RVis: An Open Access PBPK Modelling Platform

Author(s):
George Loizou

Report Number:
HG/2020/13

RVis: An Open Access PBPK Modelling Platform

Report approved by:	(Include qualifications)
Date of Issue:	
Lead Author:	George Loizou
Contributing Author(s):	(Include qualifications and other organisations if applicable) (include team name)
Customer:	Cefic - LRI
Technical Reviewer(s):	(Include qualifications)
Editorial Reviewer:	(Include qualifications)
Project number:	PE11285
Distribution list:	

Disclaimer:
This report and the work it describes were undertaken by the Health and Safety Executive (HSE) under contract to the European Chemical Industry Council (Cefic). Its contents, including any opinions and/or conclusions expressed or recommendations made, do not supersede current HSE policy or guidance.

Acknowledgements

This work was supported by The European Chemical Industry Council (Cefic) (Grant No: LRI-AIMT7-HSL).

KEY MESSAGES

RVis is an open access, free to use general purpose mathematical modelling software application for models written in R or MCSim syntax, both free, simulation and statistical modelling tools.

RVis was designed to facilitate widespread adoption of Physiologically Based Pharmacokinetic (PBPK) modelling in chemical product development and safety assessment.

A workflow for the development and analysis of model structure and quantification of uncertainty can be followed within RVis which is based on an emerging Good Modelling Practice for PBPK models.

RVis deploys an intuitive user-interface designed to shift the emphasis away from requiring high levels of mathematical expertise and programming skills to the understanding of the biology of toxicity and disease.

RVis has a parameter estimation module that may be used for “reverse dosimetry” to reconstruct human dose or exposure concentrations consistent with measured biological monitoring data or in vitro cell concentrations used as surrogates of in vivo organ or tissue concentrations.

RVis is a labour-saving device; appropriate expertise is still required to correctly configure the modules that deploy statistical algorithms and analysis.

EXECUTIVE SUMMARY

The objective of the project was to further develop RVis, a prototype application for the analysis of structure and performance of physiologically based pharmacokinetic (PBPK), and other models, written in the free, open source syntax R. The overall aim was to extend, improve and to provide more features and make them more robust. However, increasingly complex models written in R syntax have the considerable disadvantage of very slow run- times; therefore, the option of running models written in GNU MCSim, another free software platform was added.

RVis, is in fact a general purpose modelling platform providing the modules to load, run, visualise and plot graphical outputs from models, the analysis of model structure using Latin Hypercube sampling, parameter elementary effects screening (Morris Test) and global sensitivity analysis (GSA) and parameter estimation using Markov Chain Monte Carlo simulation and Bayesian inference. The latter module can be used to perform exposure or dose reconstruction, from human biological monitoring and in vitro data. The latter is commonly referred to as “reverse dosimetry”.

RVis was designed to expand the user base of PBPK modelling to include chemical safety and regulatory toxicologists. However, the correct and efficient use of the modules deploying stochastic modelling and the interpretation of the results require specialist skills and expertise. RVis is a labour saving device; it is not a substitute for specialist expertise: it is advisable that users of the software have access to such expertise.

The availability of a resource such as RVis could have a potentially significant role in three important areas: the development of internationally recognized good modelling practice (GMP) rigorous peer-review of PBPK models and software resilience.

It is envisaged that the development of RVis will continue.

CONTENTS

1. INTRODUCTION

The widespread adoption and application of physiologically based pharmacokinetic (PBPK) modelling in product development and safety assessment has been hampered by criticism that these models are data hungry, resource intensive, complex and require high levels of mathematical expertise and programming skills. Most criticisms can be addressed, as has been demonstrated, with the development of prototype, proof-of-principle, user-friendly, free to use, web-based tools such as MEGen (Loizou and Hogg, 2011), for the rapid generation of PBPK model code, and PopGen (McNally *et al.*, 2014), a virtual human population generator. Both applications shift the emphasis away from the need for high levels of mathematical expertise and programming skills to the understanding of the biology of toxicity and disease that should underpin chemical safety and risk assessment. Further development of such tools would continue to mitigate existing concerns and make this powerful approach more readily accessible to safety toxicologists and risk assessors.

However, the greatest obstacle to the more widespread adoption of PBPK modelling is most likely the availability of a common, transparent and independently auditable, free-to-use platform for running models and analysing model structure and output. In response to this need the European Partnership for Alternative Approaches to Animal testing (EPAA) and the Health and Safety Executive (HSE) funded the Health and Safety Executive Science and Research Centre (SARC) to develop a user-friendly in vitro and in vivo exposure predictor. The motivation for this tool was to contribute to the replacement, reduction and refinement (3Rs) of animal testing. This requires the ability to predict equivalent human oral, dermal or inhalation exposures that are consistent with measured in vitro target tissue concentrations; an issue which can only be achieved using PBPK modelling approaches. The output of this project was RVis, a prototype, proof-of-concept application for the analysis of structure and performance of PBPK, and other models, written in the free, open source syntax R.

The development of RVis began on June 1st 2014 and ended February 1st 2016. The scope of the project was to develop a free to use, publicly accessible 'forward-'as well as 'reverse-dosimetry' tool. This tool would serve experienced mathematical modellers as well as other scientists developing non-animal alternatives. The existing web-based tool MEGen (Loizou and Hogg, 2011) (from the CEFIC LRI toolbox) which enables a modeller to select parameters (from the literature or in-house derived) to populate (define parameter values in) a PBPK model and generate and export computer code in a number of syntaxes would be modified to export models in R syntax. The use of R, a free, open source programming language and software environment (Team, 2017) would provide the 'free to use' running of the PBPK model (actual simulation of the kinetics) as well as visualisation of outputs such as tissue concentration vs time profiles (forward dosimetry) or external exposure vs time profiles (reverse dosimetry). In order to allay the security fears of some of the EPAA industrial partners RVis was developed as an application to be installed on Windows based PCs thereby obviating the need to upload (proprietary) data over the internet.

However, larger and more complex PBPK models written in R can have very slow run times. In particular, computationally intensive analyses such as global sensitivity analysis and parameter estimation using Markov chain Monte Carlo sampling can become impractical. Therefore, RVis has been extended to accept PBPK models written in GNU MCSim syntax. This is a simulation package, written in C and therefore better suited to this type of modelling and importantly, much more rapid than R.

RVis is, in fact, a general purpose modelling platform, not just an in vitro and in vivo exposure predictor. The features of RVis include the ability to load, run, visualise and plot graphical outputs from models. Model structure may be analysed using parameter elementary effects screening (Morris Test) and global sensitivity analysis (GSA) (Hsieh *et al.*, 2018; McNally *et al.*, 2011) and parameter estimation using Markov Chain Monte Carlo simulation and Bayesian inference (McNally *et al.*, 2012). The parameter estimation feature is used to perform "reverse dosimetry" to reconstruct human dose or exposure concentrations consistent with measured biological monitoring data or in vitro cell concentrations used as surrogates of in vivo organ or tissue concentrations.

The further development of RVis was based on the recommendations made by the independent evaluation of the RVis prototype organised by ECETOC on behalf of CEFIC. The evaluations were conducted by experts from the European Chemicals Agency (ECHA), EU Joint Research Centre (Italy), US Environmental Protection Agency, US Food and Drug Administration, Texas A&M University, Sumitomo Chemical, Shell (The Netherlands), Fraunhofer ITEM (Germany) and Wageningen University (The Netherlands). Further development comprised 51 tasks under four work packages:

1. Improvements to usability,
2. Extensions to the sensitivity analysis module,
3. A new feature for batch processing operations
4. An improved parameter estimation module

In addition, the ability to run models in MCSim syntax was added although this was not an original requirement or deliverable.

RVis contributes to addressing all three priority areas of the LRI program.

1. Innovating chemical testing. RVis can help reduce chemical testing costs, time and animal use. Standard PBPK models rapidly generated using MEGen and exported in R or MCSim syntax can be exercised and analysed using RVis. The utility of in vitro and in silico derived parameters, such as metabolic rate constants and partition coefficients can be tested by incorporating into a model to assess the potential to predict bioavailability of new chemical entities in people and wildlife. Estimates of bioavailability can be used in tiered exposure assessment and integrated assessment and testing strategies (IATA) which help limit animal numbers and inform the design of specific animal bioassays to define critical dose-response information.
2. Understanding everyday exposures to chemicals. PBPK models can be used to predict consumer exposure of new and existing chemicals in commerce. Also, they are powerful tools for the

retrospective reconstruction of exposure from biological modelling data for a wide range of chemical space.

3. Translating research outcomes for product safety. The biological basis of PBPK model structure, the estimation of tissue dosimetry and the inclusion of biochemical mechanisms of toxicity provide the basis for data-informed, quantitative chemical safety and risk assessment. Scientifically supported uncertainty factors derived using quantitative, evidence-based models should increase consumer confidence in product safety.
4.

Possible Regulatory and Policy Impact

Pharmacokinetic (PK) information has an important role in pharmaceutical and non-pharmaceutical chemical safety assessment. In environmental and occupational toxicology, chemical risk assessments using PBPK models are increasingly being used in various jurisdictions (Barton *et al.*, 2007; Loizou *et al.*, 2008). In particular, PBPK modelling approaches are widely recognised as the most appropriate tools for dose–response characterization based on estimates of tissue dosimetry. Their biological basis and the ability to incorporate toxicological mechanisms contribute to a better understanding of, and precision in, assessing risks.

The determination of exposure that corresponds with biological monitoring data (parent chemical/metabolite in blood/urine) is most effectively conducted with PBPK modelling to best simulate xenobiotic disposition in complex mammalian systems. The REACH legislation acknowledges the potential for PK data to influence the development of testing strategies and optimization of study design for industrial chemicals and chemicals used in consumer goods, including food. Recent activity to improve the risk assessment of pesticides and biocides, has led to the revision of Directive 91/414/EC for pesticide use in the EU to include a requirement for the generation and use of PK information. PK also has a vital role in the safety assessment of pharmaceuticals (Baldrick, 2003; ICH, 1995) and chemicals used in cosmetic products. Indeed, the Cosmetics Directive was revised in 2013 to include a complete ban on animal testing making the safety assessment of chemical ingredients in cosmetics products exquisitely dependent on PK for the quantitative in vitro to in vivo extrapolation of concentration-response relationships.

The availability of a resource such as RVis could also have a potentially significant role in three other important areas: the development of internationally recognized good modelling practice (GMP) (Barton *et al.*, 2009; Barton, et al., 2007; Loizou, et al., 2008), rigorous peer-review of PBPK models and software resilience.

Regarding GMP, RVis was designed to capture a sensible workflow where a model structure can quickly and easily be analysed using global sensitivity analysis (GSA). GSA is the most appropriate form of sensitivity analysis for models that describe non-linear processes such as saturable metabolism and receptor binding (Loizou, et al., 2008; McNally, et al., 2011). The open source, open access, free to use philosophy provides transparency and auditability of model code and performance, and has been proposed as important elements of GMP.

The features that foster GMP could also provide a viable and convenient platform for the peer-review of models. That is, models can easily be exchanged and independently evaluated to provide a more rigorous process for publishing in the peer-reviewed literature.

Finally, there have been issues with commercial modelling software support and recently the serious issue of the discontinuation of a widely used product. Access to legacy work conducted with discontinued commercial software is highly problematic. The availability of a robust, free to use, global community supported application such as RVis should offer resilience and help address many of the issues raised and provide the confidence required by the regulatory community. It would serve the industrial, agrochemical, biocide, cosmetic and pharmaceutical chemicals sectors.

This report is a basic outline of the improvements, extended and new features of RVis and a basic user guide.

2. SOFTWARE REQUIREMENTS

Hosting of RVis

The RVis repository is hosted on GitHub here: https://github.com/GMPtk/RVis

Users can download the latest version and post issues that arise that should be addressed.

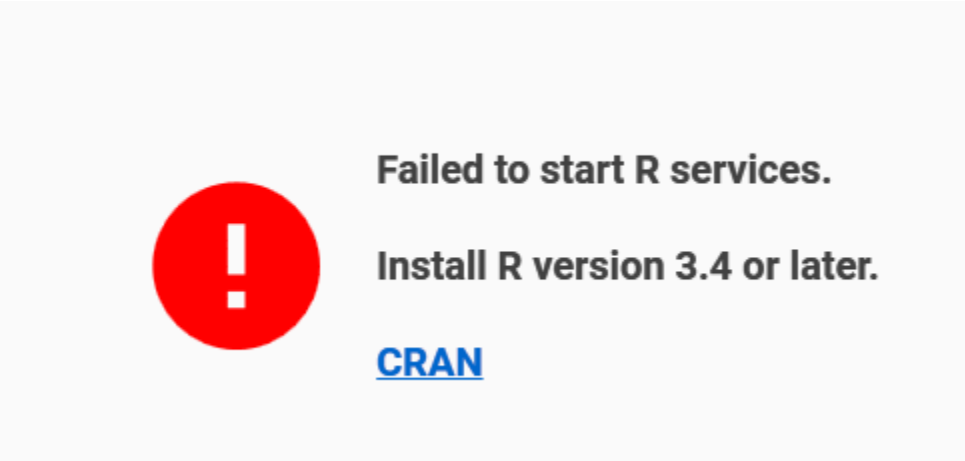
Software Requirements

The R software platform and associated packages: deSolve, Sensitivity, Coda and Rcpp, required to run RVis on a Windows PC are available from: https://cran.r-project.org/. Rtools is available from: https://cran.r-project.org/bin/windows/Rtools/

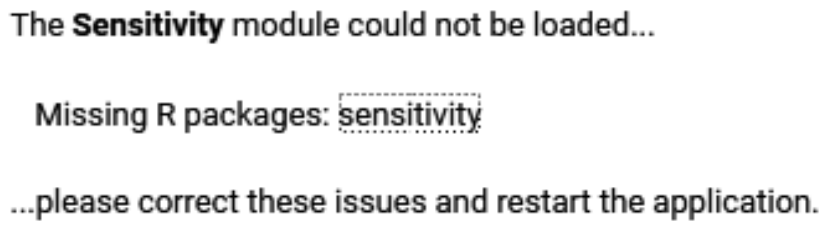
The Rcpp package allows RVis to run PBPK model scripts coded in C++. Whilst this was not part of the original project specification we were able to include this in anticipation of future requirements.

Incorrect Installation

If R is not installed the following error message should appear:



If a specific R package has not been installed e.g., the sensitivity module then an error message similar to the following should appear:



Location of RVis files

RVis (like any software) needs to be run from a directory that has (inherited) the user's default permissions (modify, read and execute, write). Sometimes some files need to be "unblocked" after download in order to avoid security warnings; this may require administrator permissions. Users would need to involve their IT support services. More information is available here: https://blogs.msdn.microsoft.com/delay/p/unblockingdownloadedfile/

Upzip the RVis folder and drag and drop to chosen location. Then click on the RVisUI (purple icon) in the folder to launch RVis.

Files to be used with this guidance

A simple PBPK model for Bisphenol A and two data sets will be made available with this user guidance document in the RVis repository on GitHub.

In addition, the code is available in the Appendix. The entire code can be copied and pasted into R or RStudio and saved. It should run in these platforms. If so, then it will run under RVis.

Data

Two data sets are also available in the Appendix. Save each dataset as a csv or txt file with the column headings exactly as they appear.

Models written in GNU MCSim

Users familiar with MCSim can build models using MCSimViaRtools. Instructions for installation and user guidance can be found here (https://github.com/GMPtk/MCSimViaRtools).

The MCSim syntax for the simple PBPK model for Bisphenol A model is available in Appendix 2 along with the simulation definition file (Template.in) and configuration file (config.R). RVis requires the latter file in order to identify model parameters and outputs specified in the Template.in file. Once installed the user guide in identical for both MCSim and R syntax models.

3. USER GUIDE

RVis is, first and foremost, a labour saving device. The analyses that are currently implemented within the software are not technically novel and could be implemented by expert modellers with the R software using available R packages or bespoke scripts. However, the tools for manipulating model inputs, interacting with model outputs and for visualising the results from high level techniques can result in substantial efficiencies even for expert modellers. Furthermore, the software exploits distributed computing to speed up computationally demanding analyses and thus offers substantial efficiencies for analyses that require many thousands of model runs.

Note that the results from high level techniques (uncertainty and sensitivity analyses and parameter estimation) will be sensitive to modelling assumptions (probability distributions ascribed to inputs, correlation structure, statistical error models etc.). The correct and efficient use of the modules deploying stochastic modelling and the interpretation of the results require specialist skills and expertise. RVis is a labour saving device; it is not a substitute for specialist expertise: it is advisable that users of the software have access to such expertise.

Figure 1 shows a suggested workflow which captures the essential sequence of tasks that constitute good PBPK modelling practice (Loizou, et al., 2008). The user can arrange the sequence of module icons to reflect this workflow (see section 3.2.2).

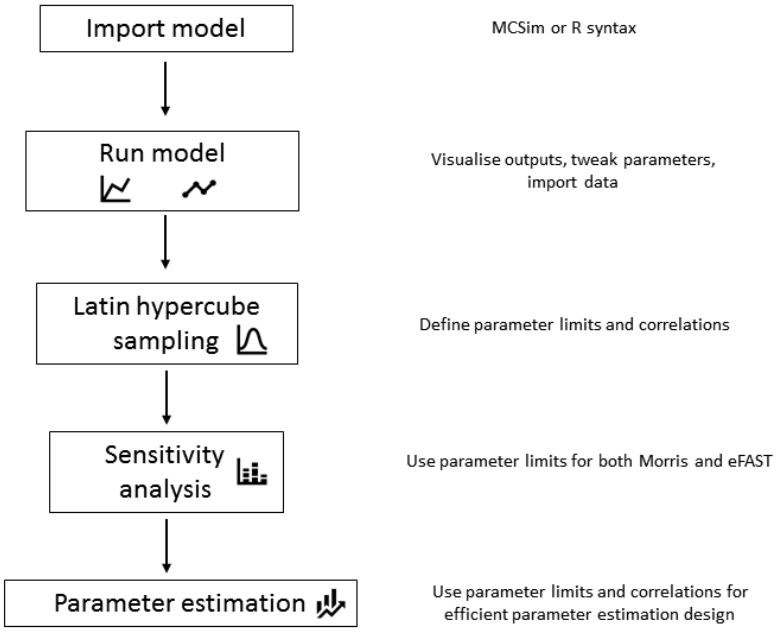


Figure 1 Suggested Good Modelling Practice Workflow

Table 1 Glossary of icons

Icon	Function	Location
	Chart Plotter – View simulation	Right-hand side tool bar
	Sensitivity Analysis – Morris Screening and eFAST	Right-hand side tool bar
	Sampling – Latin Hypercube and Monte Carlo	Right-hand side tool bar
	Parameter Estimation – Markov chain Monte Carlo	Right-hand side tool bar
	Evidence – Import data	Right-hand side tool bar
	Change module configuration	Model caption strip
	Apply/share state between modules	Model caption strip
	Export data	Model caption strip
	Close or remove chart	Model caption strip and Trace field
	Reset, or update if x<y<z entered	(In parameter dialogue box)
	Configure modules. User can set the order of modules in the right-hand margin	Status bar
	Adjust application settings (Ctrl+Shift+S)	Status bar
	Open directory containing application log files	Status bar
	Reset R services	Status bar
	Reset axis ranges to default	In Trace field
	Toggle lock axes' origin to 0,0	In Trace field
	Undo	Parameter working set
	Play	Parameter working set
	Generate samples	Samples field under Latin Hypercube sampling
	View correlation	Samples field under Latin Hypercube sampling
	Upload	Shared state Selected sample
	Download	Shared state

3.1 SIMPLE PBPK MODEL FOR BISPHENOL A

The following user guidance is demonstrated using a simple PBPK model for bisphenol A. The R code is available in the Appendix.

