



Original software publication

pk sensi: An R package to apply global sensitivity analysis in physiologically based kinetic modeling

Nan-Hung Hsieh^a, Brad Reisfeld^b, Weihsueh A. Chiu^{a,*}

^a Veterinary Integrative Biosciences, College of Veterinary Medicine and Biomedical Sciences, Texas A&M University, College Station, TX, USA

^b Chemical and Biological Engineering and School of Biomedical Engineering, Colorado State University, Fort Collins, CO, USA



ARTICLE INFO

Article history:

Received 5 February 2020

Received in revised form 8 July 2020

Accepted 6 October 2020

Keywords:

Global sensitivity analysis

Physiologically based kinetic modeling

R package

ABSTRACT

Sensitivity analysis (SA) is an essential tool for modelers to understand the influence of model parameters on model outputs. It is also increasingly used in developing and assessing physiologically based kinetic (PBK) models. For instance, several studies have applied global SA to reduce the computational burden in the Bayesian Markov chain Monte Carlo-based calibration process PBK models. Although several SA algorithms and software packages are available, no comprehensive software package exists that allows users to seamlessly solve differential equations in a PBK model, conduct and visualize SA results, and discriminate between the non-influential model parameters that can be fixed and those that need calibration. Therefore, we developed an R package, named **pk sensi**, to make global SA more accessible in PBK modeling. This package can investigate both uncertainty and sensitivity in PBK models, including those with multivariate model outputs. It also includes functions to check the convergence of the global SA results. Overall, **pk sensi** improves the user experience of performing global SA and can create robust and reproducible results for decision making in PBK model calibration.

© 2020 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

Code metadata

Current code version	1.2.0
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX_2020_29
Legal Code License	LPGL-3.0
Code versioning system used	git
Software code languages, tools, and services used	R
Compilation requirements, operating environments & dependencies	ggplot2, data.table, deSolve, dplyr, getPass, magrittr, foreach, parallel, doParallel, reshape
If available Link to developer documentation/manual	https://nanhung.rbind.io/pksensi/
Support email for questions	d99622005@ntu.edu.tw

Software metadata

Current software version	1.2.0
Permanent link to executables of this version	https://cran.r-project.org/web/packages/pksensi
Legal Software License	GPL-3
Computing platforms/Operating Systems	Linux, MacOS, Microsoft Windows
Installation requirements & dependencies	ggplot2, data.table, deSolve, dplyr, getPass, magrittr, foreach, parallel, doParallel, reshape
If available, link to user manual - if formally published include a reference to the publication in the reference list	https://cran.r-project.org/web/packages/pksensi/pksensi.pdf
Support email for questions	d99622005@ntu.edu.tw

1. Motivation and significance

Sensitivity analysis (SA) is a mathematical technique to investigate how variations in model parameters affect model outputs.

* Corresponding author.

E-mail address: wchiu@cvm.tamu.edu (W.A. Chiu).

An increasing number of studies use SA to determine which model parameters contribute to high variation in model predictions [1]. This technique has also been applied in pharmacology and toxicology research [2,3]. Pharmacokinetic models describe the changes in the concentrations or amounts of a substance in the body over time (the various terms for the kinetic models, including “pharmacokinetic”, “toxicokinetic”, and “biokinetic” models, are used interchangeably here). The goal of SA in physiologically based kinetic (PBK) research is to examine the sensitivity of output variables, such as chemical concentrations in blood or tissues, with respect to input parameters, such as anatomical, physiological, and kinetic constants [4]. In addition, SA can guide experimental design and the parameter estimation processes [5]. For instance, SA can identify “unidentifiable” parameters that can lead to problems with numerical procedures used in parameter estimation [6]. It can be further inform parameter prioritization and parameter fixing before model calibration [7].

