

### **SBPOP – A Computational Tool Supporting Model Based Drug Development - Application to the Renin-Angiotensin System**

Bruno Bieth, Novartis Pharmaceuticals, Basel, Switzerland  
Antoine Soubret, Novartis Pharmaceuticals, Basel, Switzerland  
Henning Schmidt, Novartis Pharmaceuticals, Basel, Switzerland

#### **ABSTRACT**

The practical handling of drug-based mathematical models and their analysis requires powerful and flexible, but still user-friendly computational tools, that allow the modeler to focus on the task at hand rather than on software implementation related issues. In this paper we present the SBPOP platform, a computational tool aiming at simplifying the required modeling and simulation workflow, implemented in MATLAB. We exemplify the use of the platform with a simple example model of the Renin-Angiotensin System (RAS), using data from published aliskiren trials. The use of the tool allows to integrate a wide variety of quantitative physiological information. The paper concludes by displaying results obtained from the simulation of the implemented model.

#### **INTRODUCTION**

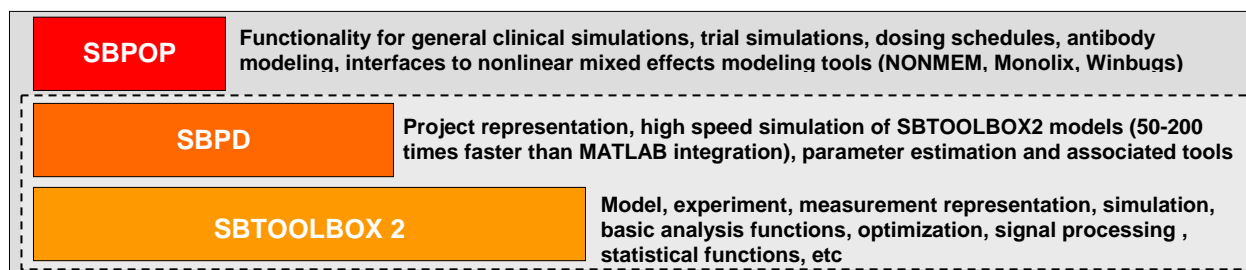
An important key to the discovery and development of new and innovative medicines is an increased understanding of the disease mechanisms and their underlying regulations. In engineering sciences advanced modeling and simulation techniques are ubiquitously used as standard tools to study physical, mechanical, electrical systems, etc. and, e.g., to design controllers that allow to influence the behavior of these systems in the desired way. The area of *systems biology*<sup>1</sup> is a fairly new direction in academic research that aims at understanding biology and physiology at the system level, by taking an engineering-like approach. This means, that rather than focusing on isolated parts of an organism as being the source of a disease or a phenotype of interest, systems biology tries to consider the interplay between the involved components of a biological or physiological system in order to understand their dynamic interactions as a source for the behavior of interest. Understanding central properties, such as robustness, emerge as central issues, and understanding these properties will have an impact on the future of medicine. However, many breakthroughs in experimental devices, advanced software, and analytical methods are required before the achievements of systems biology can live up to their full potential.

Within Pharma industry the application of methods from systems biology is still relatively limited. Modeling and simulation is mostly done in terms of empirical or physiologically based pharmacokinetic/pharmacodynamic (PK/PD) models, often comprising not more than a handful of ordinary differential equations. Nevertheless, more complex mechanistic models are coming more and more in scope also for drug discovery and development projects, requiring additional computational tools, bridging the gap between the traditional PK/PD models and the considerably more complex systems biology type of models.

In this work we present a computational framework for modeling, simulation, and analysis of biological and physiological models that originally has its roots in the area of systems biology but is continuously extended to also address the specific needs within the Pharma industry, such as the simulation of clinical trials and interfaces to nonlinear mixed effect modeling packages. We will exemplify the use of this tool based on a simple PK/PD model of Aliskiren, available in the literature<sup>2</sup>. The main reason to use this model as an example is that the regulation of blood pressure is a very complex system that lends itself nicely to a modeling and simulation approach and that we (as a next step) plan to develop a detailed mechanistic model for the renin-angiotensin-aldosterone system (RAAS).