Figure 2 is a schematic of the model showing four compartments; adipose, rapidly perfused, slowly perfused and liver. Metabolism is ascribed only to the liver compartment. An *in vitro* V_{Max} (pmol/min/mg microsomal protein) was scaled to an in vivo V_{Max} (mg/h) using the following equation:

$$Vmax_{in\ vivo} = Vmax_{in\ vitro} \times BW \times VliC \times RMM \times MPYLi$$

Where, BW is body weight (kg), VliC is liver mass (g), RMM is molecular mass (g/mol) and MPYLi is microsomal protein yield (mg/g liver). Exposure is via oral uptake represented by a first-order rate constant, K_a. Model parameters, their symbols, values and sources, where available and appropriate are listed in Table 2.

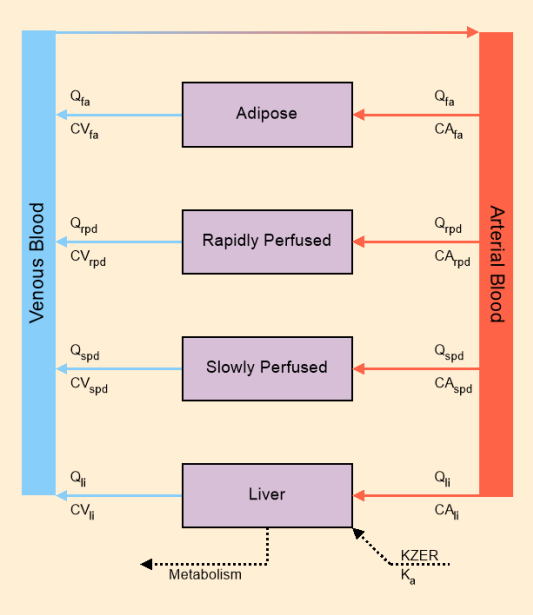


Figure 2 Schematic of a simple PBPK model for Bisphenol A

Model parameters have an associated value that is hard coded in the model script. For probabilistic modelling default parameter ranges were set as follows: the minimum is set to half, and the maximum is set to double the default unless the parameter value is zero, in which case the maximum is set to one. These defaults can be reviewed by a user and amended as appropriate. Ideally, model parameter ranges would be determined from measured distributions for organ and tissue mass and blood flow rates, experimentally measured in vitro metabolic rates and partition coefficients. Organ and tissue mass and blood flow rates are available from PopGen[1] a free to use, online web-based application.

**

[1] http://xnet.hsl.gov.uk/Popgen/

Table 2 Anatomical, physiological, kinetic and physicochemical parameters used in the PBPK model

Parameter	Symbol	Value	Reference
Oral uptake rate (/h)	K_a	0.2	
Oral dose (mg/kg)	PDOSE	10	
Dose in water (mg/kg/day)	DRINK	1.37	
Molecular mass (g/mol)	RMM	228.28	
Cardiac allometric constant (L/h/kg^CAE)	Q_{CC}	11.22	(Krishnan and Andersen, 1994)
Respiratory allometric constant (L/h/kg^RAE)	Q_{PC}	15	(Krishnan and Andersen, 1994)
Body mass (kg)	BW	70	(ICRP, 2002)
Cardiac allometric exponent	CAE	0.75	(Arms and Travis, 1988)
Respiratory allometric exponent	RAE	0.75	(Himmelstein *et al.*, 2004)
Proportion of dead space (not involved in gas exchange)	DS	0.33	(Clewell *et al.*, 2001)
Proportion of vascularised tissue	VT	0.857	(Brown *et al.*, 1997)
Metabolism (MetElim01, Liver)			
Maximum rate of metabolism (molar; in vitro; microsomal) (pmol/min/mg)	$VmaxCivM_{ME01li}$	873	(Hanioka *et al.*, 2008)
Molar Michaelis constant (μM)	$Kmmol_{ME01li}$	8.68	(Hanioka et al., 2008)
Fractional blood flow			
Adipose	Q_{faC}	0.052	(Himmelstein, et al., 2004)
Liver	Q_{liC}	0.25	(Brown, et al., 1997)
Fractional volume			
Adipose	V_{faC}	0.214	(Brown, et al., 1997)
Liver	V_{liC}	0.0257	(Pastino *et al.*, 2000)
Blood:air partition coefficient	Pba	1.43	(Shin et al., 2004)
Tissue:blood partition coefficient			
Rapidly Perfused	P_{rpdb}[2]	2.8	(Shin et al., 2004)
Slowly Perfused	P_{spdb}[3]	0.8	(Shin et al., 2004)
Adipose	P_{fab}	0.7	(Shin *et al.*, 2004)
Liver	P_{lib}	5.7	(Shin et al., 2004)
Overall fractional blood flow			
Slowly Perfused	Q_{spdAC}	0.27	(Brown, et al., 1997)
Overall fractional volume			
Slowly Perfused	V_{spdAC}	0.43	(Brown, et al., 1997)
Microsomal protein yield			
Liver	MPY_{li}	34	(Barter *et al.*, 2007; Howgate *et al.*, 2006)

**

[2] Used value for spleen
[3] Used value for muscle

3.1.1 Importing, Selecting, launching models in R syntax

The initial RVis home page should look like Figure 3. Click on “Import R” to enable the “Browse” button (Figure 4). By clicking “Browse” the user navigates to the location of a model. Make sure the model file has an “.R” extension.

Figure 3 Home page: no models imported

Select model and click “open”. You will be returned to the Import page. Click “Inspect”. RVis inspects the model for correct syntax and structure. If the model is simple inspection is rapid and the user may only see the page flicker. Inspection of a large model will display a rotating progress circle with the message “Run and Inspect”. The “?” is a help button providing a link to RVis documentation on the GitHub repository. More information about using “executive function” or “template” script types can be found here:

https://r-vis.github.io/doc/home/simulation-code/#executive-function

Figure 4 Browse, inspect and import page

The location and format of the executive function in the simple Bisphenol A model can be found on line 160 of the model script in the Appendix.

1. Click “Select”, then click downward arrow on “(unary function)” dialogue box next to “Executive function” and select “mgen_pbpk”.
2. Then click on “(list or vector)” next to to “Parameters”. Then click OK.
3. Then click “Import” to display the import simulation (exec) dialogue box (Figure 5).
4. “Parameters” should be enabled by default. Click “Use All” to make all parameters available during simulations. Alternatively, the user may select a limited number of parameters by clicking the “X” next to a parameter, which changes to “✓”.
5. Click on “Output”. All potential model outputs are displayed. Again the user may select a limited number of those outputs by clicking the “X” next to the output, which changes to “✓”. For purposes of this exercise scroll down and select all outputs (selected independent variables) starting with CV i.e., CV, CVfa, CVli etc., and rel.
6. Only selected parameters and outputs will be imported when user clicks on “Import”.
7. Click Import. The import dialogue page disappears.
8. Click “Simulation” tab at the top left of the home page. Your model should be listed under “Name” (Figure 6). The model is loaded clicking the “Load Simulation” button or deleted by

clicking the “Delete Simulation” button. Place the cursor over the button to display the function.

9. Click “Load Simulation” should display CV [mg/L] versus time (Figure 8)

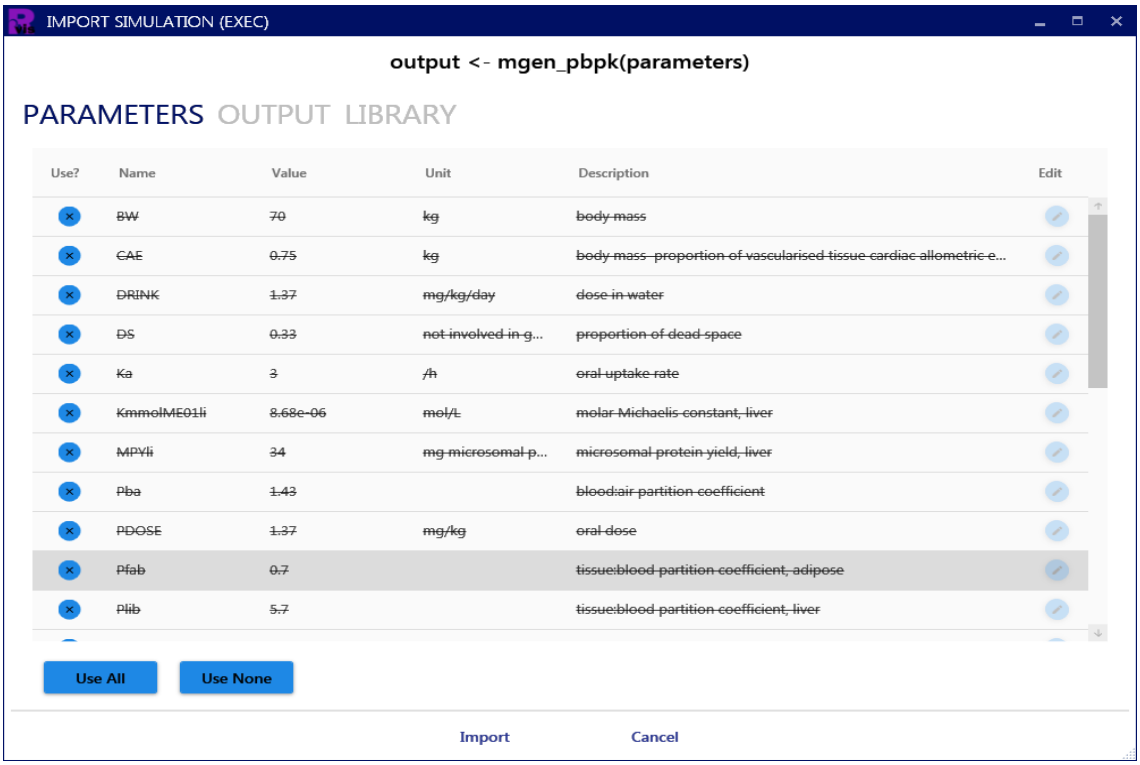


Figure 5 Import simulation (Exec) dialogue box

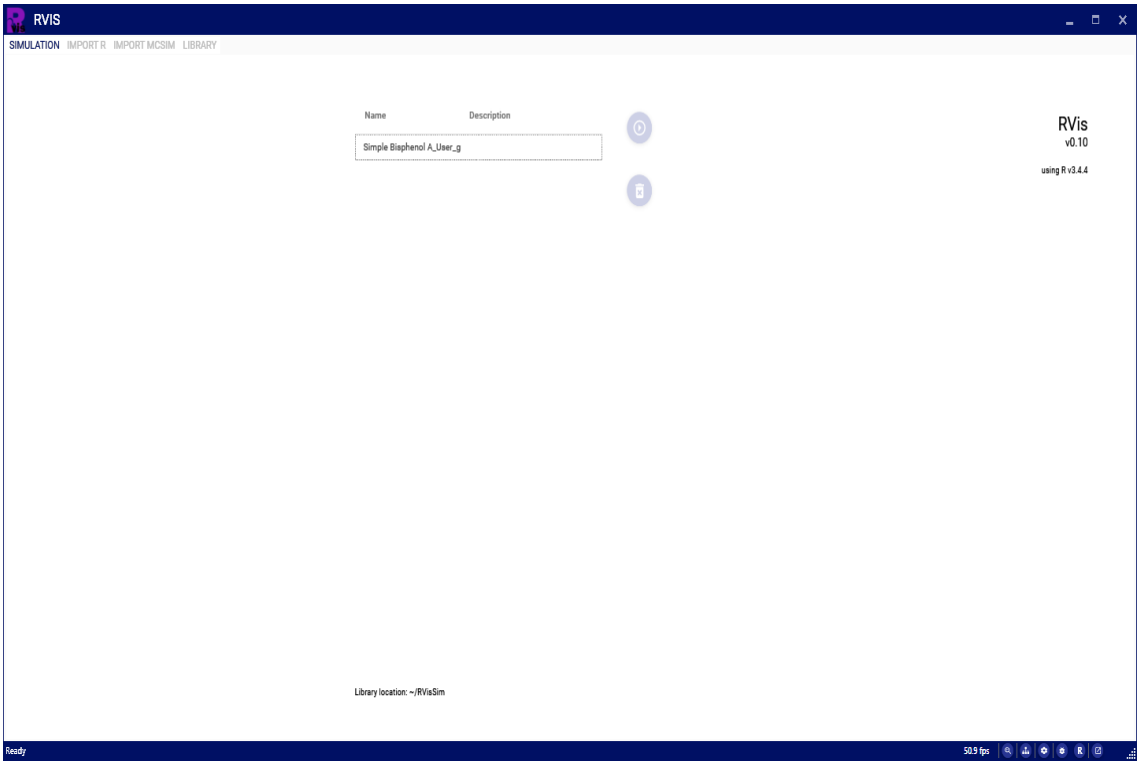


Figure 6 Load simulation page

3.1.2 Importing, Selecting, launching models in MCSIM

Click on “Import MCSIM” to enable the “Browse” button (Figure 7). By clicking “Browse” the user navigates to the location of an MCSim executable. The user should have also created and stored a configuration and template file in the same folder as the model file and executable. Tick on “Load on Import” to launch the model and display a plot (Figure 8).

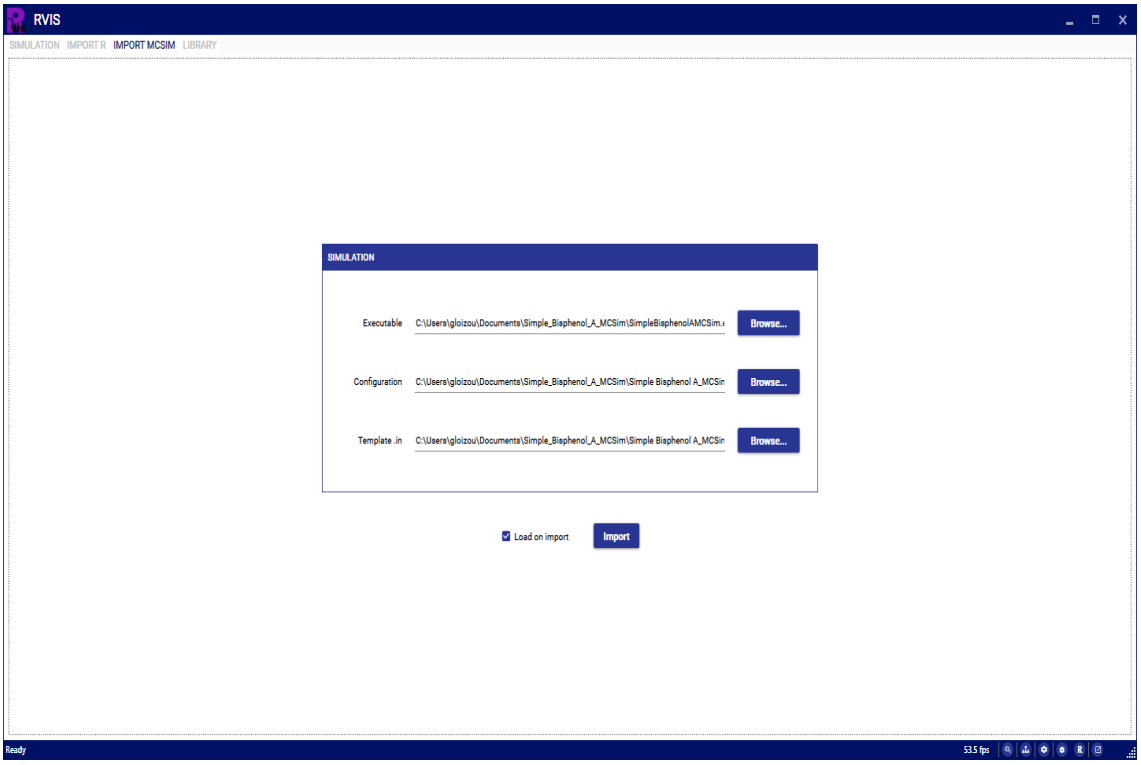


Figure 7 Importing an MCSim executable, simulation definition and configuration file

The simulation definition (Template.in) files have the same function as the dialogue box shown in Figure 5. The configuration file (config. R) is required by RVis in order to identify the model parameters and outputs. Beyond this point the user guide for R and MCSIM models is identical.

3.2 NAVIGATING AND ACCESSING FEATURES AND MODULES

3.2.1 Buttons at top and bottom right side of Trace page

Table 1 is a glossary of icons found throughout RVis. With reference to Figure 8 the name of the application "RVIS" appears in the window caption strip. The name of the uploaded model appears in the model caption strip and below that is the tab strip. The tools bars are on the right-hand side of the chart area. A description of the function of any button on any page is displayed by placing the cursor over the button e.g., "Change module configuration" is displayed when the cursor is placed over the spanner symbol. The other buttons display "Apply Share/state between modules" and "Export Data from…..[name of model currently loaded in RVis]". The latter button is only enabled in the Sensitivity, Sampling and Parameter Estimation modules.

The buttons in the status bar at the bottom of the page are "Configure Modules", "Adjust Application Settings (Ctrl+Shift+S)", "Open directory containing application log files", "Reset R services" and "Toggle full screen (F11)".