In our earlier work [7], we developed an updated workflow to apply global SA to reduce the computational burden in the Bayesian, Markov chain Monte Carlo (MCMC)-based calibration process of a physiologically based pharmacokinetic (PBPK) model. We used GNU MCSim [8], an effective simulation package for Bayesian population PBPK modeling, to calibrate the model. We found that the extended Fourier Amplitude Sensitivity Test (eFAST), a type of variance-based global SA algorithm, had the best balance of efficiency and accuracy for a sophisticated, multi-compartment, multi-dataset, and multi-metabolite PBPK model. This method was first introduced in 1973, [9] in a study of the sensitivity of coupled reaction systems that are constructed by differential equations with the numerous rate coefficients (the classic FAST method with the first-order effect only). It then evolved to estimate the Sobol’ sensitivity measure (the eFAST method with first- and total-order effects) [10]. The FAST algorithm creates the multidimensional space of input parameters through a “search-curve”. It is more computationally efficient in calculating the influence of model parameters than other global SA methods such as the Monte Carlo-based approaches [11,12]. Our previous work also found some efficient visualization approaches that can be used to distinguish between “influential” and “non-influential” parameters. This approach addresses common identifiability issues of PBPK model parameter that increase the computational burden of Bayesian analysis [13], and thereby reduce computational efficiency. We also developed approaches for communicating parameter sensitivity in decision making.

Most SA tools are constructed using scientific computing software, such as Matlab [14], python [15], and R [16]. The R software is widely used and includes numerous developed packages, such as **sensitivity** [17], which include tools to conduct local and global SA. The **sensitivity** package includes some functions to generate the parameter sequences by using different computing algorithms. Also, it provides a feasible way to integrate external modeling results. This computational approach can effectively address the heavy computing burden of numerical solutions within the pure R environment.

Several other R packages include SA tools and are freely available in the Comprehensive R Archive Network (CRAN). Although these tools provide various approaches to perform SA, they each individually have limitations for application to PBK modeling. For example, the estimated index in SA is not robust when using an inadequate sample size. However, there are no software packages that provide tools for assessing convergence. In addition, the existing eFAST function does not include the feature to generate the random sampling curve that is needed to assess convergence. We present here an R package, called **pknsensi**, which is designed to make SA more accessible and reproducible for pharmacological and toxicological research.

2. Software description

Our package can investigate both parameter uncertainty and sensitivity in a PBK model with multivariate model outputs. Moreover, it seamlessly integrates with both general ODE solvers available in R or as external C code, as well as the GNU MCSim software used for conducting sophisticated PBK modeling, including MCMC. The analytical framework in **pknsensi** includes not only global SA but also the uncertainty analysis and diagnostic tools to support the modeling process.

2.1. Installation

This package can link with the R **deSolve** package, which includes a comprehensive integrator to solve ODE. The GNU MCSim can be installed by following the instruction in GNU MCSim’s manual on <https://www.gnu.org/software/mcsim/mcsim.html> or using the built-in function in **pknsensi**.

2.2. Parameter matrix generation

We adopted eFAST, the widely used analysis of variance (ANOVA)-like global SA approach to decompose the variance to partial variances for multidimensional parameters in numerical models in this package. Most of the available eFAST functions, such as the algorithm to generate the parameter space and to set the sampling frequency, were sourced from the R **sensitivity** package [17]. The eFAST method computes the sensitivity index (SI) via ANOVA-like decomposition of the function for analysis. This type of global SA approach and the corresponding SI are also known as the Sobol’ method and Sobol’ index. Since the estimated SI is not stable (high variation) when using smaller sample sizes, we included a random phase-shift approach to conduct replicating sampling from random starting points across parameter space to test the convergence and robustness of the sensitivity indices. Considering the model that is built under $y = f(x_i)$. The sampling scheme can describe as,

$$x_i = \frac{1}{2} + \frac{1}{\pi} \arcsin(\sin(\omega_i s + \varphi_i))$$

where x_i is the nominal value of the i th parameter, and ω_i is a vector giving the set of frequencies, one frequency for each parameter, and φ is a random phase-shift coefficient ranged from 0 to 2π . For a given sample size, the SA should give similar SIs across each replication with random phase-shifts in order to be considered robust, or “converged”. For most Sobol’-based SA algorithms, a larger sample size is needed to reduce the probability of having negative SI estimates. However, eFAST estimates are always positive due to how it is constructed.

