Introduction to `jmpost`

Joint Model for Posterior Overall Survival Tranjectories

Georgios Kazantzidis Daniel Sabanés Bové Xiaoyan Fang

Introduction

In clinical development of new cancer medicines a key challenge is to decide on the next steps after having collected the initial data in Phase 1 studies. In phase 1 studies patients are followed until the cancer is growing again (progression event). Thus, reliable overall survival data are not available. However, we can use additional information from previous clinical trials or real-world data to understand the time between progression and death. Specifically, we can correlate the tumor response data, that are longitudinal measurements, (i.e. the shrinkage and growth of tumor lesions over time) with the overall survival of the patients or their hazard ratios (Beyer et al. 2020). Thereby we can predict the overall survival in our Phase 1 studies and therefore make better decisions (Kerioui et al. 2020).

The sum of the longest diameter of the cancer (SLD) provides an effective and representative biomarker for longitudinal measurements in solid cancer. SLD is relatively easy to measure with great accuracy while also being a good representative of the development of cancer.

The association between the longitudinal data and the overall survival can be achieved in different ways. The predicted SLD value or its first derivative over time can be directly treated as the linking factor. Other factors such as the time to growth, the tumor shrinkage rate or the tumor growth rate can also be used to associate longitudinal outcomes with overall survival.

Bayesian methodology provides an important tool for the analysis of similar data. Firstly, the use of historical data from previous trials is improving the performance of the models. Secondly, prior definition allows the adjustment of the models according to the specific factors of each analysis.

Although these statistical methods provide significant advantages in decision making processes in cancer drug development, their implementation is relatively intricate and time consuming. Compared to other R-packages, `jmpost` provides the flexibility of non-linear modeling for the longitudinal part of the joint model. Non-linear parametrization respects the biological implications of clinical trials in oncology where the observed tumor response usually follows non-linear patterns.

Minimal usage

```
my_long_mod <- LongModel(stan = StanModule(
    functions = "exp_long_functions.stan",
    data = "exp_long_data.stan",
    priors = long_prior(),
    inits = list(),
    parameters = "exp_long_parameters.stan",
    transformed_parameters = "exp_long_transformed_parametes.stan"
))

my_temp_os <- LogLogisticOs()
```

```
my_link <- HazardLink(
    parameters = "beta_ttg",
    contribution = "* rep_matrix(ttg(psi_ks, psi_kg, psi_phi), rows(time))",
    stan = StanModule()
)

my_os <- parametrize(osmod = my_temp_os, link = my_link)
JointModel(long = my_long_mod, os = my_os)
```

Common usage

`jmpost` is a package aiming to optimize the procedure of developing a new Bayesian joint model. Minimal use of the package provides complete examples of joint modeling code. Here we explain the workflow of the package `jmpost` using `stan` libraries as an example.

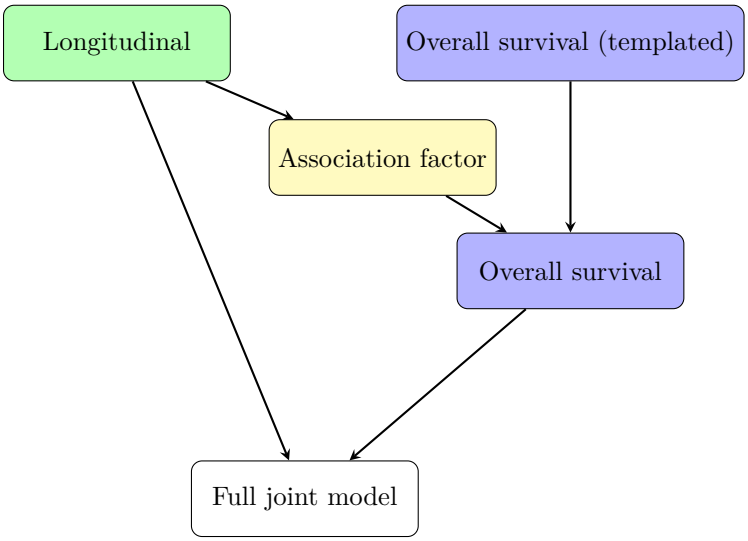

Figure 1: Package workflow. Different colours signify objects of different classes. Arrows indicate the relationships between the objects.