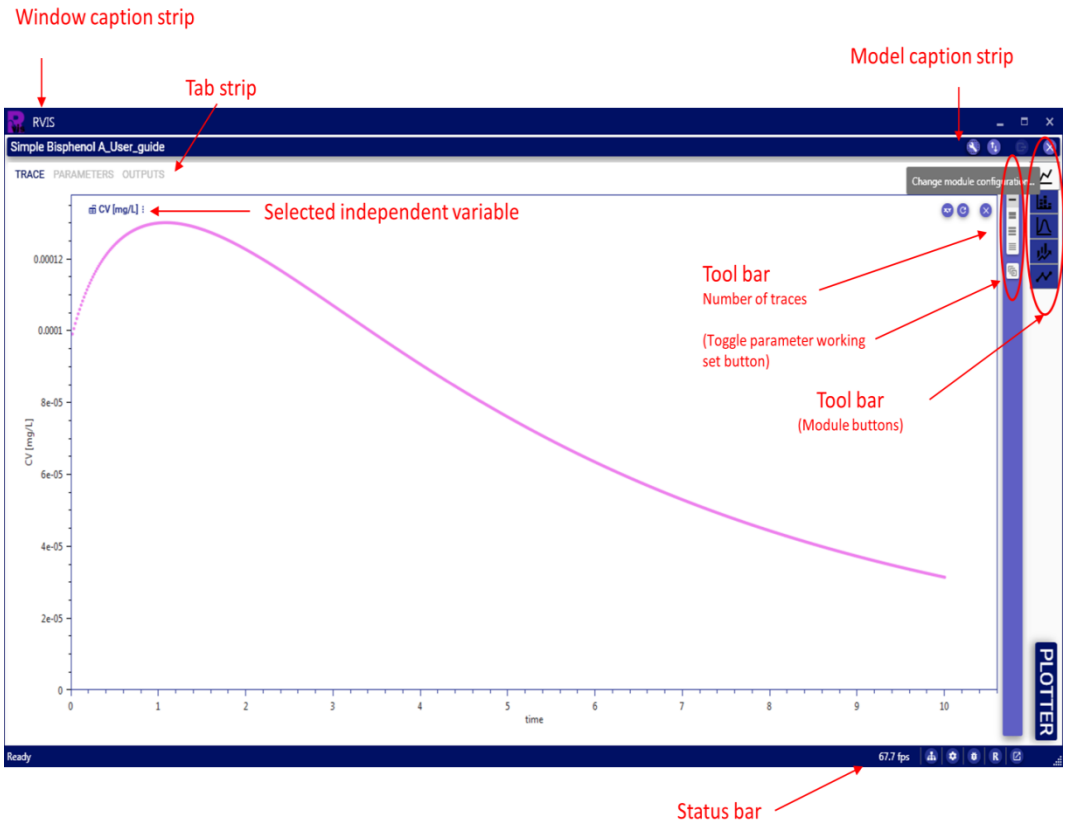


Figure 8 CV [mg/L] versus time

There are four plotter buttons represented by one to four horizontal bars. Clicking on these buttons allows up to four plots to be displayed (see Figure 14 where three are displayed).

Clicking the “Toggle parameter working set button” displays a “working set of parameters” selected by the user on the right margin (Figure 14).

There are five “Module” buttons in the top right margin (see Glossary of icons). From the top the user can switch to the “Chart Plotter” page (Figure 8), below that are the “Sensitivity”, “Sampling”, “Parameter Estimation” and “Evidence” modules. Clicking on  in the lower status bar opens the “Configure modules” page. The sequence of the module buttons can be moved up or down according to the user’s preference. A suggested sequence is; 

3.3 LOADING DATA AND MULTIPLE OUTPUTS

Click on the “Evidence” module button followed by “Browse” on the “Import Observations” dialogue box and navigate to the directory where data files are stored. Select “BPA_CV_dummy_data_10” followed by “Open”. The name of the data set should appear in “Set name” field (Figure 9). RVis currently accepts .csv or .txt files. It is important that the column titles in these data files are identical to the outputs displayed in RVis, in this case “CV” versus “time”. Only the output name and not the units must be included in the data file columns. The data values are displayed in rows under “Observations”. A green tick indicates the data are ready to be imported.

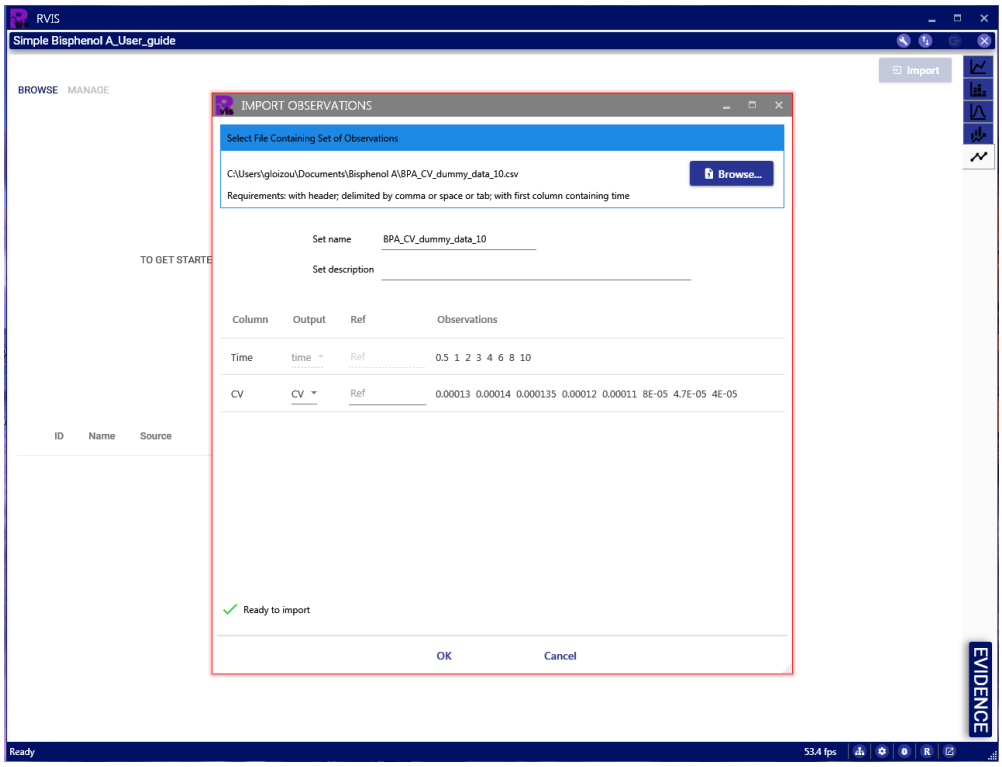


Figure 9 Import data

1. Click on CV 0/1 in the top left corner, and then click the box to select the data. The data will be displayed as in Figure 10.
2. Click on Plotter button which should return to Figure 8.
3. Click on CV [mg/L] to open Traces dialogue box (Figure 11).
4. Click box next to “CV x 8 from BPACV_dummy_data_10 in the “Observations” field.
5. Multiple outputs can also be configured from here by clicking on the boxes in the “Supplementary Traces” field (as shown in Figure 11).
6. Click on CVfa, CVli, CVrpd, CVspd and click “Close” to remove Traces dialogue box (Figure 12).

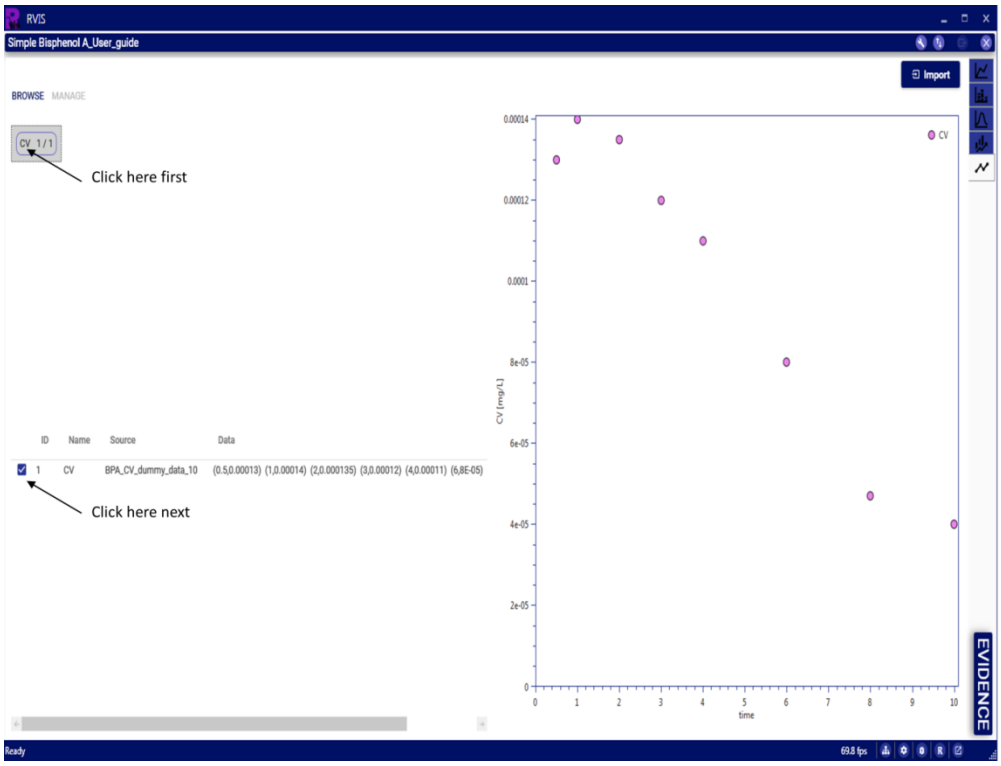


Figure 10 Accept selected data

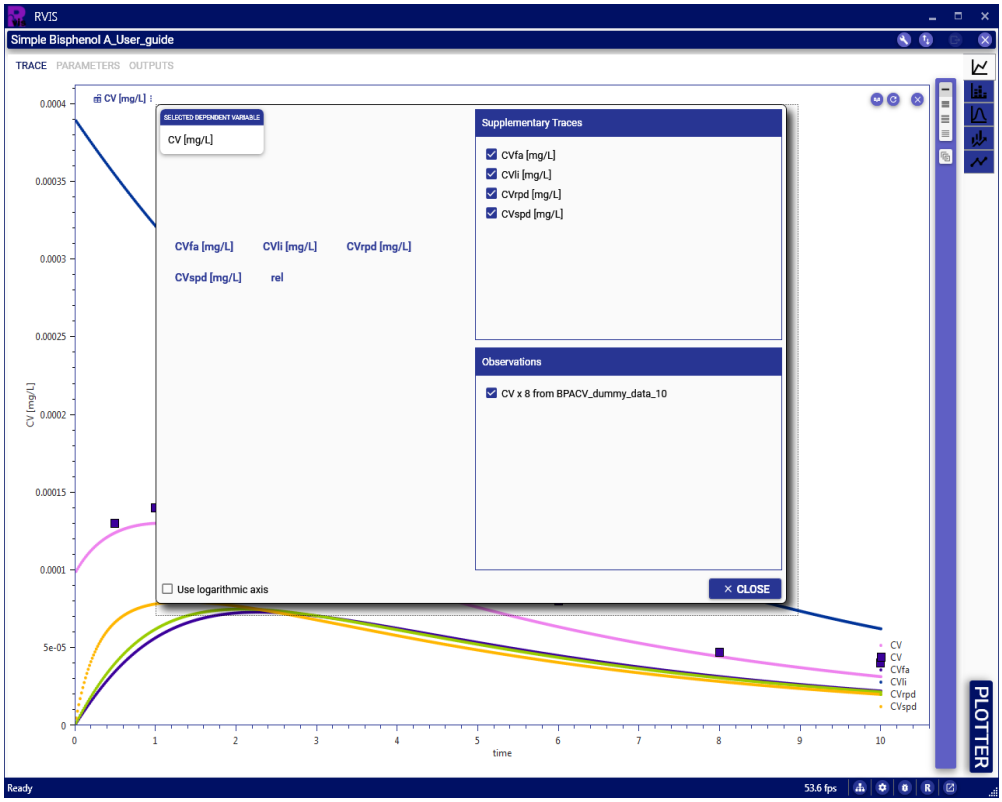


Figure 11 Imported data and multiple traces

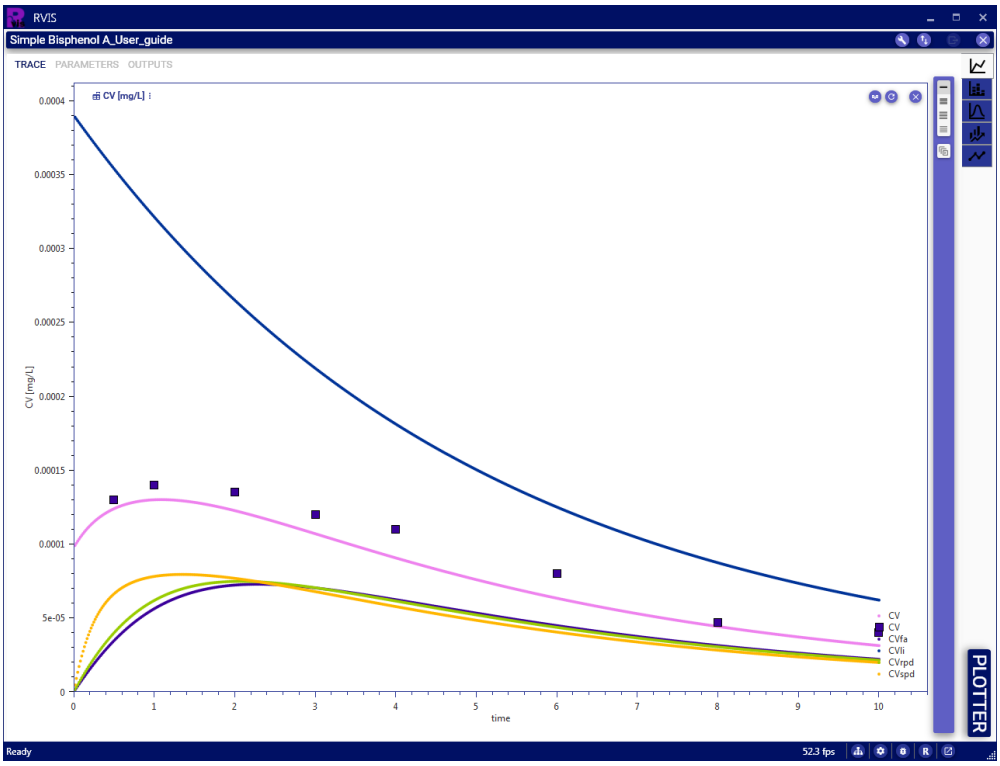


Figure 12 Plotting multiple outputs and data

3.4 SELECTING AND TWEAKING PARAMETERS

Tweaking parameters using sliders is a quick and convenient way at examining their sensitivity. It is not a replacement of a formal sensitivity analysis but can be a useful tool for testing model behaviour.

Click on “Parameters” tab. All available parameters appear in the “Parameter Pool” field. Click on “+” to select parameter. A selected parameter shifts from the parameter pool field to the “Working Set” field (Figure 13). Select all parameters listed in Table 3.

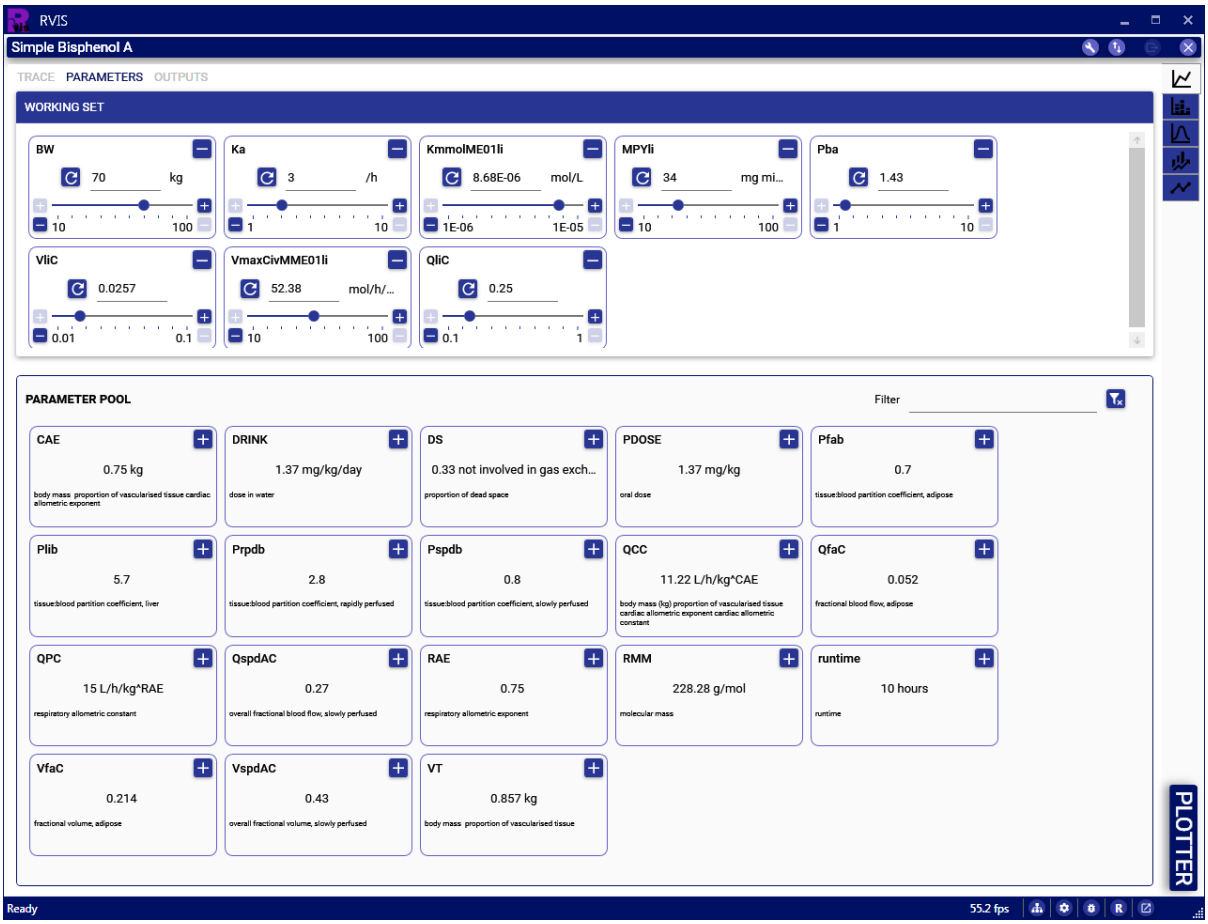


Figure 13 Parameter selection

1. Click on “Trace” to return to Figure 12.
2. Click on “Toggle parameter working set” button (see Figure 8). User’s working set of parameters will appear in the right hand margin (Figure 14).
3. Click on the three bar Plotter button (see Figure 8). Three charts appear. Click on CV [mg/L] in top chart and deselect all supplementary traces, to leave the CV simulation and imported data. Note: Set PDOSE to 1.43 when using BPA_CV_dummy_ data_ 1_43 or 10 when using BPA_CV_dummy_ data_ 10.

4. Click on “Selected independent variable” in each chart and select an independent variable and a supplementary trace to plot (User’s choice!) (You should get something similar to Figure 14. Each simulation is annotated and colour coded.

