Bayesian Pharmacometric Software Benchmarks

Jose Storopoli[1], Casey Davis[2], Mohamed Tarek[1,3], Pavan Vaddady[4], Vijay Ivaturi[1,5]

[1]Pumas-AI Inc., USA

[2]Gilead Sciences Inc., USA

[3]Business School, University of Sydney, Australia

[4]Daiichi Sankyo Inc., USA

[5]University of Maryland Baltimore, USA

Motivation

Full Bayesian modeling has become a widely accepted approach in pharmacometrics due to its flexibility, accuracy, and ability to handle complex models [1]. There are a number of available software tools for fully Bayesian methods in pharmacometrics, such as Stan [2] (with Torsten [3]), NONMEM [4], and Pumas [5]. However, despite the popularity of these tools, there is no comparison study that compares their performance, accuracy, and usability in the context of pharmacometrics. This study aims to address that gap.

Bayesian inference depends on our ability to quantify the posterior distribution of the parameters conditioned on the data. This posterior distribution combines the prior distribution (our prior knowledge), and the likelihood (knowledge obtained from the data), and it contains all of our knowledge about the parameters (here, denoted θ).

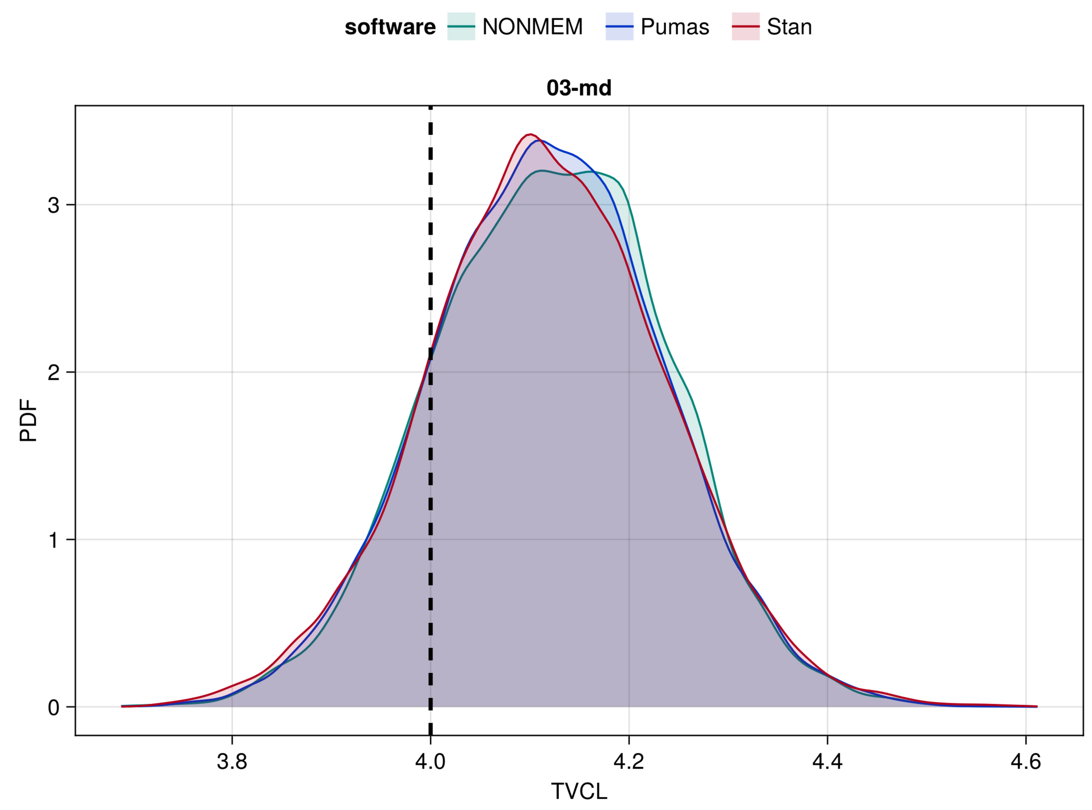
$$
\begin{aligned}
p(\theta \mid x) &= \frac{p(x, \theta)}{f(x)} \\
&= \frac{f(x \mid \theta) p(\theta)}{f(x)} \\
&= \frac{f(x \mid \theta) p(\theta)}{\int_{\Theta} f(x \mid \theta) p(\theta)} \\
&\propto f(x \mid \theta) p(\theta)
\end{aligned}
$$

In practice, the marginal distribution is often analytically intractable, and it is often just cast aside so that we have the posterior distribution *up to a constant*. Markov Chain Monte Carlo (MCMC) [6] allows us to sample from the posterior in such a way that the sample represents the true posterior distribution arbitrarily closely, and we can perform our inference based on this sample.