To investigate the convergence of SI, we adopted the “exam” approach that proposed previously [18]. This method quantitatively assesses convergence by computing the width of 95% confidence intervals of computed SI for all parameters across all time-points and output variables. Through the random phase-shift procedure, the replicated SI values can be repeatedly sampled independently. Therefore, they can be used to investigate the convergence of SI as,

$$CI_{i,t} = \max(SI_{i,t}^{ub} - SI_{i,t}^{lb})$$

where $SI_{i,t}^{ub}$ and $SI_{i,t}^{lb}$ are the upper and lower bound of the SI from the specific model output (i) at time (t). $CI_{i,t}$ is the corresponding convergence index.

In **pknsensi**, we integrate the random phase-shift with eFAST to perform the sensitivity test for the whole modeling process. A function can sample and generate the testing parameter matrix based on a given argument of sample size and probability

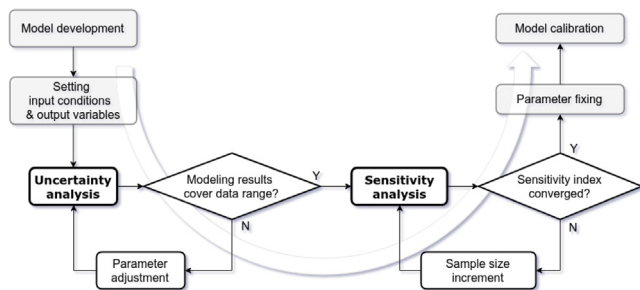


Fig. 1. The proposed workflow for performing uncertainty and sensitivity analysis in the Bayesian MCMC model calibration process.

distribution (e.g., mean/s.d. of normal distribution, meanlog/sdlog of log-normal distribution, and minimum/maximum of uniform distribution). The number of model evaluations is equal to the sample size times the number of model parameters.

2.3. Modeling

pk sensi also provides a useful function to solve ODEs in PBK models using analytical and numerical approaches. The solution can be calculated under the pure R programming environment by linking with R deSolve package [19]. However, the most efficient way to achieve high computational speed is through compiled, lower-level languages, such as FORTRAN, C, or C++. Therefore, **pk sensi** includes a function to compile and create dynamic-link libraries (.dll) on Windows and shared objects (.so) on Unix-like systems (e.g., Linux and MacOS). This compiled program can load and execute through the build-in function in R. However, Windows users need to install Rtools to compile the source code by using the GNU GCC compiler. The C implementation of the example model can be found as an example within GNU MCSim [8]. The **pk sensi** can also link with GNU MCSim to create the model program, which is used in solving each system of equations.

2.4. Uncertainty analysis

The general workflow of uncertainty and SA in **pk sensi** is illustrated in Fig. 1. The uncertainty analysis is a crucial modeling process within SA. Sometimes, it is difficult to have informative data to set up the parameter distribution. An overly wide range for a parameter distribution might cause a numerical error at extreme values. On the other hand, overly narrower ranges might lead to predictions that cannot cover the range observed in the calibration data. Uncertainty analysis using Monte Carlo is an appropriate approach in the data-driven modeling process to address this problem. Before fitting the data to model, the Monte Carlo can be applied with a given distribution of model parameters to examine whether the experimental data are within the “rational” range of the distribution of model output. The **pk sensi** also has a function that can link to GNU MCSim to conduct Monte Carlo analysis, which is built-into GNU MCSim, and then examine the simulation result. In addition, the users can also examine the model output that was generated by the parameter matrix from eFAST.

2.5. Sensitivity analysis

Through the plotting functions in **pk sensi**, time-dependent sensitivity measurements of first and total effects for all parameters can be plotted simultaneously. If the computed sensitivity indices show different values across replications under the same sample size, this indicates an inadequate sample number, which

can create an unreliable result. It is essential to use a sufficient sample size to prevent incorrect judgments in parameter fixing. The package includes functions to check the convergence and sensitivity of model parameters, providing a means to assess the robustness of the sensitivity measurement. We also developed a “cut-off”-based approach in this package to distinguish between “influential” and “non-influential” parameters. Finally, this package also provides visualization tools for effective investigation and communication of results in decision support.

