TUTORIAL

MatVPC: A User-Friendly MATLAB-Based Tool for the Simulation and Evaluation of Systems Pharmacology Models

K Biliouris[1], M Lavielle[2] and MN Trame[1]*

Quantitative systems pharmacology (QSP) models are progressively entering the arena of contemporary pharmacology. The efficient implementation and evaluation of complex QSP models necessitates the development of flexible computational tools that are built into QSP mainstream software. To this end, we present MatVPC, a versatile MATLAB-based tool that accommodates QSP models of any complexity level. MatVPC executes Monte Carlo simulations as well as automatic construction of visual predictive checks (VPCs) and quantified VPCs (QVPCs).
CPT Pharmacometrics Syst. Pharmacol. (2015) **4**, 547–557; doi:10.1002/psp4.12011; published online on 22 August 2015.

VPC is a model diagnostic tool that facilitates the evaluation of both the structural and the stochastic part of a model. It is constructed by superimposing the observations over the model simulations while accounting for both the interindividual variability as well as the residual variability.[1] Once underutilized,[2] the VPC now is recognized as one of the most valuable model diagnostics in pharmacological model evaluation.[3–5] Its superiority over comparable diagnostic tools has been established[6] and reflected by the fact that regulatory agencies recommend it as one of the central model diagnostics.[7]

The VPC has recently evolved and key features have been added to its originally simple version.[8–10] These features aid in better visualizing the data and include but are not limited to summary statistics and binning of the simulated and observed data. Extensions to the VPC have also been developed aiming for a highly objective model evaluation. A VPC extension example includes QVPC, wherein the distribution of the observations around the predicted median trend is rigorously realized.[11]

To ease the VPC construction process, several practical software products have been developed that automatically generate comprehensive VPC plots, which are challenging to generate otherwise. Such software products include Monolix[12] and Xpose-PsN (coupled with NONMEM),[13,14] among others.[15–17] Monolix carries out the simulations and uses the outcome to generate the VPC plots, whereas PsN uses NONMEM[18] as the simulation platform and performs the percentile calculations. Subsequently, the latter are plotted using Xpose. The immense practicality of these tools, however, comes with its set of shortcomings: (i) they inevitably carry out parameter estimation of a model before constructing its VPC; (ii) they do not offer extensions to the VPC; and (iii) generating VPCs of high complexity datasets may be devious. These drawbacks render these tools unattractive for constructing VPCs of high complexity models or of models that have

already been built in different software and require no further parameter estimation.

Recently, a growing body of literature has been demonstrating the catalytic role of high complexity, systems-level models in the pharmacology arena.[19–27] Traditionally, the so-called "QSP" models are not implemented in Monolix or NONMEM but in surrogate software tools, such as MATLAB or MATLAB-based software.[8,28–37] VPC plots that are generated with these tools are generally not as comprehensive[31,38] lacking important features, such as confidence intervals of simulation percentiles or data binning. As it was previously discussed, VPCs that lack these features may not be sufficiently informative.[1,9,11]

The progressive incorporation of QSP models in the regulatory arena[39] necessitates the development of flexible computational tools that are built into software commonly used for such models; these tools can be then routinely utilized in evaluating and simulating QSP models. To this end, we developed MatVPC: a flexible, user-friendly MATLAB-based tool that carries out Monte Carlo simulation as well as VPC construction of QSP models of any complexity level. MatVPC accommodates any model, independently of the software used for model development, while offering the majority of options that comparable tools list. Provided that certain features of a model may not be readily visualized in a VPC,[11] MatVPC grants the additional option of constructing QVPC plots of the available data.

MATVPC TUTORIAL

In this tool, the user simply inputs: (i) the model equations; (ii) the model parameters; and (iii) the NONMEM-like dataset with observations, and MatVPC outputs VPCs, QVPCs, or Monte Carlo simulation plots at will. The inputted model parameters here include macrokinetic or microkinetic constants, initial conditions of ordinary differential equations

[1]Center for Pharmacometrics and Systems Pharmacology, Department of Pharmaceutics, University of Florida, Orlando, Florida, USA; [2]Inria Saclay, POPIX Team, Saclay, France. *Correspondence: MN Trame (mtrame@cop.ufl.edu)

Simulation and Evaluation of Systems Pharmacology Models
Biliouris *et al*.

548

Table 1 Default values and accepted values included in MatVPC's optional input interface

Options	Default value	Accepted input
No. of datasets	200	Scalar
Automatic binning[a]	Yes	Yes, no
Manual binning[b]	No	Yes, no
Bin edges	[ ]	Vector with as many scalars as the bin edges
No. of simulated timepoints[c]	100	Scalar
Run Monte Carlo simulations[d]	No	Yes, no
ODE solver[e]	ode23tb	ode45, ode23, ode113, ode15s, ode23s, ode23t, ode23tb
Simulated time beyond observations[f]	0	Scalar
Upper percentile limit of simulations	95	Scalar
Lower percentile limit of simulations	5	Scalar
Plot observations[g]	Yes	Yes, no
Plot percentiles of observations	Yes	Yes, no
Upper percentile limit of observations	95	Scalar
Lower percentile limit of observations	5	Scalar
Plot QVPC	Yes	Yes, no
Replace negative and zero values[h]	No	Yes, no
Replacement value	0.001	Scalar
Graphical settings		
Type of median curve in simulations	–	–, – –,:, –., o, +, *, .,:, x, s, d, ^, v, >, <, p, h
Color of median curve in simulations[i]	Black	Color name
Type of percentiles curve in simulations	– –	–, – –,:, –., o, +, *, .,:, x, s, d, ^, v, >, <, p, h
Color of percentiles curve in simulations[i]	Black	Color name
Color of median CI in simulations[i]	Deep pink	Color name
Color of percentiles CI simulations[i]	Blue	Color name
Color of observations[i]	Blue	Color name
Size of observations	10	Scalar
Color of median curve in observations[i]	Red	Color name
Color of percentiles curve in observations[i]	– –	–, – –,:, –., o, +, *, .,:, x, s, d, ^, v, >, <, p, h
Type of median curve in observations	–	–, – –,:, –., o, +, *, .,:, x, s, d, ^, v, >, <, p, h
Type of percentiles curve in observations	– –	–, – –,:, –., o, +, *, .,:, x, s, d, ^, v, >, <, p, h
Color of simulation median in QVPC[i]	Red	Color name
Color of observations above median in QVPC[i]	Grey	Color name
Color of observations below median in QVPC[i]	Black	Color name
Number of rows in figure	1	Scalar
Number of columns in figure	3	Scalar
Square shaped plots	Yes	Yes, no
Width of all curves	1	Scalar
Log-y scale	No	Yes, no
X-axis label	Time	Any alphabetic input
Y-axis label	Observations/simulations	Any alphabetic input
Transparency of CI shaded areas	0.5	Scalar

CI, Confidence Intervals; ODE, ordinary differential equation; QVPC, quantified visual predictive check.