Figure~ illustrates the main workflow of the `jmpost` package. Three objects need to be combined for the construction of the full Bayesian joint model. The longitudinal sub-model, including all the required information for the modeling of the longitudinal measurements, the survival sub-model, including information of for the time to event data and the link or association factor, containing information for the conection of the two sub-models.

```
Long_mod <- LongModel(
    stan = StanModule(
        functions = "exp_long_functions.stan",
        data = "exp_long_data.stan",
        priors = long_prior(),
        model = "",
        inits = list(),
        parameters = "exp_long_parameters.stan",
        transformed_parameters = "exp_long_transformed_parametes.stan",
        generated_quantities = ""
    )
)
```

The construction of the different parts of the model is flexible. Here, we create the longitudinal part by calling the `LongModel` constructor. `LongModel` identifies the type of the sub-model whereas the helper constructor `StanModule` is responsible for the actual stan code. The created object contains six compartments containing different parts of the stan code (functions, data, priors, model, parameters, transformed parameters and generated quantities) while also `inits` which contains the initial values for the Markov Chain Monte Carlo run. The user is able to populate the different parts of the object with stan code either by providing the names of the stan files contained in the `jmpost` directory, the full paths of stan files outside the `jmpost` direcotry or by providing stan code in text form (code chunk below).

```
example_stan <- LongModel(
    stan = StanModule(
        functions = "type stan code here;",
        data = "type stan code here;"
    )
)
```

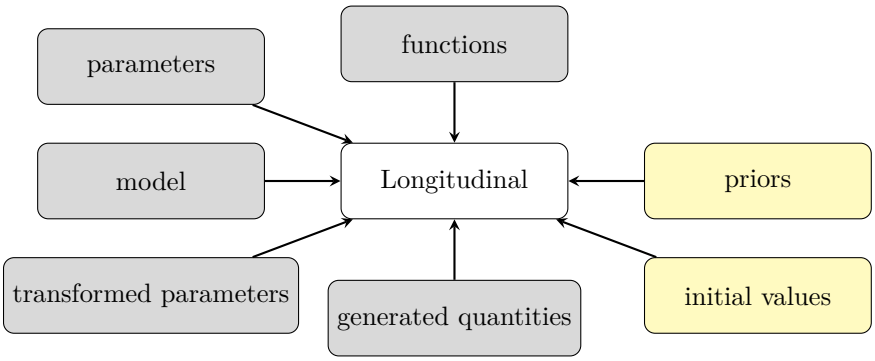

Figure 2: Structure of 'StanModule' object.

Figure~ depicts the structure of a `StanModule` object. Gray coloured cells indicate character strings containing stan code whereas yellow coloured cells mark list objects.

```
StanModule(functions = "",
           data = "",
           model = "",
           parameters = "",
           transformed_parameters = "",
           generated_quantities = "",
           priors = list(),
           inits = list()
)
```

The `jmpost` repository contains stan code for the creation of a specific stan model. The user is able to populate the longitudinal object `Long_mod` either by specifying the stan files containing the relevant code of the section or by directly typing the stan code into R. We split the model section of the stan file into two parts, the `priors` and the `model`. `Priors` contain a list with the names of the parameters and their prior distribution definitions while `model` is a character that contains model relevant information such us the definition of the likelihood. The list of priors should be a named list of strings where each element represents the stan code for the definition of the desired density.

```
prior_set <- StanModule(priors = list("param_A" = "normal(0,1);"))
```

The user can modify the prior list at any point during the workflow.

```
prior_set@priors$param_A <- "normal(1,1);"
```

Call of the object prints the different sections.

```
Long_mod
```

The overall survival object contails all ther information for the paramtetrisation of the second sub-model of the joint model. The overall survival object has identical structure as the Longitudinal object (Figure~). The slots `functions`, `parameters`, `model`, `transformed parameters` and `generated quantities` should be provided with stan code whereas `priors` and `initial values` should be filled with lists. An example with Log-logistic parametrisation of the overall survival sub-modlel object is provided in `LogLogisticOS()`.

```
temp_os <- LogLogisticOs()
```

Although the Longitudinal and the Survival object identical structures, the contained stan code has some fundamental differences. The functions contained in the overall survival object depend on the parametrisation of the longitudinal part. Thus, in order to complete the stan code of the former object, additional information is required. The required details are provided to the overall survival object through the link object. The link is an object containing the parametrisation of the longitudinal sub-model, that will be used as part of the overall survival object.

```
link <- HazardLink(
    parameters = "beta_ttg",
    contribution = "* rep_matrix(ttg(psi_ks, psi_kg, psi_phi), rows(time))",
    stan = StanModule( )
)
```

Link objects or association factor, contain the slots of `parameters`, `contribution` and `stan`. Here, the slot `parameters` is the name of the linking parameter of the two sub-models. `Contribution` contains the stan code relevant to the linking parameter. `Parameters` and `contribution` contain all the information required for the completion of the overall survival model.