#### **THE TOOL: THE SBPOP PLATFORM**

The SBPOP platform is a collection of three different packages (**Figure 1**) that together provide a powerful, yet flexible and user-friendly, environment for handling complex modeling and simulation tasks, targeting at biological and physiological systems, and addressing the whole range of tasks from the area of systems biology to clinical trial simulations.



**Figure 1: Structure and dependency of the SBPOP platform.**

The basis of this platform is the general purpose numerical analysis software MATLAB (<http://www.mathworks.com>), that since many years is a standard tool in many scientific areas. MATLAB provides numerous state-of-the-art mathematical and numerical methods and a user-programmable platform, using a simple, but powerful, high-level scripting language that avoids programming overhead usually present when developing standalone software applications.

Each package within the SBPOP platform is implemented as a MATLAB toolbox and provides an additional layer of functionality. The lowest layer is implemented in the SBTOOLBOX2<sup>3</sup> (<http://www.sbtoolbox2.org>), providing crucial functionality, such as model representation, conversion, visualization and basic analysis functionality. The next layer is the SBPD package, that requires the presence of the SBTOOLBOX2 and provides parameter estimation functionality and high speed simulations, the third layer is the SBPOP package that adds functionality that is needed for the modeling and simulation needs in the Pharma industry. In the following we will shortly summarize the main features, included in the three different packages.

## SBTOOLBOX2

The basic package is the SBTOOLBOX2<sup>3</sup>. It is the only one of the three packages that can be used without the presence of the two others.

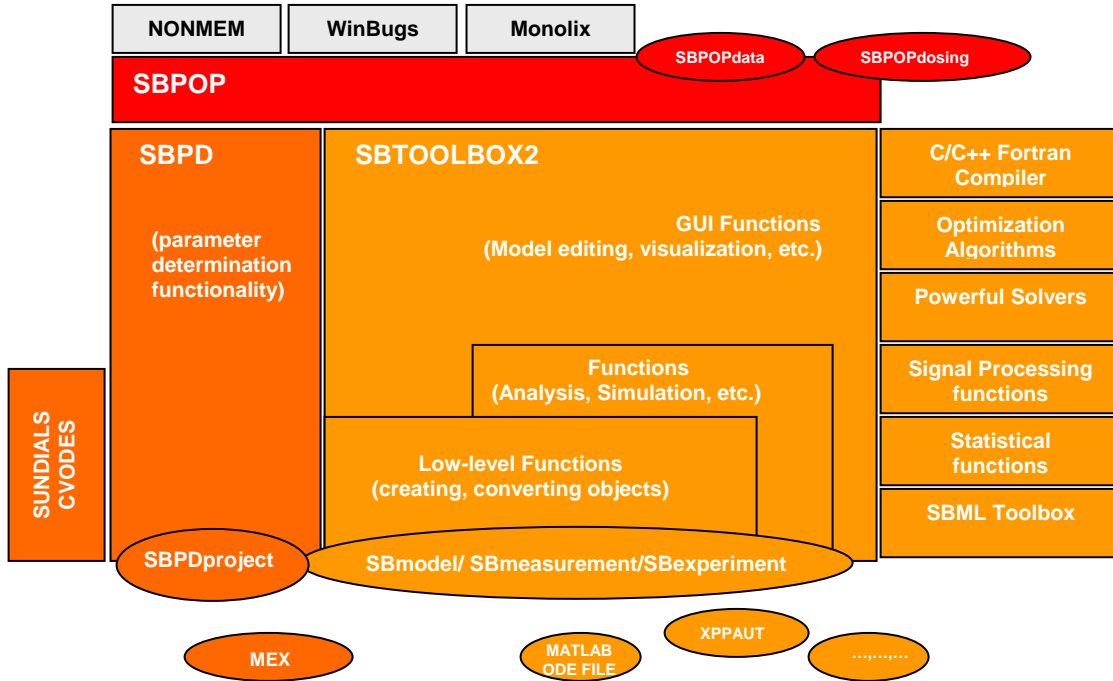
The SBTOOLBOX2 is built in a modular way, depicted in **Figure 2** (light orange color). The base elements are objects of classes SBmodel, SBexperiment, and SBmeasurement, which are used to represent models, experiment descriptions, and experimental measurement data.

