

# jmpost

## An R package to develop and use Tumor Growth Inhibition and Survival models in a Bayesian setting



<https://genentech.github.io/jmpost/>



**jpost:** An R package for Bayesian joint TGI-OS models

Francois Mercier (1), Isaac Gravestock (2), Craig Gower-Page (3), Daniel Sabanés Bové (4)

(1) Modeling and Simulation, Clinical Pharmacology, Genentech Research and Early Development, Basel, CH, (2) Statistical Engineering, Data Sciences, F. Hoffmann-La Roche AG, Basel, CH, (3) Statistical Engineering, Data Sciences, Roche Products Limited, Welwyn, UK, (4) inferential.biostatistics GmbH/ RCONIS, Basel, Switzerland.

**Genentech**  
A Member of the Roche Group

### Background

Clinical trials gather repeated measurements on endogenous markers of disease progression and on time to events, such as treatment discontinuation or death, with the ones possibly informing the other. For instance, in oncology solid tumors, changes in tumor burden as measured by the sum of lesion diameter (SLD) can be correlated with overall survival (OS).

Various modeling approaches have been considered to quantify this relationship including **two-stage** and **joint** models. In recent years, joint **TGI-OS models** have gained attention, with an implementation in a **Bayesian** framework enabling the natural handling of uncertainty through posterior samples [Ker20].

Notwithstanding, probabilistic programming languages like BUGS or Stan can have a steep learning curve.

### Objective

To facilitate fitting in a Bayesian framework:

- 1) Tumor growth inhibition (TGI) (only) models,
- 2) Survival (OS) models, including two-stage TGI-OS, or
- 3) Joint TGI-OS models,

We created a new R package called **jpost**.

The package is designed to :

- Facilitate the implementation and inference using MCMC for the aforementioned types of models,
- Support the diagnostic and extraction of model outputs
- Simulate scenarios using the fitted models

### Candidate models and algorithm

The models implemented in **jpost** include:

Process	Model
TGI	[1] Stein-Fojo [2] Generalized Stein-Fojo [3] Claret-Bruno
OS	[1] Weibull (PH) [2] Log-logistic (PH) [3] Gamma (PH)

The models are implemented in Stan, and requires companion installation of the R package **cmdstanr**. Stan supports different algorithms, and most notably an adaptive Hamiltonian Monte Carlo (HMC) sampler. Stan provides a rich set of diagnostics, such as:

- # divergent transitions during sampling,
- Effective sample sizes (ESS),
- Scale reduction factors (Rhat).

### Example workflow and outputs for a joint Stein-Fojo/Weibull model

#### 1. Setup and load data

```
library(tidyverse)
library(jmpost)
library(posterior)
```

The format of the input datasets is simple:

- **Longitudinal biomarker dataset:**

I	year	slid	arm	id
<dbl>	<dbl>	<dbl>	<fctr>	<fctr>
-0.005475702	35.8	Docetaxel	953	
0.112251882	39.5	Docetaxel	953	
0.246406571	53.0	Docetaxel	953	
-0.010951403	49.0	Docetaxel	423	
0.117727584	48.0	Docetaxel	423	
0.183436003	78.0	Docetaxel	423	

- and the **survival dataset:**

id	arm	ecog	age	race	sex	os_time	os_event
<fctr>	<fctr>	<fctr>	<dbl>	<fctr>	<fctr>	<dbl>	<dbl>
588	Docetaxel	1	61	WHITE	F	2.0506502	FALSE
330	MPDL3280A	1	56	WHITE	F	1.6755647	FALSE
791	Docetaxel	0	72	WHITE	F	0.9007529	TRUE
635	Docetaxel	0	42	OTHER	F	1.6591376	TRUE
365	MPDL3280A	0	64	WHITE	F	1.4291581	TRUE
773	Docetaxel	0	65	WHITE	M	1.6290212	FALSE

The function **DataJoint()** is used to rearrange the input data for them to be analysis-ready:

```
joint_data <- DataJoint(
  subject = DataSubject(data=subj_df, subject="id", arm="arm",
    study="study"),
  longitudinal = DataLongitudinal(data=long_df, formula=slid~year),
  survival = DataSurvival(data=os_data, formula=Surv(os_time,
    os_event) ~ ecog+age+race+sex)
)
```

#### 2. Basic model specification

Upon calling **JointModel()**, (i) we select the longitudinal model (here SteinFojo), the survival model (here WeibullPH), and the link (here linkGrowth i.e.  $\log(\text{kg}_i)$ ), and (ii) we specify the priors for the model parameters.

```
joint_mod <- JointModel(
  longitudinal = LongitudinalSteinFojo(
    mu_bslid = prior_normal(log(65), 1),
    mu_ks = prior_normal(log(0.52), 1),
    mu_kg = prior_normal(log(1.04), 1),
    omega_bslid = prior_normal(0, 3) |> set_limits(0, Inf),
    omega_ks = prior_normal(0, 3) |> set_limits(0, Inf),
    omega_kg = prior_normal(0, 3) |> set_limits(0, Inf),
    sigma = prior_normal(0, 3) |> set_limits(0, Inf)
  ),
  survival = SurvivalWeibullPH(
    lambda = prior_gamma(0.7, 1),
    gamma = prior_gamma(1.5, 1),
    beta = prior_normal(0, 20)
  ),
  link = linkGrowth(
    prior = prior_normal(0, 20)
  )
)
```