[a]For a detailed description of how the bin edges are selected, please refer to Ref. 9 and 40. Before the results of the automatic binning are accepted, a visual inspection of the calculated bin edges should be done as, in extreme cases of data point distributions, the code might not generate optimal results.

[b]In case "yes" is selected here, a vector with the bin edges should be defined in the following option.

[c]This option allows modification of the number of equally spaced timepoints that are saved during the Monte Carlo simulations. The larger the numbers, the slower the runtime.

[d]This option requests Monte Carlo simulations can be selected in other options (see footnotes c and f).

[e]The ODE solver should be carefully chosen based on the type of the model, in other words, whether the model is stiff or not.

[f]This number represents the percentage of the entire simulation time that is simulated beyond the latest observation point. For instance, in case the user desires to run a Monte Carlo simulation for an additional time equal to 20% of the entire profile, this number should be set equal to 0.2 (see Monte Carlo simulations in model one).

[g]This option allows the user to plot simulation results without plotting the observations. In case the user desires to simply plot Monte Carlo simulations of the model, "no" should be selected.

[h]This option should be exploited in case the user wants to avoid negative concentrations and associated plotting issues, for instance, when a logarithmic scale is used. In case "yes" is selected here, the desired replacement value should be defined in the following option.

[i]The color of the curves is defined by writing the name of any of the 139 available color names provided at http://www.w3.org/TR/css3-color/. Note that the first letter should be capital.

Simulation and Evaluation of Systems Pharmacology Models
Biliouris et al.

549

a)

```
1    function [V,params,proportional_part_of_RUV,additive_part_of_RUV,initial_conditions]=parameters
2        %% Please provide the model parameters here
3  -     k10=0.0916;
4  -     k12=0.0932;
5  -     k21=0.0093;
6  -     k13=0.192;
7  -     k31=0.009;
8  -     k30=0.094;
9  -     ka=0.858;
10       %% Please provide a vector with the omegas here
11 -     omega=[0.296 0.326 0.27];
12       %% Please do not change anything here
13 -     eta=normrnd(0,omega);
14
15       %% Please provide a vector with the proportional part of RUV here The dimension of this vector should be equal to the number of the compartments.
16       %% If there is no DV for a compartment please set its proportional_part_of_RUV equl to 0.
17 -     proportional_part_of_RUV=[0.104 0.0709 0.0784];
18       %% Please provide a vector with the additive part of RUV, if any. The length of the vector should be equal to the number of ODEs. If there are compartments w/
19       %and w/o additive error, please set the additive part of the latter equal to zero. If there is no additive error at all please do not change anything here.
20 -     additive_part_of_RUV=[];
21       %% Please name all model parameters as params(i) (with i=1,...,number of parameters) and add the IIV here
22 -     params(1)=k12*exp(eta(2));
23 -     params(2)=k21;
24 -     params(3)=k13*exp(eta(3));
25 -     params(4)=k31;
26 -     params(5)=k10*exp(eta(1));
27 -     params(6)=k30;
28 -     params(7)=ka;
29       %% Please provide the initial conditions as a vector here
30 -     initial_conditions=[0 0 0];
31       %% Provide the volume of each compartment here. The number of Volumes should be equal to the number of ODEs.
32       %% For compartments with no DVs, please set their Volume equal to 1.
33 -     V(1)=1; V(2)=1; V(3)=1; V(4)=1;
34 -   end
```

b)

```
1    function dy = ODEs(~,y,params,V)
2  -     n=4; %% Please provide the number of ODES here
3
4  -     dy = zeros(n,1); %% Please do not modify anything here
5
6        %% Please provide the ODEs here using params() as defined in file parameters
7  -     dy(1)=-params(1)*y(1)-params(3)*y(1)-params(5)*y(1)+params(2)*y(2)+params(4)*y(3)+params(7)*y(4);
8  -     dy(2)=params(1)*y(1)-params(2)*y(2);
9  -     dy(3)=params(3)*y(1)-params(4)*y(3)-params(6)*y(3);
10 -     dy(4)=-params(7)*y(4);
11 -   end
```

Figure 1 Required input in MatVPC for model one. (**a**) Input in file "parameters." In this file, the user inputs the values of the ODE parameters, the values of the interindividual variability parameters, the proportional and additive part of the RUV, the initial conditions of the ODEs and the volume size of each compartment. (**b**) Input in file "ODEs." In this file, the user inputs the ODEs, along with potential algebraic equations, describing the quantitative pharmacology model. In case of IV infusion dosing, an additional term should be included in the model that accounts for the dosing (for an example see model two).

(ODEs) and variability terms; MatVPC allows for implementing both interindividual variability and RUV, which can be in the form of proportional, additive, or combined error. The dataset should be provided in a comma-separated values format including, at a minimum, the following headers in no particular order and in a case-insensitive manner: (i) TIME; (ii) ID; (iii) CMT; (iv) DV (nonexistent DVs should be represented by a dot); (v) EVID; (vi) AMT; and (vii) RATE (only in the case of intravenous infusion dosing).

Once the user provides the aforementioned information, they must insert in the command line: (i) the name of the dataset; and (ii) the stratification variables, if any, as follows:

MatVPC ('dataset.csv', {'stratification1', 'stratification2'}). Common stratification variables include but are not limited to arms of a trial (placebo and active control), routes of administration, dose intervals, and, generally, covariates affecting any model parameters.[1] At present, MatVPC accepts up to two stratification variables, whereas compartment is a default stratification variable and MatVPC will always stratify on it. It should be highlighted that MatVPC treats each ODE as a "compartment," however, it only constructs VPC plots for the compartments with available observations. Once the information (i) and (ii) is provided, the user is asked (iii) to input the name of the model, which can be any alphanumeric value (e.g., Model1); and (iv) to state whether they want to run simulations and construct VPCs or to simply construct VPCs using previously saved simulation results. Please note that the former option requires the user to simply provide a desired name for the model, unlike the MATLAB file ODEs, wherein the user provides the model equations. Option 4 is expected to be useful in cases where simulations of a model have already been carried out and the user wants to simply reconstruct VPCs with modified characteristics, such as different color lines or altered binning edges. Parenthetically, when this option is exploited, the simulation-related parameters should be set to the same values as in the original VPC construction.

Upon providing all the above, a user-friendly interface commences allowing the user to modify the default values in a series of optional inputs. This input includes options related to the methods (e.g., ODE solver to be used or number of datasets to be simulated) as well as options that determine the graphical settings (e.g., color and type of

Figure 2 Optional input in MatVPC for model one. In this interface, the user can modify the default values of 40 characteristics of the VPC and QVPC plots (for a detailed description of these features see **Table 1**). The VPC and QVPC plots shown in **Figure 3** were constructed using the input values shown in this figure.

plotting curves). **Table 1** details the accepted optional input along with the default values.