The internal representation of SBmodels is based on ordinary differential equations (ODEs). The model elements are parameters, variables, reactions, functions, discrete state events, and ODEs. Low-Level Functions are implemented as methods of the two object classes and include functions for importing and exporting models, experiments, and measurement data, along with the creation of these objects. An SBmodel can be imported from and exported to two different textual descriptions of the same model. One representation is based on biochemical reaction equations, well suited for the interpretation by biochemists and biologists. The other representation is based on ordinary differential equations. Both representations are interchangeable. The toolbox can deal with SBML models (<http://sbml.org>), strongly facilitating the collaboration and the model exchange between researchers within the systems biology community and a wide variety of SBML enabled modeling and simulation tools.

While an SBmodel is the in silico representation of a biological system, SBexperiment objects are the in silico representation of experiments performed in the lab. Such an object can code changes in the initial conditions of the state variables, single changes in parameters, change parameters over time, and represent switching events when certain conditions are fulfilled. SBexperiments and SBmodels can be merged to obtain a simulatable model that reflects the defined experimental settings. Especially in the case where several different experiments are performed on the same system it is very useful to be able to only keep one copy of the model and to code the different experiments independently.

Finally, SBmeasurement do represent experimental measurement data and thus are the counterpart to the combination of a model with an experiment. SBmeasurement objects can be imported from and exported to Excel spreadsheets and comma separated value (CSV) data files.

The functions implemented in the SBTOOLBOX2 can be broken down into four different classes of functions: low-level functions, analysis functions, GUI functions and auxiliary functions.



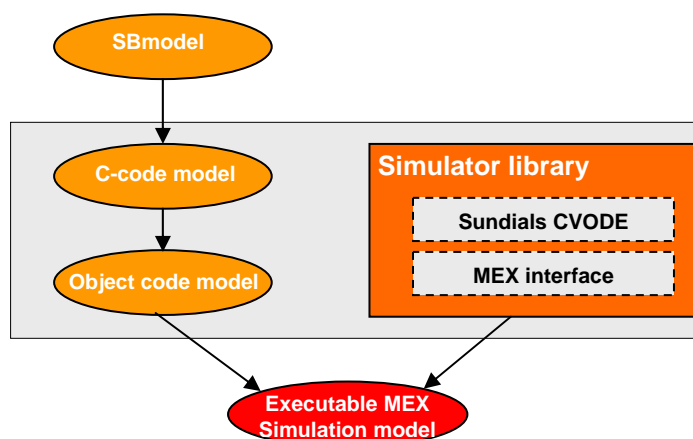
**Figure 2: Detailed modular structure of the complete SBPOP platform. Light orange color: SBTOOLBOX2, orange color: SBPD, red color: SBPOP, grey color: external software for which SBPOP provides / will provide interfaces.**

The low-level functions handle the creation of the three object types, their conversion from and to textual representations and allow to retrieve information from the objects. The analysis function add functionality for simulation, visualization and other basic analysis types, such as steady-state computation, simple model reduction, and sensitivity analysis. The GUI functions implemented several simple graphical user interfaces that are used by different functions in the toolbox to visualize results. Finally, the auxiliary functions do implement a wide variety of needed tools, such as optimization methods, linear and quadratic programming solvers, signal processing and statistical methods, etc.

**SBPD**

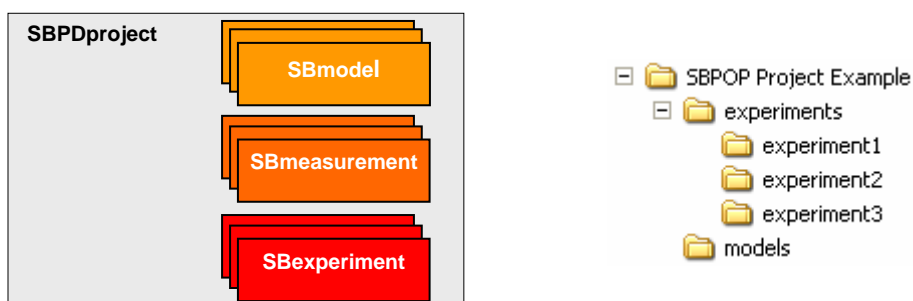
The second layer of the SBPOP platform is implemented by the SBPD package (see **Figure 2**– orange color). This package does require the presence of the SBTOOLBOX2, but does not require SBPOP to be installed.

