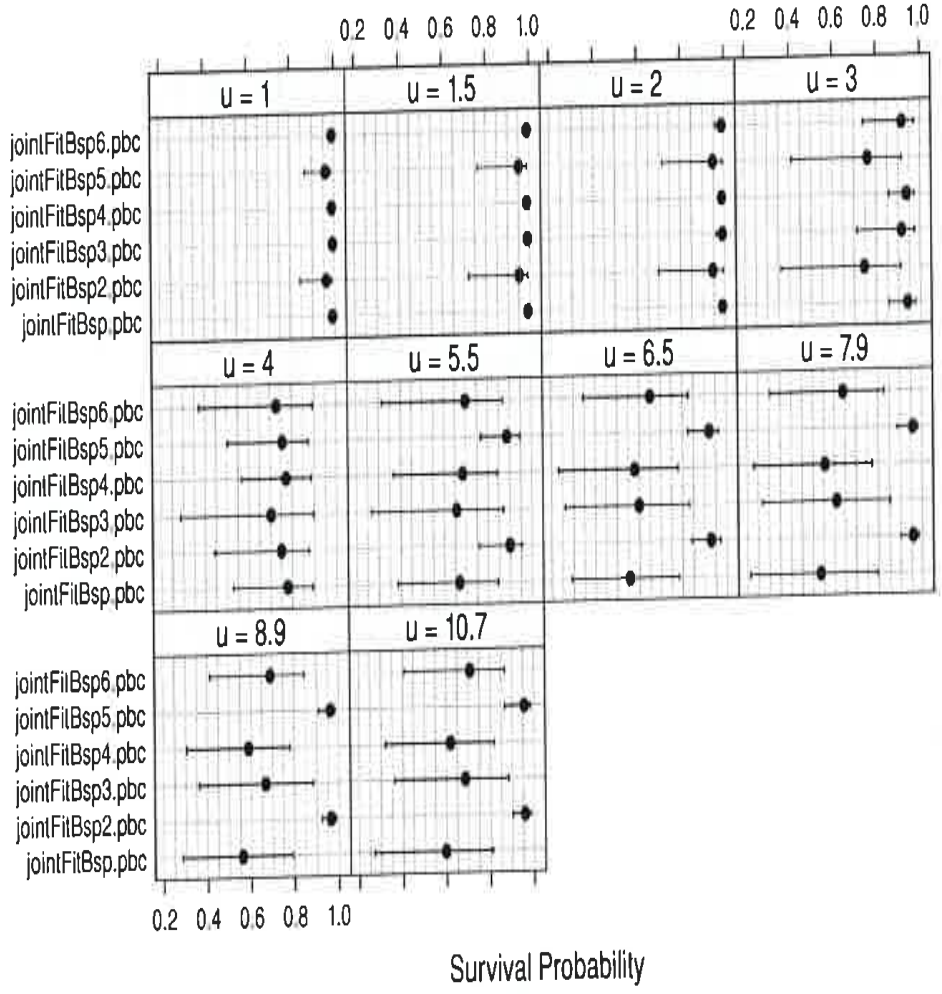


FIGURE 7.8: Dynamic predictions of survival probabilities for Patient 51 from the PBC dataset. Each panel depicts point estimates and the associated 95% confidence intervals of the probability of surviving one additional year after the last longitudinal response has been recorded, based on the six joint models.

5 Accuracy measures for JMs

- Often we are interested in determining the accuracy of the longitudinal biomarker to predict the survival outcome.

- Two measures are often used:

  - Calibration : how well the model predicts the observed data.
  - Discrimination : how well the model can discriminate between patients who will have the event in the short time frame and patients who will have it later in time.

  $\Rightarrow$ We will focus more on the discrimination measures here.

- General setting: For any t, we are interested in events in the medically relevant time interval $(t, t + \Delta t]$ (Heargety & Zheng, Bcs, 2005; Zheng & Heargety, Bcs, 2007).

- In particular, two generic patients, i and j have survived up to time t and have biomarker histories:

  $\mathcal{Y}_i(t) = \{y_i(s), 0 \leq s \leq t\}$ and $\mathcal{Y}_j(t) = \{y_j(s), 0 \leq s \leq t\}$.

- Patient i died before $t + \Delta t$, patient j lived longer.

- Goal: Use the marker to discriminate between these two patients and take an appropriate medical action.