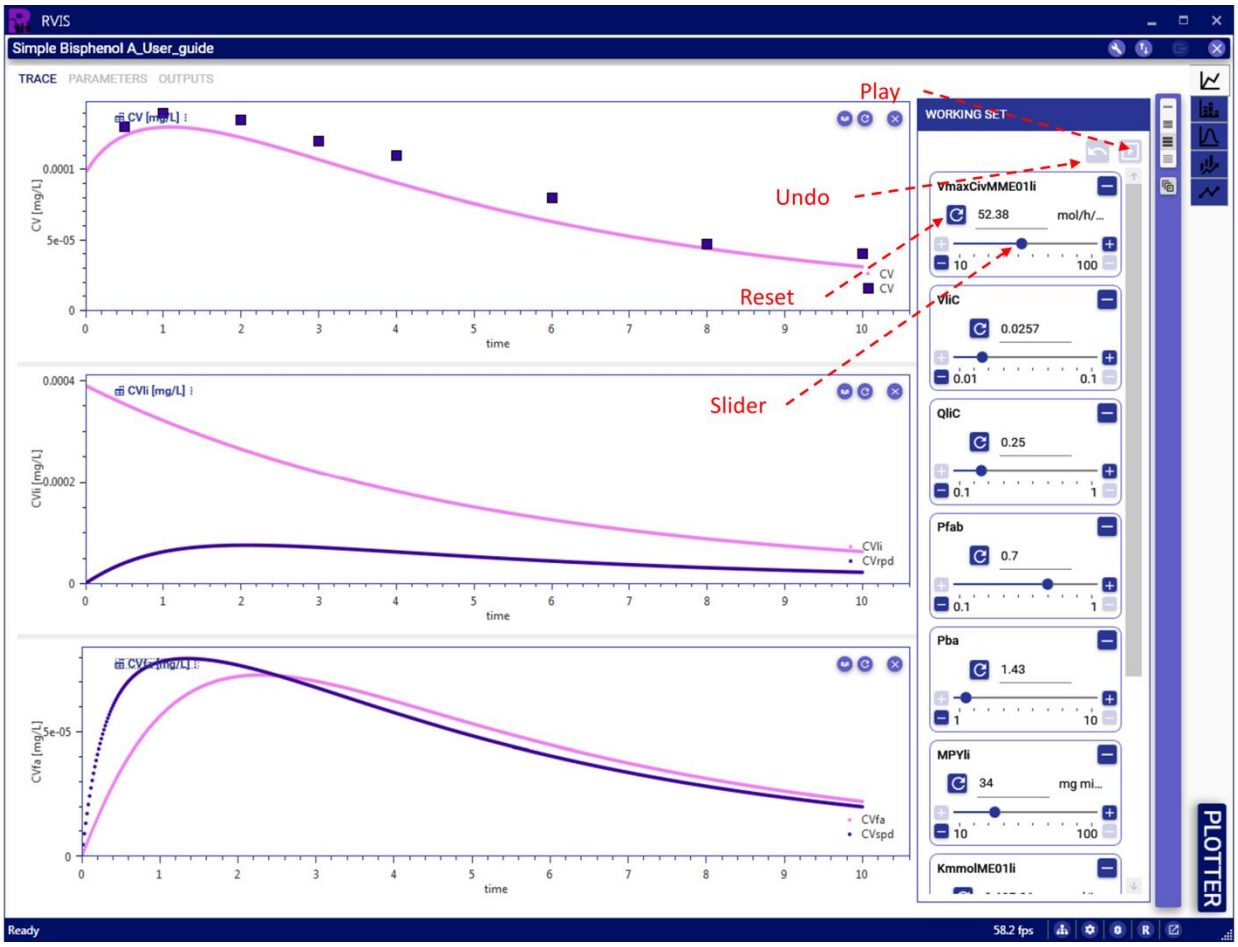


Figure 14 Tweaking parameters and multiple plots

Sliders

1. In the “Working Set” of parameters go to the VliC parameter box and right-click mouse and keep pressed on filled-circle to move slider from right to left and vice versa (Figure 14). The outputs in the Trace fields should respond.
2. Parameter default values can be restored by selecting the value, deleting and then clicking on the reset button.
3. A range of parameter values can be tested by using R code in the following format: c(1,2,3), where 1, 2 and 3 are feasible values for that parameter. For example, delete the value in the MPYli (microsomal protein yield) parameter box and type c(10,34,60) in the field. Then click “Play” button. Three simulations corresponding to the three parameter values are displayed (Figure 15). In this case the effect of three different MPYli values used to calculate whole liver metabolic rate. The simulations are annotated and colour coded in the chart field.

Figure 15 Parameter ranges

3.5 SAMPLING

The sampling module operates as a two-step process. In the first phase the user specifies probability distributions for the varying inputs in the model and creates a design. The design is a matrix of points, with N (user specified) rows corresponding to independent model runs and columns corresponding to the parameters to be varied. Once a design has been created the user may scroll through the points to check for anomalies prior to running the design. The sampling distribution of each input is displayed as a histogram once the design has been created so the ranges and probability distributions can be checked prior to execution. In the second phase the design is executed, each row of the design is used to overwrite the current values of model parameters and model outputs are acquired and stored. Two options, Monte Carlo and Latin Hypercube are available for generating a design.

Monte Carlo sampling

Monte Carlo (MC) simulation generates a random sample of N points for each uncertain input parameter variable of a model. It selects each point independently from the probability distribution ascribed to that input parameter variable. It generates a sample of N values or scenarios for each result variable in the model using each of the corresponding N points for each uncertain input parameter value. From this random sample for each result, it estimates statistical measures such as mean, standard deviation, fractiles (quantiles) and probability density curves. Because it relies on pure randomness, it can be inefficient. You might end up with some points clustered closely, while other intervals within the space get no samples. Latin hypercube sampling removes this inefficiency.

Latin hypercube sampling

Latin Hypercube sampling (LHS) aims to spread the sample points more evenly across all possible values. It partitions each input parameter distribution into N intervals of equal probability, and selects one sample from each interval. It shuffles the sample for each input so that there is no correlation between the inputs (unless you want a correlation). Centred or Median LHS uses the median value of each equi-probable interval whereas; randomised LHS selects a random point within each interval. Latin hypercube designs can also have poor space filling properties therefore a Latin hypercube with good space filling properties is selected using a (user configurable) maxi-min criterion to ensure a space filling design is generated. This option is particularly well suited for exploring the bounding behaviour of a model using only a small (several hundred) number of runs of the model.

Rank correlation

The default designs created under Monte Carlo and Latin Hypercube options assume the input parameters are statistically independent. A user-specified correlation structure can be induced prior to executing the design to change this default assumption of independence. A rank correlation structure is supported, which works for arbitrary probability distributions. A re-ordering of the values within the columns is undertaken to induce the correct user-specified rank correlation structure between model inputs. The marginal distributions are identical to those of the default (independent) design. RVis progressively limits the user's choices of correlations (to a narrower range than -1 to + 1) as a correlation

matrix is populated to ensure the correlation matrix is internally consistent (formally this is a positive definite matrix).

3.5.1 Monte Carlo Sampling

1. Click on the Sampling module icon
2. Click on “Select Parameters”. This should display all available model parameters with a box next to each. Click box to select parameter. Selected parameters are listed with an assigned sampling distribution in the field below the parameter selection field.
3. Click on a parameter to display the distribution e.g., BW in Figure 16.
4. The following distributions are available from the drop down menu to right of the parameter selection grid: normal, invariant, log normal, uniform, beta, beta scaled, gamma and student’s t. When used for the first time the mean, standard deviation, lower and upper range fields will be empty. If this is the case the user may use the distributions and values listed in Table 3. Parameters with a normal distribution are those for which we have some knowledge and have been obtained from the literature or PopGen our virtual healthy human population generator (McNally et al (2014)). Uniform distributions are ascribed to parameters for which we know little such as partition coefficients generated using a predictive algorithm. In this case the user can use half and twice the mean as the lower and upper bounds or a reasonable range above and below the point value. Truncation based upon a user specified lower and upper bound is supported for all unbounded probability distributions.
5. Click on the Samples tab, enter N=100 in the No. of samples field and 1 in the seed field (assigning a seed allows the user to reproduce the sample design).
6. Click on the “Correlation” button below the Hypercube button to enable the “Configure Rank Correlation” dialogue box (Figure 17**)**. A Correlation matrix appears where the user can set the degree of correlation (-1 to 1) between parameters. To leave all values at 0 click “Cancel” to retain the red filled circle, this indicates that no correlations have been configured.
7. Click on to “Generate samples”. One hundred rows of parameter values will be displayed and can be reviewed by scrolling (Figure 17).
8. Click on the Design tab, and then click on “Create Design” in the Design dialogue box. Then click on “Start” in the Acquire Outputs dialogue box. Progress toward 100 simulations can be monitored here.
9. Click on the Outputs tab when 100/100 simulations is reached. One hundred simulations are displayed in the chart area. Place cursor on a single simulation, left click and hold. The line number, time point along simulation and value of output (CV) at that time point are displayed. The line number corresponds to the sample number from the design and the parameter values producing that simulation are listed in the “Selected Sample” field (Figure 18). This feature allows usual runs to be quickly identified.
10. Click on upload icon to upload sample into Shared State.
11. Click on Plotter icon to return to trace window (similar to Figure 8).

12. Click on to open shared state (Figure 26). The parameters and values corresponding to the line selected in the Monte Carlo output page are listed.
13. Click on download icon, . The selected simulation form the Monte Carlo output page is reproduced allowing comparison with data.
14. Click on the Sampling icon to return to the Output page of the Sampling module.

This process can be used to test model structure and stability and to set lower and upper limits for parameter ranges to be used in sensitivity analysis and parameter estimation. Data can be uploaded to figures as described in section 3.3 to allow a comparison against the breadth of simulations corresponding to the samples.

3.5.2 Latin Hypercube Sampling

Configuring LHS is very similar to MC. Note that parameter distributions have already been specified.

1. Click on the Design tab on the MC outputs page to return to the Design page.
2. Click on "Unload Design" and then click on the Samples tab. The samples field should be empty.
3. Click on the "Hypercube" button in the Configuration box to enable the LHS configuration dialogue box (Figure 19). Ignore the various options for now. Ensure "Use simulated annealing" is not ticked. And click OK to return to the Samples page. There should be a green tick where there was a red filled circle indicating successful configuration.
4. Click on the "Correlation" button below the Hypercube button to enable the "Configure Rank Correlation" dialogue box (Figure 20**)**. A Correlation matrix appears where the user can set the degree of correlation (-1 to 1) between parameters. Click "Cancel" as in Step 6 for MC configuration to leave all values at 0.
5. Click on to generate samples. The samples field is populated as previously for MC.
6. Click on the Design tab and click on "Create Design" followed by "Start" under "Acquire Outputs" as before. When 100 simulations are complete click on the Outputs tab.
7. Repeat steps 9 to 14 under Monte Carlo Sampling.
8. Click on the Design tab and click on "Unload Design".
9. Click on the Parameters tab in the Sampling module to return to Figure 166.
10. Click on to enable the Shared State dialogue box (Figure 26**)** then click on to upload parameters to Shared State.