The main purpose of SBPD is to provide the users with important functionality that is needed for parameter estimation applications. Simulation in the SBTOOLBOX2 relies on the integration of ODEs using standard MATLAB ODE solvers, resulting in relatively poor simulation performance. While this often is not a problem for single simulations, performance does become crucial in parameter estimation applications where the same model is simulated repeatedly for different sets of parameters and/or initial conditions. Through the use of the SBPD package, simulations are sped up by a factor roughly between 50 and 200, depending on properties of the model and integrator settings. This is achieved by automatic conversion of SBmodels to compiled C-code MEX simulation functions, using the SUNDIALS CVODE<sup>4</sup> ODE solver for simulation (see **Figure 3**).



**Figure 3: Automatic conversion of an SBmodel to a compiled and MATLAB executable (MEX) function. The SBmodel is converted to C-code, compiled to object code and then linked with a static library, including the CVODE integrator and a MEX interface to MATLAB.**

Additionally, the SBPD package contains functions for parameter estimation, parameter fit analysis, manual tuning of single models and complete modeling projects (SBPDprojects). SBPDprojects are objects supported in SBPD that allow to combine models, experiments, and measurements (see **Figure 4-left**). This construct is very powerful in the sense that once a project has been setup, parameter estimation functions can immediately be run on these projects, enabling multi experiment and multi measurement parameter estimations in a most straight forward way, without the need of the development of rather complex scripts and customized functions.



**Figure 4: (left) SBPDprojects allow to combine models, experiment descriptions and measurement data. A wide variety of analysis and parameter tuning methods can directly be run on such projects. (right) Example folder structure, representing an SBPOPproject.**

The representation of SBPDprojects is based on a folder structure, where the root folder of the projects can have an arbitrary name. This root folder needs to contain the following two folders: “models” and “experiments” (see **Figure 4-right**). The models folder needs to contain at least one file with the textual description of a model for which the parameters should be fitted. Several model files can be present, for example, in the case that there exist several possible hypotheses for a modeled mechanism. The experiments folder needs to contain subfolders. One subfolder per experiment that has been performed to collect measurement data. Each of these experiment folders needs to contain a single file with the textual representation of the performed experiment and additional CSV or Excel files with the data which was collected during these experiments.

### SBPOP

The third layer of the SBPOP platform is the SBPOP package. This package does require the presence of both the SBTOOLBOX2 and SBPD, since it reuses all the functionality implemented in the two lower layers.

The main purpose of this third layer is to provide additional functionality to the platform that is needed in drug development. Examples of already available functionality are:

- Handling of physiological PK/PD type of models for the feasibility analysis of antibody therapies, including affinity and dose prediction.
- Formal representation of dosing schemes, merging of dosing schemes with SBmodels to obtain simulation models, implementing the defined dosing schemes (SBPOPdosing object), and simulation of dosing schemes on models.
- Import and export of clinical data files in the NONMEM format, handling and transformation of data (SBPOPdata object).

- Automatic generation of MLXTRAN models (for use with Monolix) from SBmodels, SBPOPdosing schemes, and information from SBPOPdata objects.

The SBPOP package is currently under heavy development. Future extensions will be: representation, handling, and simulation of clinical trials, seamless interfaces to nonlinear mixed effect modeling tools, such as NONMEM<sup>5</sup>, Winbugs<sup>6</sup> and Monolix<sup>7</sup>.

#### SOFTWARE AVAILABILITY AND INSTALLATION REQUIREMENTS

Currently, only the first two packages (SBTOOLBOX2 and SBPD) are publicly available. The SBPOP part is still heavily under development and might be made public at a later stage. The newest versions of SBTOOLBOX2 and SBPD can always be downloaded from <http://www.sbtoolbox2.org>. The same webpage contains extensive documentation and examples for each implemented function and a comprehensive tutorial, guiding the user on the first steps with the tool. The packages have been developed in order to only require plain MATLAB (>=R2006a), not requiring the presence of any other commercial MATLAB toolbox. The installation process is limited to running an installation script for each package from within the MATLAB environment and the use of MATLAB as a basis allows the toolbox to be used on Windows, Linux, Unix and MAC OS machines.