3. Illustrative examples

We show two example models to illustrate the use of **pk sensi** in PBK modeling. In the first example, we used a generic, one-compartment model from **httk** package [20] to explain the overall workflow and the details in the simulation process. Users can refer to online vignettes and user manual in the CRAN (<https://cran.r-project.org/web/packages/pksensi/index.html>) or the **pk sensi** website (<https://nanhung.rbind.io/pksensi/>) for the detailed usage examples. The partial example of the R code and its result is given in Appendix and Fig. 2 that comprises the following essential steps in the process.

1. Construct the model
2. Define initial conditions, output time steps and variables
3. Set parameter distributions
4. Conduct simulation
5. Uncertainty analysis
6. Check and visualize SA result

The first step is to prepare and construct the model. The model code can be written based on R **deSolve** and GNU MCSim format to provide the flexibility for the user. Then, one must set up the model variables and their initial conditions. Furthermore, the simulation time points have to provided in this part.

The next step is to identify the model parameters that will be examined in global SA and create the parameter matrix. After parameter selection, we have to assign each parameter a probability distribution over which to evaluate, such as a uniform distribution with specified min and max. The users then can use **rfast99()** with the above assignments (e.g., parameter names, sample number, parameter distribution, and its corresponding properties) to create the parameter matrix.

The final step is to solve the ODEs in the model by using the **solve_fun()** function and visualize the results. The result of the one-compartment model shows that body weight (BW) is a non-influential parameter for the blood concentration in the simulation (see Appendix). In addition, use of the model code and solving engine from GNU MCSim (**solve_mcsim()**) can provide the fastest computational speed.

The second example uses the acetaminophen (APAP)-PBPK model, a complex model with multiple compartments, to demonstrate the reproducibility of our previous publication [7] and the capability of **pk sensi**. The APAP-PBPK model describes the pharmacokinetics of parent APAP and its two metabolites, a glucuronide AG and a sulfate AS. We applied the comprehensive global SA workflow to the original published model with 21 model parameters [21]. Fig. 3 shows the time-dependent total SI computed for the global SA method. The heat map provides an alternative way to visualize the estimated results. Some parameters (e.g., *CYP_Km*) showed non-influential impact across all three output variables across the investigation time interval.

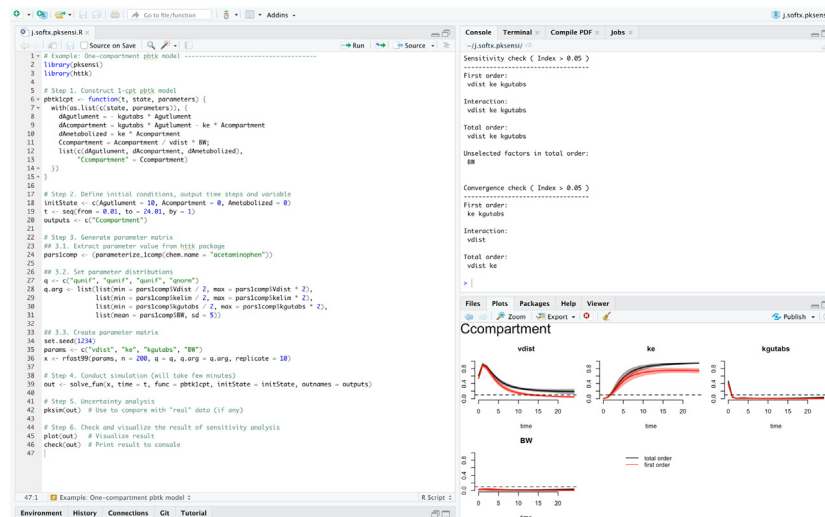


Fig. 2. Example R script and its corresponding results for the application of the **pksensi** in one-compartment pharmacokinetic modeling. Results can be checked directly in the R console through **check()** function, and the visualization result can output by **plot()** function.

