

Population Pharmacokinetic Reanalysis of a Diazepam PBPK Model

A Comparison of *Stan* and *GNU MCSim*

Periklis Tsiros · Frederic Y. Bois · Aristides Dokoumetzidis · Haralambos Sarimveis

the date of receipt and acceptance should be inserted later

Abstract The aim of this study is to benchmark two Bayesian software tools, namely *Stan* and *GNU MCSim*, that use different Markov Chain Monte Carlo (MCMC) methods for the estimation of Physiologically Based Pharmacokinetic (PBPK) model parameters. The software tools were applied and compared on the problem of updating the parameters of a Diazepam PBPK model, using time-concentration human data. Both tools produced very good fits at the individual and population levels, despite the fact that *GNU MCSim* is not able to consider multivariate distributions. *Stan* outperformed *GNU MCSim* in sampling efficiency, due to its almost uncorrelated sampling. However, *GNU MCSim* exhibited much faster convergence and performed better in terms of effective samples produced per unit of time.

Keywords Diazepam · Population Pharmacokinetics · PBPK · Bayesian · GNU MCSim · Stan

P. Tsiros

School of Chemical Engineering, National Technical University of Athens, 9 Heron Polytechniou Street, 15780 Zografou Campus, Athens, Greece.

Tel.: +30-210-7723236

E-mail: ptsirostsib@gmail.com

F. Bois

Institut National de l'Environnement Industriel et des Risques (INERIS), Unit Modles pour l'Ecotoxicologie et la Toxicologie (METO), Parc ALATA BP2, 60550 Verneuil en Halatte, France.

E-mail: Frederic.BOIS@ineris.fr

A. Dokoumetzidis

Department of Pharmacy, University of Athens, Panepistimiopolis Zografou, 15784 Athens, Greece.

Tel.: +30-210-7274122

E-mail: adokoum@pharm.uoa.gr

H. Sarimveis

School of Chemical Engineering, National Technical University of Athens, 9 Heron Polytechniou Street, 15780 Zografou Campus, Athens, Greece.

Tel.: +30-210-7723242

E-mail: hsarimv@central.ntua.gr

Introduction

Pharmacokinetics is the field that studies the time evolution of absorption, distribution, metabolism and excretion (ADME) of drugs in humans and animals. Prof. Macheras has contributed extensively and opened new research directions in the field [1]. He actually introduced and developed this field in Greece and founded the Laboratory of Biopharmaceutics and Pharmacokinetics at the University of Athens, Faculty of Pharmacy. Since then, many researchers from the pharmaceuticals research area and beyond have been motivated by his work to become engaged in the field of pharmacokinetics.

One of the main tools in studying the pharmacokinetics of an organism is compartmental modeling, which models a biological system through interconnected subunits. Depending on the available data and the goal of the modeling process, there are two broad classes of compartmental models: Empirical models, which are build to fit observations and follow a "top down" approach, and the physiologically-based pharmacokinetic (PBPK) models, which make use of prior physiological knowledge and take a "bottom up" approach [2]. In empirical models, the number of compartments is chosen based on goodness of fit. Thus, parameter values have no physiological interpretation and those models can mainly be used for interpolation of drug concentration within the studied population and statistical evaluation of its apparent kinetics [3]. On the other hand, PBPK models are mechanistic; they break a complex system into individual parts whose functioning and interaction is determined by the underlying physiology and mechanistic data on their specific parameters.

The main parameters of a PBPK model are tissue volumes and blood flows and as a consequence, the structural model is confined to the physiology of the species under examination [4]. Besides these drug-independent physiologi-

cal parameters, PBPK models incorporate information about the drug as well, through the drug-related parameters. Whole body PBPK models (WBPBPK) are a category of particular interest where the body is represented by a closed circulation system consisting of several tissues than are important for the ADME processes. Mass balance equations are used to describe regional blood flows, the arterial and venous blood pools connect the tissues, and a lung compartment closes the blood loop [5]. The number and type of compartments included is determined by several factors such as the target compound, the animal species and the administration route. Increasing the number of compartments and sub-compartments increases complexity since more data are needed to determine the model parameters, making large models impractical.

The mechanistic nature of the PBPK models offers many advantages. First of all, it allows the prediction not only of plasma but also of tissue pharmacokinetic profiles, before conducting *in vivo* experiments. This is enabled by the modular scheme; information can be gathered for each compartment separately through *in vitro* experiments or from literature sources. Adding to that, the complex structure that includes both drug-related and physiological parameters permits the creation of more informative models, where data from independent sources (especially for the physiological parameters) can be integrated together with the drug specific experimental results to create a model with more robust inference capabilities [4]. The fact that most physiological parameters can be found in literature, coupled with the fact that most mammals share the same PBPK structure leads to the ability to perform both inter and intra-species extrapolation. These tools allow for example to extrapolate pharmacokinetics from rats to humans and from adults to children. PBPK modeling can also be used for predicting drug-drug interactions [6].

As stated above, the physiological parameters of a PBPK model can be derived from several sources. The drug-related parameters can be obtained from *in vitro* experiments or *in silico* methods. Therefore, PBPK modeling can be used in the drug development phase as a simulation tool [7]. However, extrapolation to humans is a sensitive process and fine tuning to clinical data is needed in order to produce more robust inferences.

There are several methods for estimating drug-related parameters based on the information provided by clinical data [2]. Among them, Bayesian inference is a powerful natural route for refining an estimation based on new data. In this case, preclinical data constitute the prior belief which is updated with the clinical data, resulting in a new estimation in the form of a (posterior) distribution. One of the major advantages of the Bayesian approach is the inherent way in which the quantification of the estimation uncertainty is handled, through the shape of the posterior distribution.

This increases the estimation reliability and is also very important in the extrapolation procedure, where it enables the incorporation of uncertainty in the model predictions. Moreover, if the Bayesian framework is combined with hierarchical modeling, then both the individual and population levels can be described by the model [8,9]. In that way, the inter-individual variability can be quantified and thus it can be separated from the inherent uncertainty of the data [10,11]. One way to account for inter-individual variability is to create a covariate model for the physiological parameters and to assign them random effects, considering all drug-related parameters as fixed effects having a "global" impact [12]. An alternative approach is to use again a covariate model for explaining a part of the inter-individual variability, with physiological parameters as fixed effects and drug-related parameters as random effects [6]. This can be extended to include intra-individual or inter-study variability [13,14].

The cases where Bayesian inference can be implemented directly through the use of the Bayesian theorem are limited due to the difficulty of approximating the denominator, which -in most cases- is a multidimensional integral. This is the reason why Bayesian inference is usually conducted through Markov Chain Monte Carlo (MCMC). MCMC is a class of algorithms that, under certain assumptions, sample directly from the unknown target distribution (posterior), which is summarized at the end of the procedure by the collected samples [15]. There are two main groups of MCMC algorithms: the random walk MCMC and the non-random walk MCMC. In the first group the most popular methods are the Metropolis-Hastings algorithm, the Gibbs sampler and the reversible jump MCMC, while the most known non-random walk MCMC is the Hamiltonian Monte Carlo (HMC).

There exist many free software platforms that allow the user to design the statistical model and perform Bayesian inference through variants of the MCMC method. Among them, the *GNU MCSim* software [16] has been used successfully in estimating parameters in PBPK modeling problems. *GNU MCSim* uses a Metropolis-Hastings algorithm to perform stochastic simulations. The *GNU MCSim* project began in 1991, but is consistently updated and extended with updated releases each year. *Stan* [17] is a newer software language for statistical inference, and was initially released in 2012. MCMC sampling in *Stan* is typically performed using the no-U-turn sampler [18], which is an adaptive variant of Hamiltonian Monte Carlo, but the user can also choose to implement static HMC. *Stan* has been applied in various areas, such as social sciences and finance, but to the best of our knowledge, no applications have been published on the use of *Stan* in PBPK modeling. In this paper, the two software platforms are used to deploy a hierarchical Bayesian model and estimate the drug-related parameters of an existing WBPBPK model, based on published clinical data, ex-

tending the comparison between NONMEM and Winbugs performed by Langdon et al. [19].

Materials and methods

The modeling approach of this work builds on the work of Gueorguieva et al. [20]. The same clinical data were used [21] in order to update prior pharmacokinetic knowledge gained from preclinical rat experiments. We use a slightly different statistical model and a covariate model to explain part of the inter-individual variability of the study population.

Clinical data

The available concentration data refer to two different groups of healthy adults, which have been described in a previous study [21], one group of 12 healthy male and female volunteers and another group of 11 healthy male volunteers. All patients received a single dose of diazepam via intravenous infusion over 15-30 seconds. The dose was 5 to 10 mg and was adjusted for body weight. Following the administration of the drug, venous plasma samples were collected at 0.083, 0.25, 0.5, 0.75, 1, 1.5, 2, 3, 4, 6, 8, 10, 12, 24, 30, 36, 48 and 72 hours. In addition to the plasma data, the age, body weight and gender of each individual were available.

Structural model

The detailed description of the PBPK model used can be found in [22]. Its schematic representation can be seen in Fig. 1. It consists of 11 compartments describing the concentration of the drug in various tissues, namely liver (*LI*), kidney (*KI*), brain (*BR*), intestine (*IN*), stomach (*ST*), muscle (*MU*), adipose (*AD*), skin (*SK*), gonads (*GO*), heart (*HT*) and lungs (*LU*), one compartment to model the rest of the body (*RE*) as well as two blood pools; venous (*VEN*) and arterial (*ART*). The parameters of the model are divided into drug-dependent and physiological (drug-independent) parameters. The first category comprises eleven tissue-to-plasma partition coefficients (*Kp*), fraction unbound in plasma (*f_u*), blood-to-plasma ratio (*R*), and intrinsic hepatic clearance (*CL_{int}*, in L/h). The physiological parameters of the model are simply the regional tissue blood flows (*Q*) and tissue volumes (*V*).

For the current analysis, the same parameterization as in [19] was followed. Specifically, *f_u* was set to 0.015 and *R* was assumed to be equal to 0.65. Moreover, the *Kp*'s were scaled by assuming that the unbound equilibrium tissue-to-venous blood concentration ratio (*Kp_{UB}*) are identical for humans and rats. Therefore, *Kp* values were obtained by

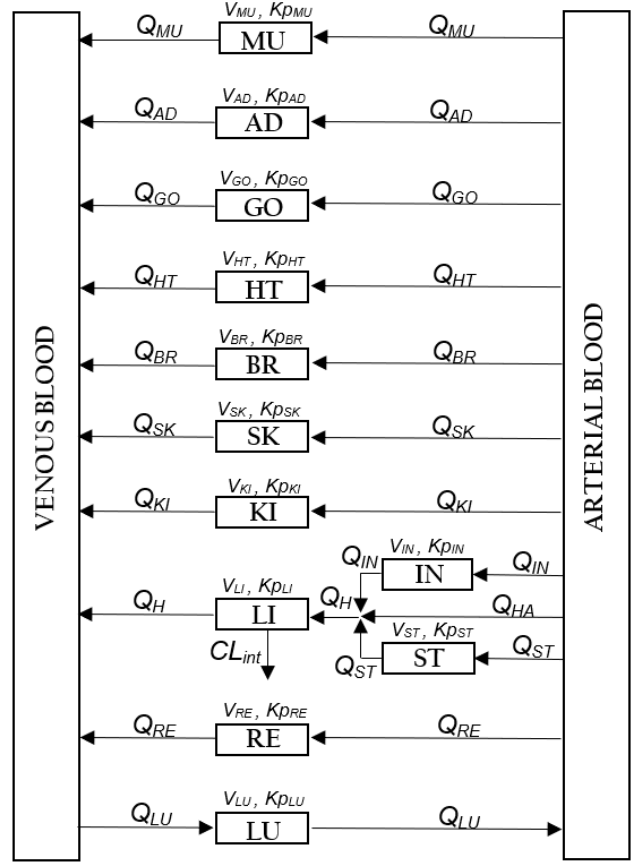


Fig. 1 Schematic presentation of the WBPBPK model of diazepam [22].

scaling the rat posterior results of winbugs (Table III of [20]) using Eq. 1 (except for *Kp* for the rest of the body, which was set to 2.2 as in [19]). The *Kp* values are presented in Table 1.

$$\left. \begin{aligned} f_{u_{rat}} &= 0.15, R_{rat} = 1 \text{ [19]} \\ Kp_{UB_{rat}} &= Kp_{UB_{human}} \\ Kp_{UB} &= Kp \cdot R \end{aligned} \right\} \Rightarrow Kp_{human} = \frac{Kp_{rat}}{6.5} \quad (1)$$

Covariate model for body mass-dependent parameters

Physiological parameters were calculated for each subject using body mass and sex. Blood flows were calculated as a percentage of total cardiac output (TCO), using the coefficients provided by [23] (Table 2). The total cardiac output of each individual was in turn considered to be a function of the body mass calculated as:

$$TCO = 11.22 \cdot BM_{tot}^{0.81} \quad (2)$$

Table 1 Kp values after scaling the posterior results of the rat model of [20].