The SBTOOLBOX2 is published under the GNU General Public License (Free Software Foundation 1991; <http://www.gnu.org>), while the SBPD package is distributed under a creative commons license model.

#### THE MODEL: APPLICATION OF THE SBPOP PLATFORM TO THE ALISKIREN EXAMPLE

In the previous section we gave a brief overview over the computational tool. In the following we apply the tool to the modeling and analysis of a simple example model in order to demonstrate some of the available features. The considered model is a PK/PD model to evaluate the effects of aliskiren on the renin-angiotensin system (RAS) in humans, published in Hong et al<sup>2</sup>. The PD part of the model tests the current hypothesis that RAS is inhibited by direct renin antagonism. The data to which the model was to be fitted were extracted from a study conducted by Nussberger et al.<sup>8</sup> From their study, we analyzed mean data after administration of placebo and different aliskiren doses (40mg, 80mg, 160mg, 640mg) in 9 healthy volunteers over 8 days. Note that this dose range is wider than the therapeutic doses of aliskiren (150 and 300mg), for treating and controlling hypertension. We chose it to be large in order to test the model on the SBPOP platform.

We reproduced here the simulation of the PK/PD models, using the SBPOP platform in order to show how to represent the model and the measurement data, and how to compare the model to the measurement data by using the construct of insilico experiments.

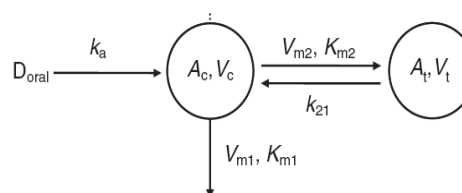
#### PK/PD MODEL (from Hong et al.<sup>2</sup>)

**PK Model:** The drug is characterized by a two-compartment model (**Figure 5**) with nonlinear capacity-limited distribution and elimination. The differential equations used for describing the PK models are as follows:

$$\frac{dA_p}{dt} = -k_a \cdot A_p$$

$$\frac{dA_c}{dt} = k_a \cdot A_p - \frac{V_{m1}}{K_{m1} \cdot V_c + A_c} \cdot A_c + k_{21} \cdot A_t - \frac{V_{m2}}{K_{m2} \cdot V_c + A_c} \cdot A_c$$

$$\frac{dA_t}{dt} = \frac{V_{m2}}{K_{m2} \cdot V_c + A_c} \cdot A_c - k_{21} \cdot A_t$$



**Figure 5: Pharmacokinetic model for aliskiren in humans.**

where  $A_p$  represents the amount of drug at the absorption site ( $A_p(0) = \text{Dose}$ ), and  $A_c$  and  $A_t$  are the drug amounts in the central and peripheral compartments ( $A_c(0) = A_t(0) = 0$ ), respectively. Plasma aliskiren concentrations are defined by  $C_p = A_c/V_c$ . The remaining constants in the system are parameters of the PK model.

**Reduced PD Model:** The PD part of the model utilizes the plasma drug concentration determined in the PK part ( $C_p$ ) and represents the effect of the drug on the secretion of AR (active renin). A simplified representation of the PD model is shown below.

$$\frac{dAR}{dt} = K_s(t) \cdot \left( 1 + \frac{E_{\max} \cdot C_p^h}{EC_{50}^h + C_p^h} \right) - k_{out} \cdot AR$$

with 
$$E_{\max} = E_{\max0} + \left( \frac{S_{\max} \cdot \text{DOSE}}{SD_{50} + \text{DOSE}} - E_{\max0} \right) \cdot \left( 1 - e^{-k_t t} \right)$$

## PhUSE 2009

In order to take into account the placebo effect in the model, this effect on the concentrations of active renin has been determined as a periodic function with a periodicity of 24 hours (daily rhythm). The representation of the placebo effect is described by:

$$K_s(t) = k_{out} \cdot a_0 + \sum_{n=1}^2 \left[ \left( k_{out} \cdot a_n + \frac{b_n \cdot 2\pi \cdot n}{24} \right) \cdot \cos\left(\frac{2\pi \cdot n \cdot t}{24}\right) + \left( k_{out} \cdot b_n - \frac{a_n \cdot 2\pi \cdot n}{24} \right) \cdot \sin\left(\frac{2\pi \cdot n \cdot t}{24}\right) \right]$$

The calculated Fourier coefficients are given in Hong et al.<sup>2</sup>.