It is important to stress that, unlike analogous tools that are traditionally used in pharmacometrics, MatVPC constitutes an all-in-one package that integrates the following:

1. It is publicly open (at https://sourceforge.net/projects/matvpc/);
2. Constructs VPC plots of complex QSP models;
3. Offers automatic binning of the observed and simulated data using a rigorous approach[9,40];
4. Constructs QVPC plots of complex QSP models;
5. Performs Monte Carlo simulations of the model and plots the results with any requested summary statistics;
6. Does not require prior implementation (e.g., parameter estimation) of the model within MATLAB;
7. Endows the user with the option to perform post-processing of the VPC, QVPC, or simulation plots.

In what follows, we implement three models with varying characteristics to demonstrate the functionality of MatVPC. These models include: (i) a linear three compartment pharmacokinetic (PK) model with single oral and intravenous (IV) bolus dosing; (ii) a nonlinear two compartment PK model with multiple IV infusion dosing[41]; and (iii) a highly

nonlinear pharmacodynamic (PD) model that describes the time-course of body weight.[42] To validate the accuracy of the presented VPCs, we directly compare them with VPCs generated either by Xpose version 4 (coupled with PsN version 4.2.0 and NONMEM version 7.2) or by Monolix version 4.3.3.

Model one: three-compartment PK model with oral and IV bolus dosing

The first model describes simulated data of a linear three-compartment PK model with linear elimination from the first and third compartment. The data were assumed to stem from two different studies. The difference between the two studies is nothing but the dosing type; in study one, the dosing type was IV bolus, whereas in study two, the dosing type was oral. A total of 50 individuals were simulated for each study. A single dose was administered at time zero and observations were taken at asymmetric timepoints. The initial conditions of all compartments were set to zero and the parameters of the model were estimated with NON-MEM. The ODEs of the model are shown in Eqs. 1–4:

$$\frac{dy_1}{dt} = -k_{10} \cdot y_1 - k_{12} \cdot y_1 + k_{21} \cdot y_2 - k_{13} \cdot y_1 + k_{31} \cdot y_3 + k_a \cdot y_4 \quad (1)$$

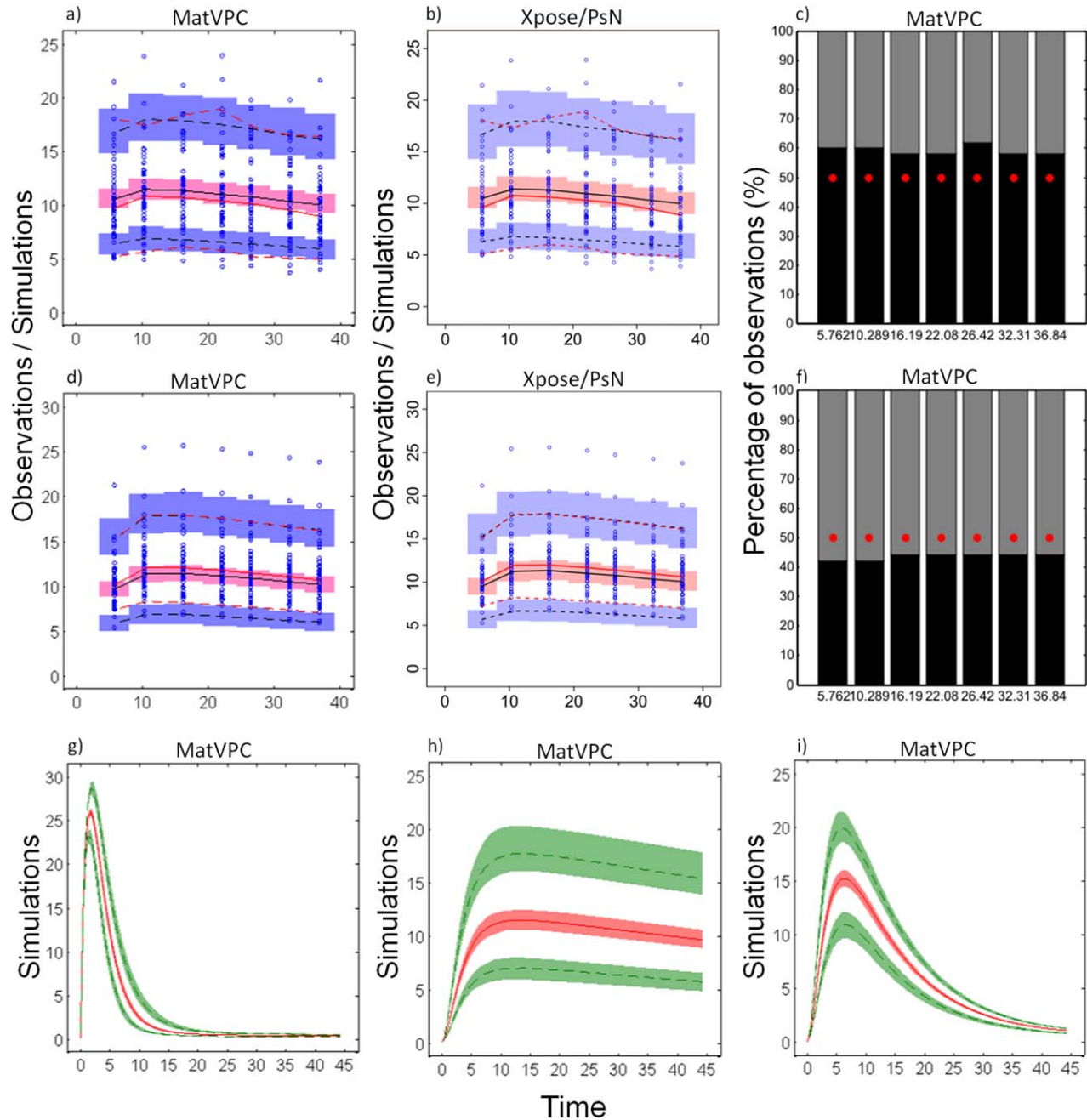$$\frac{dy_2}{dt} = k_{12} \cdot y_1 - k_{21} \cdot y_2 \quad (2)$$

Figure 3 VPC and QVPC plots for model one. (**a,d**) VPC plots generated by MatVPC showing the results from compartment two and study one (**a**) and two (**d**). (**b,e**) VPC plots generated by Xpose-PsN showing the results from compartment two and study one (**b**) and two (**e**). Blue dots correspond to the observations. Red dashed lines correspond to the 5th and 95th percentiles of the observations, whereas red solid lines correspond to the median of the observations. Black dashed lines correspond to the 5th and 95th percentiles of the simulations, whereas black solid lines correspond to the median of the simulations. Blue shaded areas represent the 90% confidence intervals of the simulation 5th and 95th percentiles, whereas pink shaded areas represent the 90% confidence intervals of the simulation median. (**c,f**) QVPC plots of compartment two and study one (**c**) and two (**f**), as constructed by MatVPC. At each timepoint, the black bar presents the observed data below the model predicted median (red dots), whereas the dark grey bar shows the observed data above the model predicted median. The total of the black and grey bar combined presents the percentage of available data (here 100%). (**g–i**) The 200 Monte Carlo simulations of model one and study two (oral dosing) for compartment one (**g**), two (**h**), and three (**i**). Green dashed lines correspond to the median of the 5th and 95th percentiles of the Monte Carlo simulations. Red lines correspond to the median of the 50th percentile of the Monte Carlo simulations. Shaded areas represent the 90% prediction intervals of the Monte Carlo simulation percentiles.