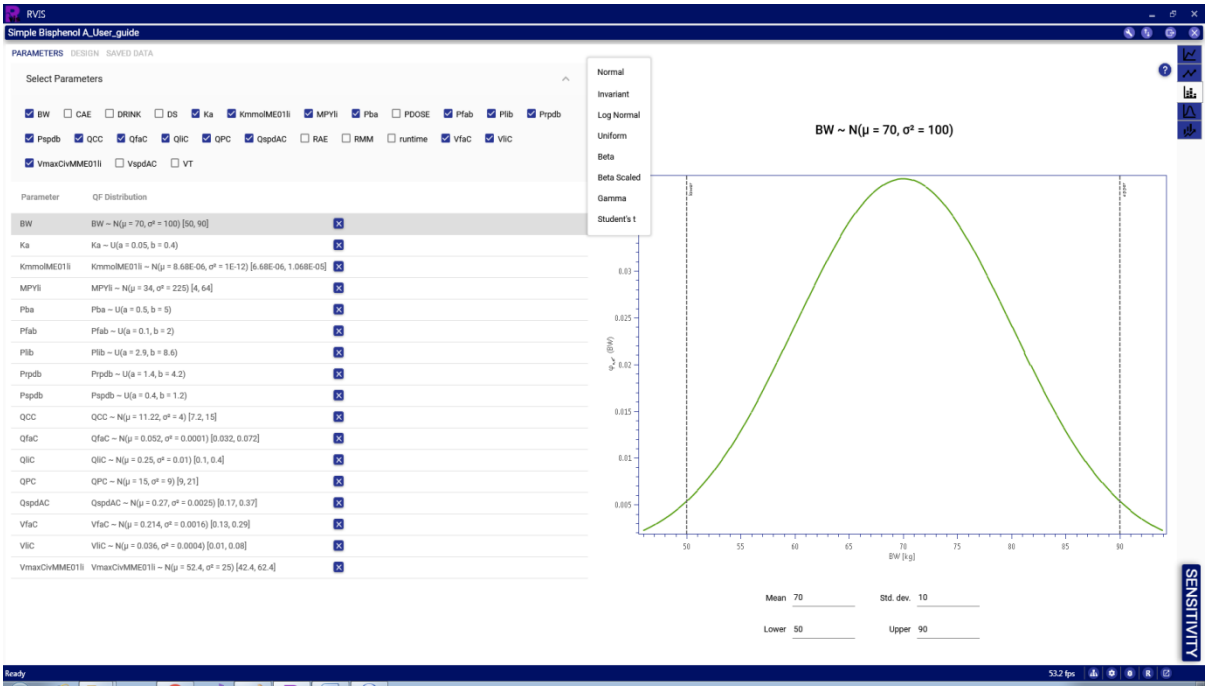


Figure 16 Sampling configuration page

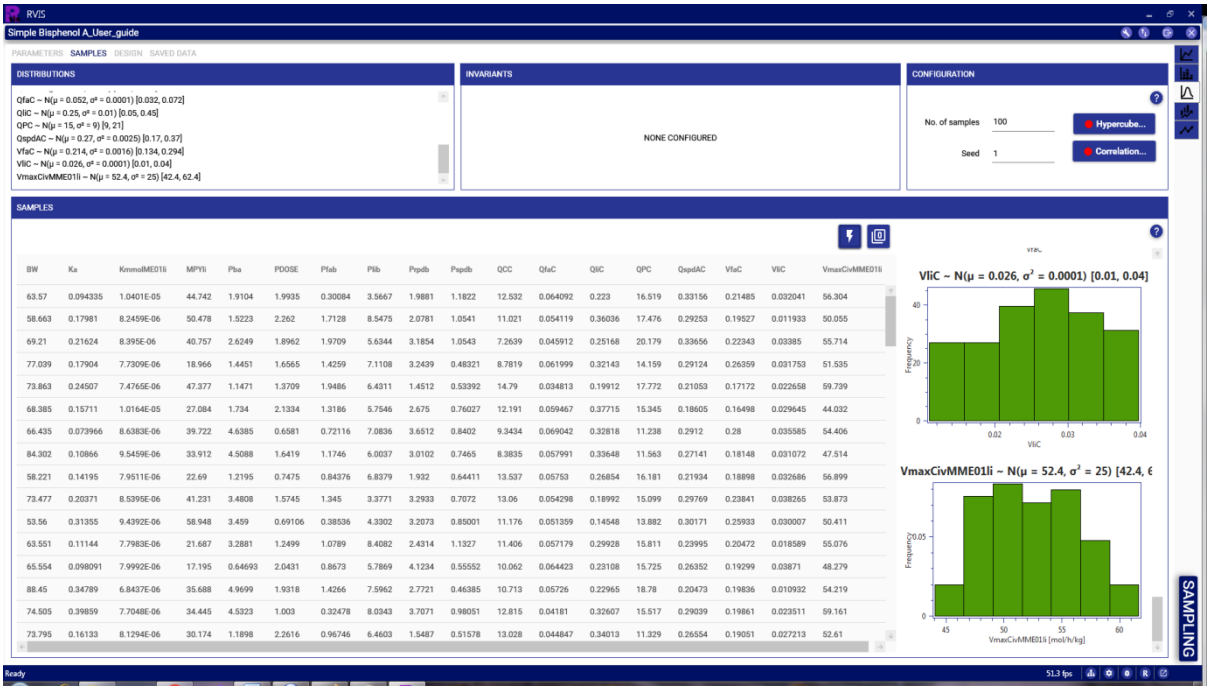


Figure 17 Sampling: Samples

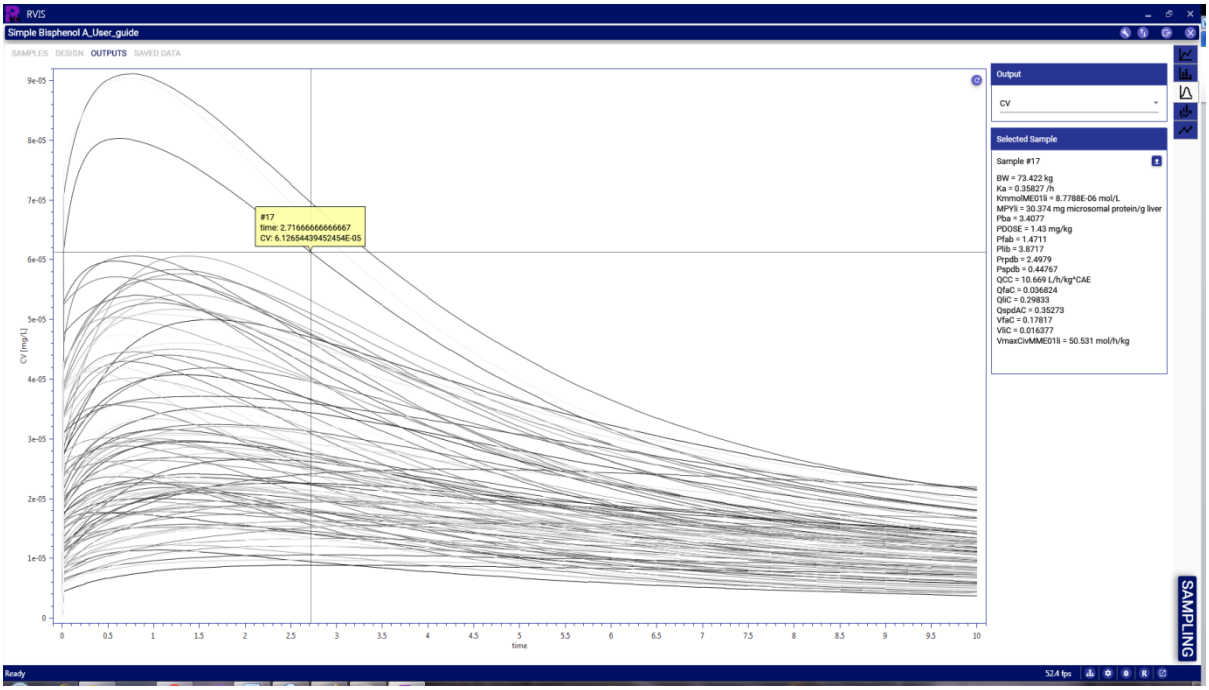


Figure 18 Selection of simulation

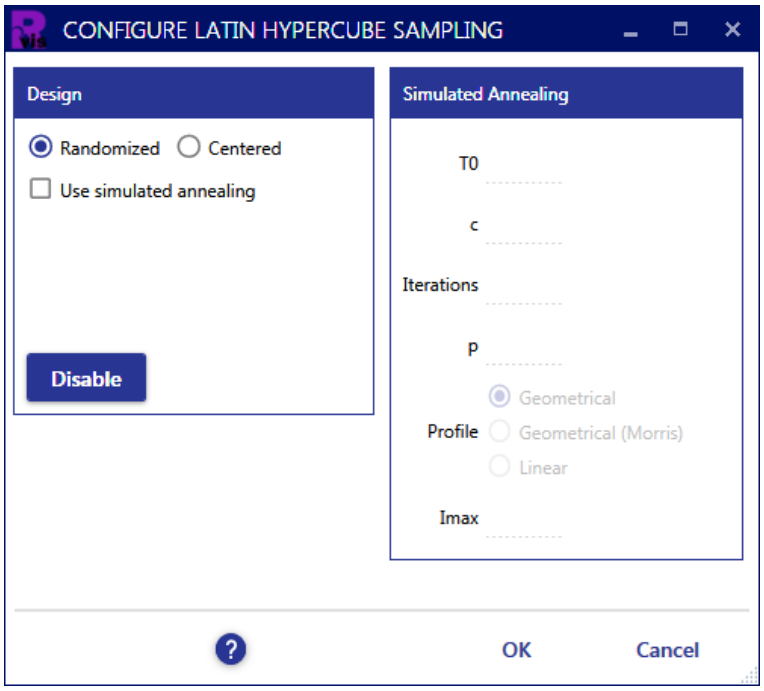


Figure 19 Configure Latin Hypercube Sampling

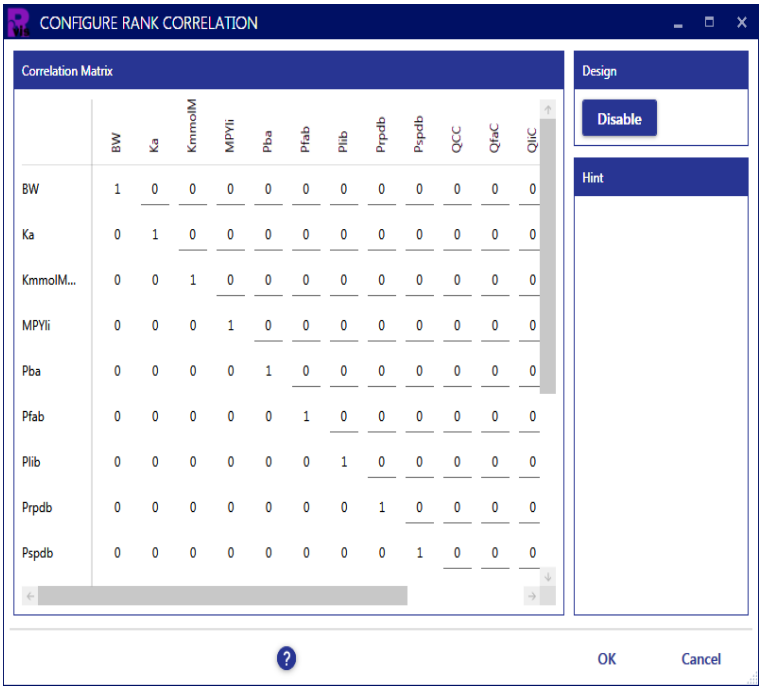


Figure 20 Configure Rank Correlation

3.6 SENSITIVITY ANALYSIS

Two methods for global sensitivity analysis are supported in RVis: elementary effects screening, widely known as the Morris Test and the extended Fourier Amplitude Sensitivity Test (eFAST), which is a variance based method. The workflow described below is the two-step process is described in McNally et al. (2011) and comprises of elementary effects screening to efficiently screen a subset of parameters with a negligible influence on the outputs under study, prior to undertaking a more computationally expensive variance-based analysis using eFAST.

1. Click on the sensitivity module icon and then on and then on to download parameters from Shared State. Click on BW to display distribution. You should have a similar page to Figure 16.
2. Click box to select parameter. Add QPC with values listed in Table 3. Some parameters are not included in the Morris screening as they may be inappropriate for sensitivity analysis e.g., Molecular mass (RMM), runtime. Note that whilst shared-state allows the parameter distributions that have been previously ascribed to be quickly downloaded, distributions can be ascribed (and shared) within the sensitivity analysis module, or downloaded and modified prior to running the sensitivity analysis
3. Click on "Design" tab, selected parameters and distributions are listed in the Factors/Distributions field. Ensure the Morris radio button is selected. The number of runs is set to 6 as default, if not enter 6. The Morris test is a stochastic process and RVis is able to rank the consistently most significant parameters over the six iterations.

Table 3 Morris screening test

Parameter	Distribution	Mean	SD	Lower	Upper
BW	Normal	76.2	2.95	70.3	82.1
Ka	Uniform	-	-	0.05	0.4
KmmolME01li	Normal	8.68E-06	1E-06	6.68E-06	1.068E-05
MPYli	Normal	32.3	2.3	27.7	36.9
Pba	Uniform	-	-	0.5	5
Pfab	Uniform	-	-	0.1	2
PDOSE	Invariant	1.43	-	0.5	3
PDOSE	Invariant	10	-	5	15
Plib	Uniform	-	-	2.9	8.6
Prpdb	Uniform	-	-	1.4	4.2
Pspdb	Uniform	-	-	0.4	1.2
QCC	Normal	11.22	2	7.2	15
QfaC	Normal	0.052	0.01	0.032	0.072
QliC	Normal	0.25	0.1	0.1	0.4
QPC	Normal	15	3	9	21
QspdAC	Normal	0.27	0.05	0.17	0.37
VfaC	Normal	0.214	0.04	0.13	0.29
VliC	Normal	0.026	0.01	0.01	0.04
VmaxCivMME01li	Normal	52.4	5	42.4	62.4

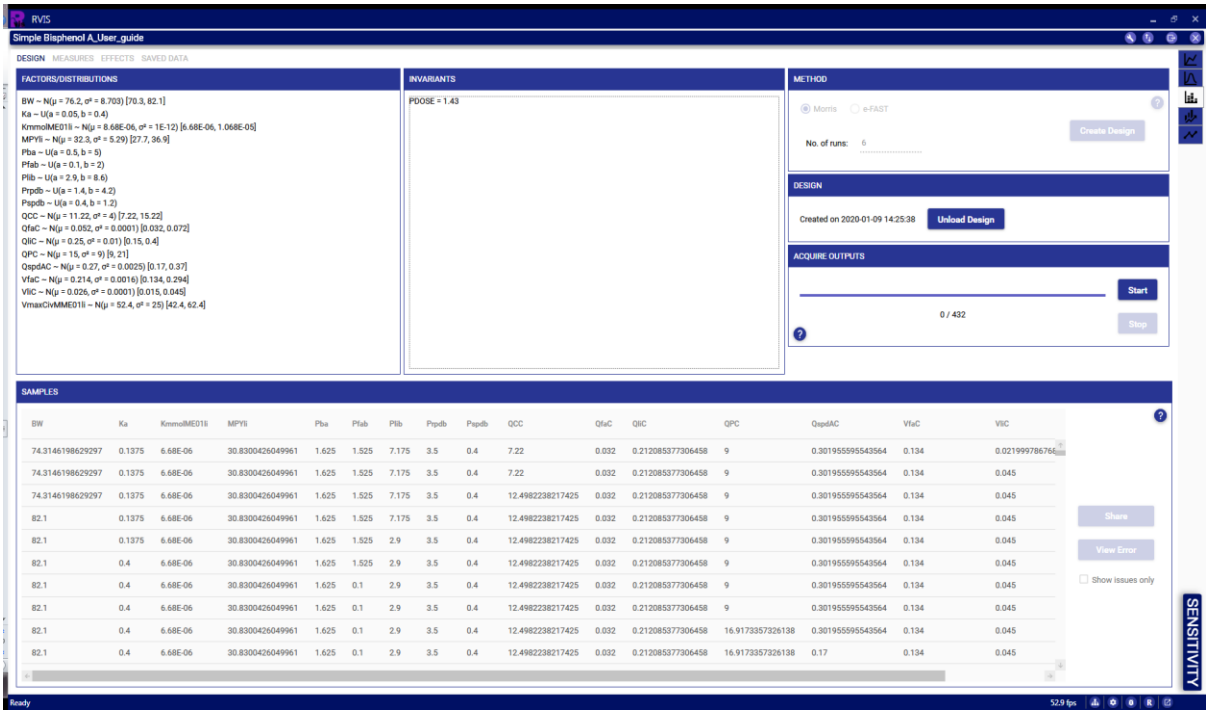


Figure 21 Morris test design page

4. Click “Create Design”. The design is displayed in the “Samples” field (Figure 21)
5. Click “Start” in the “Acquire Outputs” field. The target number of iterations and progress to the target can be monitored. When the target is achieved Morris output measures are generated and a chart a “Measures” chart is plotted (Figure 22).
6. Click on the “Effects” tab above the Measures chart to display the output measures, σ versus μ* (Figure 22)
7. Click the play button in the “Simulation” field. The simulation speed may be increased as required. Changes in parameter importance with time can be quickly ascertained. Stop simulation.
8. In the lower panel right click on the dashed vertical line in the plot of CV versus time and drag the line left or right to determine the maximum spread of parameters in the upper panel. Expanding the spread of parameters should help identify clusters and the most consistently important parameters. Aim to place Ka as close as possible to the upper right hand corner as in Figure 23.
9. Click on “Measures” to return to previous plot. Ensure Output is CV. In the “Ranking” dialogue box enter 0.3 and 5 in the “From” and “To” field. The area from 0.3 to 5 in the plot is highlighted (Figure 22). RVis will select the most important parameters throughout the user defined time period.
10. Click on the ⚙ in the Ranking dialogue box. The “Rank Parameters dialogue box is enabled (Figure 24). Click on CV and select the highest ranking parameters. User judgement is required.

This will require a combination of looking for clusters in the Effects (σ versus μ*) plot and the value in brackets next to the ranked parameters. In Figure 24, VliC has a value of 0.632 and QPC a value of 0.124. The next parameter is Prpdb with a value of 0.0663 which is almost half. This may mark a cut-off point.

Note: the Morris test is a stochastic process which means that the parameter ranking and values will differ slightly each time the test is run.

11. Click "OK" to close the Ranking dialogue box. A green tick appears next to each selected and ranked parameter (Figure 22).
12. Click "Use" and RVis returns to the Sensitivity design page (Figure 21) except that the selected and ranked parameters only appear in the Factors/Distributions field (Ka, KmmolME01li, MPYli, Pba, QCC, QliC, QPC, VliC, VmaxCivMME01li).
13. Click on the e-Fast radio button and click on "Create Design". The design for the e-FAST test is displayed in the "Samples" field.
14. Click "Start" in the "Acquire Outputs" field. The number of samples is displayed (default = 1000). The number of iterations is calculated as follows: No. of samples × No. of parameters. In this case it should be 9000. Progress toward 9000 iterations is displayed in the "Acquire Outputs" field. When complete a "Generate eFAST Output measures" is displayed and a graph of "Measures" i.e., Main or Total Effect or Variance versus Time plotted (no image provided).

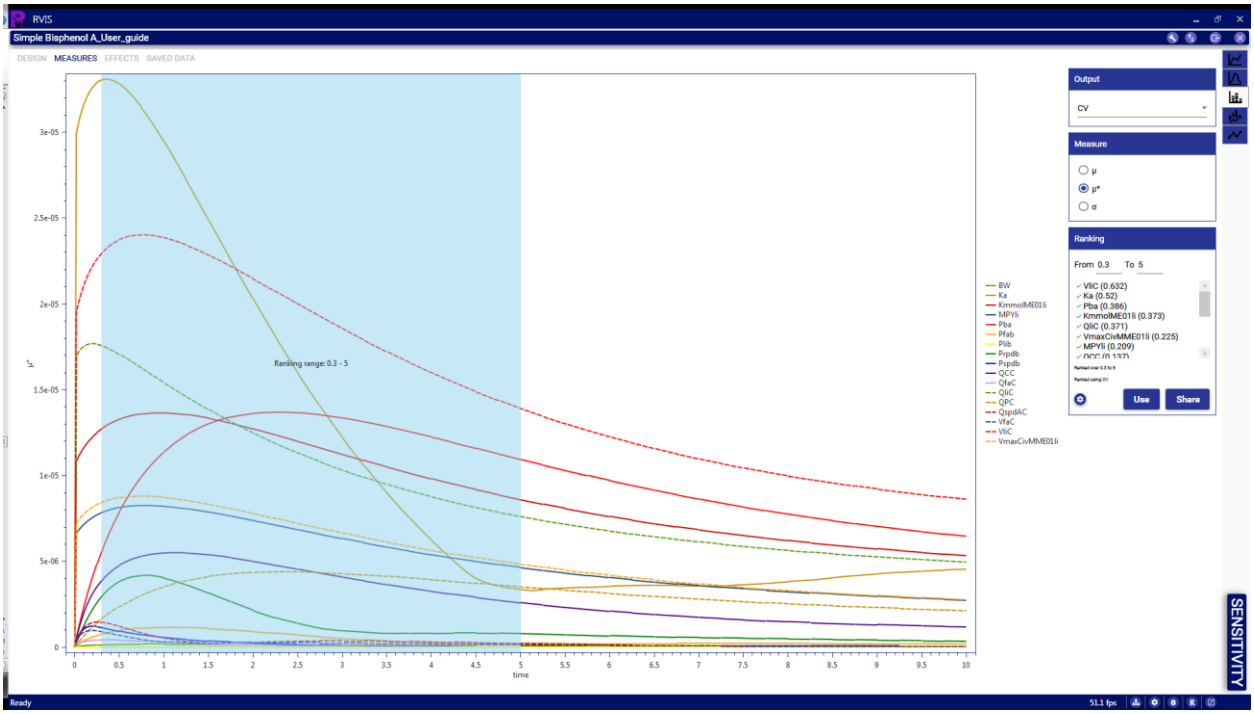


Figure 22 Share selected and ranked parameters

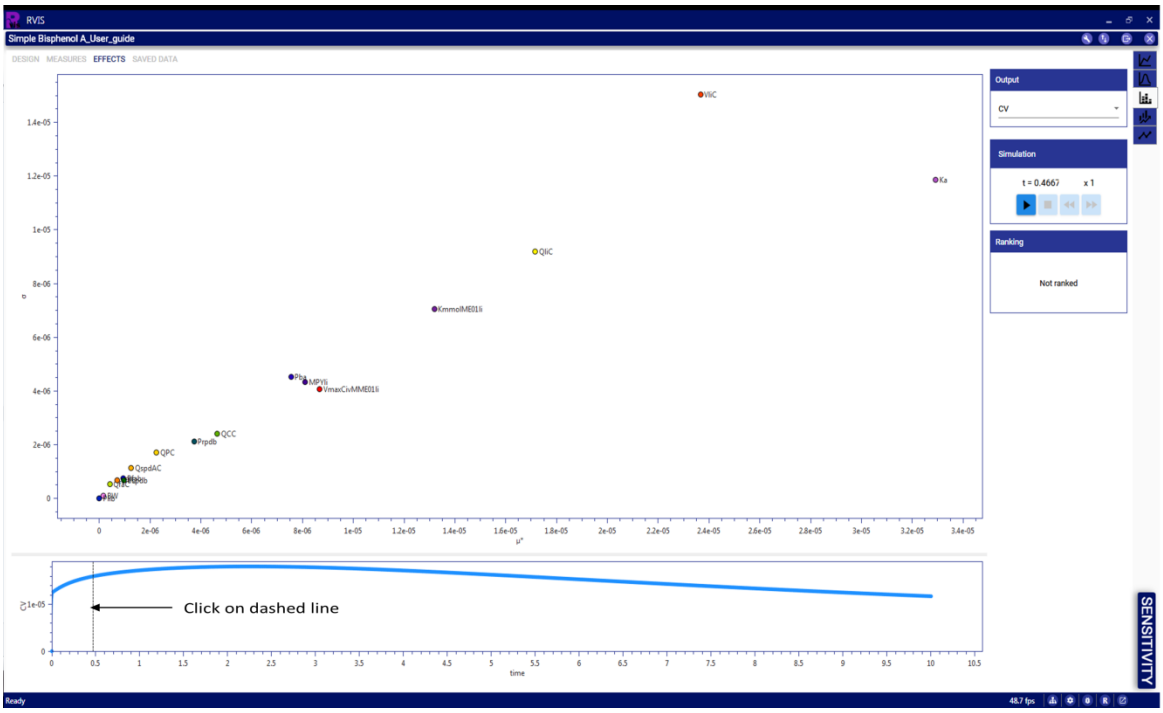


Figure 23 Morris Effects page: σ versus μ*

15. Click on the "Effects" tab. The eFAST measures are expressed as a Lowry plot (Figure 25) (see McNally et al (2011) for a more detailed description).
16. The change in eFAST sensitivity indices (Total = Main Effect + Interaction) with time can also be viewed by clicking on the play button in the "Simulation" field. Changes in the magnitude of the histograms, the size of the plume or even the ranking of the parameters which decrease in significance from left to right can be visualised. The number of parameters that account for 100% of variance in CV can be determined by running a line from 1 on the y axis to the plume and then a vertical line down to the x axis. All parameters to the left of that line account for 100% variance. In this simple example all the parameters appear to the left!
18. The changes in parameter Total Effects with time can be viewed where a range of time over which the average "Total Effect" for each parameter should be estimated can be set. The approach is similar to setting the range on the "Measures" page of the Morris test.
19. Click on the "Measures" tab and in the "From" and "To" fields in the "Ranking" box enter 0 and 7. An area from 0 to 7 hours is highlighted in the chart area.
20. Click on the ⚙ in the Ranking dialogue box and tick the box next to "CV". Select all the important parameters and click OK to close the "Rank Parameters" dialogue box. The selected parameters have a green tick and are listed in the Ranking dialogue box.
21. Click "Share" to upload selected parameters into "Shared State". This operation can also be conducted from the Effects (Lowry plot) page.
22. Click on ⇅ to open Shared State dialogue box, and then click ⬆ to upload selected parameters.