Tissue	Kp
Lungs	0.71
Intestines	0.53
Stomach	0.75
Liver	1.35
Brain	0.32
Heart	0.86
Kidney	0.73
Skin	0.46
Muscle	0.56
Adipose	3.34
Gonads	0.76
Rest of the Body	2.2

where CO is the total cardiac output in L/h and BM_{tot} the subject's body mass in kg [24].

Table 2 Percentage of TCO received by each tissue [23].

Tissue	percentage of TCO (%)
Lungs	100
Intestines	14
Stomach	1
Liver	6.5
Brain	12
Heart	4
Kidney	19
Skin	5
Muscle	17
Adipose	5
Testes	0.5
Uterus	0.2

To calculate tissue volumes, the method described in [25] was used. A 5th degree polynomial gave the percentage of total body mass of each tissue (Eq. 3) as a function of the body mass, up to 75 kg for males and 65 kg for females.

$$BM_i = X_0 + X_1 \cdot BM_{tot} + X_2 \cdot BM_{tot}^2 + X_3 \cdot BM_{tot}^3 + X_4 \cdot BM_{tot}^4 + X_5 \cdot BM_{tot}^5 \quad (3)$$

where BM_i is the percentage of tissue i with respect to total body mass, and BM_{tot} is the body mass of the subject in *grams*. For higher body masses, constant percentages were used as described in [25]. The polynomial coefficients, X_i , are different for males and females and can be found in Appendix A.

Statistical model

Statistical inference was made on three parameters: CL_{int} , the metabolic clearance of the drug; Kp_{AD} , the adipose tissue-

to-plasma partition coefficient; and fKp , a scaling factor multiplying the rest of the Kp 's.

A three stage hierarchical model was adopted to describe the data collection process. This enabled the description of both the inter-individual variability and uncertainty of the selected drug-related parameters, as well as the quantification of the residual variability which is mainly a result of model misspecification and measurement errors [26]. The first stage of the model described the model likelihood, where proportional errors were modeled by a log-normal distribution (Eq. 4). The second stage of the model, presented in Eq. 5, refers to the individual level, where log-normality around a population mean was assumed for all parameters. The third stage of the model consisted of distributional assumptions of the population level, *i.e.*, priors placed on the population parameters (Eq. 6).

Suppose that N blood samples, indexed by j , were drawn for each of the M individuals indexed by i . Let the j th measurement of individual i be denoted by y_{ij} , the associated time by t_{ij} and the related individual dose by D_i . Denote by θ_i the p -dimensional vector of drug-related parameters of individual i , by σ^2 the residual variance of the model, and by $f(\cdot)$ the structural model. The hierarchical model is then:

First stage

$$p(\log(y_{ij})|\theta_i, \sigma^2) \propto N(\log(f(D_i; t_{ij}; \theta_i)), \sigma^2) \quad (4)$$

Second stage

$$p(\theta_{tr_i}|\mu, \Omega) = MVN_p(\mu, \Omega) \quad (5)$$

$$\theta_i = e^{\theta_{tr_i}}$$

Third stage

$$p(\mu) = MVN_p(\eta, H)$$

$$\Omega = \text{diag}(s) \cdot C \cdot \text{diag}(s)$$

$$p(s) = N_{half}(0, 1) \quad (6)$$

$$p(C) = LKJ(a)$$

$$p(\sigma) = N_{half}(0, 1)$$

where MVN_p stands for the p -variate normal distribution, Ω is the population variance-covariance matrix (size 3×3), s a vector of dimension 3, N_{half} is the half-normal distribution, and C a prior correlation matrix. As suggested in [27], a Lewandowski-Kurowicka-Joe (LKJ) prior was assigned to the correlation matrix [28]. The hyper-parameter values were:

$$\begin{aligned}
\tilde{\eta} &= (70.8, 1, 3.34) \\
\tilde{\eta}_{std} &= (62.5, 1, 0.44) \\
\eta &= \log\left(\frac{\tilde{\eta}^2}{\sqrt{\tilde{\eta}_{std}^2 + \tilde{\eta}^2}}\right) \\
H_{mn} &= \begin{cases} \log\left(\frac{\tilde{\eta}_{std}^2}{\tilde{\eta}^2} + 1\right), & \text{for } m = n \\ 0, & \text{otherwise} \end{cases} \\
a &= 5
\end{aligned} \tag{7}$$

The standard deviations, as well as the prior mean of the clearance were identical to the ones used in [19] for comparison reasons. The above approach describes a centered parameterization of the statistical model. Besides this most common parameterization, a non-centered parameterization was also investigated to assess its impact on results. Model parameterization, especially in hierarchical structures, has a decisive role in the performance of the sampler [29]. The centered form can be converted to the non-centered one through the transformation given by Eq. 8. The non-centered parameterization was used to model the random effects, i.e. the individual level.

$$\begin{aligned}
X &\sim N(\mu, \sigma^2) \rightarrow X = \mu + \sigma \cdot y \\
y &\sim N(0, 1)
\end{aligned} \tag{8}$$

The above model was slightly simplified for the *GNU MCSim*, which does not offer MVN or LKJ sampling. In that case, Eq. 5 first line was simplified as:

$$p(\theta_{tr} | \mu, \Omega) = N(\mu, \Omega) \tag{9}$$

while Eq. 6 became:

$$\begin{aligned}
p(\mu) &= N(\eta, H) \\
\Omega &= N_{half}(0, 1) \\
p(\sigma) &= N_{half}(0, 1)
\end{aligned} \tag{10}$$

Software Tools for Bayesian Inference

The first Bayesian tool used for obtaining the posterior distribution of the parameters was *Stan*, which is a statistical modeling language that can be used through many interfaces such as *R*, *Python* and *Matlab* and includes MCMC sampling, variational inference and penalized maximum likelihood estimation with optimization. Its major difference with other MCMC tools is that it avoids random walk. This is managed through the utilization of Hamiltonian dynamics for the proposal. Imagine a particle with some position and momentum moving over time in a frictionless space. The

model parameters correspond to the position vector and an auxiliary vector is also introduced, representing the momentum vector. The potential and kinetic energy related to the position and momentum vector constitute the total energy (*Hamiltonian*), which remains constant over time due to the absence of friction. The position and momentum of the particle can be calculated by solving the Hamilton's equations (a system of partial derivatives) [30]. In each iteration, the particle is given a random momentum and is left to move in space for a certain amount of time. The position at the end of the trajectory followed by the particle, composes the proposal of the algorithm. The solution of the Hamiltonian system requires a discretization, so the trajectories are characterized by a step size and a number of steps. These parameters are crucial for the success of the algorithm and need to be tuned by the user in static HMC implementations. If the trajectory is too small, the algorithm will use a very small pace or will not be able to explore the entire parametric space. On the contrary, if the trajectory is too long, it will backtrack and waste computational resources [31]. *Stan* uses the No-U-Turn sampler (NUTS), a dynamic sampler that automatically tunes these parameters (together with others like the Mass matrix) through the warm-up period.

The nature of the HMC algorithm allows *Stan* to offer some very helpful diagnostic tools, which generate alerts in the process of model building that are reported at the end of each run. These diagnostic tools reveal pathologies that are model-inherent but cannot be recognized by tools that do not apply HMC. Divergence transitions is one diagnostic tool that indicates pathological neighborhoods of the posterior which are not sufficiently explored by the simulated Hamiltonian trajectories. In this case the Markov chains do not completely explore the posterior and the MCMC estimators are biased. Another tool is the maximum tree depth (*check_tree_depth* in *Stan*) that is related with NUTS. A non-zero value of this diagnostic metric suggests that the posterior includes regions of very low curvature, which is another type of pathology [32]. Finally, reviewing the energy Bayesian Fraction of Missing Information (E-BFMI) (*option check_energy* in *Stan*) informs the user about problems in the pace of exploration. In general, if any of these tools suggests that there are pathologies, the user can rebuild the statistical model, and change parameterizations or priors. For divergent transitions, there is a chance of observing false positives, in which case an increase of the step size (*adapt_delta* in *Stan*) will reduce them to zero.

Another major advantage of the HMC algorithm is that it informs the proposal about the target distribution through numerical approximation of its gradient. This is in contrast to usual MCMC methods where the proposal is unrelated to the target distribution (posterior). Therefore, HMC performs better in high dimensional problems where the posterior has high curvature compared to other MCMC methods which

get stuck or tend to explore the posterior at a very low rate. *Stan* offers both static HMC and NUTS. The reader is referred to [30, 31, 33] for a detailed description of HMC and to [18] for an in-depth review of the NUTS sampler. The interface used in this study was *RStan* v.2.17.3. For sampling, NUTS was preferred to (static) HMC.

Stan offers three approaches for solving a system of differential equations: numerical solution via a non-stiff solver (*integrate_ode_rk45*), a stiff solver (*integrate_ode_bdf*) and the matrix exponential method (*matrix_exp*) [27]. In the latter case, the system must be linear (which is the case for the diazepam model) and has to be cast in a matrix form, A , whose solution is given by Eq.11:

$$\begin{aligned} x(t) &= e^{tA} \cdot x_0 \\ e^{tA} &= \sum_{k=0}^{\infty} \frac{1}{k!} (tA)^k \end{aligned} \quad (11)$$

The second Bayesian tool was *GNU MCSim* [16, 34]. It offers a simulation language, Monte Carlo and MCMC sampling, tempered MCMC sampling, stochastic optimization and experimental design optimization. User-defined models are compiled, as in *Stan*. Numerical integration of differential equations can be performed by the very efficient stiff ODE solvers *Lsodes* or *Cvodes* [35]. We used here *Lsodes* (for stiff ODEs, with relative and absolute tolerances of 10^{-6}) and the simple component by component Metropolis-Hasting sampler. The (Gaussian) proposal kernels are adapted automatically to yield a satisfactory jump frequency for each sampled parameter.

Model specifications

The models on both platforms were run with 4 chains with 5,000 samples each. The model that was built with *GNU MCSim* had a warm-up period (initial samples that are discarded, also termed burn-in period) of 1,000 iterations, while the model in *Stan* used 400 warm-up iterations. The role of warm-up period is different in the two software tools: in *GNU MCSim*, its length is chosen by the user and simply reduces the output size; the user has to check convergence to the target distribution independently. In *Stan* the warm-up period spans both convergence and tuning of the HMC parameters if NUTS is used. The warm-up period in *Stan* is more time consuming per iteration than the sampling period because of the adaptation of computational parameters, such as the target acceptance rate (termed *adapt_delta*), which are then used in all sampling iterations.