The merge of the `Longitudinal`, `Overall survival` and `link` object takes place in two parts. First, the parameterisation of the overall survival sub-model needs to be completed using the information from the link object (part 1).

```
os <- parametrize(osmod = temp_os, link = link)
```

Secondly, the `Longitudinal` and `Survival` sub-models need to be combined (part 2). Function `JointModel` merges the two sub-models, creates a character containing the full stan model, creates a stan file with the produced model and checks the functionality of the produced file. Mistakes in the stan file will be reported from the `JointModel` call.

```
jm <- JointModel(long = Long_mod, os = os)
```

`jm` object contains all the complete stan code for the joint model. `jmpost` uses the package `cmdstanr` for the compilation of the stan model. Although the stan code of the model is obtained in `jm`, additional information for the model run should be provided.

```
jm_options <- mcmc_options(chains = 1,
                           parallel_chains = 1,
                           iter_warmup = 200,
                           iter_sampling = 500,
                           max_treedepth = 12,
                           adapt_delta = .9,
                           gauss_legendre = gauss_legendre()
)
```

`mcmc_options` object contains technical information relevant to the model run. Most of the options are

borrowed from the package `cmdstanr` except from the `gauss_legendre` that defines the integration parameters.

Example

SLD model

$$y_{ij} \sim N(SLD_{ij}, SLD_{ij}^2 \sigma^2) \tag{1}$$

Where:

- y_{ij} is the observed tumor mesasurements
- SLD_{ij} is the expected sum of longest diameter for subject i at time point j

Expected SLD

$$SLD_{ij} = b_i[\phi_i e^{-s_i t_{ij}} + (1 - \phi_i)e^{g_i t_{ij}}] \tag{2}$$

Where:

- i is the subject index
- j is the visit index
- t_{ij} the time since first treatment for subject i at visit j
- SLD_{ij} is the observed SLD measurement for subject i at visit j
- b_i is the Baseline sld measurement
- s_i is the kinetics shrinkage parameter
- g_i is the kinetics tumor growth parameter
- ϕ_i is the proportion of cells affected by the treatment

The SLD parameters

$$\begin{aligned} \phi_i &= logit^{-1}(logit(m_{\phi l_i}) + \eta_{\phi i} * \omega_\phi) \\ s_i &= exp(ln(m_{sl_i}) + \eta_{si} * \omega_s) \\ g_i &= exp(lm(m_{gl_i}) + \eta_{gi} * \omega_g) \\ b_i &= exp(ln(m_{bl_i}) + \eta_{bi} * \omega_b) \end{aligned}$$

Where:

- i is the subject index
- l_i is the group/treatment index for subject i
- m_{xl_i} is the mean for parameter x in group l_i
- η_{xi} is a random effects offset on parameter x for subject i
- ω_x is the variance for the random effects on parameter x

The SLD Hyper-parameters

$$
\begin{aligned}
m_{sl_i} &\sim Lognormal(\mu_s, \sigma_s) \\
m_{gl_i} &\sim Lognormal(\mu_g, \sigma_g) \\
logit(m_\phi l_i) &\sim N(logit(\mu_\phi), \sigma_\phi)
\end{aligned}
$$

Where:

- l_i is the group/treatment index of subject i
- μ_x is the mean of the parameter distribution
- σ_x is the variance of the parameter distribution

Survival model

$$h_i(t) = h_0(t) e^{\beta f_i(t)} e^{\gamma G_i} e^{\beta_c X_i}$$

Where:

- $h_i(t)$ is the overall survival hazard function
- $h_0(t)$ is the baseline hazard function
- $f_i(t)$ is the derivative of the SLD trajectory for subject i at time t
- G_i is the time to growth for subject i
- X_i is the matrix of covariates at each visit for subject i
- $\beta, \gamma \& \beta_c$ are the strength coefficients for the derivative of the SLD trajectory, the time to growth and the subjects covariates respectively

Baseline hazard

$$
\begin{aligned}
t &\sim LogLogistic(\lambda, p) \\
f_0(t) &= \frac{\lambda p (\lambda t)^{p-1}}{(1 + (\lambda t)^p)^2} \\
h_0(t) &= \frac{\lambda p (\lambda t)^{p-1}}{1 + (\lambda t)^p} \\
S_0(t) &= \frac{1}{1 + (\lambda t)^p}
\end{aligned}
$$

Where:

- $f(t)$ is the probability density function
- $h_0(t)$ is the hazard distribution function
- $S_0(t)$ is the survival distribution function

Time to growth

$$G_i = \frac{logit(\phi_i) + log(s_i g_i^{-1})}{s_i + g_i}$$

Where:

- G_i is the time to growth for subject i
- b_i is the baseline SLD measurement
- s_i is the kinetics shrinkage parameter
- g_i is is the kinetics tumor growth parameter
- ϕ_i is the proportion of cells affected by the treatment

References

Beyer, Ulrich, David Dejardin, Matthias Meller, Kaspar Rufibach, and Hans Ulrich Burger. 2020. "A Multistate Model for Early Decision-Making in Oncology." *Biometrical Journal* 62 (3): 550–67.

Kerioui, Marion, Francois Mercier, Julie Bertrand, Coralie Tardivon, René Bruno, Jérémie Guedj, and Solène Desmée. 2020. "Bayesian Inference Using Hamiltonian Monte-Carlo Algorithm for Nonlinear Joint Modeling in the Context of Cancer Immunotherapy." *Statistics in Medicine* 39 (30): 4853–68.

Appendix