- General prediction rule: We define "success" as

$$\mathcal{S}_i(t, k, c) = \{y_i(s) \geq c_s; k \leq s \leq t\},$$

where

- $y_i(s)$ is the value of the marker at time s,
- c vector of threshold values,
- k specifies which past values of the marker we are using.

- Note: In case lower $y_i(t)$ values are more predictive for an event, then

$$\mathcal{S}_i(t, k, c) = \{y_i(s) \leq c_s; k \leq s \leq t\},$$

- Simple/Standard Prediction Rule

$$\mathcal{S}_i(t, k = t, c) = \{y_i(t) \geq c\},$$

$\Rightarrow$ we use the most recent measurement to guide decision making.

- Composite Prediction Rule

$$\mathcal{S}_i(t, k = t - 1, c) = \{y_i(t - 1) \geq c\} \cup \{y_i(t) \geq c\},$$

$\Rightarrow$ we use the last two measurements, or,

$$\mathcal{S}_i(t, k = t - 1, c) = \{y_i(t - 1) \geq c\} \cup \{y_i(t) \geq (1 + v)c\},$$

$\Rightarrow$ the same threshold or $(100v)\%$ increase from the pre-last to the last one.

- Sensitivity

$$\mathrm{TP}_t^{\Delta t} = P\{\mathcal{S}_i(t,k,c)|T_i^* > t, T_i^* \in (t, t+\Delta t]\}.$$

- Specificity

$$1 - \mathrm{FP}_t^{\Delta t} = P\{\mathcal{F}_i(t,k,c)|T_i^* > t, T_i^* > t+\Delta t\},$$

where

T_i^* is the true failure time

$\mathcal{F}_i(t,k,c) = \mathbb{R}^{r(k,t)} \setminus \{y_i(s) \leq c_s; k \leq s \leq t\}$, and $r(k,t)$ is the number of longitudinal measurements in the interval $[k,t]$.

- Note that these measures are time-dependent and they also depend on the length of the medically relevant time interval Δt.

$\Rightarrow$ It means that for the same t, different models may exhibit different discrimination power for different Δt.

- The overall discrimination of the longitudinal biomarker for all threshold values $c \in \mathbb{R}_y$ can be assessed by the ROC curve and the AUC at specific follow-up times.

- Time-dependent ROC

$$\text{ROC}_t^{\Delta t}(p) = \text{TP}_t^{\Delta t}\{[FP_t^{\Delta t}]^{-1}(p)\}$$

- AUC

$$\text{AUC}_t^{\Delta t} = \int_0^1 \text{ROC}_t^{\Delta t}(p)dp$$

- For the estimation of the accuracy measures we need to account for censoring.

- We take full advantage of the JM framework. JM is estimated using maximum likelihood.

- Based on the fitted joint model we can estimate the sensitivity and specificity using a Monte Carlo simulation scheme similar to the one used for $\pi_i(u|t)$. The technical details are explained in Rizopoulos' book and Rizopoulos (2011, Biometrics).

- The time-dependent measures of sensitivity, specificity and the corresponding ROC curve and the AUC for the biomarker can be calculated with the function rocJM() of the *R* package JM.

- Its main argument is the fitted JM and the data frame that contains the baseline covariates as well as the time points at which the longitudinal

measurements were supposed to be taken. The argument dt specifies the length of Δt of the medically relevant time interval(s).

- Example: We calculate time-dependent ROCs for the PBC data for individual 2.

```
pbc2.id2<-pbc2[pbc2$id==2, ]
pbc2.id2[,c("id","drug","year","Pro")]

   id      drug      year    Pro
3   2 D-penicil 0.0000000 Normal
4   2 D-penicil 0.4983025 Normal
5   2 D-penicil 0.9993429 Normal
6   2 D-penicil 2.1027270 Normal
7   2 D-penicil 4.9008871 Normal
8   2 D-penicil 5.8892783 Normal
9   2 D-penicil 6.8858833 Normal
10  2 D-penicil 7.8907020 Normal
11  2 D-penicil 8.8325485 Normal

roc <- rocJM(jointFitBsp.pbc, data = pbc2.id2, dt = c(1, 2))
roc

Areas under the time-dependent ROC curves

Estimation: Monte Carlo (200 samples)
Difference: absolute, lag = 1 (0)
Thresholds range: (-1.61, 6.47)

Case: 2
Recorded time(s): 0, 0.5, 1, 2.1, 4.9, 5.89, 6.89, 7.89, 8.83
 dt t + dt    AUC    Cut
  1   9.83 0.7376 1.2331
  2  10.83 0.7649 1.2331

plot(roc, legend=T)
```

- We notice that the higher Δt, the better the discrimination.