Methods

This study uses a set of common pharmacometric models to compare the performance, accuracy, and overall experience of using Torsten/Stan [3], NONMEM [4], and Pumas [5] in the inference step of the Bayesian workflow. The following software versions were used in this study: Pumas [5] 2.5.0 with Julia [7] 1.9.3; NONMEM [4] 7.5.1; and Torsten 0.90 (Stan 2.29.2). All of the codes used in this study are public and available at github.com/Pumas-AI/Bayesian-Benchmarks. In this poster, we summarize the preliminary results from benchmarking only 4 of the chosen test models as this is still a work in progress. The 4 models benchmarked so far are: IV 2-compartment (model 01), depot 1- and 2- compartment (models 02 and 03), and the Friberg-Karlsson model for myelosuppression [8] (model 05). The benchmarking environment for this study is a cloud-based virtual machine with an Intel Xeon 8375C CPU, 128GB of RAM with 32 threads. All the models' parameters were inferred using the No U-Turn Sampler (NUTS) [9] with 4 parallel chains and 8 threads per chain. We run each model 5 times, each time with different initial values for each chain. The initial values were kept fixed across all three software implementations. All chains had 1,500 iterations with the first 500 discarded as warmup. To calculate benchmark and convergence metrics, we concatenated all 5 runs $\cdot$ 4 chains $\cdot$ 1000 iterations $=$ 20000 iterations into a single chain for comparison for all four models across all three software implementations; as shown as an example in Figure 1

Figure 1: Posterior Density Comparison – Model 03 – TVCL – 20,000 iterations



Results

The Rhat convergence diagnostic was well below the 1.1 mark so the different chains all seem to have mixed well. The Effective Sample Size (ESS) per second, a measurement of runtime inference performance, varied across models and softwares, as shown in Figure 2. In less complex models (models 01, 02, and 03), NONMEM [4] was faster than both Pumas [5] and Torsten/Stan [3]. Whereas, in the Friberg model (model 05), NONMEM [4] performance was comparable to both Pumas [5] and Torsten/Stan [3] in most of the parameter estimates. From a usability standpoint, a complete Bayesian workflow with parallelization could be performed within Pumas [5] while Torsten/Stan [3] and NONMEM [4] required additional tools such as R and bash scripts.

Conclusions

This evaluation helped compare the performance, accuracy, and usability of these tools in the context of pharmacometrics. The preliminary benchmarking results using four models and 32 threads suggest that NONMEM [4] had the highest ESS per second on average across the models tested, followed by Torsten/Stan [3] and Pumas [5]. In accuracy, all three software were found to be sufficiently accurate. In the final study, additional models and settings will also be tested to better understand the performance differences when using different numbers of threads across the three software implementations.

References

[1] M. Tarek et al., A Practitioner's Guide to Bayesian Inference in Pharmacometrics using Pumas, arXiv:2304.04752, 2023.

[2] B. Carpenter, et al., Stan: A Probabilistic Programming Language, *Journal of Statistical Software*, 2017.

[3] Metrum Research Group (2022). Torsten 0.90.0 [Computer software].

[4] ICON (2022). NONMEM 7.5.1 [Computer software].

[5] C. Rackauckas, et al., Accelerated predictive healthcare analytics with Pumas, a high performance pharmaceutical modeling and simulation platform, 2020.