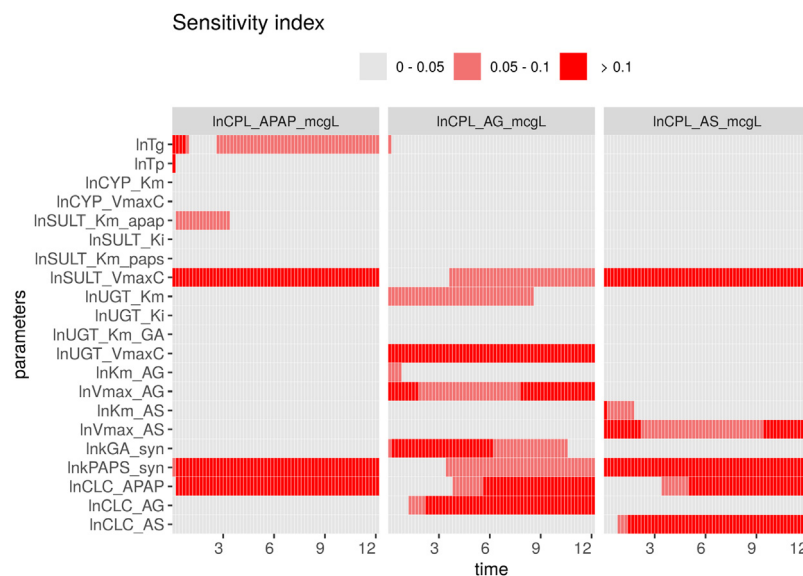


Fig. 3. Heat map representation of time-dependent total SI computed from 21 substance-specific parameters in APAP-PBPK model.

4. Impact and conclusions

Computational modeling has become an essential technique in modern scientific research to support hypotheses, experiments, and discoveries. Uncertainty and SA are essential tools for modelers in the conduct of model development and evaluation. Several packages on the CRAN repository and other software platforms had been built to perform global SA with a variety of algorithms, but none have included integration with ODE-based models, such as PBK models.

We initiated **pksensi** in order to facilitate a comprehensive and efficient global SA workflow in PBK modeling, filling a crucial gap for open-source modeling communities. It provides a straightforward application to investigate the impact of parameters on model outputs across simulation time-points and variables. We adopted the eFAST method, a common variance-based SA that we found to have the best balance of efficiency and accuracy. In addition, the package also includes the ability to assess the convergence of SA results, which is rarely addressed in most global SA studies and software packages. We also developed

functions to visualize the output results and help distinguish “influential” parameters from “non-influential” parameters that can be fixed in model calibration. Our package makes global SA more accessible in PBK modeling.

The current version of **pksensi** includes both GNU MCSim and **deSolve** package as the main ODE solvers. We chose to integrate SA with GNU MCSim because the latter is a powerful open-source software package for dynamical simulation of biological-based models, mainly built for PBK research with probabilistic approaches [8]. Compared with the **deSolve** package, the GNU MCSim can provide better speed and efficiency to solve the ODEs in PBK models. Although **pksensi** provides the essential features and functions to install and link with GNU MCSim, conducting the uncertainty and SA in PBK modeling, the overall learning curve of this workflow might be steep for users that are not familiar with GNU MCSim from model development, debugging, and testing. Therefore, the general package, such as **deSolve**, can be an alternative option for R users. The flexibility of our package can help modelers quantify the influence of model parameters on model outputs. In addition, **pksensi** also has the

potential to be applied in an interactive PBPK modeling platform that was mentioned in the recent study [22].

It is worth noting that the current global SA results are conditional on the specific model and the given range of the model parameters. Other factors, such as the number of parameters examined, the parameter distributions/ranges, and the given dosing level in the PBK model might also have a potential impact on the level of importance of each parameter, but these “meta-uncertainties” are not currently addressed. Complex models with high parameter dimensionality are also challenging in global SA as they have higher computer demands. As approaches are developed to address these complexities, we hope to incorporate them into **pknsensi**.

Overall, **pknsensi** provides a comprehensive suite of features and functions for performing global SA for PBK models and can create robust and reproducible results for decision making in model development and calibration. Although **pknsensi** used here is mainly for PBK modeling, it can also be applied to other ODE-based dynamic models in order to investigate the sensitivity of model outputs to input parameters.

CRediT authorship contribution statement

Nan-Hung Hsieh: Conceptualization, Software, Writing - original draft. **Brad Reisfeld:** Funding acquisition, Project administration. **Weihsueh A. Chiu:** Supervision, Methodology, Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was funded, in part, by grant 1U01FD005838 from the U.S. Food and Drug Administration (FDA) and grant P42 ES027704 from the U.S. National Institute of Environmental Health Sciences. This article reflects the views of the author and should not be construed to represent FDA's views or policies. We thank Dr. Yasha Hartberg and Dr. Barbara Gastel at the Texas A&M University, Dr. Frederic Bois from Certara, and Dr. Eleftheria Tsakalozou from U.S. FDA for reviewing the manuscript and consultation.