Both models were run on a computer with an Intel Core i7-8700 (3.2 GHz), 16 GB RAM and windows 10 OS. Their computational efficiency was measured in terms of effective sample size (N_{eff}), an estimate of the number of independent

samples drawn. Smaller effective sample sizes lead to less accurate posterior distribution estimates. Since each parameter has a different N_{eff} , two metrics were used: the average effective samples over all parameters (85 parameters in total: 3 population means, 3 population variances, 9 correlation coefficients, 69 individual parameters, and model error) and the minimum number of effective samples (*i.e.*, N_{eff} for the least efficiently sampled parameter). There are also several ways to calculate N_{eff} : to be consistent, the same *RStan* function *monitor()*, which uses both cross-chain and within-chain calculations [36], was used to obtain N_{eff} for both outputs.

Results

Comparison on computational efficiency

The two MCMC samplers were compared with respect to the posterior parameter distributions they yielded, as well as their computational efficiency. Efficiency results are presented in Table 3, where time refers to the computer time (in seconds) taken to compute all 4 chains in parallel. It is evident that the NUTS sampler is more efficient than the sampling method used in *GNU MCSim*. It is indicative that in all but two parameters the *Stan* output had $N_{\text{eff}} = N_{\text{sample}}$, which demonstrates the low autocorrelation in the produced chains. Nevertheless, the execution time of *GNU MCSim* is much shorter, resulting in a higher overall computational efficiency of produced effective samples per unit time. Note, however, that the model assumed by *GNU MCSim* is somewhat simpler than the *Stan* model which uses multivariate distributions in the upper levels of the hierarchy.

Table 3 Comparison of computational efficiency between *Stan* and *GNU MCSim*.

Platform	time (s)	$\overline{N_{\text{eff}}}$	$\overline{N_{\text{eff}}}/\text{s}$	$N_{\text{eff}_{\text{min}}}$	$N_{\text{eff}_{\text{min}}}/\text{s}$
<i>Stan</i>	5400	19926	3.69	14256	2.64
<i>GNU MCSim</i>	160	2533	15.8	1221	7.63

With regard to solving the system of linear differential equations in *Stan*, the most robust method proved to be the stiff solver (*integrate_ode_bdf*), with relaxed tolerances. Specifically, the lowest time record of 5,400 s for 5,000 samples, was obtained for relative tolerance, absolute tolerance and maximum number of steps equal to 10^{-4} , 10^{-2} and 10^5 respectively. Use of *Stan*'s default values (10^{-6} , 10^{-6} and 10^{-6}) deteriorated the performance and registered an execution time around 42,100 s. On the other hand, the matrix exponential method (*matrix_exp*) was much slower, recording 31,000 s execution time. Use of the non-stiff solver (*integrate_ode_rk45*)

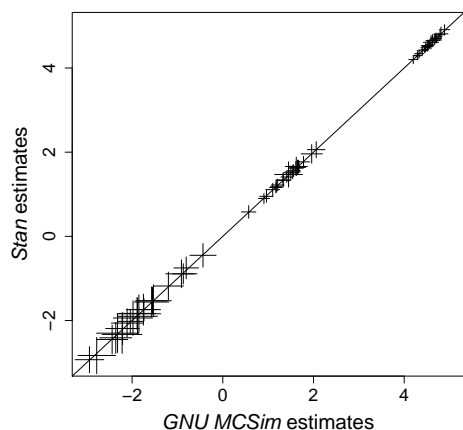


Fig. 2 Comparison of individual parameter estimates between *GNU MCSim* and *Stan*.

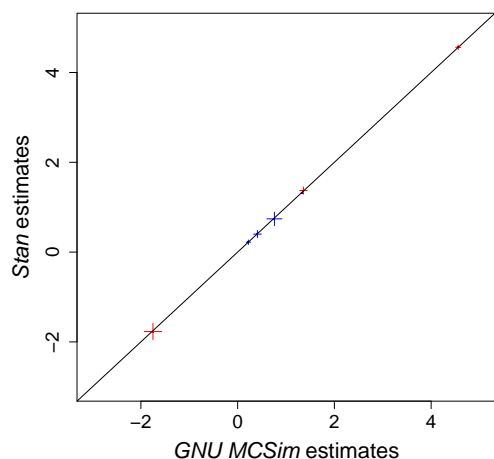


Fig. 3 Comparison of population estimates between *GNU MCSim* and *Stan*. The red crosses represent the mean population estimates while the blue crosses represent the population variance estimates.

was extremely slow and did not yield any results after a significant amount of time. Note that in [19] computation times were 7,200 s with NONMEM and 601,200 s with WINBUGS, but on a different computer, clocked at 450 MHz.

Comparison on Parameter Estimation

The parameter estimates obtained with the two software tools were practically identical. This can be seen in Fig. 2, where the log-transformed estimates of the individual parameters of *Stan* are plotted against those of *GNU MCSim* and all fall on the identity line. The size of the crosses quantifies estimation uncertainty, which is virtually the same for both programs. Fig. 3 depicts the population estimates, where the red crosses represent the population means and the blue crosses

represent the population variances. Detailed estimation results can be found in Appendix B.

Traceplots were used to graphically inspect convergence of the simulated Markov chains. Fig. 4 presents the traceplots of the population parameters in *Stan* (*GNU MCSim* produced similar traceplots). The absence of patterns and regions where the chains remain static as well as the similar behaviour for all chains suggest that convergence has been achieved. However, graphical diagnostics of convergence are usefully complemented by quantitative criteria. One such criterion is the potential scale reduction factor (\hat{R}), which suggests that convergence has been practically reached if all model parameters have $\hat{R} < 1.2$ [37]. In this case, split \hat{R} was used, which splits the chains in half (after discarding the warm-up samples) and checks both mixing and stationarity. All parameters had a value of $\hat{R} < 1.005$, with both *Stan* and *GNU MCSim*. The HMC-related diagnostic tools did not reveal any pathology, which further enhanced the validity of the produced results.

Fig. 5 illustrates the posterior vs the prior distributions of the population means. It is evident that the information contained in the data reduced significantly the prior uncertainty of CL_{int} and fKp . On the contrary, the variance of Kp_{AD} was considerably increased by the clinical data integration. So in fact, our prior on that parameter was too precise. These results also mean that the parameters chosen were identifiable, given the available data.

Regarding the model fit to the data, Figs. 6 and 7 present the individual plots for all patients. Most of the patients' data were adequately described by the PBPK model.

Fig. 8 and 9 present visual predictive check (VPC) plots corresponding to the two models, which assess their efficiency in describing inter-individual variability at the population level. The circles represent the data from all subjects, the green dotted line depicts the 5th percentiles of the data, the solid red line the median and the dotted blue line the 95th percentiles. These lines were obtained by ordering all observations at each time instance and extracting the respective quantiles. The colored ribbons were created by simulating 1,000 sets of 23 virtual patient with physiological characteristics identical to the real patients. For the *Stan* model, in each set, population means (μ) and variances (Ω) for CL_{int} , fKp , and Kp_{AD} were drawn from their posterior distributions and then, for each virtual patient, a triplet of individual parameters was drawn from $MVN_p(\mu, \Omega)$. The same procedure was followed in the *GNU MCSim* VPC plot, with the difference that both the population and individual parameters were drawn from univariate distributions. In each set, the 5th, 50th and 95th prediction percentiles were calculated and by ordering the percentiles produced by all 1,000 sets, the 95 % confidence interval (CI) was calculated for each of the three percentiles. Thus the pink ribbon on the bottom of the VPC represents the 95 % CI of the 5th percentile of the

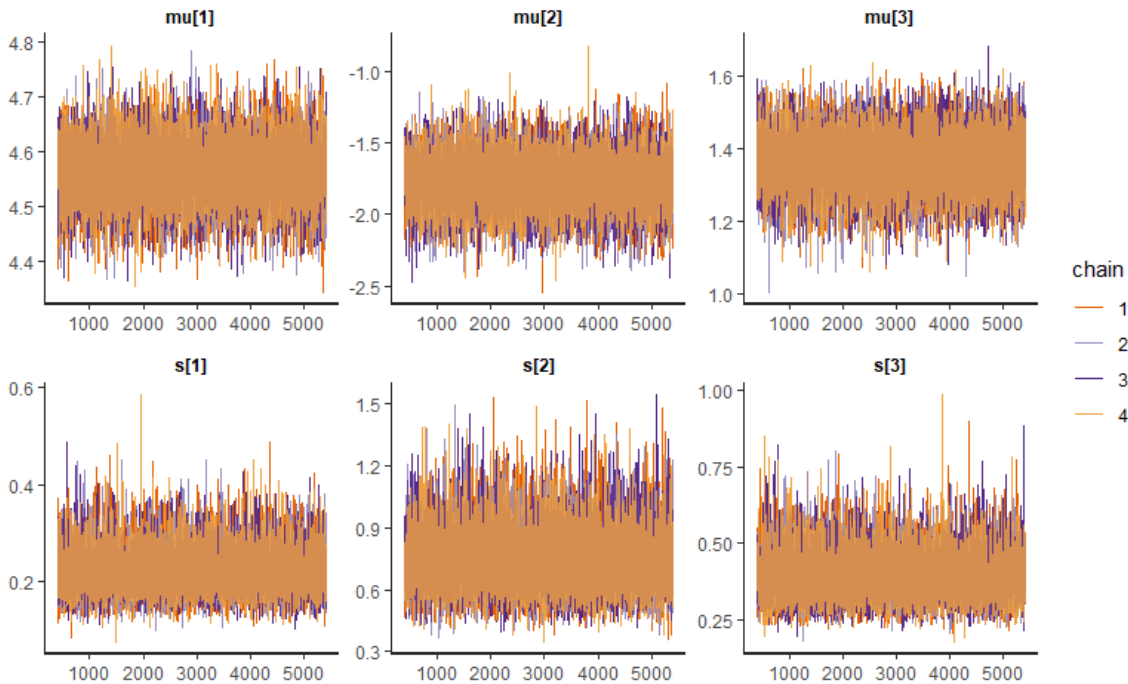


Fig. 4 Traceplots of the three population parameters ($\mu[1] = \log(CL_{int})$, $\mu[2] = \log(fKp)$, $\mu[3] = \log(Kp_{AD})$ and $s[1]$, $s[2]$, $s[3]$ the corresponding population variances)*.

model, the gray ribbon represents the 95 % CI of the 50th percentile of the model and finally the green ribbon depicts the 95 % CI of the 95th percentile of the model.

Similarly to parameter estimation results, the *Stan* and the *GNU MCSim* VPC plots are almost identical, which implies that the population structure is correctly modeled when an independence assumption is made at the population level. In both plots, the 5th, 50th and 95th percentiles of the observations fall within the 95% CI of the respective percentiles of the model predictions, which illustrates that the models describe adequately inter-individual variability.