- The column cut indicates the value of the biomarker that maximizes the product of sensitivity and specificity.

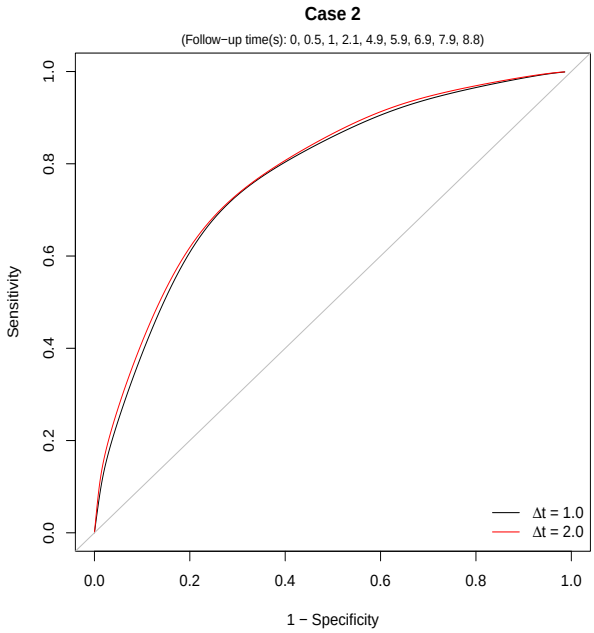


Figure 8: ROC curves at time $t = 8.83$ years for patient 2 and two options for Δt under the simple prediction rule (based on the fitted JM on the PBC data). The black curve corresponds to $\Delta t = 1$ and the red curve to $\Delta t = 2$.

- To examine how the predictive performance of the biomarker evolves during follow-up, we can produce the ROC curves at different time points t. This can be achieved by updating the data argument of the rocJM() function using a for-loop.

```
# time-dependent ROCs
ROCs <- vector("list", 4)
for (i in seq_along(ROCs)) {
  print(i)
  set.seed(123) # we set the seed for reproducibility
  ROCs[[i]] <- rocJM(jointFitBsp.pbc, dt = c(1, 2, 4),
                     data = pbc2.id2[seq_len(2*i-1), ])
}

# plot of time-dependent ROCs
pdf("roc3.pdf")
par(mfrow = c(2, 2), oma = c(0, 0, 2, 0))
for (i in 1:4) {
```

```
  plot(ROCs[[i]], legend = TRUE)
}
mtext("Prediction rule: Simple", side = 3, line = -1, outer = TRUE)
dev.off()
```