Appendix. Example code

```
library(pknsensi)
library(httk)

# Example: One-compartment pbtk model
#-----
# Step 1. Construct 1-cpt pbtk model for deSolve package
pbtk1cpt <- function(t, state, parameters) {
  with(as.list(c(state, parameters)), {
    dAgutlument = - kgutabs * Agutlument
    dAcompartment = kgutabs * Agutlument - ke *
      Acompartment
    dAmetabolized = ke * Acompartment
    Ccompartment = Acompartment / vdist * BW;
    list(c(dAgutlument, dAcompartment, dAmetabolized),
         "Ccompartment" = Ccompartment)
  })
}

# Step 2. Define initial conditions, output time steps
and variable
```

```
initState <- c(Agutlument = 10, Acompartment = 0,
               Ametabolized = 0)
t <- seq(from = 0.01, to = 24.01, by = 1)
outputs <- c("Ccompartment")

# Step 3. Generate parameter matrix
## 3.1. Extract parameter value from httk package
pars1comp <- (parameterize_1comp(chem.name = "
  acetaminophen"))

## 3.2. Set parameter distributions
q <- c("qunif", "qunif", "qunif", "qnorm")
q.arg <- list(list(min = pars1comp$Vdist / 2, max =
  pars1comp$Vdist * 2),
              list(min = pars1comp$kelim / 2, max =
  pars1comp$kelim * 2),
              list(min = pars1comp$kgutabs / 2, max =
  pars1comp$kgutabs * 2),
              list(mean = pars1comp$BW, sd = 5))

## 3.3. Create parameter matrix
set.seed(1234)
params <- c("vdist", "ke", "kgutabs", "BW")
x <- rfast99(params, n = 200, q = q, q.arg = q.arg,
             replicate = 10)

# Step 4. Conduct simulation (will take few minutes)
out <- solve_fun(x, time = t, func = pbtk1cpt,
                 initState = initState, outnames = outputs)

# Step 5. Uncertainty analysis
pksim(out) # Use to compare with "real" data (if any)

# Step 6. Check and visualize the result of
sensitivity analysis
plot(out) # Visualize result
check(out) # Print result to console
```

References

- [1] Ferretti F, Saltelli A, Tarantola S. Trends in sensitivity analysis practice in the last decade. *Sci Total Environ* 2016;568:666–70. <http://dx.doi.org/10.1016/j.scitotenv.2016.02.133>.
- [2] Loizou GD, McNally K, Jones K, Cocker J. The application of global sensitivity analysis in the development of a physiologically based pharmacokinetic model for m-xylene and ethanol co-exposure in humans. *Front Pharmacol* 2015;6:135. <http://dx.doi.org/10.3389/fphar.2015.00135>.
- [3] McNally K, Cotton R, Cocker J, Jones K, Bartels M, Rick D, et al. Reconstruction of exposure to m-xylene from human biomonitoring data using PBPK modelling, Bayesian inference, and Markov chain Monte Carlo simulation. *J Toxicol* 2012. <http://dx.doi.org/10.1155/2012/760281>.
- [4] McNally K, Cotton R, Loizou GD. A workflow for global sensitivity analysis of PBPK models. *Front Pharmacol* 2011;2:31. <http://dx.doi.org/10.3389/fphar.2011.00031>.
- [5] Zhang X-Y, Trame M, Lesko L, Schmidt S. Sobol sensitivity analysis: a tool to guide the development and evaluation of systems pharmacology models. *CPT Pharmacomet Syst Pharmacol* 2015;4(2):69–79. <http://dx.doi.org/10.1002/psp4.6>.
- [6] Chu Y, Hahn J. Quantitative optimal experimental design using global sensitivity analysis via quasi-linearization. *Ind Eng Chem Res* 2010;49(17):7782–94. <http://dx.doi.org/10.1021/ie9009827>.
- [7] Hsieh N-H, Reisfeld B, Bois FY, Chiu WA. Applying a global sensitivity analysis workflow to improve the computational efficiencies in physiologically-based pharmacokinetic modeling. *Front Pharmacol* 2018;9:588. <http://dx.doi.org/10.3389/fphar.2018.00588>.
- [8] Bois FY. GNU MCSim: Bayesian statistical inference for SBML-coded systems biology models. *Bioinformatics* 2009;25(11):1453–4. <http://dx.doi.org/10.1093/bioinformatics/btp162>.
- [9] Cukier R, Fortuin C, Shuler KE, Petschek A, Schaibly J. Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I theory. *J Chem Phys* 1973;59(8):3873–8. <http://dx.doi.org/10.1063/1.1680571>.
- [10] Saltelli A, Tarantola S, Chan K-S. A quantitative model-independent method for global sensitivity analysis of model output. *Technometrics* 1999;41(1):39–56. <http://dx.doi.org/10.2307/1270993>.
- [11] Jansen MJ. Analysis of variance designs for model output. *Comput Phys Comm* 1999;117(1–2):35–43. [http://dx.doi.org/10.1016/S0010-4655\(98\)00154-4](http://dx.doi.org/10.1016/S0010-4655(98)00154-4).