The above proposed PK/PD model was fitted to the available data using a two-stage approach. First, the parameters of the PK part of the model were fitted to the aliskiren concentration-time data (after multiple dosing). Subsequently, the parameters of the PD part of the model were fitted to the PD data, while holding the previously optimized PK parameters constant.

Since the PD model, in the second stage, only requires to add one more equation to the PK model (AR activity), we will in the following only discuss the implementation, simulation, and analysis of the PK model. The handling of the full PK/PD model is analogous and uses the same constructs from the SBPOP platform. However, we will show the final simulation results for the complete PK/PD model.

### IMPLEMENTATION OF THE PK MODEL

In the following we outline the implementation of the PK model, presented in the previous section, in the textual representation format of the SBTOOLBOX2:

```

***** MODEL STATES
d/dt(Ap) = - R1 % ODE sytem (model reaction are written below)
d/dt(Ac) = R1-R2+R3-R4
d/dt(At) = R4-R3
Ap(0) = 40 % Initial condition for Ap (if 0, IC definition can be left out)
***** MODEL PARAMETERS
ka = 0.0748 % /h % Settings for the PK parameters
Vc = 176000 % ml
Vm1 = 812000 % ng/h
Km1 = 0.516 % ng/ml
Vm2 = 87480000 % ng/h
Km2 = 135 % ng/ml
k21 = 0.00248 %^h-1
***** MODEL VARIABLES
Cp = Ac/Vc % Aliskiren concentration is the output of the simulation
***** MODEL REACTIONS
R1 = -ka*Ap % Definition of the reaction kinetic expressions
R2 = Vm1/(Km1*Vc+Ac)*Ac
R3 = k21*At
R4 = Vm2/(Km2*Vc+Ac)*Ac

```

### IMPLEMENTATION OF THE MEASUREMENTS

Here we choose to display the CSV representation of measurements of mean plasma concentration at given time instants, after oral administration of the drug. Measurements are listed as “mean, mean plus SEM (standard mean error), mean minus SEM” at given time.

```

[Name] % Measurement name
Measurement Aliskiren

[Notes] % Notes about the measurement

[Components]
time,Cp,Cp+,Cp- % Names of the measured components:
% time of measurement (time), aliskiren concentration (Cp), upper (Cp+)
% and lower bound (Cp-) of the concentration

[Componentnotes]
Cp: component Ac/Vc % Aliskiren concentration

[Values] % Measurement data
0,0,0,0
0.5,4,4.6,3.4
1,4.1,4.77,3.43
3,2.3,2.75,1.85
6,1.7,2.4,1
10,0.6,0.78,0.42
24,0.26,0.31,0.21

```

## PhUSE 2009

### IMPLEMENTATION OF THE EXPERIMENTS

The dosing schedule for the different experiments from the study conducted by Nussberger et al.<sup>8</sup>, is a dose of the drug given to subject every 24 hours for 8 days. It is equivalent to re-initializing the value of  $A_p$  at regular interval of time (every 24 hours). The textual representation of the SBexperiment (when the aliskiren given dose is 40mg over 8 days) is given by:

```
***** EXPERIMENT NAME
Aliskiren PK Model
***** EXPERIMENT INITIAL PARAMETER AND STATE SETTINGS
***** EXPERIMENT PARAMETER CHANGES
***** EXPERIMENT STATE CHANGES
time=24, Ap=Ap+40           % At time =24, a new dose of aliskiren is given to the patient
                           % and added to the current value of the amount of drug at
                           % absorption site
time=48, Ap=Ap+40
...
```