a)

```
1    ☐function [V,params,proportional_part_of_RUV,additive_part_of_RUV,initial_conditions]=parameters
2        %% Please provide the model parameters here
3  -     CL=0.296;
4  -     Q=0.167;
5  -     Vm=7.04;
6  -     Km=1.84;
7
8        %% Please provide a vector with the omegas here
9  -     omega=[0.37 0.299 0.155 0.15];
10
11       %% Please do not change anything here
12 -     eta=normrnd(0,omega);
13
14       %% Please provide a vector with the proportional part of RUV here The dimension of this vector should be equal to the number of the compartments.
15       %% If there is no DV for a compartment please set its proportional_part_of_RUV equl to 0.
16 -     proportional_part_of_RUV=[0.315 0];
17       %% Please provide a vector with the additive part of RUV, if any. The length of the vector should be equal to the number of ODEs. If there are compartments w/
18       %and w/o additive error please set the additive part of the latter equal to zero. If there is no additive error at all please do not change anything here.
19 -     additive_part_of_RUV=[0.563 0];
20
21       %% Please name all model parameters as params(i) (with i=1,...,number of parameters) and add the IIV here
22 -     params(1)=CL*exp(eta(1));
23 -     params(2)=Q;
24 -     params(3)=Vm*exp(eta(4));
25 -     params(4)=Km;
26 -     params(5)=0;
27       %% Please provide the initial conditions as a vector here
28 -     initial_conditions=[0 0];
29
30       %% Provide the volume of each compartment here. The number of Volumes should be equal to the number of ODEs.
31       %% For compartments with no DVs, please set their Volume equal to 1.
32 -     V(1)=4.15*exp(eta(2));
33 -     V(2)=3.14*exp(eta(3));
34 -   └end
```

b)

```
1    ☐function dy = ODEs(~,y,params,V)
2  -     n=2; %% Please provide the number of ODES here
3
4  -     dy = zeros(n,1); %% Please do not modify anything here
5
6        %% Please provide the ODEs here using params() as defined in file parameters
7  -     K10 = params(1)/V(1);
8  -     K12 = params(2)/V(1);
9  -     K21 = params(2)/V(2);
10 -     dy(1) = -K10*y(1)-K12*y(1)+K21*y(2)-(y(1)/V(1))*params(3)/(params(4)+(y(1)/V(1)))+params(5);
11 -     dy(2) = K12*y(1)-K21*y(2);
12 -   └end
```

Figure 4 Required input in MatVPC for model two. (**a**) Input in file "parameters." Please notice the extra term, "params(5)," that has been added to the parameter list. (**b**) Input in file "ODEs." Please note that "params(5)" has been added to Eq. dy(1) to account for IV infusion dosing.

$$\frac{dy_3}{dt} = k_{13} \cdot y_1 - k_{31} \cdot y_3 - k_{30} \cdot y_3 \qquad (3)$$

$$\frac{dy_4}{dt} = -k_a \cdot y_4 \qquad (4)$$

To generate VPC and QVPC plots of model one using MatVPC, the user is first required to open the MATLAB file parameters and provide the model parameters therein. Once the parameters have been provided, the user must open the MATLAB file ODEs and type the Eqs. 1–4. A snapshot of the parameters and the ODEs file upon inserting all the pertinent information of model one is shown in **Figures 1a,b**, respectively.

Upon completion of the above steps, the following command must be typed in the command window: MatVPC('obs.csv',{'study'}). This command requests a VPC that is based on the dataset called obs.csv and is stratified by study. Once the user inserts this command and presses "enter," they are asked to provide the name of the model and whether they want to simply create a VPC from previously saved simulation results or to simulate new data and plot the VPCs. Following their response (y/n), MatVPC opens a user-friendly interface with optional

input that bears default values. **Figure 2** shows a snapshot of this interface, which includes the optional input used for constructing VPC and QVPCs for model one. After pressing "OK" at the bottom of the interface, MatVPC runs all the required simulations, calculates the respective percentiles, constructs the VPC and QVPC plots, and saves them as MATLAB figures (.fig) in a folder that is named after the model name. Unlike comparable software, by saving the plots as MATLAB figures, MatVPC allows for postprocessing of the plots. More specifically, the user can open the figures (VPC, QVPC, or Monte Carlo simulation plots) that are saved in .fig format and interactively modify their characteristics, such as: (i) the type, color, or size of the curves; (ii) the axes labels and their font size; (iii) the figure title and its font size; (iv) the size, shape, or resolution of the figure; and (v) the axes limits. Modifying these characteristics can also be done using the command line, but it is not recommended as it is usually rather laborious. More details about this can be found on MATLAB's website (http://www.mathworks.com/help/symbolic/edit-graphs. html?searchHighlight=edit).