Choice of Parameterizations and Priors

As stated above, two different model parameterizations were tested in *Stan*; the centered and the non-centered one. The centered parameterization proved to be more efficient, as it was faster and produced a higher N_{eff} metric for the estimated parameters. More specifically, computation time was 11,000 s for the non-centered model *vs.* 5400 s for the centered one. Different computational time affected the $\frac{N_{\text{eff}}}{s}$ metric, which for the non-centered model was $1.73 \frac{N_{\text{eff}}}{s}$, i.e. less than half of this metric for the centered model. Increased computational time for the non-centered model was due to the extra parameters at the individual level. If the data did not contain significant information, then the original posterior would have a geometry with high curvature and the

non-centered parameterization would result in simpler geometries [39], contributing to a speed-up of the posterior exploration. For the studied combination of PBPK model, data and statistical model, the resulting geometries did not include any regions of high curvature (which was suggested by several graphical inspections) and this is most probably the reason why the non-centered model was slower. In addition, the parameter estimates of the centered and non-centered models were all the same with the exception of fKp , which differed in both the population and individual level. More specifically, the non-centered model considered a higher fKp , which resulted in slightly worse individual plots and VPC plots.

We also tested the impact of assigning different distributions to the correlation matrix and to the standard deviation of the population level (denoted by C and s respectively in Eq. 6). For s , we compared the results obtained when using a half-normal *vs.* a half-Cauchy distribution (see Fig. 10). The half-normal distribution was tested with scale 1, 10, 100 and the half-Cauchy with scale 2.5 and 5. No differences can be seen in the posterior means, but posterior variances are affected. Naturally, the flatter the prior, the higher the posterior uncertainty.

Regarding C , many values for the shape parameter(a) of the LKJ distribution were tested. The impact of the shape

* The plots correspond to the *Stan* model. Similar plots were generated from the *GNU MCSim* model.

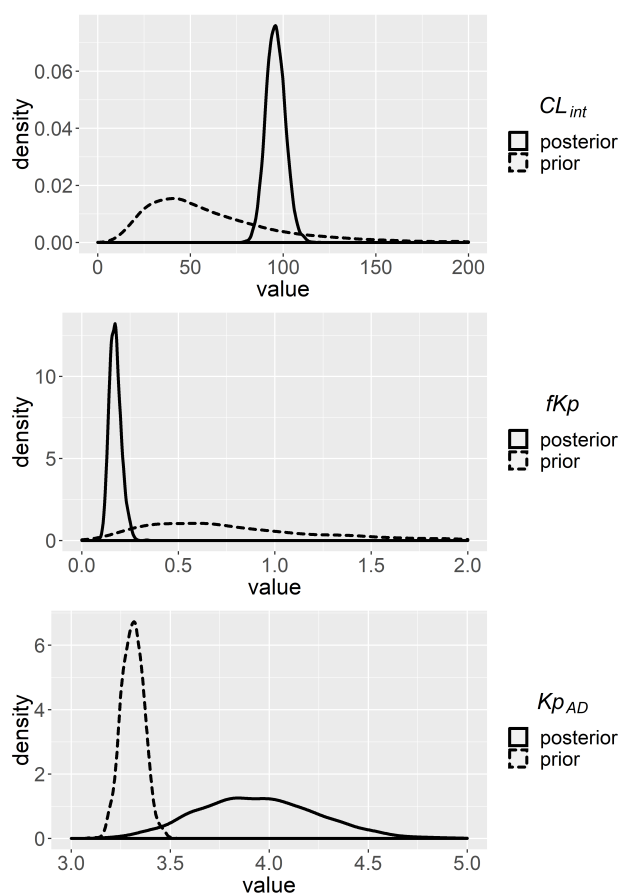


Fig. 5 Posterior vs prior densities of the three population parameters*.

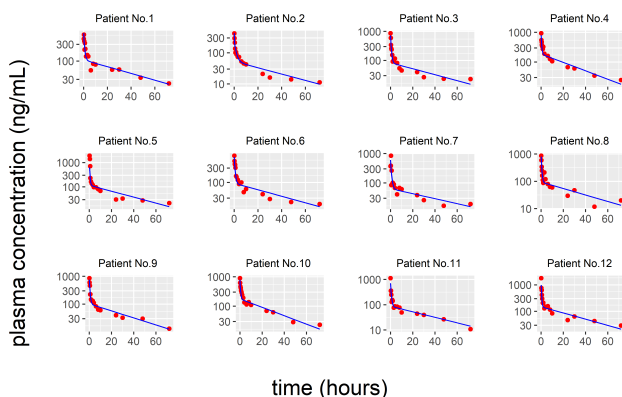


Fig. 6 Individual plots presenting the fitted model and the observations for the first 12 patients*.

parameter can be seen in Fig. 11, where the plot presents the prior distribution assigned to the non-diagonal element $C[1, 2]$ for each correlation matrix. As a increases, the prior assigns less correlation between parameters and gets closer to the unit correlation matrix, while for values equal or less than 1, it yields a distribution with shape close to uniform. The effect of the prior selection on the posterior of the correlation matrix can be seen in Fig. 12. The posterior is heavily dependent on the prior shape, but the uncertainty of the es-

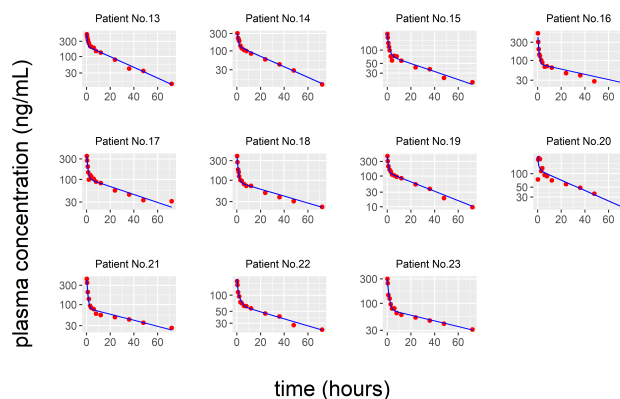


Fig. 7 Individual plots presenting the fitted model and the observations for the last 11 patients*.

timation is such that it does not allow any robust inference; zero correlation among parameters cannot be ruled out. The only suggestion that can be supported by the graph is the higher correlation between fKp and Kp_{AD} with respect to the other two correlation pairs, which was expected beforehand. Moreover, the different priors did not affect significantly any of the remaining parameters, only small variations were observed. Examination of the individual plots and the VPC graph for each scenario did not reveal significant changes either. This comes in good agreement with the similarity between the results of the multivariate model followed in *Stan* and the univariate produced by *GNU MCSim*. The results presented above refer to $a = 15$, because that shape value produced an output that was found to be slightly more computationally efficient, as seen in Appendix C.

Finally, another option for building the statistical model is the Cholesky factorization: One can declare the correlation matrix to be a *cholesky_factor_corr* and then adopt *multi_normal_cholesky* instead of *multi_normal* for the 2nd stage of the model. This approach yielded the same results but the resulting distribution of the correlation matrix had a diagonal that slightly deviated from the unit diagonal and this is the reason why it was not preferred.

Discussion

We re-analyzed with two software programs diazepam population kinetic data with a PBPK model in a purely Bayesian framework. We chose to infer on diazepam metabolic clearance, its the adipose tissue-to-plasma partition coefficient, and a scaling factor multiplying the rest of the Kp 's. This choice was based on the fact that clearance had the least informed prior, the adipose tissue had the slowest distribution and the highest partition coefficient *a priori* (making it an important storage compartment), and that all other partition coefficients should be scaled with it for coherence (prior information was as much about the relative values of the

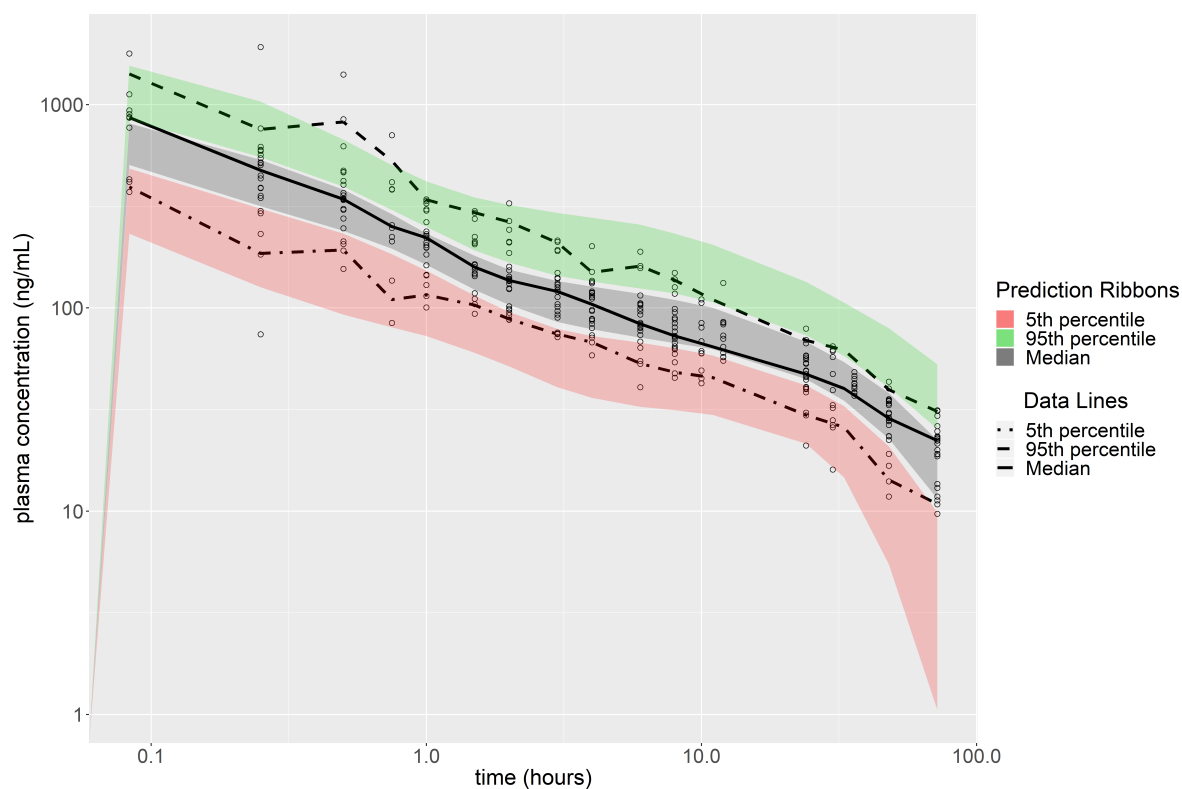


Fig. 8 Visual predictive check of inter-individual variability produced by the *Stan* model. Circles represent the data on all individuals (note the log-log scale). The Ribbons correspond to model predictions (see text).