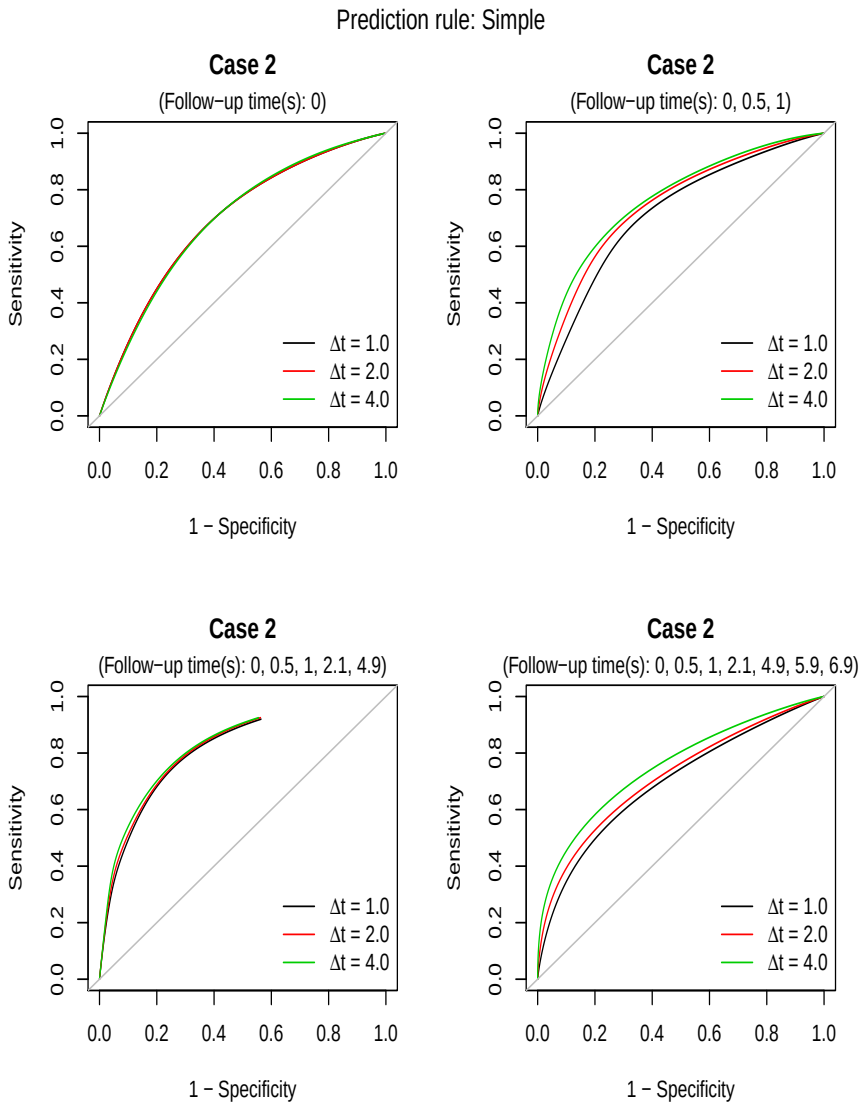


Figure 9: ROC curves for patient 2 and 3 options for Δt under the simple prediction rule (based on the fitted JM on the PBC data).

- In general, the biomarker cannot separate patients who will die before $t + \Delta t$ from those who will not.

- Instead of a real individual, some fictive individuals can also be created. In the following, we create 4 individuals having a follow-up of 5 time points, each of them having a different combination of the Drug × Pro variables.

```
NewData <- expand.grid(
  year = c(0, 0.5, 2, 3, 5),
  drug = c("placebo", "D-penicil"),
  Pro = c("Abnormal", "Normal")
  )
NewData$id <- rep(1:4, each = 5)
```

```
> NewData
   year      drug      Pro id
1   0.0   placebo Abnormal  1
2   0.5   placebo Abnormal  1
3   2.0   placebo Abnormal  1
4   3.0   placebo Abnormal  1
5   5.0   placebo Abnormal  1
6   0.0 D-penicil Abnormal  2
7   0.5 D-penicil Abnormal  2
8   2.0 D-penicil Abnormal  2
9   3.0 D-penicil Abnormal  2
10  5.0 D-penicil Abnormal  2
11  0.0   placebo   Normal  3
12  0.5   placebo   Normal  3
13  2.0   placebo   Normal  3
14  3.0   placebo   Normal  3
15  5.0   placebo   Normal  3
16  0.0 D-penicil   Normal  4
17  0.5 D-penicil   Normal  4
18  2.0 D-penicil   Normal  4
19  3.0 D-penicil   Normal  4
20  5.0 D-penicil   Normal  4
```

```
roc2 <- rocJM(jointFitBsp.pbc, data = NewData, dt = c(1, 2, 4))
```

```
roc2
```

```
Case: 1
Recorded time(s): 0, 0.5, 2, 3, 5
 dt t + dt    AUC    Cut
  1      6 0.6751 1.6207
  2      7 0.6913 1.6207
```

```
  4      9 0.7124 1.5238

Case: 2
Recorded time(s): 0, 0.5, 2, 3, 5
 dt t + dt    AUC     Cut
  1      6 0.5412 -0.7696
  2      7 0.5640 -0.7050
  4      9 0.5808 -0.6727

Case: 3
Recorded time(s): 0, 0.5, 2, 3, 5
 dt t + dt    AUC    Cut
  1      6 0.6844 1.7822
  2      7 0.6951 1.7176
  4      9 0.7074 1.5884

Case: 4
Recorded time(s): 0, 0.5, 2, 3, 5
 dt t + dt    AUC    Cut
  1      6 0.6975 1.8791
  2      7 0.7170 1.8468
  4      9 0.7436 1.7176

pdf("roc2.pdf")
par(mfrow=c(2,2), oma=c(0,0,2,0))
plot(roc2)
dev.off()
```