The following script, after combining the SBmodel with the SBexperiment object, produces a new model where the experimental settings have been added, and runs the simulation on the merged model:

```
>> model = SBmodel('model.txt')           % Import the model
>> experiment = SBexperiment('experiment.exp') % Import the experiment
>> modelexp = SBmergemodelexp(model,experiment) % Merge model and experiment
>> simulation = SBsimulate(modelexp,[0:1:216]) % Simulate the merged model
                                                % from time =0 to time = 216
```

The result of the simulation is the time course of mean plasma concentration of the drug after multiple dosing over 8 days (see **Figure 6**)

### IMPLEMENTATION OF THE DOSING SCHEME

The SBPOPdosing scheme is as an alternative to representing the dosing schemes in terms of experiments. It is a new feature that has been added for clinical simulations.

Since the PK model corresponds to a first order absorption in the central compartment, corresponding to state  $A_c$ ; the dosing scheme has the following textual representation:

```
***** INPUT1
type:          ABSORPTION1
time:          0           % (hr) time for first application
deltaT:        24          % (hr) time in-between applications
nr_repetitions: 8          % number of applications
D:             40          % (unit) dose
ka:            0.0748      % (1/hr) absorption rate coefficient
```

The SBmodel that integrates the absorption states is defined by the SBmodel object as stated before where the system of differential equations is then replaced by:

```
***** MODEL STATES
d/dt(Ac) = INPUT1-R2+R3-R4 % INPUT1 is a placeholder for the dosing input
d/dt(At) = R4-R3
```

INPUT1, written in the differential equation for  $A_c$  is a placeholder for the implementation of the dosing scheme, defined in the SBPOPdosing object.

The model can be merged with dosing schemes to result in a simulatable model implementing the dosing schemes. What essentially is done during the merging in this example is to add a dosing compartment and the necessary mathematics to apply the defined dose to the  $A_c$  compartment and a reaction that empties the dosing compartments contents into the  $A_c$  compartment according to first order kinetics.