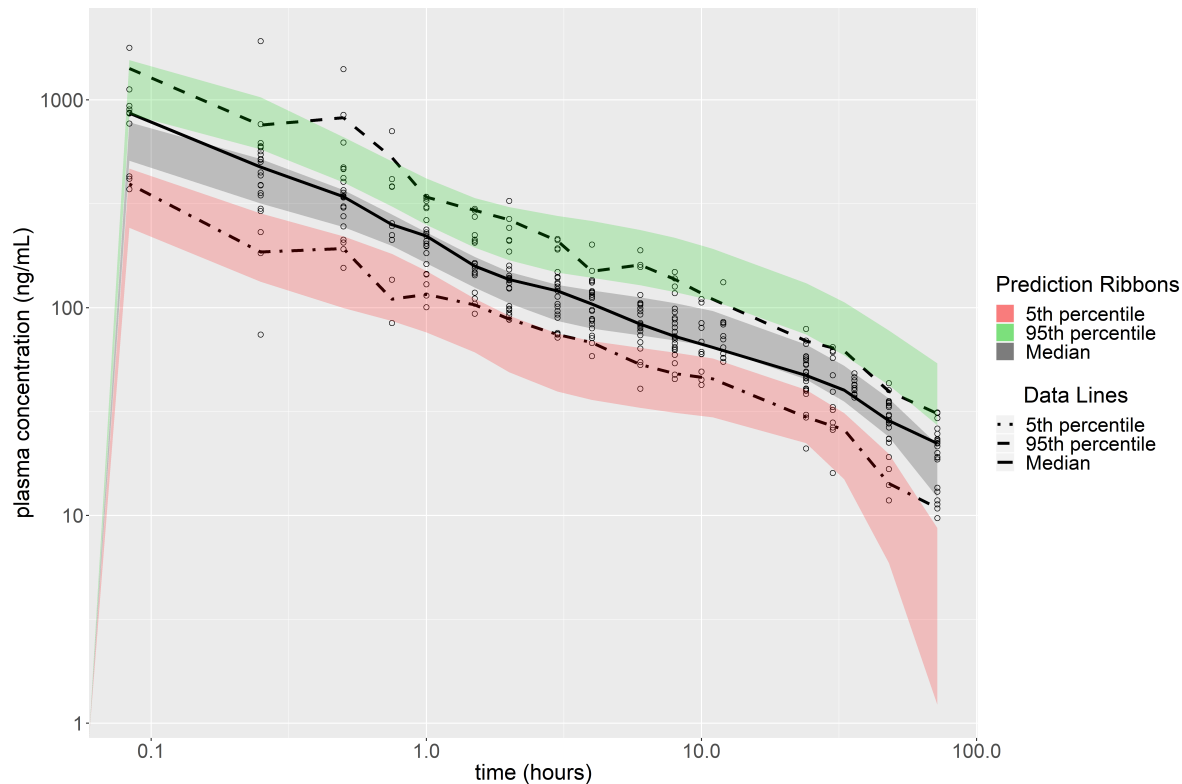


Fig. 9 Visual predictive check of inter-individual variability produced by the *GNU MCSim* model. Circles represent the data on all individuals (note the log-log scale). The Ribbons correspond to model predictions (see text).

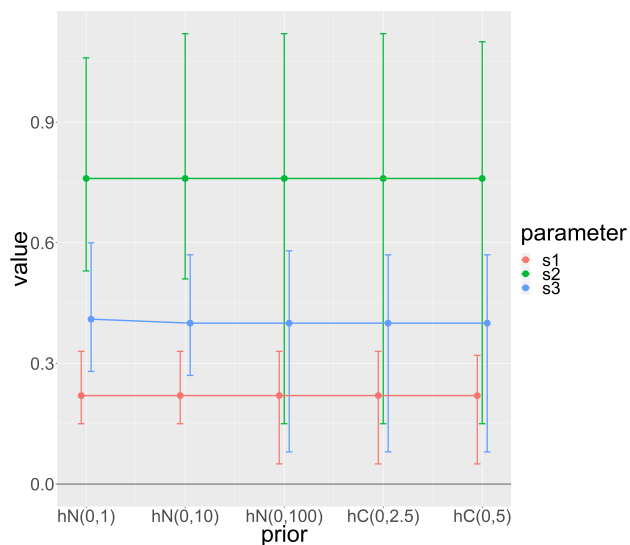


Fig. 10 Impact of different priors on the standard deviation of the population level. The x-axis contains the priors, where hN and hC encode the half-normal and half-Cauchy distribution respectively.

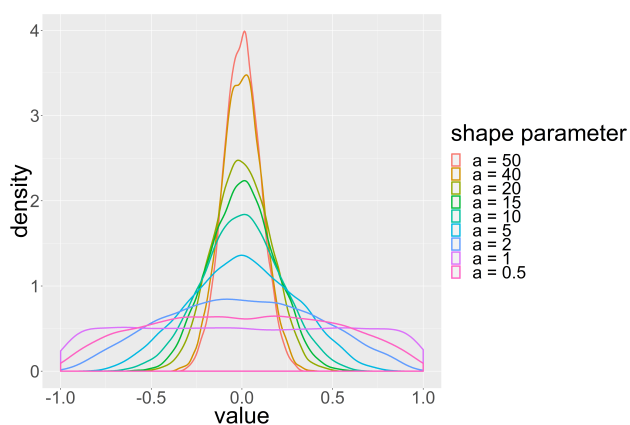


Fig. 11 Impact of shape parameter α on the LJK prior. The appearing distribution is applied on each non-diagonal element of the correlation matrix C .

Kp 's than about their magnitude). The small number of selected parameters is also directly connected to the structural and statistical identifiability problems potentially affecting complex PBPK models [40,41]. We exercised our judgment here, and another approach would be to select the most influential parameters through sensitivity analysis [42]. We did try alternative choices. For instance, we added the rest of the body Kp to the parameters to infer on (prior information about it is weak, as there was no such parameter in the rat model), but we gained no insight about its true value (its prior remained unchanged).

The models used in this study have two major differences with the approach followed in Langdon et al. [19]. The first is the use of a covariate model for body mass-dependent parameters in order to partly explain inter-individual variability. The second concerns the choice of the parameters

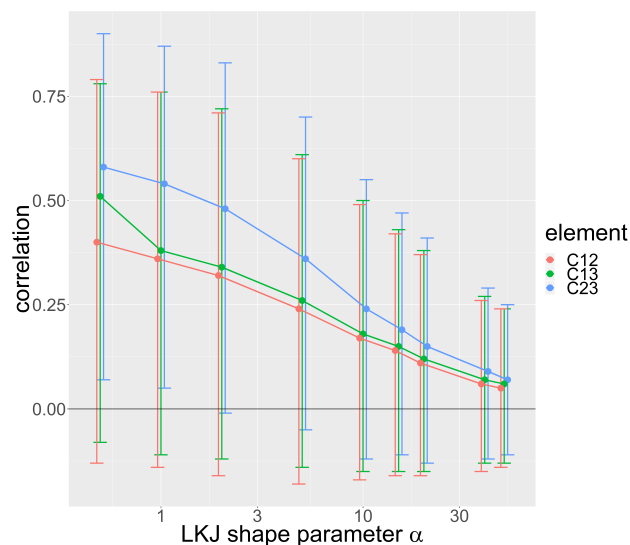


Fig. 12 Impact of the prior shape selection on the posterior of the correlation matrix. The bars represent the 95% credible interval. The LKJ shape parameter α values tested were: 0.5, 1, 2, 5, 10, 15, 20, 40, 50.

for the statistical model. These two differences are responsible for obtaining different results. More specifically, CL_{int} was found to be 96 L/h instead of about 75 L/h in [19] and all Kp s had different posterior estimations (see Table 3 in [19]). Almost all partition coefficients in [19] had posterior partition coefficient values lower than their (rat) prior central estimates. In particular, the partition coefficient of the "carcass" (a large compartment by volume) had a vague prior and very low posterior values (around 0.07) in [19], much different from the others, which were typically between 0.5 and 1. Our partition coefficient estimates are all lowered equally by the scaling coefficient fKp , and therefore they are more coherent, while avoiding identifiability problems. Still, our estimate of the partition coefficient for adipose tissue, Kp_{AD} , is similar to the one reported in [19], which indicates that this parameter is reasonably identified from the data.

Several parameterizations of the hierarchical model were also tested and we retained the one giving the most stable results. More specifically, we tried different prior combinations for the correlation matrix and the standard deviation of the population level. The shape parameter of the LKJ prior had a substantial impact on the posterior of the correlation matrix, which, however, did not affect the posterior of any other parameter and, therefore, the goodness-of-fit of the model on both individual and population levels. Regarding the priors assigned to the population standard deviations, they did not affect the produced posterior. The comparison of the centered and non-centered model revealed that the centered model was more efficient. For the specific combination of model, data and priors, the posterior of the centered model proved to be easily explored. This is not always the case. Non-centered parameterizations are supposed to per-

form better in hierarchical structures where the data does not carry strong information [39]. The problem arises when small variations of the bottom layer of the structure introduce high variations at the top levels. That leads to a posterior distribution of high curvature that is difficult to explore. In such cases, a non-centered parameterization introduces transformations between layers, ending up with less correlated variables and a simpler posterior geometry.

One of the main advantages of *Stan* lies in its HMC-specific diagnostic tools. A user can receive significant help by consulting the output of the diagnostics. If the posterior contains pathologies, then the user is alerted and is advised to reparameterize the model or change priors. These pathologies cannot be located by global diagnostics, so users of other MCMC methods cannot be informed by potential biases included in the produced fits.

However, for models based on differential equations, such as PBPK models, the algorithms implemented in *Stan* appear quite slow. This is in part inherent to the HMC algorithm: The Jacobian of the ODE system, with respect to the parameters sampled, must be computed at each step, either by finite differencing or by integration of additional differential equations. Since integration time typically increases quadratically with the number of ODEs, HMC incurs a high computational cost when dealing with ODE models. In this case, straightforward MCMC simulations are a competitive alternative. But for simpler algebraic models *Stan* is usually faster. Moreover, it performs better in cases where the posterior geometry is difficult, with regions of high curvature, where algorithms like Metropolis-Hastings explore in a slow pace or even get stuck [33].

Note again that *GNU MCSim* does not offer multivariate sampling and therefore assumed independence, at the second and third levels of the hierarchy, between the three parameters sampled. That must have contributed to better computational performance. However, more importantly, the results obtained under either independence or correlation assumptions were practically the same (see Figs. 3 and 2). Therefore, it seems superfluous to use multivariate distributions for population PBPK models. This is probably due to the mechanistic nature of the PBPK parameters which do not lump several processes into a unique value. This can be taken as a further advantage of PBPK model. The parameters of empirical models (*e.g.* classical compartmental models) get more easily entangled because they share lumped processes. In that case, more complex multivariate models are required for correct inference.

Conclusions

Two Bayesian software inference tools namely, *Stan* and *GNU MCSim*, were compared in this work on updating the parameters of a Diazepam population PBPK model. The two

software tools produced almost identical parameter estimates and both exhibited very good individual and population fits. Taking into account that *GNU MCSim*'s is based on univariate modeling, it was shown that multivariate distributional assumptions of *Stan* are superfluous on PBPK parameter estimation problems where there is either strong prior information or many data available at the individual level. It was also observed that centered parameterization is better suited than non-centered parameterization in this kind of parameter estimation problems. The goodness-of-fit of the model was not affected by the change in the posterior results of the correlation matrix, which was completely dependent on the prior, indicating the lack of information about parameter correlation in the data. *Stan* was very efficient in terms of the effective sample size $\overline{N}_{\text{eff}}$, which in all but two parameters was equal to the total number of samples. However, *GNU MCSim* converged much faster and outperformed *Stan* in the overall computation efficiency $\overline{N}_{\text{eff}}/s$ metric. The model building phase was found to be more user-friendly for *Stan* due to its HMC-specific diagnostic tools that alert the user for potential biases that are included in the final model.

Acknowledgements We would like to thank Gueorgieva I. for granting us access to the Diazepam data. H. Sarimveis and F. Bois acknowledge financial support by OpenRiskNet (Grant Agreement 731075), a project funded by the European Commission under the Horizon 2020 Programme.