As the sampling times in this case were asymmetric, we exploited the automatic binning option of MatVPC to bin the
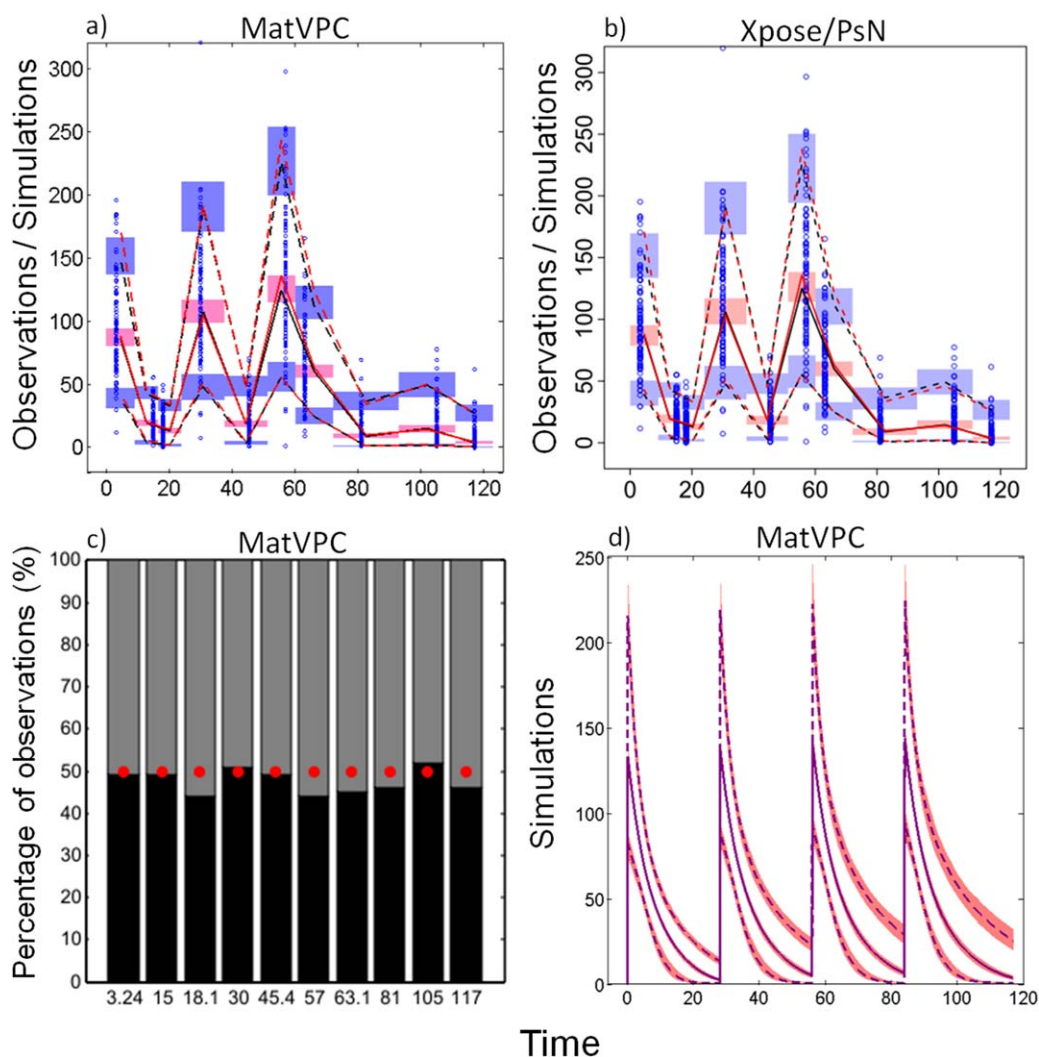
Figure 5 VPC and QVPC plots for model two. (**a**) VPC plots of compartment one generated by MatVPC. (**b**) VPC plots of compartment one generated by Xpose-PsN. Blue dots correspond to the observations. Red dashed lines correspond to the 5th and 95th percentiles of the observations whereas red solid lines correspond to the median of the observations. Black dashed lines correspond to the 5th and 95th percentiles of the simulations, whereas black solid lines correspond to the median of the simulations. Blue shaded areas represent the 90% confidence intervals of the simulation 5th and 95th percentiles, whereas pink shaded areas represent the 90% confidence intervals of the simulation median. (**c**) QVPC plots of compartment one as generated by MatVPC. At each timepoint, the black bar presents the observed data below the model predicted median (red dots), whereas the dark grey bar shows the observed data above the model predicted median. The total of the black and grey bar combined presents the percentage of available data (here 100%). (**d**) The 200 Monte Carlo simulations of model two for compartment one. Purple dashed lines correspond to the median of the 5th and 95th percentiles of the Monte Carlo simulations. Purple solid lines correspond to the median of the 50th percentile of the Monte Carlo simulations. Red shaded areas represent the 90% prediction intervals of the Monte Carlo simulation percentiles.

data of model one (i.e., "yes" was selected in the "automatic binning" option). The VPC and QVPC plots of model one as generated by MatVPC are shown in **Figure 3**. VPC plots of model one were also constructed with Xpose and PsN (running the simulations in NONMEM and using the bin edges calculated by MatVPC) and are also provided in **Figure 3**. For simplicity, we only show the results from compartment two of each study but the agreement between MatVPC plots and Xpose plots was consistent for all compartments.

As discussed above, one of the key characteristics of MatVPC is the option of performing Monte Carlo simulations of a QSP model, either accounting for or neglecting the resid-

ual error. Here, we carried out Monte Carlo simulations of model one (only for study two) without taking into consideration the residual error. To do so, the RUV terms in the "parameters" file were set equal to zero (see **Supplementary Figure S1**). In addition, in the optional input interface, for "plot observations" was selected "no" and the "simulated time beyond observations" was set equal to 0.2, as we were interested in the dynamics of the model from time zero to 20% additional time beyond the timepoint of the latest observation (see **Supplementary Figure S2**). The results of the simulations of the three compartments of study two (oral dosing) are presented in **Figure 3**.

a)

```
1  ⊟function [V,params,proportional_part_of_RUV,additive_part_of_RUV,initial_conditions]=parameters
2   %% Please provide the model parameters here
3 -  kout    =0.163;
4 -  krel    =0.0258;
5 -  kde     =1;
6 -  kpro    =0.0127;
7 -  DSTIM   =0.0331;
8 -  base_BW =99.6;
9
10   %% Please provide a vector with the omegas here
11 - omega=[1.03 1.56 1.68 0.934 0.32];
12
13   %% Please do not change anything here
14 - eta=normrnd(0,omega);
15   %% Please provide a vector with the proportional part of RUV here The dimension of this vector should be equal to the number of the compartments.
16   %% If there is no DV for a compartment please set its proportional_part_of_RUV equl to 0.
17 - proportional_part_of_RUV=[0];
18
19   %% Please provide a vector with the additive part of RUV, if any. The length of the vector should be equal to the number of ODEs. If there are compartments w/
20   %and w/o additive error please set the additive part of the latter equal to zero. If there is no additive error at all please do not change anything here.
21 - additive_part_of_RUV=[1.18];
22   %% Please name all model parameters as params(i) (with i=1,...,number of parameters) and add the IIV here
23 - params(1)= kout*exp(eta(1));
24 - params(2)= krel*exp(eta(2));
25 - params(3)=kde;
26 - params(4)=kpro*exp(eta(3));
27 - params(5)=DSTIM*exp(eta(4));
28 - params(6)=base_BW*exp(eta(5));
29   %% Please provide the initial conditions as a vector here
30 - initial_conditions=params(6);
31   %% Provide the volume of each compartment here. The number of Volumes should be equal to the number of ODEs.
32   %% For compartments with no DVs, please set their Volume equal to 1.
33 - V(1)=1;
34 - └end
```

b)

```
1  ⊟function dy = ODEs(t,y,params,V)
2 -  n=1; %% Please provide the number of ODES here
3
4 -  dy = zeros(n,1); %% Please do not modify anything here
5
6    %% Please provide the ODEs here using params() as defined in file parameters
7 -  BASE=params(6)+params(4)*t;
8 -  kin=params(1)*BASE;
9 -  dy(1) =kin- params(1)*y(1)*(1+params(5)*params(3)*(exp(-params(2)*t)-exp(-params(3)*t))/(params(3)-params(2)));
10
11 - └end
```

Figure 6 Required input in MatVPC for model three. (**a**) Input in file "parameters." (**b**) Input in file "ODEs." For a detailed description about the input see caption in **Figure 1**.

