

Memoranda to Reviewers for:

“BayesSPsurv: An R Package to Estimate Bayesian (Spatial) Split-Population Survival Models” R Journal 2021-29

Memo to Reviewer 1

We thank you for your kind words and for providing constructive as well as incisive comments. We learned a lot from your review. We have implemented revisions to our paper, code and built-in documentation to incorporate all your suggestions. We have also addressed the queries you raised in your review. Below we briefly describe how we responded to your comments and suggestions.

1. “The from-scratch C++ implementation of these algorithms is adequate in lieu of an external MCMC sampling dependency built on C++ subroutines...STAN...may be beneficial for future development...but is not obviously superior or necessary for the existing use case.”

Response: Following your suggestion, we emphasize in our revised paper’s final paragraph that users can implement MCMC estimation of the spatial and non-spatial split-population survival models in our paper by using STAN that employs the Hamiltonian Monte Carlo algorithm (**page 16**). We note here that a key advantage of the Hamiltonian Monte Carlo algorithm in STAN is that it permits sampling from high dimensional posterior distributions.

2. “...It is not clear whether W and V are accepted or rejected within the same Metropolis-Hastings proposal step...there is a single proposal variance parameter (`prop.var`) in...`spatialSPsurv()`. Clarity on this...may be valuable. It would seem better for tuning acceptance rates to assign separate steps...the introduction of separate proposal variance variables (`prop.var`) for parameters W and V to optimize Metropolis-Hastings acceptance rates.”

Response: This is an excellent suggestion. We apologize for the lack of clarity presenting the MCMC algorithm’s Metropolis-Hastings step for sampling the spatial random effects W and V in `spatialSPsurv()` and `exchangeSPsurv()` in the earlier version of our paper. We have addressed this point in a few ways. First, following your advice, we have introduced in our updated code separate proposal variance arguments (`prop.varW` and `prop.varV`) for the parameters W and V that users can specify in `spatialSPsurv()` and `exchangeSPsurv()`, and we have assigned separate proposal steps to estimate these parameters to optimize the Metropolis-Hastings acceptance rates. This adjustment to the code is included in the latest version of our **BayesSPsurv** package (V 0.1.3), which is already available on CRAN R (<https://cran.r-project.org/web/packages/BayesSPsurv/index.html>.) The code is also available on the package’s public GitHub repository (<https://github.com/Nicolas-Schmidt/BayesSPsurv>).

In our revised paper, we emphasize (i) the assignment of separate proposal steps for sampling W and V when discussing our MCMC algorithm for estimating the Spatial SP survival model (**page 6**) and (ii) using `prop.varW` and `prop.varV` in our MCMC routine for `spatialSPsurv()` (**see**

pages 9-14). Further, we estimate the Spatial SP survival model on the post-civil war peace data after making the adjustments to the code described above and then report the results (trace and density plots from the single and multiple [parallel] chains and convergence diagnostics) from this exercise in the paper (pages 9-10). Finally, we also incorporated separate proposal variance variables for the nonspatial frailties W and V and assigned separate proposal steps for estimating these parameters for the nonspatial SP survival model with i.i.d frailties (pages 14-15). The updated code available on Github and CRAN R reflects this change.

4. “Only minor improvements to user experience could be suggested, namely a progress bar to monitor completion of MCMC sampling in `spatialSPsurv()`.”

Response: We have addressed this suggestion by including an automatic progress bar for the three models in the updated R package (also mentioned on page 10 of manuscript). This new feature provides users with information about the progress of the MCMC sampling for each of our sampling functions: `spatialSPsurv()`, `exchangeSPsurv()` and `pooledSPsurv()`.

Memo to Reviewer 2

We thank you very much for providing extremely useful comments. We have implemented numerous revisions in our paper and code to ensure that we address all the issues you raised in your review. We briefly discuss below how we incorporated your comments.