References

1. Macheras P, Iliadis A (2016) Modeling in Biopharmaceutics, Pharmacokinetics and Pharmacodynamics: Homogeneous and Heterogeneous Approaches. Springer, New York.
2. Tsamandouras N, Rostami-Hodjegan A, Aarons L (2015) Combining the 'bottom up' and 'top down' approaches in pharmacokinetic modelling: fitting PBPK models to observed clinical data. *Br J Clin Pharmacol* 79(1):48-55. <https://doi.org/10.1111/bcp.12234>
3. Clewell HJ, Clewell RA, Andersen ME (2011) Physiologically-Based Pharmacokinetic (PBPK) Modeling and Risk Assessment. In: Nriagu J (ed) *Encyclopedia of Environmental Health*. Elsevier, pp 536-570. <https://doi.org/10.1016/B978-0-444-52272-6.00599-7>
4. Nestorov I (2003) Whole body pharmacokinetic models. *Clin Pharmacokinet* 42(10):883-908. <https://doi.org/10.2165/00003088-200342100-00002>
5. Edginton AN, Theil FP, Schmitt W, Willmann S (2008) Whole body physiologically-based pharmacokinetic models: their use in clinical drug development. *Expert Opin Drug Metab Toxicol* 4(9):1143-1152. <https://doi.org/10.1517/17425255.4.9.1143>
6. Wendling T, Dumitras S, Ogungbenro K, Aarons L (2015) Application of a Bayesian approach to physiological modelling of mavoglurant population pharmacokinetics. *J Pharmacokinet Pharmacodyn* 42(6):639657. <https://doi.org/10.1007/s10928-015-9430-4>
7. Zhuang X, Lu C (2016) PBPK modeling and simulation in drug research and development. *Acta Pharmaceutica Sinica B* 6(5):430440. <https://doi.org/10.1016/j.apsb.2016.04.004>
8. Gelman A, Bois FY, Jiang JM (1996) Physiological pharmacokinetic analysis using population modeling and informative prior dis-

- tributions. *JASA* 91(436):1400-12. <https://doi.org/10.2307/2291566>
9. Wakefield J (1996) The Bayesian analysis of population pharmacokinetic models. *J Am Stat Assoc* 91(433):6275. <https://doi.org/10.2307/2291383>
10. Bois FY, Jamei M, Clewell HJ (2010) PBPK modelling of inter-individual variability in the pharmacokinetics of environmental chemicals. *Toxicology* 278(3):256267. <https://doi.org/10.1016/j.tox.2010.06.007>
11. Krauss M, Schuppert A (2016) Assessing interindividual variability by Bayesian-PBPK modeling. *Drug Discovery Today: Disease Model* 22:15-19. <https://doi.org/10.1016/j.ddmod.2017.08.001>
12. Krauss M, Tappe K, Schuppert A, Kuepfer L, Goerlitz L. (2015) Bayesian Population Physiologically-Based Pharmacokinetic (PBPK) Approach for a Physiologically Realistic Characterization of Interindividual Variability in Clinically Relevant Populations. *PLoS One* 10(10):e0139423. <https://doi.org/10.1371/journal.pone.0139423>
13. Mezzetti M, Ibrahim JG, Bois FY, Ryan LM, Ngo L, Smith TJ (2003) A Bayesian compartmental model for the evaluation of 1,3-butadiene metabolism. *J R Stat Soc Ser C* 52:291-305. <https://doi.org/10.1111/1467-9876.00405>
14. Zurlinden TJ, Reisfeld B (2016) Physiologically based modeling of the pharmacokinetics of acetaminophen and its major metabolites in humans using a Bayesian population approach. *Eur J Drug Metab Pharmacokinet* 41(3):267-80. <https://doi.org/10.1007/s13318-015-0253-x>
15. Geyer CJ (2011) Introduction to Markov Chain Monte Carlo. In: Brooks S, Gelman A, Jones GL, Meng XL (eds) *Handbook of Markov Chain Monte Carlo*, Vol. 2. CRC Press, Boca Raton pp 348
16. Bois FY (2009) GNU MCSim: Bayesian statistical inference for SBML-coded systems biology models. *Bioinformatics* 25(11):1453-1454. <https://doi.org/10.1093/bioinformatics/btp162>
17. Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017) Stan: A probabilistic programming language. *J Stat Softw* 76(1). <https://doi.org/10.18637/jss.v076.i01>
18. Hoffman MD, Gelman A (2014) The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J Mach Learn Res* 15:1593-1623.
19. Langdon G, Gueorguieva I, Aarons L, Karlsson M (2007) Linking preclinical and clinical whole-body physiologically based pharmacokinetic models with prior distributions in NONMEM. *Eur J Clin Pharmacol* 63(5):485-498. <https://doi.org/10.1007/s00228-007-0264-x>
20. Gueorguieva I, Aarons L, Rowland M (2006) Diazepam pharmacokinetics from preclinical to Phase I using a Bayesian population physiological model with informative prior distributions in WINBUGS. *J Pharmacokinet Pharmacodyn* 33(5):571594. <https://doi.org/10.1007/s10928-006-9023-3>
21. Greenblatt DJ, Allen MD, Harmatz JS, Shader RI (1980) Diazepam disposition determinants. *Clin Pharmacol Ther* 27(3):301312.
22. Gueorguieva II, Nestorov IA, Rowland M (2004) Fuzzy Simulation of Pharmacokinetic Models: Case Study of Whole Body Physiologically Based Model of Diazepam. *J Pharmacokinet Pharmacodyn* 31(3):185213. <https://doi.org/10.1023/B:JOPA.0000039564.35602.78>
23. Leggett RW, Williams LR, Eckerman KF (1995) A blood circulation model for reference man. *Health Phys* 69(2):187-201
24. Nestorov IA (2001) Modelling and simulation of variability and uncertainty in toxicokinetics and pharmacokinetics. *Toxicol Lett* 120(1-3):411-420.
25. Luecke RH, Wosilait WD, Slikker WJr, Young JF, Pearce BA (2007) Postnatal Growth Considerations for PBPK Modeling. *J Toxicol Environ Health A* 70(12):1027-1037. <https://doi.org/10.1080/15287390601172056>
26. Mould DR, Upton RN (2013) Basic Concepts in Population Modeling, Simulation, and Model-Based Drug Development-Part 2: Introduction to Pharmacokinetic Modeling Methods. *CPT Pharmacometrics Syst Pharmacol* 2(4): e38. <https://doi.org/10.1038/psp.2013.14>
27. Stan Development Team (2018) Stan Modeling Language Users Guide and Reference Manual, Version 2.18.0. <http://mc-stan.org>
28. Lewandowski D, Kurowicka D, Joe H (2009) Generating random correlation matrices based on vines and extended onion method. *J Multivar Anal* 100(9):1989-2001. <https://doi.org/10.1016/j.jmva.2009.04.008>
29. Papaspiliopoulos O, Roberts GO, Skld M (2007) A General Framework for the Parametrization of Hierarchical Models. *Sta Sci* 22(1):59-73
30. Neal RM (2011) MCMC using Hamiltonian dynamics. In: Brooks S, Gelman A, Jones GL, Meng XL (eds) *Handbook of Markov Chain Monte Carlo*, Vol. 2. CRC Press, Boca Raton pp 113162
31. Monnahan CC, Thorson JT, Branch TA (2018) Faster estimation of Bayesian models in ecology using Hamiltonian Monte Carlo. *Methods Ecol Evol* 8:339-348. <https://doi.org/10.1111/2041-210X.12681>
32. Betancourt M (2016) Diagnosing suboptimal cotangent disintegrations in Hamiltonian Monte Carlo. arXiv:1604.00695
33. Betancourt M (2017) A Conceptual Introduction to Hamiltonian Monte Carlo. arXiv:1701.02434v2
34. Bois FY, Maszle DR (1997) MCSim: A simulation program. *J Stat Soft*, 2(9). <https://doi.org/10.1093/bioinformatics/btp162>
35. Hindmarsh AC, Brown PN, Grant KE, Lee SL, Serban R, Shumaker DE, Woodward CS (2005) SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers. *ACM Trans Math Soft* 31(3):363-96.
36. Gelman A, Carlin JB, Stern HS, Dunson DB, Vehtari A, Rubin DB (2001) *Bayesian Data Analysis*. (3rd edition) CRC PRESS, Boca Raton, pp 284-287
37. Gelman A, Rubin DB (1992) Inference from Iterative Simulation Using Multiple Sequences. *Statist Sci* 7(4):457-472
38. Fabrega F, Nadal M, Schuhmacher M, Domingo JL, Kumar V (2016) Influence of the uncertainty in the validation of PBPK models: A case-study for PFOS and PFOA. *Regul Toxicol Pharmacol* 77:230-9. <https://doi.org/10.1016/j.yrtph.2016.03.009>
39. Betancourt MJ, Girolami M (2013) Hamiltonian Monte Carlo for Hierarchical Models. arXiv:1312.0906
40. Garcia RI, Ibrahim JG, Wambaugh JF, Kenyon EM, Setzer RW (2015) Identifiability of PBPK models with applications to dimethylarsinic acid exposure. *J Pharmacokinet Pharmacodyn* 42(6):591-609. <https://doi.org/10.1007/s10928-015-9424-2>
41. Yates JWT (2006) Structural identifiability of physiologically based pharmacokinetic models. *J Pharmacokinet Pharmacodyn* 33(4):421-39. <https://doi.org/10.1007/s10928-006-9011-7>
42. Hsieh NH, Reisfeld B, Bois F, Chiu WA (2018) Applying a Global Sensitivity Analysis Workflow to Improve the Computational Efficiencies in Physiologically-Based Pharmacokinetic Modeling. *Front Pharmacol* 9:588. <https://doi.org/10.3389/fphar.2018.00588>

A Coefficients of organ weight polynomials for humans

Table 4 Coefficients of organ weight polynomials for male humans [25].

Tissue	X_0	X_1	X_2	X_3	X_4	X_5
Muscle	9.61e-02	-4.88e-06	3.05e-10	-3.62e-15	1.22e-20	0.00
Adipose	3.95e-02	1.59e-05	-6.99e-10	1.09e-14	-5.26e-20	0.00
Gonads	1.67e-04	6.20e-10	-6.54e-13	2.48e-17	-2.85e-22	1.03e-27
Skin	1.07e-01	-3.26e-06	6.11e-11	-5.43e-16	1.83e-21	0.00
Heart	8.53e-03	-4.07e-07	1.40e-11	-1.90e-16	1.05e-21	-1.94e-27
Brain	1.19e-01	-3.51e-06	4.28e-11	-1.82e-16	0.00	0.00
Kidney	7.31e-03	-8.29e-08	5.71e-13	0.00	0.00	0.00
Stomach	1.88e-03	8.76e-08	-2.52e-12	1.86e-17	0.00	0.00
Splachnic	1.74e-02	-5.30e-07	1.18e-11	-6.74e-17	0.00	0.00
Lung	1.67e-02	-9.96e-08	-1.09e-13	1.13e-17	0.00	0.00
Liver	3.49e-02	-3.23e-07	2.13e-12	0.00	0.00	0.00
Arterial Blood	3.66e-02	-3.44e-07	5.00e-12	-2.59e-17	0.00	0.00
Venous Blood	5.49e-02	-5.15e-07	7.50e-12	-3.87e-17	0.00	0.00

Table 5 Coefficients of organ weight polynomials for female humans [25].