Model two: two-compartment PK model with multidose IV infusion

The second model was adopted from a previously published study[41] and describes simulated PK data of a drug administered via IV infusion. In this case, a total of 100 individuals were simulated with asymmetrically sampled observations. The parameters of this nonlinear PK model were estimated in NONMEM. The model consists of Eqs. 5 and 6:

$$\frac{dy_1}{dt} = -k_{10} \cdot y_1 - k_{12} \cdot y_1 + k_{21} \cdot y_2 - \frac{V_m \cdot \left(\frac{y_1}{V_1}\right)}{K_m + \frac{y_1}{V_1}} \qquad (5)$$

$$\frac{dy_2}{dt} = k_{12} \cdot y_1 - k_{21} \cdot y_2 \qquad (6)$$

It is important to note that in case IV infusion dosing is simulated, an additional term should be added in the model to account for IV infusion dosing. This term is simply "params(n)," where "n" is equal to the number of model parameters plus one (please note that all parameters are inserted as "params(i)"); "params(n)" should invariably be: (i) the last parameter listed in the file "parameters,"; (ii) equated to zero; and (iii) added to the equation describing

the dynamics of the compartment in which IV dosing is administered. In this example, the term "params(5)" was added to Eq. 5 as the IV infusion is given in compartment one (see file "parameters" and "ODEs" in **Figure 4**).

Similarly to model one, in order to generate the VPC and QVPC plots of model two using MatVPC, the user must open the MATLAB file "parameters" and insert the model parameters. Once the parameters have been provided, the user must open the MATLAB file "ODEs" and type the ODEs shown in Eqs. 5 and 6, along with the additional term that accounts for IV infusion dosing. A snapshot of the completed MATLAB files "parameters" and "ODEs" file is depicted in **Figures 4a,b**, respectively. Please note that in file "parameters" an additional parameter, "params(5)=0," has been listed. As shown in file "ODEs," this parameter has then been incorporated in the first ODE of the model (see last term in first ODE).

Upon inserting this information in the two files, the user should type in the command window the following: "MatVPC('obs.csv')". MatVPC then asks the user to provide the model name and to define whether they want to construct VPC plots from scratch or to capitalize on previously generated simulation data. Upon completing these steps, a MATLAB interface with the optional input pops up. A
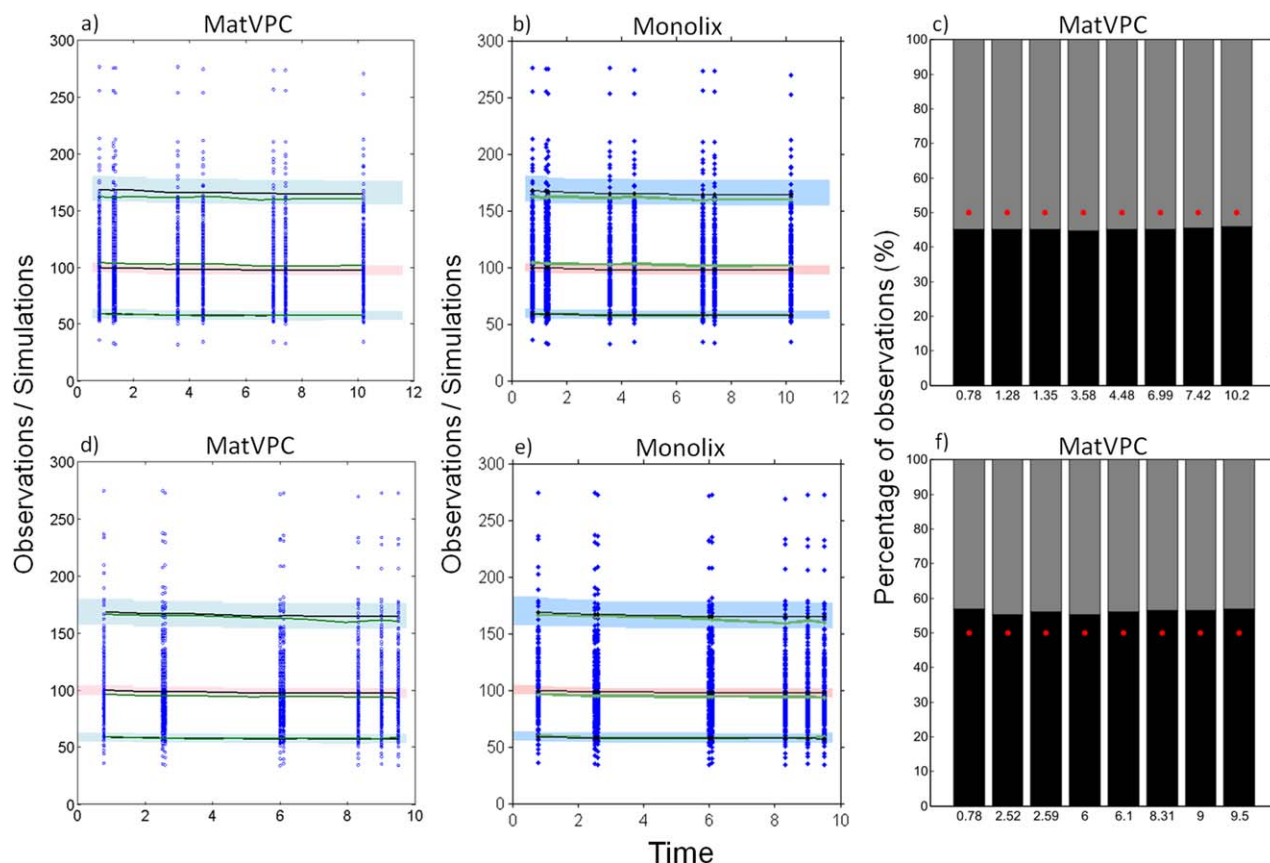
Figure 7 VPC and QVPC plots for model three. (**a,d**) VPC plots of study one (**a**) and study two (**d**) generated by MatVPC. (**b,e**) VPC plots of study one (**b**) and study two (**e**) generated by Monolix. Blue dots correspond to the observations. Green lines correspond to the 5th, 50th, and 95th percentiles of the observations. Black lines correspond to the 5th, 50th, and 95th percentiles of the simulations. Light blue shaded areas represent the 90% confidence intervals of the simulation 5th and 95th percentiles whereas pink shaded areas represent the 90% confidence intervals of the simulation median. (**c,f**) QVPC plots of study one (**c**) and two (**f**), as generated by MatVPC. At each timepoint, the black bar presents the observed data below the model predicted median (red dots), whereas the dark grey bar shows the observed data above the model predicted median. The total of the black and grey bar combined presents the percentage of available data (here 100%).

snapshot of this interface with the optional input used for model two is shown in **Supplementary Figure S3**. Please note that automatic data binning has been requested through this interface as the sampling timepoints were not consistent and manual binning of data would be challenging. After pressing "OK" at the bottom of this interface, MatVPC undergoes all the necessary steps and constructs the VPC and QVPC plots of model two. **Figures 5a,c** illustrate the VPC and QVPC plots, respectively, as constructed by MatVPC. **Figure 5b** shows the VPC of model two as generated with PsN and Xpose (with the simulations conducted in NONMEM and using bin edges calculated by MatVPC). For simplicity, only the plots from compartment one are shown.