- [12] Owen AB. Better estimation of small sobol'sensitivity indices. *ACM Trans Model Comput Simul* 2013;23(2):11. <http://dx.doi.org/10.1145/2457459.2457460>.
- [13] Garcia RI, Ibrahim JG, Wambaugh JF, Kenyon EM, Setzer RW. Identifiability of PBPK models with applications to dimethylarsinic acid exposure. *J Pharmacokinet Pharmacodyn* 2015;42(6):591–609. <http://dx.doi.org/10.1007/s10928-015-9424-2>.
- [14] Pianosi F, Sarrazin F, Wagener T. A Matlab toolbox for global sensitivity analysis. *Environ Model Softw* 2015;70:80–5. <http://dx.doi.org/10.1016/j.envsoft.2015.04.009>.
- [15] Herman J, Usher W. SALib: an open-source Python library for sensitivity analysis. *J Open Source Softw* 2017;2(9). <http://dx.doi.org/10.21105/joss.00097>.
- [16] Iooss B, Saltelli A. Introduction to sensitivity analysis. In: *Handbook of uncertainty quantification*. Springer; 2017, p. 1103–22. http://dx.doi.org/10.1007/978-3-319-12385-1_31.
- [17] Iooss B, Janon A, Pujol G, Boumhaout K, DaVeiga S, Delage T, et al. Sensitivity: Global sensitivity analysis of model outputs. 2018, R package version 1.15.1, URL <https://CRAN.R-project.org/package=sensitivity>.
- [18] Sarrazin F, Pianosi F, Wagener T. Global sensitivity analysis of environmental models: convergence and validation. *Environ Model Softw* 2016;79:135–52. <http://dx.doi.org/10.1016/j.envsoft.2016.02.005>.
- [19] Soetaert K, Petzoldt T, Setzer RW. Solving differential equations in R: Package deSolve. *J Stat Softw* 2010;33(9):1–25. <http://dx.doi.org/10.18637/jss.v033.i09>.
- [20] Pearce R, Setzer RW, Strobe C, Sipes N, Wambaugh J. Httk: R package for high-throughput toxicokinetics. *J Stat Softw* 2017;79(4):1–26. <http://dx.doi.org/10.18637/jss.v079.i04>.
- [21] Zurlinden TJ, Reisfeld B. Physiologically based modeling of the pharmacokinetics of acetaminophen and its major metabolites in humans using a Bayesian population approach. *Eur J Drug Metab Pharmacokinet* 2016;41:267–80. <http://dx.doi.org/10.1007/s13318-015-0253-x>.
- [22] Li M, Cheng Y-H, Chittenden JT, Baynes RE, Tell LA, Davis JL, et al. Integration of food animal residue avoidance databank (FARAD) empirical methods for drug withdrawal interval determination with a mechanistic population-based interactive physiologically based pharmacokinetic (iPBPK) modeling platform: example for flunixin meglumine administration. *Arch Toxicol* 2019;1–16. <http://dx.doi.org/10.1007/s00204-019-02464-z>.