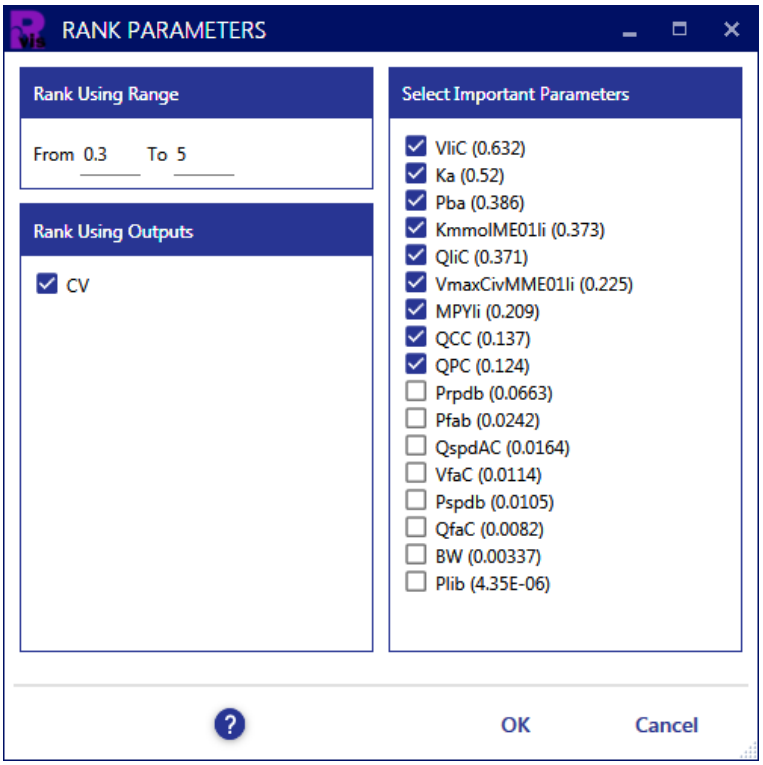


Figure 24 Ranked parameters following Morris screening

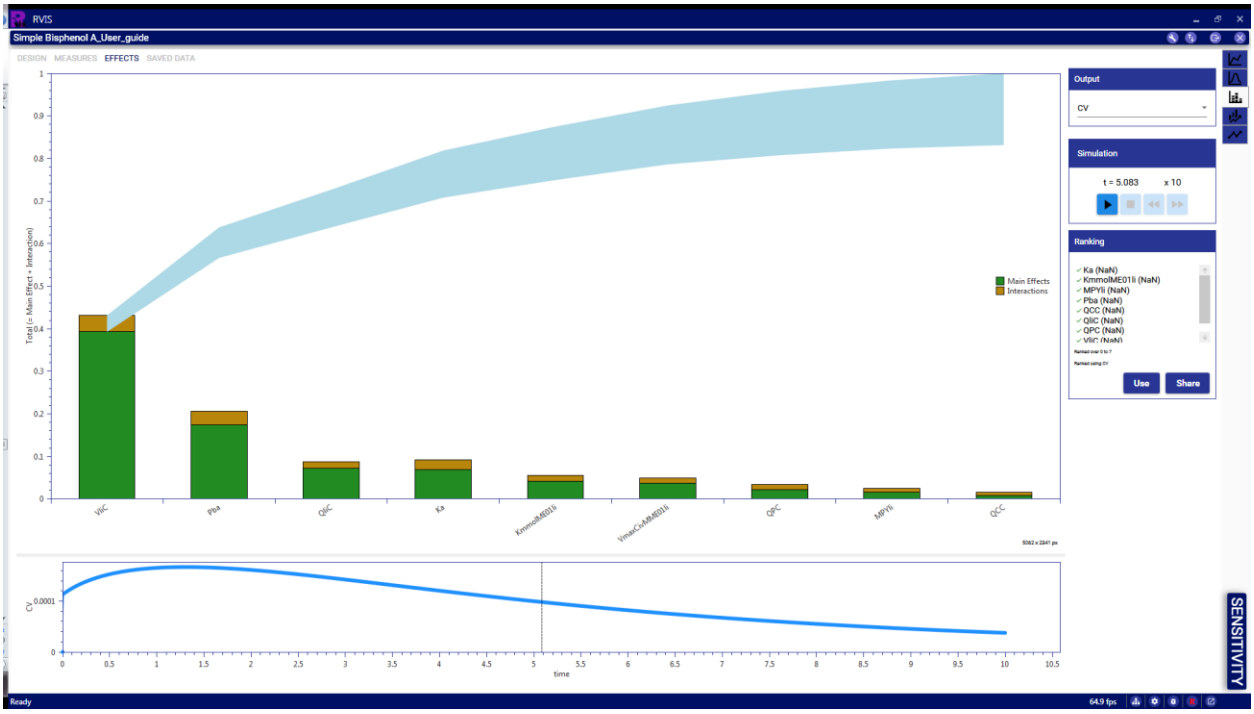


Figure 25 Effects: The Lowry Plot

3.6.1 Shared state

The shared state is a feature which can minimise repetition of various actions. This is particularly useful during the selection and configuration of parameters in the modules that use stochastic processes.

1. Click on "Apply/share state between modules" to open the "Shared state" dialogue box (Figure 26). All parameters selected within any module can be uploaded into the Shared State. For example, the selection of parameters uploaded from the eFAST module should be listed in the "Parameters" field. If not then click on left upload button, "Store selected module's parameter state in shared state" at the bottom of the "Parameters" field to upload parameters in to shared state.

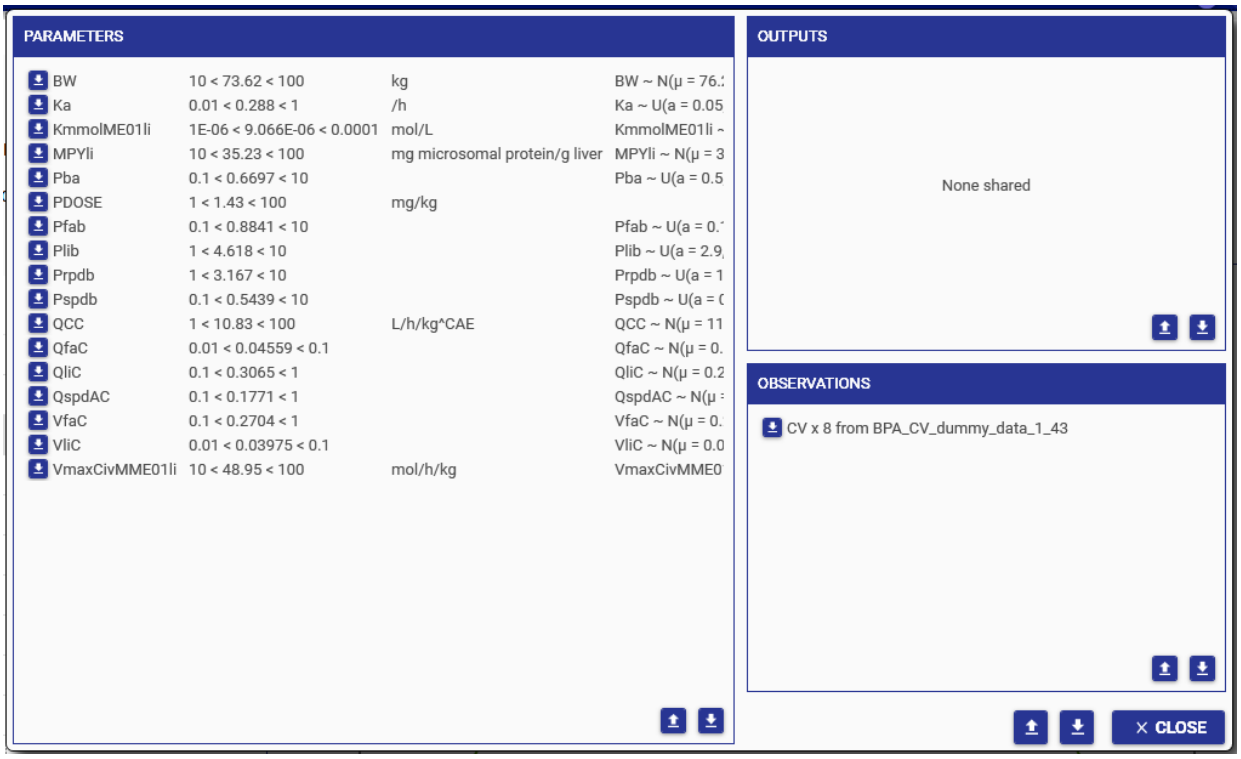


Figure 26 Shared state dialogue box

3.7 PARAMETER ESTIMATION

Parameter estimation in RVis is within a Bayesian framework. The user initially supplies prior distributions for the input parameters under study and these are refined through the comparison of predictions from the model against comparable measurements within a statistical error model. Parameter estimation is undertaken using Markov chain Monte Carlo (MCMC) sampling implemented by a single component Metropolis-Hastings algorithm... A choice of four statistical error models is available for characterising the discrepancy: normal, log normal, heteroscedastic (power) and heteroscedastic (exponential). These latter two error models both assume normally distributed errors but the variance is proportional to magnitude. These models encode different statistical assumptions about the distribution of prediction errors.

Please note: RVis is, first and foremost, a labour saving device. The correct and efficient use of the modules deploying stochastic modelling and the interpretation of the results require specialist skills and expertise. Whilst RVis makes this process easier the software is not a substitute for specialist expertise. It is advisable that the user has access to such expertise.

1. Click on the Parameter Estimation module icon
2. Click on "Priors" tab.
3. Click on "Apply/share state between modules" to open the "Shared state" dialogue box, and then click on to download selected parameters. The parameter selection from the eFAST module should apear in the "Select Priors" field. These can be adjusted.
4. Make sure PDOSE is selected, if not add it and ascribe an appropriate range (See Table 3**)**
5. Check each parameter distribution by clicking on the parameter. The distribution is displayed with dashed vertcal lines indicating the lower and upper limits. The user should exercise judgement regarding the truncation of distributions, noting that if priors exclude a region of parameter space then the posteriors will also exclude the same region of parameter space, regardless of what measurement data are used.
6. Click on the "Likelihood" tab.
7. In the "Error Model" field the user has a choice of error models: normal, log normal, heteroscedastic (power) and heteroscedastic (exponential) (Figure 27).
8. Select the "normal" error model.
9. In the standard deviation field enter 0.0001 if is PDOSE is 1.43, or 0.00005 if PDOSE is 10. Note that these are only suggestions for initial values for the error standard deviation, the parameter is updated within the MCMC algorithm alongside the model parameters.
10. Select a data set from the "Observations" field. If PDOSE is 1.43 mg/kg select BPA_CV_dummy_data_1_43 or if PDOSE is 10 mg/kg select BPA_CV_dummy_data_10. Note that the column names in these files correspond to 'Time' and the model output being calibrated to in parameter estimation 'CV'.
11. Click on the "Design" tab. The priors, output error models, observations are listed on the left. The number of iterations, burn-in and the number chains are selected on the right by the user.
12. Enter, 1000 into the No. of iterations, 50 into the burn-in and 3 into Chains fields.

13. Click “Create Design” followed by clicking the “Simulation” tab.
14. Click on “Iterate” in the “Chains” field to initiate the parameter estimation. The user can follow the progress of each of the three chains (each chain has a different colour) during parameter estimation by clicking on the desired parameter in the “Trace” field. After around 50 iterations (the burn-in period) stop the simulation and click on the ⚙ in the Chains field to open the “Iteration Options” dialogue box.
15. Enter 0.4 into the “Set target accept rate to” field. Click OK to close Iterations Options box. The (Gaussian) proposal distribution is adapted if this option is selected in order to achieve the user-defined acceptance rate (0.2 to 0.4 is a generally accepted range for acceptance)
16. Click “Iterate” in the “Chains” field to resume simulations. Progress toward convergence for any selected parameter can be monitored by clicking on the down-arrow icon in the “Trace” dialogue box.

Figure 27 Selection of likelihood model

17. To review the posterior distributions set the range over which they are to be calculated. Enter a “Begin” and “End” point in the fields in the “Convergence” dialogue box (The range will be highlighted in chart field. In Figure 28 a range from simulation 400 to 1000 was selected.
18. Click on the “Posterior” tab.
19. In Figure 29 the prior distribution is represented by the green line and the posterior by the histogram. Click on the down-arrow icon in the “Show Posterior for” dialogue box to view the posterior for any selected parameter.
20. Click on the “Fit” tab to view the fit to the selected data set (Figure 30).

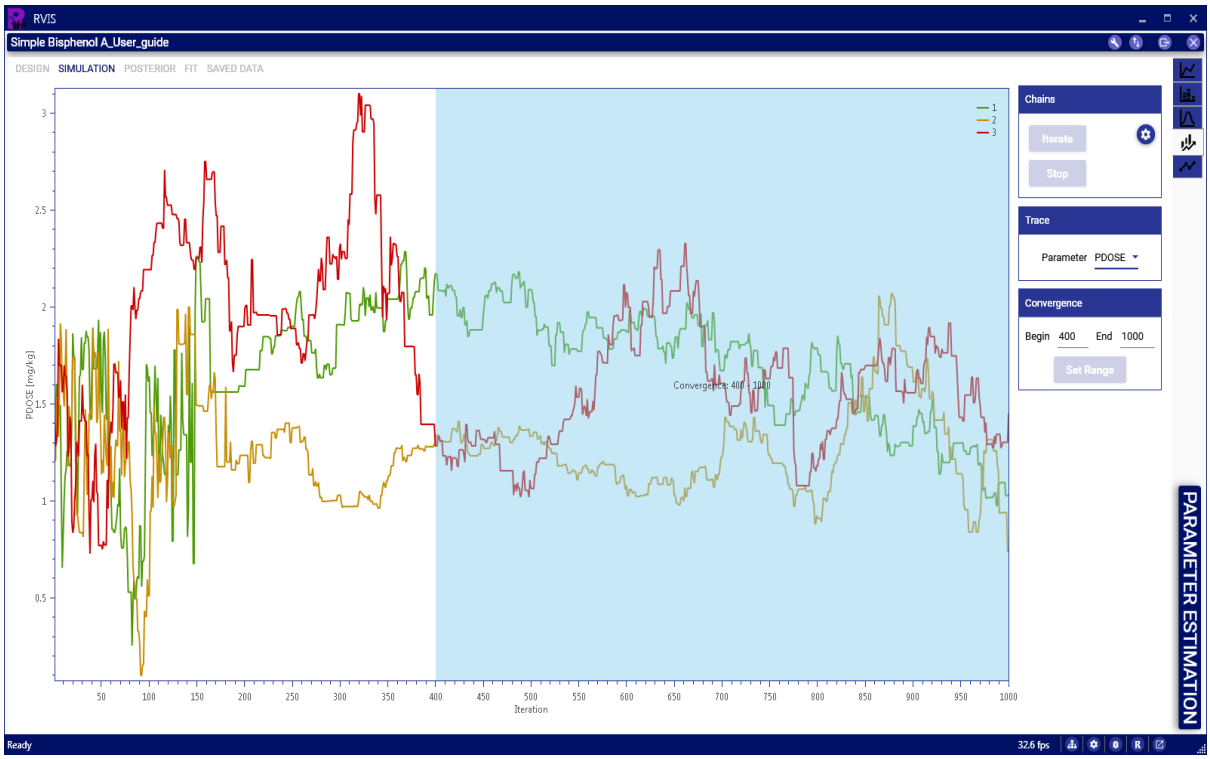


Figure 28 Parameter estimation convergence range

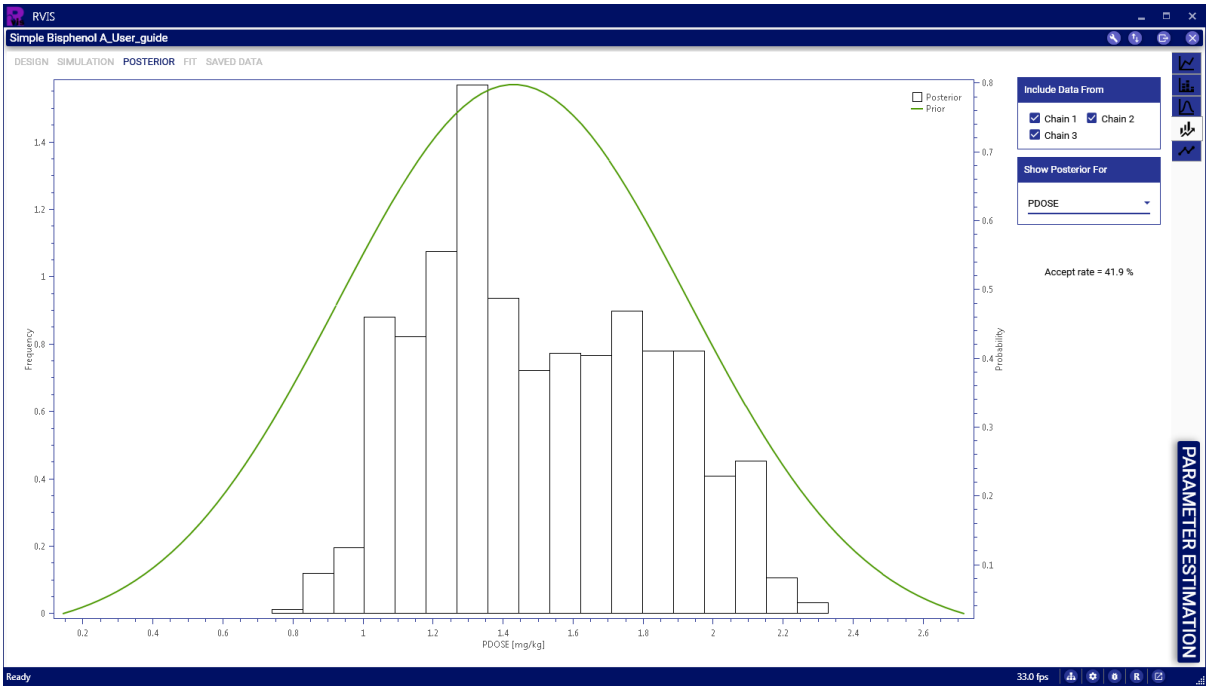


Figure 29 Posterior Distributions

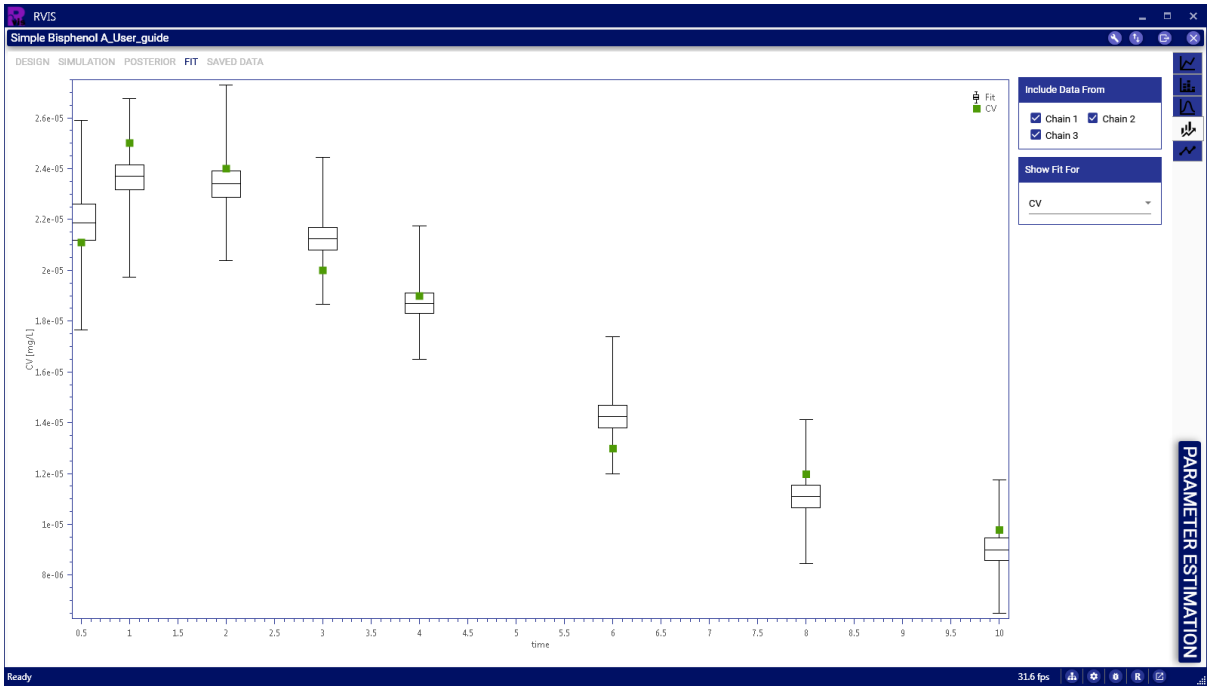


Figure 30 Fit to data: box and whisker (median and interquartile range) plots with min/max values

3.8 EXPORT DATA

Data may be exported from any module where the Export icon is enabled.