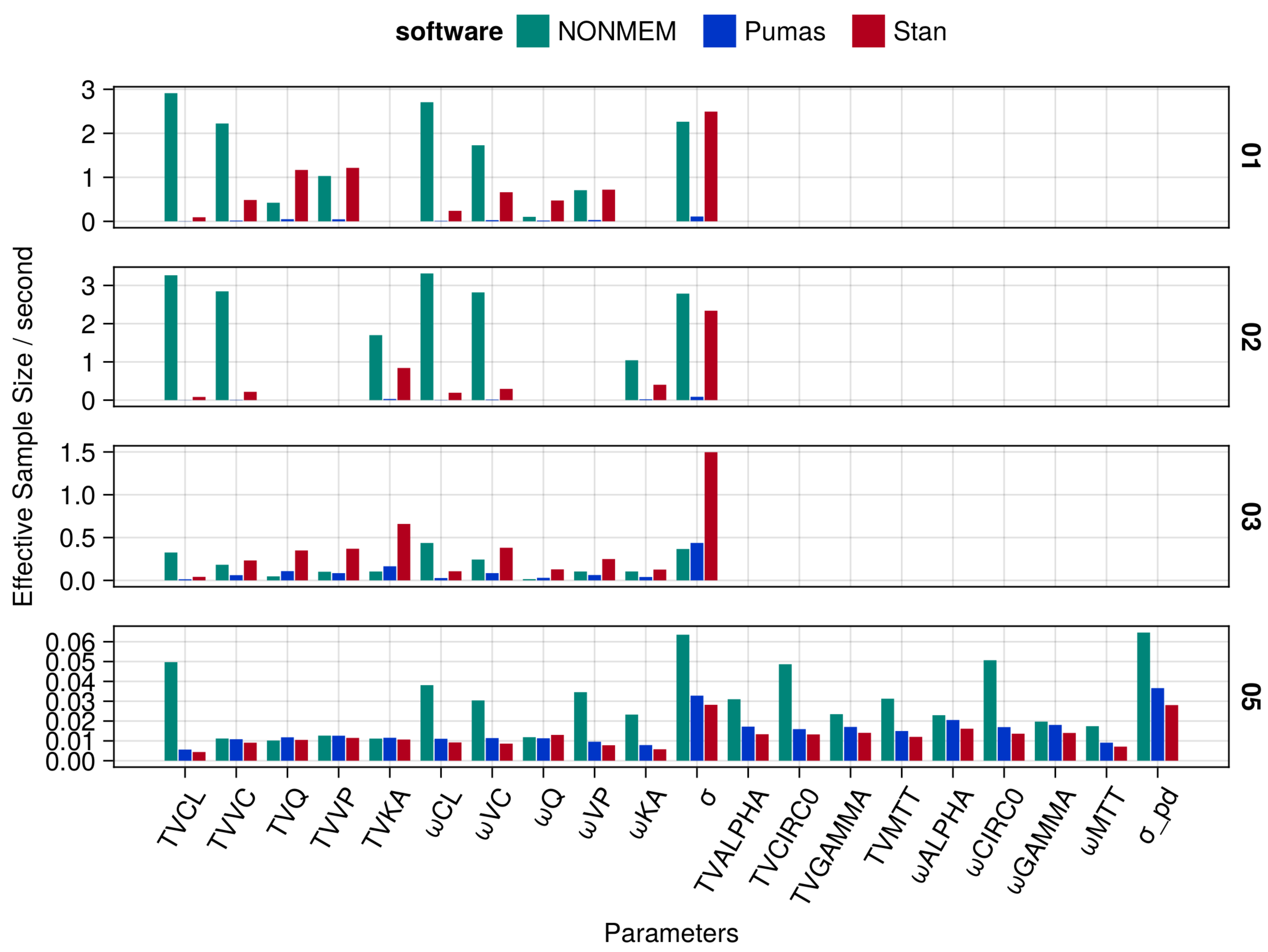
[6] S. Brooks, et al., eds. *Handbook of Markov Chain Monte Carlo*. CRC press, 2011.

[7] J. Bezanson, A. Edelman, S. Karpinski, V. Shah Julia: A fresh approach to numerical computing, *SIAM review*, 59, 1, 65–98, 2017.

[8] L. Friberg, et al., Model of chemotherapy-induced myelosuppression with parameter consistency across drugs, *Journal of Clinical Oncology*, 2002.

[9] M. Hoffman, A. Gelman, The No-U-Turn Sampler: Adaptively Setting Path Lengths in Hamiltonian Monte Carlo, *Journal of Machine Learning Research*, 2014.

Figure 2: Comparison of Effective Sample Size per second across Models and Softwares



ACoP, November 2023, National Harbor, MD, USA