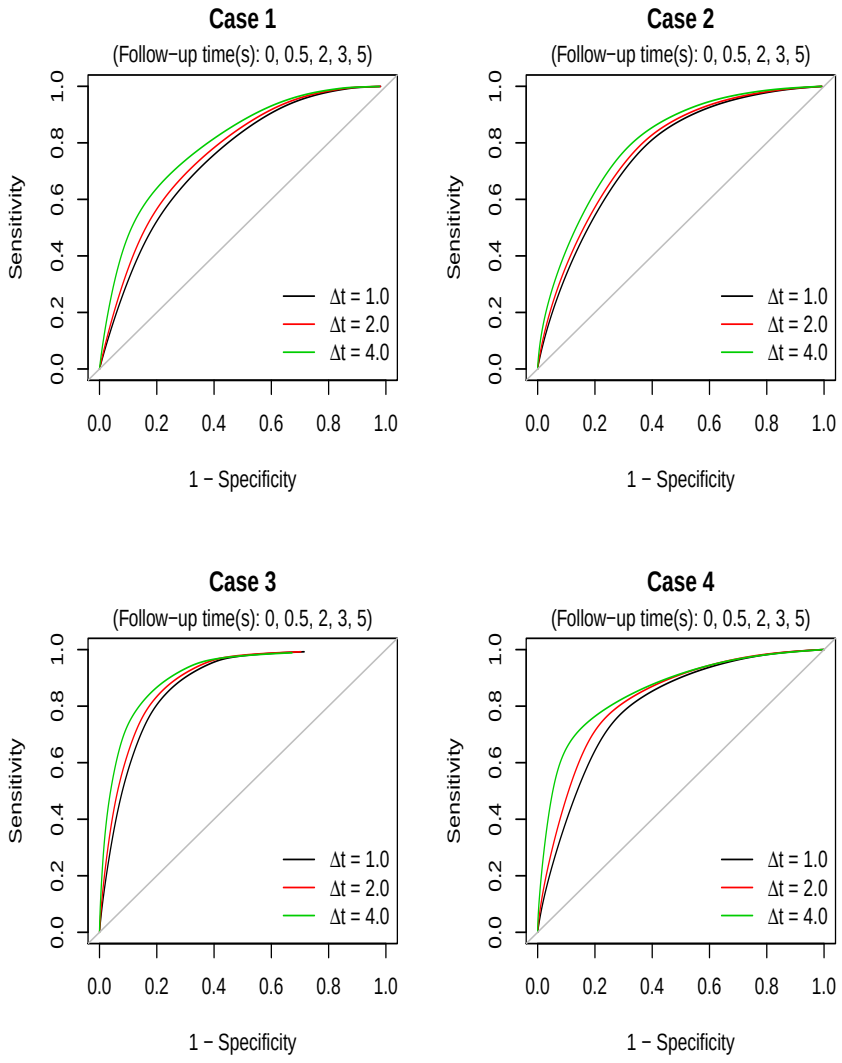


Figure 10: ROC curves at time $t = 5$ years for 4 fictive individuals and 3 options for Δt under the simple prediction rule (based on the fitted JM on the PBC data). Case 1 corresponds to placebo with abnormal prothrombin; Case 2 treatment group with abnormal prothrombin; Case 3 corresponds to placebo with normal prothrombin; Case 4 corresponds to treatment group with normal prothrombin.

- We now investigate whether we can improve the discrimination by considering a composite prediction rule with the following form:

$$\mathcal{S}_i(t, k, c) = \{y_i(t - k) \leq c\} \cap \{y_i(t) \leq 0.8c\},$$

which postulates that there is a higher chance for the patient to experience the event within the time interval $(t, t + \Delta t[$ when he/she shows a 20% decrease in the biomarker level between two subsequent visits, where $t - k$ denotes the time point of the next to last visit.

These ROC curves are produced the argument diffType and rel.diff of rocJM().

```
# ROCs based on the composite rule assuming a 20% decrease
# in biomarker
set.seed(123)

ROCplcb.Rel <- rocJM(jointFitBsp.pbc, data = pbc2.id2, dt = c(1, 2,4),
    diffType = "relative", rel.diff = c(1, 0.8))

# estimated AUCs and optimal thresholds under the composite rule
ROCplcb.Rel

Areas under the time-dependent ROC curves

Estimation: Monte Carlo (200 samples)
Difference: relative, lag = 2 (1, 0.8)
Thresholds range: (-1.61, 6.47)

Case: 2
Recorded time(s): 0, 0.5, 1, 2.1, 4.9, 5.89, 6.89, 7.89, 8.83
 dt t + dt    AUC  Cut.1  Cut.2
  1   9.83 0.7943 1.7499 1.3999
  2  10.83 0.8108 1.6207 1.2966
  4  12.83 0.8112 1.3623 1.0898

# ROC curves under the composite rule
plot(ROCplcb.Rel, legend = TRUE)
```