Data 36are stored in a sub-folder identified by a date and time stamp assigned (at the time of execution) by RVis and stored in a sub-folder with the name of the imported model within a library directory module called, RVisData. For example:

~RVisData\estimation\Simple Bisphenol A_User_guide\20191227084802

Data are stored as csv files within the sub-directory allowing import into a variety of software environments for further analysis. In addition an R script 'load_data.R' is written to the directory and may be viewed by loading in R or RStudio. The script reads the .csv files into the working environment and performs some operations of the data. These files can be edited as appropriate. Figures 26 and 27 are examples of visualisations of MCMC output that are coded within the R scripts associated with this module.

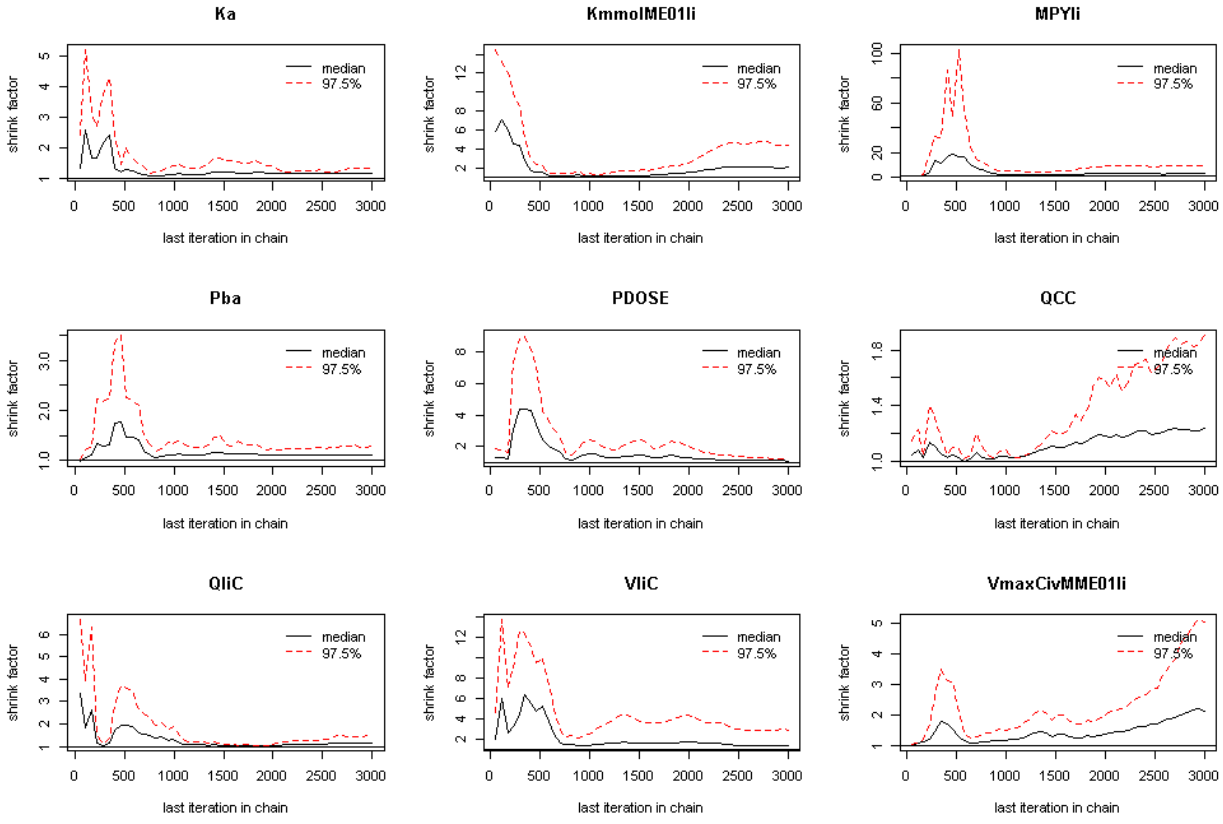


Figure 31 Exported data from Estimation module

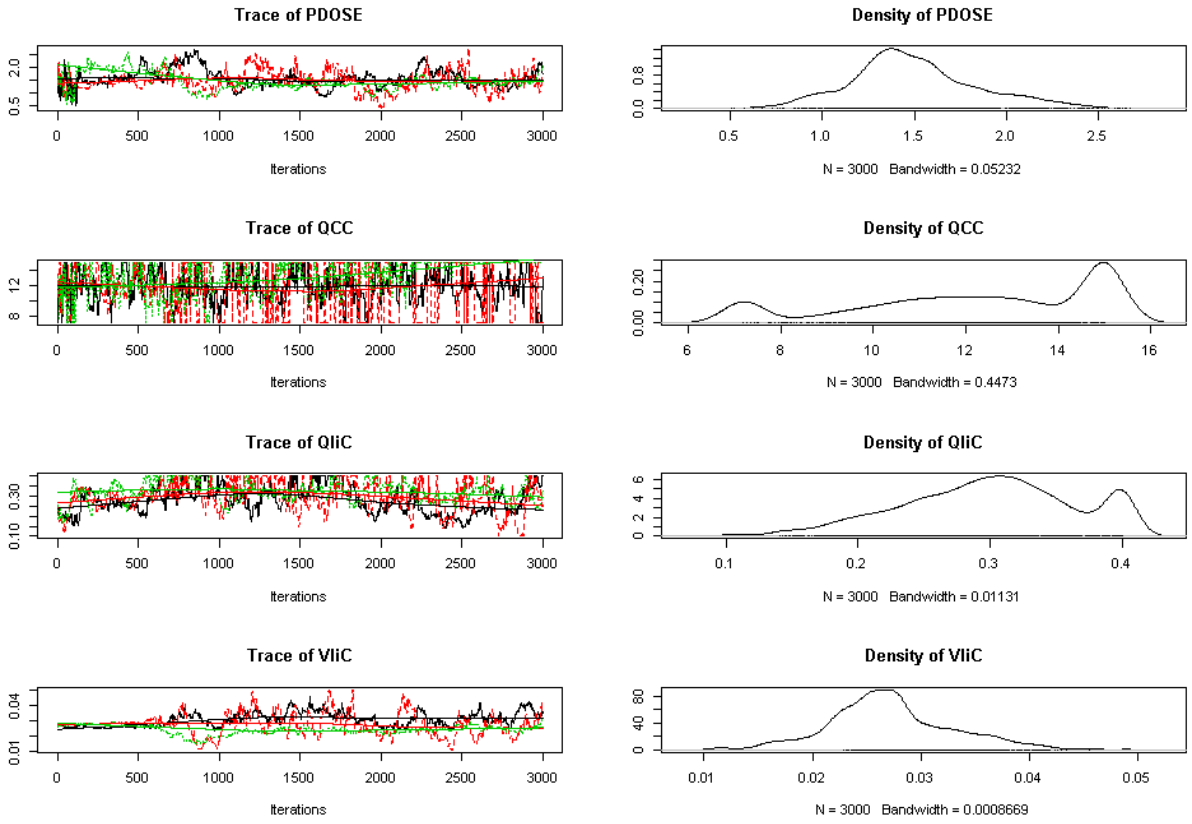


Figure 32 Exported data from Estimation module

4. CONCLUSIONS

We have made efforts to ensure that RVis has been thoroughly tested but we make no claim that it is perfect-far from it! This user guide illustrates the core functionality of RVis using a simple model for Bisphenol A. The step by-step process described in section 3 should allow a user to replicate our findings and gain a feel for the software. In practical use the features for uncertainty and sensitivity analysis have more functionality than can be demonstrated using a step by step guide. The authors suggest that in the first instance users work with the PBPK model and datasets provided in the appendix (and the Github directory with the RVis download) and change the distributions etc. to build their familiarity of the software. Edits to the PBPK model and the creation of additional synthetic data for other model outputs (for simple comparisons or formal calibration) are suggested tasks for building familiarity. Use bespoke PBPK models for serious study only once you know what you are doing.

RVis is software-in-development. Features are being refined and embellished in response to in-lab testing. Wider testing and use from the research community will assist in the further development of features. Suggestions, gripes (and compliments!) can be made within the Github directory.

We envisage that further functionality will be added in the future; in particular a module for in-vitro to in-vivo extrapolation is in the pipeline. The user guide will be updated alongside any significant future developments

5. REFERENCES

Arms, A. D., and Travis, C. C. (1988). Reference physiological parameters in pharmacokinetic modeling. *Office of Health and Environmental Assesment, U.S. EPA, Washington, D.C. NTIS PB88-196019*.

Baldrick, P. (2003). Toxicokinetics in preclinical evaluation. *Drug Discov Today* **8**(3), 127-33.

Barter, Z. E., Bayliss, M. K., Beaune, P. H., Boobis, A. R., Carlile, D. J., Edwards, R. J., et al. (2007). Scaling factors for the extrapolation of in vivo metabolic drug clearance from in vitro data: reaching a consensus on values of human microsomal protein and hepatocellularity per gram of liver. *Curr Drug Metab* **8**(1), 33-45.

Barton, H. A., Bessems, J., Bouvier d'Yvoire, M., Buist, H., Clewell III, H., Gundert-Remy, U., et al. (2009). Principles of Characterizing and Applying Physiologically-Based Pharmacokinetic and Toxicokinetic Models in Risk Assessment. *IPCS project on the Harmonization of Approaches to the Assessment of Risk from Exposure to Chemicals.*

Barton, H. A., Chiu, W. A., Setzer, R. W., Andersen, M. E., Bailer, A. J., Bois, F. Y., et al. (2007). Characterizing uncertainty and variability in physiologically-based pharmacokinetic (PBPK) models: state of the science and needs for research and implementation. *Toxicol Sci* **99**, 395-402.

Brown, R. P., Delp, M. D., Lindstedt, S. L., Rhomberg, L. R., and Beliles, R. P. (1997). Physiological parameter values for physiologically based pharmacokinetic models. *Toxicol. Ind. Health* **13**(4), 407-484.

Clewell, H. J., Gentry, P. R., Gearhart, J. M., Allen, B. C., and Andersen, M. E. (2001). Comparison of cancer risk estimates for vinyl chloride using animal and human data with a PBPK model. *Sci. Total Environ.* **274**(1-3), 37-66.

Hanioka, N., Naito, T., and Narimatsu, S. (2008). Human UDP-glucuronosyltransferase isoforms involved in bisphenol A glucuronidation. *Chemosphere* **74**(1), 33-36.

Himmelstein, M. W., Carpenter, S. C., Evans, M. V., Hinderliter, P. M., and Kenyon, E. M. (2004). Kinetic modeling of beta-chloroprene metabolism: II. The application of physiologically based modeling for cancer dose response analysis. *Toxicol Sci* **79**(1), 28-37.

Howgate, E. M., Rowland Yeo, K., Proctor, N. J., Tucker, G. T., and Rostami-Hodjegan, A. (2006). Prediction of in vivo drug clearance from in vitro data. I: impact of inter-individual variability. *Xenobiotica* **36**(6), 473-97.

Hsieh, N.-H., Reisfeld, B., Bois, F. Y., and Chiu, W. A. (2018). Applying a Global Sensitivity Analysis Workflow to Improve the Computational Efficiencies in Physiologically-Based Pharmacokinetic Modeling. *Frontiers in Pharmacology* **9**(588), 10.3389/fphar.2018.00588.

ICH (1995). CPMP/ICH/384/95. Note for guidance on toxicokinetics: a guidance for assessing systemic exposure in toxicology studies.

ICRP (2002). Basic Anatomical and Physiological Data for Use in Radiological Protection: Reference Values. *ICRP Publication 89. Ann. ICRP 32 (3-4).*

Krishnan, K., and Andersen, M. E. (1994). Physiologically based pharmacokinetic modeling in toxicology. In *Principals and Methods of Toxicology* (A. W. Hayes, Ed.) Eds.)3 ed., pp. 149-188. Raven Press Ltd., New York.

Loizou, G. D., and Hogg, A. (2011). MEGen: A Physiologically Based Pharmacokinetic Model Generator. *Frontiers in Pharmacology: Predictive Toxicity* **2** Article 56, 1-14, 10.3389/fphar.2011.00056.

Loizou, G. D., Spendiff, M., Barton, H. A., Bessems, J., Bois, F. Y., Bouvier, d. Y., et al. (2008). Development of Good Modelling Practice for Physiologically Based Pharmacokinetic Models for Use in Risk Assessment: The First Steps. *Reg. Toxicol. Pharmacol.* **50**, 400-411.

McNally, K., Cotton, R., Cocker, J., Jones, K., Bartels, M., Rick, D., et al. (2012). Reconstruction of Exposure to m-Xylene from Human Biomonitoring Data Using PBPK Modelling, Bayesian Inference, and Markov Chain Monte Carlo Simulation. *Journal of Toxicology* **2012**, 18, 10.1155/2012/760281.

McNally, K., Cotton, R., Hogg, A., and Loizou, G. (2014). PopGen: A virtual human population generator. *Toxicology* **315**(0), 70-85.

McNally, K., Cotton, R., and Loizou, G. (2011). A workflow for global sensitivity analysis of PBPK models. *Frontiers in Pharmacology: Predictive Toxicity* **2**, Article 31, 1-21, 10.3389/fphar.2011.00031.

Pastino, G. M., Flynn, E. J., and Sultatos, L. G. (2000). Genetic polymorphisms in ethanol metabolism: Issues and goals for physiologically based pharmacokinetic modeling. *Drug and Chemical Toxicology* **23**(1), 179-201.

Shin, B., Kim, C., Jun, Y., Kim, D., Lee, B., Yoon, C., et al. (2004). Physiologically based pharmacokinetics of bisphenol a. *J Toxicol Environ Health A* **67**(23-24), 1971-85.

Team, R. C. (2017). R Core Team (2017). R: A language and environment for statistical computing. *R Found. Stat. Comput. Vienna, Austria. URL http://www. R-project. org/., page R Foundation for Statistical Computing*.