In addition to generating the VPC and QVPC plots of model two, we also carried out 200 Monte Carlo simulations and the results are presented in **Figure 5d**. The "parameters" and "ODEs" files, along with the selected options in MatVPC interface, that were used to generate the Monte Carlo simulation plots are shown in **Supplementary Figures S4 and S5**, respectively.

Model three: PD model describing the time-course of body weight

The third case example is a formerly presented highly nonlinear PD model that describes the time-course of body weight.[42] The simulated dataset includes observations from a total of 500 subjects that are involved in two different studies, 250 subjects belonging to study one and 250 subjects belonging to study two. The difference between the two studies here is the sampling times. The PD model is provided in Eq. 7:

$$\frac{dy_1}{dt} = k_{in} - k_{out} \cdot y_1 \cdot \left(1 + \frac{DSTIM \cdot k_{de}}{k_{de} - k_{rel}} \cdot \left(e^{-k_{rel} \cdot Time} - e^{-k_{de} \cdot Time} \right) \cdot kde \right)$$

(7)

whereby k_{in} is the rate of weight gain, k_{out} is the rate of weight loss, k_{de} and k_{rel} are associated with the onset and loss of the lifestyle intervention effect and *DSTIM* is the maximum fractional increase in k_{out} caused by the lifestyle intervention. The values of these parameters were estimated using Monolix.

The "parameters" and "ODEs" files that were utilized in MatVPC to construct the VPC and QVPC plots of model

Simulation and Evaluation of Systems Pharmacology Models
Biliouris *et al*.

556

three are shown in **Figures 6a,b**, respectively. As the data originate from two distinct studies, the user would need to stratify on the study number. To do so, the following should be typed in the command window: "MatVPC ('obs.csv',{'study'})." This command requests VPC and QVPC plots of the PD model shown in Eq. 7, using the dataset called "obs.csv" while stratifying on the variable "study." Once the user defines the name of the model, and whether they want to use previous simulation results, the MatVPC interface with the optional input appears. The optional input used in the construction of VPCs for model three is provided in **Supplementary Figure S6**. Upon pressing "OK" in this interface, the VPCs and QVPCs of the model are generated and saved. These VPC and QVPC plots are illustrated in **Figure 7**. The respective VPC plots were also generated with Monolix and are provided in **Figures 7b,e**. It should be underlined that the automatic binning approach implemented in MatVPC and Monolix is identical.[9,40]

DISCUSSION

We have presented a MATLAB-based tool, dubbed MatVPC, that simulates and generates VPC and QVPC plots of QSP models. The plots generated by MatVPC were compared with plots constructed by the gold standard tools in the pharmacometrics community, PsN (with NONMEM) and Monolix,[43] and the results were interchangeable across all case studies.

Despite its flexibility, MatVPC does have its drawbacks. At present, MatVPC generates QVPC plots using only the available data and does not account for missing data.[11] In addition, it can currently run only on a single computer and is not designed for running in parallel processors. However, MatVPC will be continuously advanced and its future versions will address these imperfections.

MatVPC is publicly available and can be utilized by users with little or no prior MATLAB experience. This computational tool can be potentially expanded to perform key analyses in systems pharmacology,[44] such as sensitivity analysis[45] or model reduction.[38,46] Collectively, MatVPC constitutes a useful addition to the openly available toolboxes exploited by quantitative as well as clinical pharmacologists.

1. Karlsson, M. & Holford, N. A tutorial on visual predictive checks, PAGE meeting, Marseille, France, 2008.

2. Brendel, K. *et al.* Are population pharmacokinetic and/or pharmacodynamic models adequately evaluated? A survey of the literature from 2002 to 2004. *Clin. Pharmacokinet.* **46**, 221–234 (2007).

3. Byon, W. *et al.* Establishing best practices and guidance in population modeling: an experience with an internal population pharmacokinetic analysis guidance. *CPT Pharmacometrics Syst. Pharmacol.* **2**, e51 (2013).

4. Keizer, R.J., Karlsson, M.O. & Hooker, A. Modeling and simulation workbench for NONMEM: tutorial on Pirana, PsN, and Xpose. *CPT Pharmacometrics Syst. Pharmacol.* **2**, e50 (2013).

5. Comets, E., Brendel, K. & Mentré, F. Model evaluation in nonlinear mixed effect models, with applications to pharmacokinetics. *J. Soc. Stat. Paris* **151**, 106–128 (2010).

6. Holford N. The visual predictive check – superiority to standard diagnostic (Rorschach) plots. *PAGE Abstracts* Abstr 738, p 14 (2005).

7. European Medicines Agency. Guideline on reporting the results of population pharmacokinetic analyses, London, 21 June 2007. <http://www.ema.europa.eu/docs/en_GB/document_library/Scientific_guideline/2009/09/WC500003067.pdf>. Accessed March 2015.

8. Bergstrand, M., Hooker, A.C., Wallin, J.E. & Karlsson, M.O. Prediction-corrected visual predictive checks for diagnosing nonlinear mixed-effects models. *AAPS J.* **13**, 143–151 (2011).

9. Lavielle, M. & Bleakley, K. Automatic data binning for improved visual diagnosis of pharmacometric models. *J. Pharmacokinet. Pharmacodyn.* **38**, 861–871 (2011).

10. Wang, D.D. & Zhang, S. Standardized visual predictive check versus visual predictive check for model evaluation. *J. Clin. Pharmacol.* **52**, 39–54 (2012).

11. Post, T.M., Freijer, J.I., Ploeger, B.A. & Danhof, M. Extensions to the visual predictive check to facilitate model performance evaluation. *J. Pharmacokinet. Pharmacodyn.* **35**, 185–202 (2008).

12. Monolix Lixoft. <http://www.lixoft.eu/products/monolix>.

13. Jonsson, E.N. & Karlsson, M.O. Xpose–an S-PLUS based population pharmacokinetic/pharmacodynamic model building aid for NONMEM. *Comput. Methods Programs Biomed.* **58**, 51–64 (1999).

14. Lindbom, L., Ribbing, J. & Jonsson, E.N. Perl–speaks–NONMEM (PsN)–a Perl module for NONMEM related programming. *Comput. Methods Programs Biomed.* **75**, 85–94 (2004).

15. The MathWorks. SimBiology. <http://www.mathworks.com/products/simbiology/>.

16. Wojciechowski, J., Hopkins, A.M. & Upton, R.N. Interactive pharmacometric applications using R and the shiny package. *CPT Pharmacometrics Syst. Pharmacol.* **4**, 146–159 (2015).

17. Guiastrennec, B., Wollenberg, L., Forrest, A. & Ait-Oudhia, S. AMGET, an R-based postprocessing tool for ADAPT 5. *CPT Pharmacometrics Syst. Pharmacol.* **2**, e61 (2013).