Tissue	X_0	X_1	X_2	X_3	X_4	X_5
Muscle	1.17e-01	-3.59e-06	3.19e-10	-3.55e-15	-7.58e-22	0.00
Adipose	5.91e-02	1.20e-05	-5.80e-10	1.12e-14	-6.36e-20	0.00
Gonads	1.94e-04	-8.32e-09	3.15e-13	0.00	0.00	0.00
Skin	9.54e-02	-1.70e-06	-1.64e-13	2.64e-16	-1.49e-21	0.00
Heart	5.72e-03	-1.02e-07	2.53e-12	-2.71e-17	9.29e-23	0.00
Brain	1.12e-01	-3.33e-06	4.04e-11	-1.70e-16	0.00	0.00
Kidney	8.04e-03	-1.38e-07	2.19e-12	-1.34e-17	0.00	0.00
Stomach	1.88e-03	8.76e-08	-2.52e-12	1.86e-17	0.00	0.00
Splachnic	1.89e-02	-6.62e-07	1.56e-11	-9.87e-17	0.00	0.00
Lung	1.74e-02	-7.14e-08	-6.78e-14	0.00	0.00	0.00
Liver	3.59e-02	-4.76e-07	8.50e-12	-5.45e-17	0.00	0.00
Arterial Blood	3.66e-02	-3.44e-07	5.00e-12	-2.59e-17	0.00	0.00
Venous Blood	5.49e-02	-5.15e-07	7.50e-12	-3.87e-17	0.00	0.00

B Posterior estimates

Table 6: Posterior estimates obtained with *Stan*. Indices [1], [2] and [3] correspond to CL_{im} , fKp and Kp_{AD} respectively. Omega (variance-covariance matrix) was not a model parameter; it was generated indirectly through the generated quantities block using the samples of the correlation matrix, C, and the samples of the standard deviation, s.

parameter	mean	se_mean	sd	2.5 %	25%	50%	75%	97.5%	N_{eff}	\hat{R}
sigma	0.282	0.0001	0.012	0.259	0.273	0.281	0.290	0.306	20000	1
mu[1]	4.563	0.0004	0.054	4.455	4.527	4.563	4.598	4.669	20000	1
mu[2]	-1.770	0.0013	0.178	-2.128	-1.886	-1.769	-1.654	-1.423	20000	1
mu[3]	1.368	0.0005	0.078	1.209	1.318	1.370	1.421	1.516	20000	1
s[1]	0.222	0.0003	0.046	0.146	0.190	0.217	0.248	0.325	20000	1
s[2]	0.746	0.0012	0.146	0.507	0.644	0.729	0.831	1.078	14265	1
s[3]	0.395	0.0005	0.077	0.270	0.341	0.387	0.440	0.569	20000	1
C[1,1]	1.000	0.0001	0.000	1.000	1.000	1.000	1.000	1.000	20000	1
C[1,2]	0.131	0.0011	0.152	-0.173	0.030	0.134	0.235	0.421	20000	1
C[1,3]	0.144	0.0011	0.151	-0.157	0.043	0.146	0.249	0.430	20000	1
C[2,1]	0.131	0.0011	0.152	-0.173	0.030	0.134	0.235	0.421	20000	1
C[2,2]	1.000	0.00011	0.000	1.000	1.000	1.000	1.000	1.000	20000	1
C[2,3]	0.187	0.0011	0.152	-0.119	0.086	0.192	0.293	0.475	20000	1
C[3,1]	0.144	0.0011	0.151	-0.157	0.043	0.146	0.249	0.430	20000	1
C[3,2]	0.187	0.0011	0.152	-0.119	0.086	0.192	0.293	0.475	20000	1
C[3,3]	1.000	0.0001	0.000	1.000	1.000	1.000	1.000	1.000	18831	1
Omega[1,1]	0.051	0.0002	0.022	0.021	0.036	0.047	0.062	0.106	20000	1
Omega[1,2]	0.021	0.0002	0.027	-0.031	0.005	0.020	0.037	0.079	20000	1
Omega[1,3]	0.012	0.0001	0.014	-0.015	0.003	0.012	0.021	0.042	20000	1
Omega[2,1]	0.021	0.0002	0.027	-0.031	0.005	0.020	0.037	0.079	20000	1
Omega[2,2]	0.578	0.0020	0.235	0.257	0.415	0.532	0.690	1.162	14221	1
Omega[2,3]	0.053	0.0003	0.048	-0.039	0.023	0.052	0.081	0.156	20000	1
Omega[3,1]	0.012	0.0001	0.014	-0.015	0.003	0.012	0.021	0.042	20000	1
Omega[3,2]	0.053	0.0003	0.048	-0.039	0.023	0.052	0.081	0.156	20000	1
Omega[3,3]	0.162	0.0005	0.066	0.073	0.116	0.149	0.193	0.323	20000	1
theta_tr[1,1]	4.529	0.0008	0.110	4.296	4.459	4.535	4.605	4.729	20000	1
theta_tr[1,2]	-1.740	0.0020	0.278	-2.336	-1.914	-1.725	-1.549	-1.242	20000	1
theta_tr[1,3]	1.342	0.0010	0.147	1.055	1.242	1.341	1.441	1.631	20000	1
theta_tr[2,1]	4.908	0.0008	0.113	4.669	4.837	4.914	4.985	5.110	20000	1
theta_tr[2,2]	-0.754	0.0018	0.254	-1.295	-0.915	-0.742	-0.579	-0.293	20000	1
theta_tr[2,3]	2.063	0.0014	0.193	1.697	1.935	2.057	2.189	2.454	20000	1
theta_tr[3,1]	4.483	0.0007	0.103	4.270	4.417	4.487	4.553	4.672	20000	1
theta_tr[3,2]	-2.061	0.0024	0.339	-2.764	-2.276	-2.044	-1.829	-1.441	20000	1
theta_tr[3,3]	1.767	0.0010	0.135	1.508	1.676	1.764	1.856	2.039	20000	1
theta_tr[4,1]	4.286	0.0006	0.088	4.110	4.228	4.287	4.346	4.456	20000	1
theta_tr[4,2]	-2.411	0.0026	0.361	-3.220	-2.626	-2.374	-2.160	-1.796	20000	1
theta_tr[4,3]	0.954	0.0010	0.144	0.662	0.859	0.957	1.051	1.227	20000	1
theta_tr[5,1]	4.508	0.0007	0.093	4.318	4.449	4.510	4.572	4.682	20000	1
theta_tr[5,2]	-2.926	0.0022	0.304	-3.584	-3.116	-2.905	-2.718	-2.384	20000	1
theta_tr[5,3]	1.666	0.0009	0.128	1.420	1.578	1.665	1.752	1.919	20000	1
theta_tr[6,1]	4.704	0.0007	0.102	4.492	4.639	4.707	4.773	4.896	20000	1
theta_tr[6,2]	-2.017	0.0020	0.284	-2.618	-2.199	-2.003	-1.822	-1.494	20000	1
theta_tr[6,3]	1.565	0.0010	0.145	1.285	1.468	1.561	1.660	1.860	20000	1
theta_tr[7,1]	4.706	0.0009	0.122	4.451	4.628	4.712	4.790	4.927	20000	1
theta_tr[7,2]	-1.536	0.0022	0.318	-2.203	-1.737	-1.523	-1.318	-0.948	20000	1
theta_tr[7,3]	1.650	0.0011	0.150	1.363	1.548	1.647	1.750	1.947	20000	1
theta_tr[8,1]	4.701	0.0007	0.095	4.507	4.638	4.704	4.767	4.880	20000	1
theta_tr[8,2]	-2.295	0.0033	0.461	-3.266	-2.583	-2.276	-1.981	-1.450	20000	1
theta_tr[8,3]	1.180	0.0009	0.128	0.934	1.094	1.178	1.265	1.438	20000	1
theta_tr[9,1]	4.549	0.0006	0.089	4.367	4.491	4.550	4.609	4.718	20000	1
theta_tr[9,2]	-2.188	0.0026	0.371	-2.989	-2.415	-2.166	-1.932	-1.518	20000	1
theta_tr[9,3]	1.413	0.0009	0.128	1.165	1.327	1.411	1.498	1.669	20000	1
theta_tr[10,1]	4.295	0.0006	0.087	4.121	4.238	4.297	4.354	4.461	20000	1
theta_tr[10,2]	-2.453	0.0025	0.347	-3.204	-2.664	-2.424	-2.212	-1.852	20000	1
theta_tr[10,3]	1.157	0.0009	0.133	0.891	1.068	1.159	1.247	1.413	20000	1

theta_tr[11,1]	4.788	0.0007	0.104	4.576	4.721	4.792	4.859	4.982	20000	1
theta_tr[11,2]	-1.952	0.0030	0.426	-2.916	-2.198	-1.911	-1.660	-1.243	20000	1
theta_tr[11,3]	1.223	0.0010	0.141	0.953	1.125	1.221	1.316	1.510	20000	1
theta_tr[12,1]	4.196	0.0007	0.097	3.995	4.133	4.198	4.263	4.379	20000	1
theta_tr[12,2]	-2.830	0.0031	0.434	-3.795	-3.089	-2.788	-2.528	-2.103	20000	1
theta_tr[12,3]	0.896	0.0009	0.120	0.664	0.815	0.895	0.976	1.135	20000	1
theta_tr[13,1]	4.437	0.0006	0.079	4.278	4.384	4.437	4.489	4.591	20000	1
theta_tr[13,2]	-2.330	0.0032	0.456	-3.389	-2.582	-2.275	-2.013	-1.598	20000	1
theta_tr[13,3]	0.582	0.0011	0.152	0.268	0.484	0.590	0.686	0.863	20000	1
theta_tr[14,1]	4.550	0.0006	0.085	4.383	4.494	4.550	4.608	4.711	20000	1
theta_tr[14,2]	-1.838	0.0032	0.447	-2.869	-2.089	-1.784	-1.529	-1.111	20000	1
theta_tr[14,3]	1.330	0.0011	0.156	1.007	1.234	1.337	1.436	1.617	20000	1
theta_tr[15,1]	4.647	0.0008	0.115	4.406	4.576	4.653	4.727	4.858	20000	1
theta_tr[15,2]	-0.888	0.0023	0.327	-1.578	-1.099	-0.871	-0.663	-0.292	20000	1
theta_tr[15,3]	1.955	0.0016	0.223	1.471	1.822	1.971	2.107	2.353	20000	1
theta_tr[16,1]	4.813	0.0011	0.155	4.484	4.714	4.821	4.922	5.091	20000	1
theta_tr[16,2]	-1.829	0.0027	0.385	-2.682	-2.059	-1.801	-1.563	-1.167	20000	1
theta_tr[16,3]	1.619	0.0012	0.163	1.308	1.506	1.616	1.727	1.947	20000	1
theta_tr[17,1]	4.412	0.0008	0.112	4.178	4.340	4.418	4.489	4.618	20000	1
theta_tr[17,2]	-1.532	0.0028	0.389	-2.378	-1.765	-1.500	-1.261	-0.860	20000	1
theta_tr[17,3]	1.531	0.0011	0.162	1.193	1.429	1.537	1.640	1.833	20000	1
theta_tr[18,1]	4.673	0.0009	0.121	4.417	4.597	4.679	4.757	4.895	20000	1
theta_tr[18,2]	-1.558	0.0026	0.362	-2.340	-1.778	-1.531	-1.309	-0.929	20000	1
theta_tr[18,3]	1.506	0.0011	0.159	1.191	1.400	1.506	1.613	1.820	20000	1
theta_tr[19,1]	4.749	0.0006	0.084	4.583	4.693	4.750	4.805	4.913	20000	1
theta_tr[19,2]	-1.895	0.0030	0.427	-2.878	-2.135	-1.851	-1.602	-1.182	20000	1
theta_tr[19,3]	1.111	0.0010	0.146	0.813	1.016	1.114	1.210	1.389	20000	1
theta_tr[20,1]	4.339	0.0007	0.099	4.134	4.275	4.342	4.407	4.526	20000	1
theta_tr[20,2]	-0.889	0.0017	0.243	-1.372	-1.053	-0.884	-0.724	-0.417	20000	1
theta_tr[20,3]	1.468	0.0021	0.294	0.791	1.302	1.502	1.670	1.955	20000	1
theta_tr[21,1]	4.529	0.0009	0.127	4.261	4.449	4.537	4.619	4.755	20000	1
theta_tr[21,2]	-1.731	0.0026	0.369	-2.535	-1.951	-1.707	-1.478	-1.078	20000	1
theta_tr[21,3]	1.662	0.0010	0.147	1.377	1.562	1.662	1.760	1.955	20000	1
theta_tr[22,1]	4.653	0.0010	0.147	4.343	4.558	4.660	4.757	4.918	20000	1
theta_tr[22,2]	-0.455	0.0020	0.279	-1.038	-0.634	-0.441	-0.265	0.063	20000	1
theta_tr[22,3]	1.657	0.0016	0.223	1.177	1.517	1.669	1.810	2.060	20000	1
theta_tr[23,1]	4.613	0.0011	0.161	4.274	4.513	4.623	4.723	4.904	20000	1
theta_tr[23,2]	-1.185	0.0024	0.333	-1.895	-1.396	-1.164	-0.955	-0.589	20000	1
theta_tr[23,3]	1.643	0.0012	0.169	1.297	1.533	1.648	1.758	1.962	20000	1