6. APPENDIX 1

Code in R Syntax for a simple PBPK Model for bisphenol A

```
# *** Bisphenol A ***
# Shin et al (2004) J Toxicol Env Health Part A
# George Loizou (HSL) 2019-06-30T23:00:00.0000000Z
# Compiled on: 2019-07-01T15:48:18.5123353Z
#pragma exec mgen_pbpk parameters
# ************************************************************************************
# Set parameter values
# ************************************************************************************
define_parameters <- function()
{
BW <- 70 #@p body mass (kg)
CAE <- 0.75 #@p body mass (kg) proportion of vascularised tissue cardiac allometric exponent
QCC <- 11.22 #@p body mass (kg) proportion of vascularised tissue cardiac allometric exponent cardiac
allometric constant (L/h/kg^CAE)
DS <- 0.33 #@p proportion of dead space (not involved in gas exchange)
RAE <- 0.75 #@p respiratory allometric exponent
QPC <- 15 #@p respiratory allometric constant (L/h/kg^RAE)
Ka <- .2 #@p oral uptake rate (/h)
PDOSE <- 10 #@p oral dose (mg/kg)
DRINK <- 1.37 #@p dose in water (mg/kg/day)
RMM <- 228.28 #@p molecular mass (g/mol)

VT <- 0.857 #@p body mass (kg) proportion of vascularised tissue
VliC <- 0.0257 #@p fractional volume, liver
VfaC <- 0.214 #@p fractional volume, adipose
VspdAC <- 0.43 #@p overall fractional volume, slowly perfused

QliC <- 0.25 #@p fractional blood flow, liver
QfaC <- 0.052 #@p fractional blood flow, adipose
QspdAC <- 0.27 #@p overall fractional blood flow, slowly perfused

Pba <- 1.43 #@p blood:air partition coefficient
Pfab <- 0.7 #@p tissue:blood partition coefficient, adipose
Prpdb <- 2.8 #@p tissue:blood partition coefficient, rapidly perfused
Pspdb <- 0.8 #@p tissue:blood partition coefficient, slowly perfused
Plib <- 5.7 #@p tissue:blood partition coefficient, liver
MPYli <- 34 #@p microsomal protein yield, liver (mg microsomal protein/g liver)
VmaxCivMME01li <- 52.379999999999995 #@p maximum rate of metabolism (molar; in vitro; microsomal), liver
(mol/h/kg)
KmmolME01li <- 0.00000868 #@p molar Michaelis constant, liver (mol/L)
runtime <- 10 #@p runtime (hours)
#Return all variables in this function's environment
as.list(sys.frame(sys.nframe()))
}
```

```
# ************************************************************************************
# Set solver's initial values
# ************************************************************************************
define_initial_values <- function(parameters)
{
        with(parameters,
        {

                PPDOSE <- PDOSE * BW  # scaled oral dose (mg/day)


                KZER <- (DRINK / 24) * BW # zero order uptake rate constant

                VmaxME01li <- VmaxCivMME01li * BW * VliC * RMM * MPYli
                # maximum rate of metabolism (entire organism), liver (mg/h)
                KmME01li <- KmmolME01li * RMM * 1000.0 # Michaelis constant, liver (mg/L)
                P <- 1 – DS  # proportion of inhaled gas involved in gas exchange

                BWc <- BW ^ CAE  # cardiac scaling output factor (kg)

                QC <- QCC * BWc  # cardiac output (L/h)

                BWr <- BW ^ RAE  # respiratory scaling output factor (kg)

                QP <- QPC * BWr  # ventilation rate (L/h)

                QPa <- QP * P  # alveolar ventilation rate (L/h)

                VspdC <- VspdAC - VfaC  # fractional volume, slowly perfused
                VrpdAC <- VT - VspdAC  # overall fractional volume, rapidly perfused
                VrpdC <- VrpdAC – VliC  # fractional volume, rapidly perfused

                QspdC <- QspdAC – QfaC  # fractional blood flow, slowly perfused
                QrpdAC <- 1 - QspdAC  # overall fractional blood flow, rapidly perfused
                QrpdC <- QrpdAC - QliC  # fractional blood flow, rapidly perfused


#Gelman reparameterisations

Qcci <- QliC+QfaC+QspdC+QrpdC
Qlici <- QliC/Qcci
Qfaci <- QfaC/Qcci
Qspdci <- QspdC/Qcci
Qrpdci <- QrpdC/Qcci
Qrpd <- Qrpdci * QC # scaled fractional blood flow, rapidly perfused
Qspd <- Qspdci * QC # scaled fractional blood flow, slowly perfused
Qli <- Qlici * QC # scaled fractional blood flow, liver
Qfa <- Qfaci * QC # scaled fractional blood flow, adipose
Vti <- (1-VT)+VliC+VfaC+VrpdC+VspdC
```

```
 96
 97    Vlici <- VliC/Vti
 98    Vfaci <- VfaC/Vti
 99    Vrpdci <- VrpdC/Vti
100    Vspdci <- VspdC/Vti
101    Vli <- Vlici * BW # scaled fractional volume, liver
102    Vfa <- Vfaci * BW # scaled fractional volume, adipose
103    Vrpd <- Vrpdci * BW # scaled fractional volume, rapidly perfused
104    Vspd <- Vspdci * BW # scaled fractional volume, slowly perfused
105
106
107
108    #Return all variables in this function's environment
109    as.list(sys.frame(sys.nframe()))
110    })
111    }
112    # ************************************************************************************
113    # Model
114    # ************************************************************************************
115    pbpk <- function(t, y, parms, ...)
116    {
117    with(as.list(c(y,parms)),
118    {
119    (kg)MRS <- PPDOSE * exp((-Ka * t)) # amount remaining in stomach
120
121    TIS <- Ka * MRS # total input from stomach (kg)
122    Ruptake <- TIS + KZER # uptake derivative (kg)
123    Cfa <- Afa / Vfa  # cellular concentration, adipose (mg/L)
124    Cspd <- Aspd / Vspd  # cellular concentration, slowly perfused (mg/L)
125    Cli <- Ali / Vli # cellular concentration, liver (mg/L)
126    mass <- Afa + Arpd + Aspd + Ali + AMli + AX  # mass in system (kg)
127    CVfa <- Cfa / Pfab  # venous organ concentration, adipose (mg/L)
128    CVspd <- Cspd / Pspdb # venous organ concentration, slowly perfused (mg/L)
129    CVli <- Cli / Plib  # venous organ concentration, liver (mg/L)
130
131    # mass balance
132    if (t>0)
133    {
134    rel <- mass / uptake
135    }
136    Else
137    {
138    rel <- 1
139    }
140    MRli <- (VmaxME01li * CVli) / (KmME01li + CVli) # rate of change of metabolism, liver (mg/h/kg)
141    Crpd <- Arpd / Vrpd # cellular concentration, rapidly perfused (mg/L)
```

```
142     RAMli <- MRli # amount metabolised derivative, liver (kg)
143     CVrpd <- Crpd / Prpdb # venous organ concentration, rapidly perfused (mg/L)
144     CV <- ((CVfa * Qfa) + (CVrpd * Qrpd) + (CVspd * Qspd) + (CVli * Qli)) / QC  # venous concentration (mg/L)
145     CA <- (QC * CV) / (QC + (QPa / Pba)) # arterial concentration (mg/L)
146     RAfa <- Qfa * (CA - CVfa) # cellular compartment derivative, adipose (mg/h/kg)
147     RArpd <- Qrpd * (CA - CVrpd) # cellular compartment derivative, rapidly perfused (mg/h/kg)
148     RAspd <- Qspd * (CA - CVspd) # cellular compartment derivative, slowly perfused (mg/h/kg)
149     RAX <- QPa * CX # amount exhaled derivative (kg)
150     RAli <- (Qli * (CA - CVli) + (TIS + KZER)) - MRli # cellular compartment derivative, liver (mg/h/kg)
151     CX <- CA / Pba  # exhaled concentration (mg/L)
152     list(c(RAfa, RAli, RAMli, RArpd, RAspd, RAX, Ruptake),
153     c(MRS=MRS, TIS=TIS, Cfa=Cfa, Cspd=Cspd, Cli=Cli, mass=mass, CVfa=CVfa, CVspd=CVspd, CVli=CVli, rel=rel,
154     MRli=MRli, Crpd=Crpd, CVrpd=CVrpd, CV=CV, CA=CA, CX=CX))
155             })
156     }
157     # ************************************************************************************
158     # Solver
159     # ************************************************************************************
160     mgen_pbpk <- function(parameters)
161     {
162             if(!require(deSolve))
163             {
164                     stop("The 'deSolve' package is required.  Please install it.")
165             }
166      parms <- define_initial_values(parameters)
167      t_range<-c(0, parameters$runtime)
168             # Solve ODE system
169             y <- rep.int(0, 7)
170             names(y) <- c("Afa", "Ali", "AMli", "Arpd", "Aspd", "AX", "uptake")
171             times <- seq.int(t_range[1], t_range[2], 1/60)
172             deSolve::ode(y, times, pbpk, parms, method = "lsodes")
173     }
174
175     parameters <- define_parameters()
176     res <- mgen_pbpk(parameters) #@o
177     # cellular concentration, adipose (mg/L)
178     #Cfa <- res[, "Afa"] / with(initial_values, Vfa)
179     # cellular concentration, rapidly perfused (mg/L)
180     #Crpd <- res[, "Arpd"] / with(initial_values, Vrpd)
181     # cellular concentration, slowly perfused (mg/L)
182     #Cspd <- res[, "Aspd"] / with(initial_values, Vspd)
183     # cellular concentration, liver (mg/L)
184     #Cli <- res[, "Ali"] / with(initial_values, Vli)
185     # example code for creating conc vs time plots
186     #plot(res[, "time"], res[, "CV"], type="l", xlab="time", ylab="conc", main="Liver Conc")
187
```

Data

BPA_CV_dummy_data_10_mg	
Time	CV
0.5	1.30E-04
1	1.40E-04
2	1.35E-04
3	1.20E-04
4	1.10E-04
6	8.00E-05
8	4.70E-05
10	4.00E-05

BPA_CV_dummay_data_1_43_mg	
Time	CV
0.5	2.11E-05
1	2.50E-05
2	2.40E-05
3	2.00E-05
4	1.90E-05
6	1.30E-05
8	1.20E-05
10	9.80E-06

7. APPENDIX 2

Code in MCSim Syntax for a simple PBPK Model for bisphenol A

```
# MCSim 5.x script

# *** Bisphenol A ***

# Shin et al (2004) J Toxicol Env Health Part A
# George Loizou (HSL) 2019-06-30T23:00:00.0000000Z

# Compiled on: 2019-07-01T15:48:18.5123353Z

States =
{
        Afa,
        Arpd,
        Aspd,
        Ali,
        AMli,
        uptake,
        AX
};

Outputs =
{
        Cfa,
        Cspd,
        Cli,
        mass,
        rel,
        CVfa,
        CVspd,
        CVli,
        Crpd,
        CVrpd,
        CV,
        CA,
        CX,
        TIS,
        MRS

};

Inputs =
{
};

# Parameters
# ==========
#
```

```
BW = 70;                    # body mass (kg)
QPC = 15;                   # respiratory allometric constant (L/h/kg^RAE)
QCC = 11.22;                # cardiac allometric constant (L/h/kg^CAE)
CAE = 0.75;                 # cardiac allometric exponent
RAE = 0.75;                 # respiratory allometric exponent
DS = 0.33;                  # proportion of dead space (not involved in gas exchange)

Ka = 3;                     # oral uptake rate (/h)
PDOSE = 1.43;               # oral dose (mg/kg)
DRINK = 1.37;               # dose in water (mg/kg/day)
RMM = 228.28;               # molecular mass (g/mol)

QspdAC = 0.27;              # overall fractional blood flow
QliC = 0.25;                # fractional blood flow
QfaC = 0.052;               # fractional blood flow

VT = 0.857;                 # proportion of vascularised tissue
VliC = 0.0257;              # fractional volume
VspdAC = 0.43;              # overall fractional volume
VfaC = 0.214;               # fractional volume

Pba = 1.43;                 # blood:air partition coefficient
Plib = 5                    # tissue:blood partition coefficient
Pspdb = 0.8;                # tissue:blood partition coefficient
Prpdb = 2.8;                # tissue:blood partition coefficient
Pfab = 0.7;                 # tissue:blood partition coefficient

VmaxCivMME01li = 52.38; # maximum rate of metabolism (molar; in vitro; microsomal) (mol/h/kg)
KmmolME01li = 0.00000868;       # molar Michaelis constant (mol/L)
MPYli = 34;                 # microsomal protein yield (mg microsomal protein/g liver)

Vfa = 0;
Vspd = 0;
Vrpd = 0;
Vli = 0;
Qfa = 0;
Qrpd = 0;
Qspd = 0;
Qli = 0;
QC = 0;
QP = 0;
Uptake = 0;
CV = 0;
VmaxME01li = 0;
KmME01li = 0;
QPa = 0;
TIS = 0;
MRS = 0;
PPDOSE = 0;
KZER = 0;

# Run settings
# ============
```

```
#

#STARTt = 0;
#STOPt = 10;

Initialize
{
VmaxME01li = VmaxCivMME01li * BW * VliC * RMM * MPYli;    # maximum rate of metabolism (entire organism)
(mg/h)

KmME01li = KmmolME01li * RMM * 1000.0;# Michaelis constant (mg/L)
P = 1 - DS;                    # proportion of inhaled gas involved in gas exchange

QrpdAC = 1 - QspdAC;              # overall fractional blood flow
QrpdC = QrpdAC - QliC;             # fractional blood flow
QspdC = QspdAC - QfaC;             # fractional blood flow

VrpdAC = VT - VspdAC;              # overall fractional volume
VrpdC = VrpdAC - VliC;             # fractional volume
VspdC = VspdAC - VfaC;             # fractional volume

PPDOSE = PDOSE * BW;                    # scaled oral dose (mg/day)
KZER = (DRINK / 24) * BW;               # zero order uptake rate constant

BWc = pow(BW,CAE);                      # cardiac scaling output factor (kg)

BWr = pow(BW,RAE);              # respiratory scaling output factor (kg)
QP = QPC * BWr;                         # ventilation rate (L/h)
QC = QCC * BWc;                         # cardiac output (L/h)
QPa = QP * P;                           # alveolar ventilation rate (L/h)

## Gelman reparameterisations
Qcci = QliC+QfaC+QspdC+QrpdC;
Qlici = QliC/Qcci;
Qfaci = QfaC/Qcci;
Qspdci = QspdC/Qcci;
Qrpdci = QrpdC/Qcci;

Qrpd = Qrpdci * QC;                     # scaled fractional blood flow, rapidly perfused
Qspd = Qspdci * QC;                     # scaled fractional blood flow, slowly perfused
Qli = Qlici * QC;                       # scaled fractional blood flow, liver
Qfa = Qfaci * QC;                       # scaled fractional blood flow, adipose

Vti = (1-VT)+VliC+VfaC+VrpdC+VspdC;
Vlici = VliC/Vti;
Vfaci = VfaC/Vti;
Vrpdci = VrpdC/Vti;
Vspdci = VspdC/Vti;

Vli = Vlici * BW;                       # scaled fractional volume, liver
Vfa = Vfaci * BW;                       # scaled fractional volume, adipose
Vrpd = Vrpdci * BW;                     # scaled fractional volume, rapidly perfused
Vspd = Vspdci * BW;                     # scaled fractional volume, slowly perfused
```

```
} # End of model initialization

Dynamics
{
        MRS = PPDOSE * exp((-Ka * t));      # amount remaining in stomach (mg)
        TIS = Ka * MRS;                     # total input from stomach (mg)

    # cellular concentrations (mg/L)

        Cfa = Afa / Vfa;
        Cspd = Aspd / Vspd;
        Crpd = Arpd / Vrpd;
        Cli = Ali / Vli;

    # venous organ concentrations (mg/L)

        CVfa = Cfa / Pfab;
        CVspd = Cspd / Pspdb;
        CVrpd = Crpd / Prpdb;
        CVli = Cli / Plib;

    CX = CA / Pba;                                  # exhaled concentration (mg/L)

        mass = Afa + Arpd + Aspd + Ali + AMli + AX;  # mass in system (kg
        rel = ((t>0) ? mass / (uptake + 1e-10): 1);      # mass balance

    # venous concentration (mg/L)
        CV = ((CVfa * Qfa) + (CVrpd * Qrpd) + (CVspd * Qspd) + (CVli * Qli)) / QC;
        CA = (QC * CV) / (QC + (QPa / Pba));   # arterial concentration (mg/L)

        dt (uptake) = TIS + KZER;               # uptake derivative (mg)

    # rate of change of metabolism (mg/h/kg)
        dt (AMli) = (VmaxME01li * CVli) / (KmME01li + CVli);

    # cellular compartment derivatives (mg/h/kg)

    dt (Ali) = (Qli * (CA - CVli) + (TIS + KZER)) - dt (AMli)

    dt (Afa) = Qfa * (CA - CVfa);

    dt (Arpd) = Qrpd * (CA - CVrpd);

    dt (Aspd) = Qspd * (CA - CVspd);
    dt (AX) = QPa * CX;

} # End of Dynamics

CalcOutputs
{
}
End.
```

Template.in file

```
Integrate (Lsodes, 1e-6, 1e-6, 1);
Simulation {
  BW = {{BW}};
  DS = {{DS}};
  Ka = {{Ka}};
  QCC = {{QCC}};
  QPC = {{QPC}};
  CAE = {{CAE}};
  RAE = {{RAE}};
  RMM = {{RMM}};
  PDOSE = {{PDOSE}};
  DRINK = {{DRINK}};
  VT = {{VT}};
  VfaC = {{VfaC}};
  VspdAC = {{VspdAC}};
  VliC = {{VliC}};
  QliC = {{QliC}};
  QfaC = {{QfaC}};
  QspdAC = {{QspdAC}};
  Pfab = {{Pfab}};
  Prpdb = {{Prpdb}};
  Pspdb = {{Pspdb}};
  Plib = {{Plib}};
  MPYli = {{MPYli}};
  KmmolME01li = {{KmmolME01li}};
  VmaxCivMME01li = {{VmaxCivMME01li}};

  PrintStep(
    CV, rel, uptake, CVli, CVfa, CVspd, CVrpd, MRS, TIS, AMli, KZER, PPDOSE,
    {{TStart}}, {{TEnd}}, {{Tint}}
    );

}

END
```

Configuration. R file

```
import <- list(
  simulationName = "Simple Bisphenol A MCSim",
  description = "Simple Bisphenol A MCSim",
  importName = "Simple Bisphenol A MCSim"
  )
parameters <- list(

BW = 70,                  # body mass (kg)
QPC = 15,                 # respiratory allometric constant (L/h/kg^RAE)
QCC = 11.22,              # cardiac allometric constant (L/h/kg^CAE)
CAE = 0.75,               # cardiac allometric exponent
RAE = 0.75,               # respiratory allometric exponent
DS = 0.33,                # proportion of dead space (not involved in gas exchange)
Ka = 3,                   # oral uptake rate (/h)
PDOSE = 1.43,             # oral dose (mg/kg)
DRINK = 1.37,             # dose in water (mg/kg/day)
RMM = 228.28,             # molecular mass (g/mol)
QspdAC = 0.27,            # overall fractional blood flow
QliC = 0.25,              # fractional blood flow
QfaC = 0.052,             # fractional blood flow
VT = 0.857,               # proportion of vascularised tissue
VliC = 0.0257,            # fractional volume
VspdAC = 0.43,            # overall fractional volume
VfaC = 0.214,             # fractional volume
Plib = 5.7,               # tissue:blood partition coefficient
Pspdb = 0.8,              # tissue:blood partition coefficient
Prpdb = 2.8,              # tissue:blood partition coefficient
Pfab = 0.7,               # tissue:blood partition coefficient

VmaxCivMME01li = 52.38, # maximum rate of metabolism (molar, in vitro, microsomal) (mol/h/kg)
KmmolME01li = 0.00000868, # molar Michaelis constant (mol/L)
MPYli = 34,               # microsomal protein yield (mg microsomal protein/g liver)


  TStart = 0,
  TEnd = 10,
  Tint = 0.1                    # [min]
)

independentVariable <- list(

  # elapsed
  Time = NA                     # [min]

)

outputs <- list(
```

```
  # Venous blood concentration
  CV = NA,                      # [mg/l]
  CVli = NA,                    # [mg/l]
  CVfa = NA,                    # [mg/l]
  CVspd = NA,                   # [mg/l]
  CVrpd = NA,                   # [mg/l]
  MRS = NA,                     # [mg]
  TIS = NA,                     # [mg]
  rel = NA,
  uptake = NA,
  AMli = NA,                    # [mg]
  KZER = NA,
  PPDOSE = NA                   # [mg]
)
```

HSE, through its Science Division, is one of the world's leading providers of health and safety solutions to industry, government and professional bodies. The main focus of our work is on understanding and reducing health and safety risks. We provide health and safety consultancy, research, specialist training and products to our customers worldwide. Our long history developing health and safety solutions means that we're well placed to understand the changing industrial, regulatory and societal landscape, and to anticipate future issues. We employ over 360 scientific, medical and technical specialists, drawing on their wealth of knowledge and experience to deliver evidence-based solutions to our clients. Our work is supported by accredited management systems.

ISO 9001 OHSAS 18001

Health and Safety Executive
Science Division

Harpur Hill
Buxton
Derbyshire
SK17 9JN
UK

1.2 Redgrave Court
Merton Road
Bootle
L20 7HS

www.hsl.gov.uk
www.hse.gov.uk/research

T: +44 (0)20 3028 2000
E: hslinfo@hsl.gsi.gov.uk