18. Bauer, R.J. NONMEM users guide: introduction to NONMEM 7.2.0 (ICON Development Solutions, Ellicott City, MD, 2011).

19. Bai, J.P., Fontana, R.J., Price, N.D. & Sangar, V. Systems pharmacology modeling: an approach to improving drug safety. *Biopharm. Drug Dispos.* **35**, 1–14 (2014).

20. Cucurull-Sanchez, L., Spink, K.G. & Moschos, S.A. Relevance of systems pharmacology in drug discovery. *Drug Discov. Today* **17**, 665–670 (2012).

21. Iyengar, R., Zhao, S., Chung, S.W., Mager, D.E. & Gallo, J.M. Merging systems biology with pharmacodynamics. *Sci. Transl. Med.* **4**, 126ps7 (2012).

22. Sorger, P.K. *et al.* Quantitative and systems pharmacology in the post-genomic era: new approaches to discovering drugs and understanding therapeutic mechanisms (ed. Ward, R. ed) 1–48. An NIH white paper by the QSP Workshop Group (NIH, Bethesda, MD, 2011).

23. Visser, S.A., de Alwis, D.P., Kerbusch, T., Stone, J.A. & Allerheiligen, S.R. Implementation of quantitative and systems pharmacology in large pharma. *CPT Pharmacometrics Syst. Pharmacol.* **3**, e142 (2014).

24. Lei, T.A. & Bertz, R. Quantitative systems pharmacology can reduce attrition and improve productivity in pharmaceutical research and development. *Front. Pharmacol.* **5**, 247 (2014).

25. Mavroudis, P.D., Corbett, S.A., Calvano, S.E. & Androulakis, I.P. Circadian characteristics of permissive and suppressive effects of cortisol and their role in homeostasis and the acute inflammatory response. *Math. Biosci.* **260**, 54–64 (2015).

26. van der Graaf, P.H. & Benson, N. Systems pharmacology: bridging systems biology and pharmacokinetics-pharmacodynamics (PKPD) in drug discovery and development. *Pharm. Res.* **28**, 1460–1464 (2011).

27. Boran, A.D. & Iyengar, R. Systems approaches to polypharmacology and drug discovery. *Curr. Opin. Drug Discov. Devel.* **13**, 297–309 (2010).

28. Ballesta, A., Zhou, Q., Zhang, X., Lv H. & Gallo, J.M. Multiscale design of cell-type-specific pharmacokinetic/pharmacodynamic models for personalized medicine: application to temozolomide in brain tumors. *CPT Pharmacometrics Syst. Pharmacol.* **3**, e112 (2014).

29. Ait-Oudhia, S., Straubinger, R.M. & Mager, D.E. Systems pharmacological analysis of paclitaxel-mediated tumor priming that enhances nanocarrier deposition and efficacy. *J. Pharmacol. Exp. Ther.* **344**, 103–112 (2013).

30. Hendrix, A.O., Hughes, C.L. & Selgrade, J.F. Modeling endocrine control of the pituitary-ovarian axis: androgenic influence and chaotic dynamics. *Bull. Math. Biol.* **76**, 136–156 (2014).

31. Madrasi, K., Burns, R.N., Hendrix, C.W., Fossler, M.J. & Chaturvedula, A. Linking the population pharmacokinetics of tenofovir and its metabolites with its cellular uptake and metabolism. *CPT Pharmacometrics Syst. Pharmacol.* **3**, e147 (2014).

32. Wajima, T., Isbister, G.K. & Duffull, S.B. A comprehensive model for the humoral coagulation network in humans. *Clin. Pharmacol. Ther.* **86**, 290–298 (2009).

33. Gadkar, K., Budha, N., Baruch, A., Davis, J.D., Fielder, P. & Ramanujan, S. A mechanistic systems pharmacology model for prediction of LDL cholesterol lowering by PCSK9 antagonism in human dyslipidemic populations. *CPT Pharmacometrics Syst. Pharmacol.* **3**, e149 (2014).

34. Benson, N. *et al.* Systems pharmacology of the nerve growth factor pathway: use of a systems biology model for the identification of key drug targets using sensitivity analysis and the integration of physiology and pharmacology. *Interface Focus* **3**, 20120071 (2013).

35. Toni, T., Dua, P. & van der Graaf, P.H. Systems pharmacology of the NGF signaling through p75 and TrkA receptors. *CPT Pharmacometrics Syst. Pharmacol.* **3**, e150 (2014).

36. Pichardo-Almarza, C., Metcalf, L., Finkelstein, A. & Diaz-Zuccarini, V. Using a systems pharmacology approach to study the effect of statins on the early stage of atherosclerosis in humans. *CPT Pharmacometrics Syst. Pharmacol.* **4**, 41–50 (2014).

37. Gieschke, R. & Serafin, D. Development of innovative drugs via modeling with MATLAB (Springer, New York, NY, 2013).

38. Gulati, A., Isbister, G.K. & Duffull, S.B. Scale reduction of a systems coagulation model with an application to modeling pharmacokinetic-pharmacodynamic data. *CPT Pharmacometrics Syst. Pharmacol.* **3**, e90 (2014).

39. Peterson, M.C. & Riggs, M.M. FDA advisory meeting clinical pharmacology review utilizes a quantitative systems pharmacology (QSP) model: a watershed moment? *CPT Pharmacometrics Syst. Pharmacol.* **4**, 189–192 (2015).

40. Lavielle, M. Mixed effects models for the population approach: models, tasks, methods and tools (Chapman and Hall/CRC Biostatistics Series, 2014).

41. Combes, F.P., Retout, S., Frey, N. & Mentré, F. Prediction of shrinkage of individual parameters using the Bayesian information matrix in non-linear mixed effect models with evaluation in pharmacokinetics. *Pharm. Res.* **30**, 2355–2367 (2013).

42. Van Wart, S., Tsai, M., Chan, J. & Cirincione, B.B. Modeling the time-course of body weight for subjects given placebo in an obesity trial. ACOP meeting. San Diego, CA, October 7–10, 2012.

43. Vlasakakis, G. *et al.* White paper: landscape on technical and conceptual requirements and competence framework in drug/disease modeling and simulation. *CPT Pharmacometrics Syst. Pharmacol.* **2**, e40 (2013).

44. Schmidt, H. & Jirstrand, M. Systems biology toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics* **22**, 514–515 (2006).

45. Zhang, X.-Y., Trame, M.N., Lesko, L.J. & Schmidt, S. Sobol sensitivity analysis: a tool to guide the development and evaluation of systems pharmacology models. *CPT Pharmacometrics Syst. Pharmacol.* **4**, 69–79 (2015).

46. Dokoumetzidis, A. & Aarons, L. Proper lumping in systems biology models. *IET Syst. Biol.* **3**, 40–51 (2009).

Supplementary information accompanies this paper on the *CPT: Pharmacometrics & Systems Pharmacology* website (http://www.wileyonlinelibrary/psp4)