1. “Implementation of a joint distribution for the frailty terms is not...straightforward or required here, but there are approaches, like having a shared frailty term with factor loadings. It might be helpful just to comment on this and to indicate whether this is something that might be considered in the future.”

Response: This is a constructive point that has been incorporated (as per your advice) in two main ways in our revised paper. First, we note on page 3 that if researchers believe that units in their survival data are clustered such that units within the same cluster share the same frailty while frailties are independent across clusters, then they should incorporate shared frailty terms for each cluster in their data. Further, on page 3, we emphasize that our study focuses on incorporating unit-specific (i.e., individual) frailty terms in the model's split-stage and survival-stage since units exhibit their own unique frailties in the survival data employed for our empirical application. Second, we emphasize in the paper's conclusion that in future work we plan to build on Yin's (2005) approach to model shared frailty terms with factor loadings in cure models and apply it in our setting (page 16).

2. “...users will prefer to run multiple chains to examine multi-chain convergence diagnostics...if they can be run in parallel it may decrease overall running time. It may be helpful to add some guidance on how or if multiple chains might be used here...they should just plug straight into the coda package functions...”

Response: This is an excellent point, and it has led us to update the code in the **BayesSPsurv** package and resubmit the revised version (V 0.1.3) to CRAN R (now available at <https://cran.r->

project.org/web/packages/BayesSPsurv/index.html). The user can now specify their preferred starting values for the survival-stage and split-stage covariates as well as W and V using the `ini.beta`, `ini.gamma`, `ini.W`, and `ini.V` arguments, respectively. We illustrate the usage of these arguments in our updated application section (**pages 6-16**). This update on its own is a significant improvement in the package's flexibility for the user.

Per your advice, we then created a routine for generating multiple chains in parallel for each of the three models in **BayesSPsurv** and provide the code on the GitHub repository for the package (<https://github.com/Nicolas-Schmidt/BayesSPsurv>). The routine for simultaneously generating the multiple chains with different initial starting values uses the **do.parallel** and **doRNG** R packages, while the trace and density plots from these multiple chains and their Gelman-Rubin diagnostics is from the **coda** package. After generating multiple chains, we demonstrate how a user can collect them into a single `mcmc.list` object, plot the trace and density plots of the multiple chains, and examine Gelman-Rubin diagnostics to assess convergence.

Due to space constraints, we provide the routine for estimating multiple chains on GitHub rather than in the manuscript and then provide the code for (and output from) plotting the multiple chains and calculating Gelman-Rubin diagnostics (using the **coda** package) for the Spatial SP survival model on **pages 11-13** of the manuscript. Additionally, we suggest in our revised paper that users can also employ the multiple chain routine available in our GitHub repository to extract and assess (e.g. convergence) these parallel chains for the split-population survival model with nonspatial frailties and without any frailty terms (**pages 14 and 16**).

3. "It might be useful in the text to give a rough indication of running time for the sampling functions..?"

Response: Following your suggestion, we used `system.time()` to obtain the running times for the three sampling functions as applied to our example data as well as the multiple chain routine for the Spatial SP survival model. The running times for our application were as follows: (i) `spatialSPsurv()` was about 23 minutes (**see page 10**), (ii) `exchangeSPsurv()` was approximately 16 and a half minutes (**page 14**), (iii) `pooledSPsurv()` was about 10 minutes (**page 15**), and the parallel multiple chain routine for the Spatial SP survival model took approximately 70 minutes (**page 11**).

4. "While beyond the scope of revisions necessary for this paper, a verbose option for the sampling functions that provides feedback on sampling progress would also be very useful if possible?"

Response: We have included a progress bar (as requested by the other reviewer as well) in the updated version of the **BayesSPsurv** R package (V 0.1.3). While we will include a verbose option in future iterations of the package, the progress bar provides the percentage of the computation completed in real time for `spatialSPsurv()`, `exchangeSPsurv()` and `pooledSPsurv()`. This addition provides a significant improvement to user experience.