Table 7: Posterior estimates obtained with *GNU MCSim*. Indices [1], [2] and [3] correspond to CL_{int} , fKp and Kp_{AD} respectively.

parameter	mean	se_mean	sd	2.5 %	25%	50%	75%	97.5%	N_{eff}	\hat{R}
sigma(geometric)	1.327	0.0003	0.016	1.296	1.317	1.327	1.338	1.361	2095	1.002
mu[1]	4.560	0.0012	0.053	4.453	4.525	4.560	4.595	4.663	2444	1.002
mu[2]	-1.743	0.0037	0.179	-2.100	-1.857	-1.740	-1.625	-1.392	2130	1.002
mu[3]	1.361	0.0017	0.079	1.195	1.311	1.364	1.413	1.513	1221	1.004
s[1]	0.223	0.0013	0.046	0.146	0.190	0.218	0.252	0.327	1375	1.007
s[2]	0.760	0.0042	0.153	0.509	0.653	0.740	0.847	1.114	1349	1.000
s[3]	0.410	0.0022	0.080	0.279	0.353	0.400	0.458	0.598	2353	1.005
theta_tr[1,1]	4.527	0.0021	0.108	4.303	4.456	4.530	4.599	4.733	2840	1.001
theta_tr[1,2]	-1.735	0.0051	0.290	-2.361	-1.920	-1.714	-1.528	-1.219	3318	1.001
theta_tr[1,3]	1.334	0.0027	0.152	1.038	1.235	1.335	1.440	1.628	3210	1.000
theta_tr[2,1]	4.892	0.0025	0.117	4.645	4.821	4.898	4.970	5.101	2187	1.001
theta_tr[2,2]	-0.808	0.0051	0.274	-1.408	-0.987	-0.787	-0.622	-0.327	2885	1.002
theta_tr[2,3]	2.053	0.0041	0.194	1.672	1.924	2.050	2.178	2.435	2196	1.002
theta_tr[3,1]	4.472	0.0020	0.106	4.253	4.403	4.478	4.543	4.669	2759	1.001
theta_tr[3,2]	-2.109	0.0060	0.348	-2.842	-2.329	-2.097	-1.876	-1.476	3417	1.000
theta_tr[3,3]	1.776	0.0024	0.134	1.515	1.685	1.778	1.867	2.041	3005	1.002
theta_tr[4,1]	4.292	0.0020	0.091	4.110	4.230	4.293	4.357	4.468	2159	1.004
theta_tr[4,2]	-2.362	0.0069	0.353	-3.136	-2.570	-2.336	-2.111	-1.750	2691	1.001
theta_tr[4,3]	0.962	0.0029	0.148	0.661	0.866	0.964	1.063	1.242	2682	1.002

theta_tr[5,1]	4.505	0.0019	0.095	4.309	4.443	4.509	4.571	4.683	2682	1.001
theta_tr[5,2]	-2.940	0.0065	0.316	-3.648	-3.132	-2.916	-2.724	-2.383	2449	1.002
theta_tr[5,3]	1.680	0.0025	0.132	1.434	1.591	1.677	1.766	1.949	2804	1.002
theta_tr[6,1]	4.702	0.0020	0.104	4.485	4.634	4.707	4.772	4.900	2796	1.005
theta_tr[6,2]	-2.035	0.0051	0.280	-2.612	-2.213	-2.025	-1.846	-1.519	3184	1.002
theta_tr[6,3]	1.563	0.0028	0.145	1.286	1.465	1.559	1.662	1.850	3013	1.002
theta_tr[7,1]	4.701	0.0025	0.123	4.445	4.622	4.710	4.785	4.921	2503	1.002
theta_tr[7,2]	-1.553	0.0057	0.318	-2.215	-1.758	-1.541	-1.329	-0.961	3169	1.001
theta_tr[7,3]	1.649	0.0029	0.154	1.347	1.548	1.647	1.752	1.953	2727	1.002
theta_tr[8,1]	4.708	0.0017	0.097	4.503	4.649	4.712	4.772	4.893	3061	1.001
theta_tr[8,2]	-2.314	0.0086	0.467	-3.287	-2.611	-2.293	-1.989	-1.460	2914	1.000
theta_tr[8,3]	1.179	0.0022	0.129	0.935	1.089	1.176	1.266	1.439	3544	1.000
theta_tr[9,1]	4.550	0.0016	0.088	4.366	4.493	4.554	4.611	4.713	3071	1.001
theta_tr[9,2]	-2.213	0.0064	0.369	-3.019	-2.430	-2.192	-1.966	-1.545	3381	1.000
theta_tr[9,3]	1.414	0.0022	0.128	1.166	1.327	1.413	1.499	1.668	3460	1.001
theta_tr[10,1]	4.297	0.0017	0.088	4.123	4.239	4.298	4.359	4.468	2708	1.003
theta_tr[10,2]	-2.440	0.0065	0.342	-3.189	-2.644	-2.412	-2.210	-1.842	2805	1.003
theta_tr[10,3]	1.169	0.0026	0.139	0.891	1.074	1.171	1.261	1.434	2749	1.003
theta_tr[11,1]	4.792	0.0020	0.106	4.576	4.722	4.798	4.865	4.992	2690	1.002
theta_tr[11,2]	-1.989	0.0080	0.431	-2.952	-2.248	-1.934	-1.686	-1.285	2938	1.002
theta_tr[11,3]	1.210	0.0025	0.140	0.945	1.117	1.205	1.302	1.496	3087	1.002
theta_tr[12,1]	4.206	0.0020	0.100	4.000	4.141	4.211	4.276	4.395	2513	1.002
theta_tr[12,2]	-2.770	0.0080	0.416	-3.685	-3.017	-2.734	-2.477	-2.055	2712	1.001
theta_tr[12,3]	0.907	0.0023	0.123	0.670	0.824	0.906	0.993	1.144	2870	1.000
theta_tr[13,1]	4.445	0.0016	0.081	4.284	4.392	4.446	4.500	4.599	2374	1.003
theta_tr[13,2]	-2.225	0.0100	0.436	-3.240	-2.460	-2.173	-1.921	-1.519	1907	1.003
theta_tr[13,3]	0.570	0.0042	0.158	0.245	0.472	0.575	0.675	0.873	1492	1.003
theta_tr[14,1]	4.548	0.0016	0.083	4.383	4.491	4.547	4.604	4.705	2816	1.001
theta_tr[14,2]	-1.850	0.0107	0.485	-2.987	-2.104	-1.789	-1.506	-1.065	2034	1.002
theta_tr[14,3]	1.327	0.0035	0.165	0.975	1.225	1.336	1.436	1.634	2279	1.002
theta_tr[15,1]	4.624	0.0022	0.119	4.377	4.551	4.627	4.706	4.840	2949	1.001
theta_tr[15,2]	-0.918	0.0081	0.347	-1.660	-1.138	-0.900	-0.684	-0.297	1796	1.002
theta_tr[15,3]	1.963	0.0062	0.233	1.454	1.826	1.987	2.120	2.371	1446	1.003
theta_tr[16,1]	4.814	0.0036	0.157	4.478	4.720	4.825	4.927	5.089	1895	1.001
theta_tr[16,2]	-1.888	0.0073	0.406	-2.780	-2.133	-1.857	-1.601	-1.182	3056	1.001
theta_tr[16,3]	1.614	0.0036	0.165	1.296	1.501	1.612	1.722	1.939	2139	1.001
theta_tr[17,1]	4.402	0.0022	0.114	4.163	4.332	4.408	4.478	4.607	2715	1.002
theta_tr[17,2]	-1.520	0.0081	0.391	-2.347	-1.767	-1.485	-1.258	-0.842	2376	1.001
theta_tr[17,3]	1.535	0.0033	0.170	1.167	1.432	1.544	1.650	1.845	2563	1.002
theta_tr[18,1]	4.669	0.0023	0.120	4.424	4.591	4.675	4.751	4.891	2708	1.002
theta_tr[18,2]	-1.569	0.0065	0.373	-2.379	-1.804	-1.540	-1.307	-0.912	3413	1.001
theta_tr[18,3]	1.496	0.0034	0.167	1.163	1.388	1.500	1.609	1.818	2500	1.004
theta_tr[19,1]	4.750	0.0017	0.085	4.577	4.693	4.752	4.806	4.915	2528	1.001
theta_tr[19,2]	-1.874	0.0088	0.430	-2.873	-2.123	-1.823	-1.589	-1.171	2527	1.002
theta_tr[19,3]	1.094	0.0030	0.152	0.777	0.994	1.100	1.195	1.374	2629	1.002
theta_tr[20,1]	4.326	0.0020	0.099	4.127	4.260	4.328	4.393	4.515	2609	1.002
theta_tr[20,2]	-0.870	0.0065	0.243	-1.358	-1.038	-0.863	-0.706	-0.411	1366	1.001
theta_tr[20,3]	1.445	0.0086	0.309	0.731	1.269	1.475	1.665	1.959	1254	1.003
theta_tr[21,1]	4.513	0.0028	0.136	4.224	4.428	4.519	4.606	4.763	2300	1.003
theta_tr[21,2]	-1.748	0.0067	0.378	-2.563	-1.976	-1.719	-1.489	-1.098	3113	1.002
theta_tr[21,3]	1.671	0.0031	0.152	1.377	1.568	1.671	1.773	1.972	2424	1.002
theta_tr[22,1]	4.633	0.0030	0.155	4.307	4.534	4.642	4.743	4.906	2643	1.001
theta_tr[22,2]	-0.438	0.0068	0.288	-1.045	-0.624	-0.426	-0.236	0.090	1769	1.001
theta_tr[22,3]	1.623	0.0059	0.241	1.085	1.474	1.641	1.785	2.053	1686	1.002
theta_tr[23,1]	4.591	0.0034	0.167	4.232	4.487	4.600	4.704	4.894	2571	1.003
theta_tr[23,2]	-1.201	0.0066	0.333	-1.923	-1.404	-1.188	-0.980	-0.598	2547	1.001
theta_tr[23,3]	1.647	0.0042	0.176	1.290	1.529	1.654	1.773	1.976	1939	1.003

C Impact of LKJ prior on computational efficiency

Table 8 Computational efficiency for different values of the shape parameter a of the LKJ prior.

a	time (s)	$\overline{N_{\text{eff}}}$	$N_{\text{eff}_{\text{min}}}$	$\overline{N_{\text{eff}}}/s$	$N_{\text{eff}_{\text{min}}}/s$
0.5	5760	18537.29	6410	3.22	1.11
1	6000	18160.41	6465	3.03	1.08
2	5500	19193.47	10410	3.49	1.89
5	5465	19475.82	10460	3.56	1.91
10	5500	19985.18	17350	3.63	3.16
15	5400	19920.41	16240	3.69	3.00
20	5500	19839.59	12290	3.61	2.23
40	5450	19747.41	14205	3.62	2.61
50	5400	19903.88	15270	3.69	2.83