#### 3. Model fitting

**sampleStanModel()** function can be used to fit the model (here using default MCMC options)

```
joint_results <- sampleStanModel(joint_mod, data=joint_data)
```

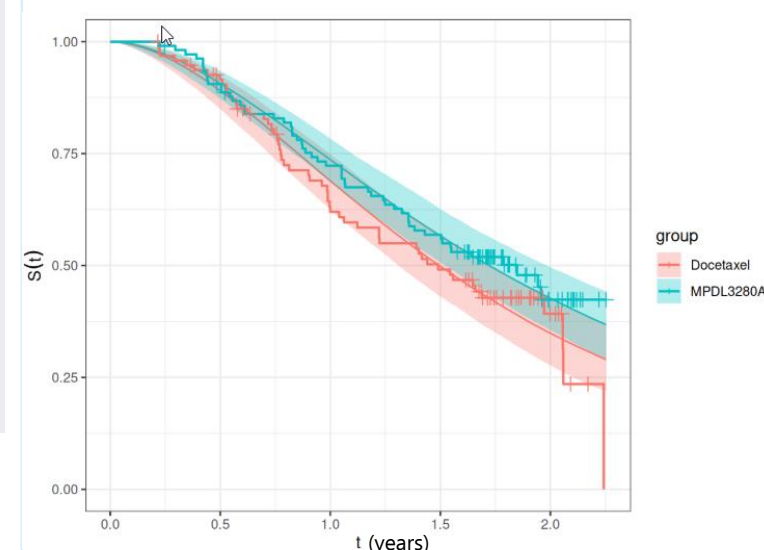
Let's check the convergence of the population parameters.

```
mcmc_joint_results <- cmdstanr::as.CmdStanMCMC(joint_results)
mcmc_joint_results$summary(vars)
```

```
# A tibble: 19 x 10
  variable      mean  median    sd    mad    q5    q95  rhat  ess_bulk
  <chr>         <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl> <dbl>  <dbl>
1 lm_sf_mu_bs...  3.76    3.76    0.0371  0.0355  3.69  3.82  1.02  192.
2 lm_sf_mu_ks... -0.0188  0.00943  0.235  0.228  -0.445  0.339  1.00  462.
3 lm_sf_mu_ks... -1.06    -1.03   0.330  0.314  -1.66  -0.591  1.01  559.
4 lm_sf_mu_kg... -0.585   -0.576  0.139  0.146  -0.837  -0.375  0.999  500.
5 lm_sf_mu_kg... -0.892   -0.890  0.139  0.138  -1.14  -0.656  1.00  552.
6 lm_sf_sigma...  0.129    0.129  0.00371  0.00366  0.123  0.135  1.00  840.
7 lm_sf_omega...  0.530    0.529  0.0265  0.0256  0.488  0.576  1.01  340.
8 lm_sf_omega...  0.936    0.913  0.178  0.169  0.680  1.27  1.01  409.
9 lm_sf_omega...  1.36    1.33  0.252  0.247  0.989  1.81  1.00  566.
10 lm_sf_omega...  0.684    0.677  0.0812  0.0744  0.561  0.831  1.00  781.
11 lm_sf_omega...  0.956    0.952  0.0953  0.0973  0.810  1.12  1.01  585.
12 beta_os_cov...  0.842    0.829  0.234  0.229  0.464  1.24  1.00  812.
13 beta_os_cov...  0.00316  0.00296  0.00981  0.00996  -0.0128  0.0197  0.998  975.
14 beta_os_cov...  0.658    0.656  0.431  0.428  -0.0678  1.36  1.00  901.
15 beta_os_cov... -0.00443  -0.00251  0.237  0.247  -0.371  0.371  1.00  760.
16 beta_os_cov...  0.313    0.305  0.215  0.217  -0.0363  0.677  1.00  735.
17 sm_weibull...  1.81    1.81  0.167  0.170  1.55  2.09  1.00  794.
18 sm_weibull...  0.261    0.213  0.176  0.132  0.0733  0.635  0.998  805.
19 link_growth...  0.880    0.861  0.235  0.242  0.525  1.29  1.00  571.
# 1 more variable: ess_tail <dbl>
```

Using the functions **SurvivalQuantities()** and **autoplot()**, we can visualize the posterior predictive survival curves, and overlay the Kaplan-Meier estimates.

```
time_grid <- seq(from = 0, to = max(os_data$os_time), length = 100)
os_surv_group_grid <- GridGrouped(
  times = time_grid,
  groups = with(
    subj_df,
    split(as.character(id), arm)
  )
)
os_surv_pred <- SurvivalQuantities(
  object = joint_results,
  grid = os_surv_group_grid,
  type = "surv"
)
```



[Ker20] Kerioui M, Mercier F, et al. (2020). Bayesian inference using Hamiltonian Monte-Carlo algorithm for nonlinear joint modeling in the context of cancer immunotherapy. *Statistics in Medicine*, 39: 4853-4868