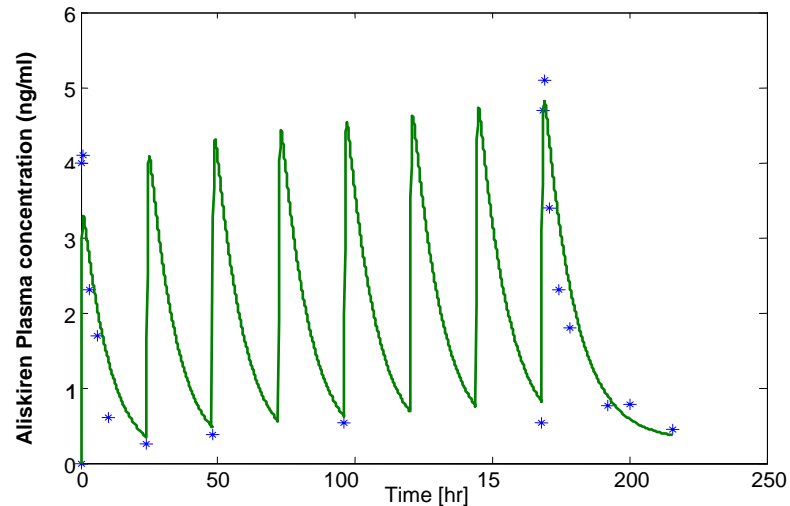
### SIMULATION OF THE PK MODEL

The simulation is run on the model that is produced after merging either with the SBexperiment or with the SBPOPdosing object. The following example code loads the model and a dosing description, merges both, simulates and plots the simulation results. Additionally, a measurement is imported and the data are plotted in the same figure (**Figure 6**).

```

>> model = SBmodel('Pkmodel.txt') % Import model
>> dosing = SBPOPdosing('dosing.dos'); % Import dosing scheme
>> modeldosing = SBPOPmergemoddos(model,dosing); % Merge model and dosing description
>> measurement = SBmeasurement('Measurement.csv') % Import Measurement
>> [time,name,data] = SBmeasurementdata(measurement) % Import time, name and data measurement
>> simdata = SBsimulate(modeldosing,[0:1:216]) % Simulation
>> plot(time, data(:,1), Simdata.time,Simdata.statevalues(:,1)) % Plot

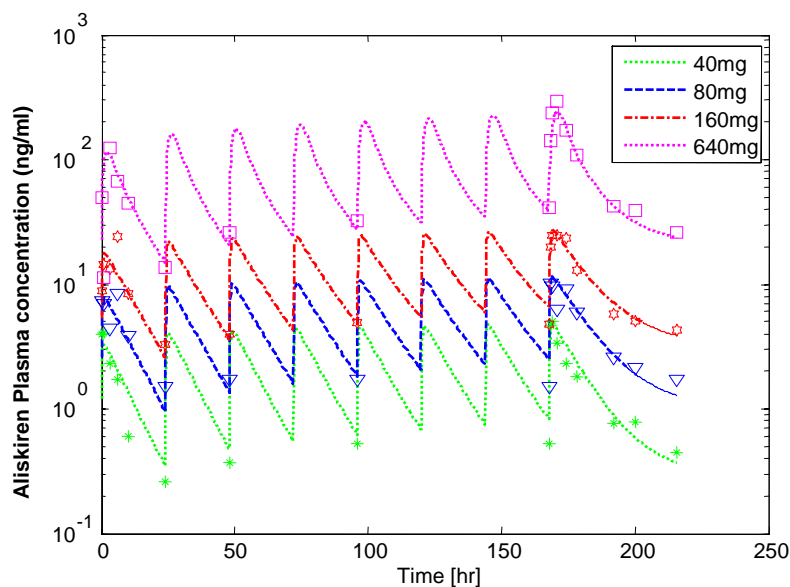
```



**Figure 6:** Time course of mean aliskiren concentration after multiple dosing (40 mg) for 8 days. The symbols represent the data extracted from Nussberger et al.<sup>8</sup> and the line is the model-predicted profile.

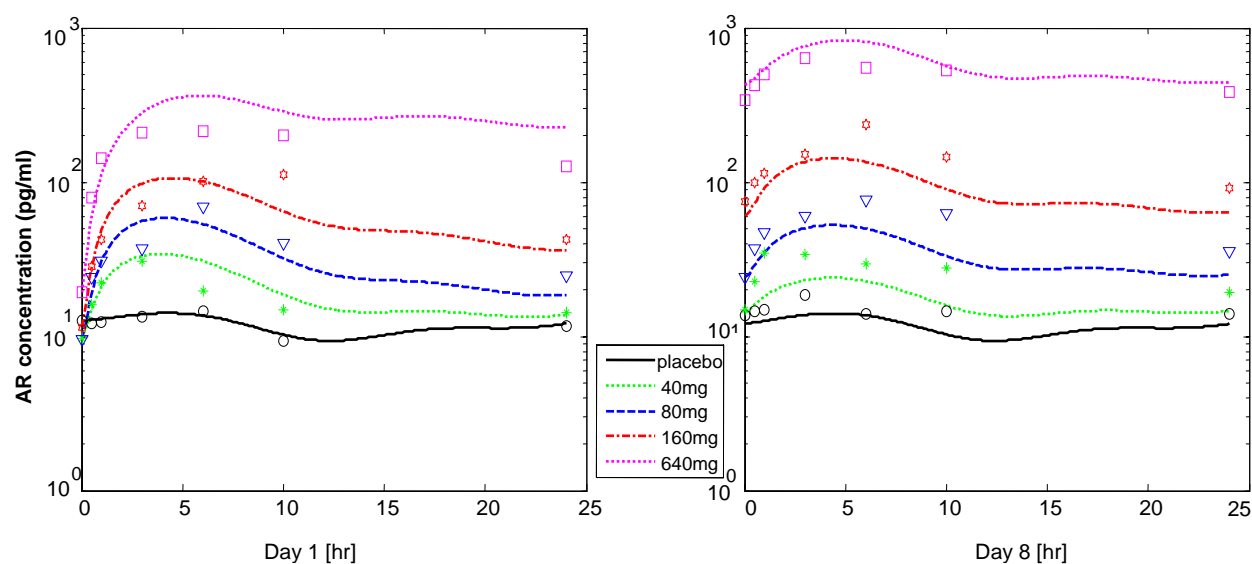
#### ESTIMATION OF THE PK/PD MODEL

Since different dose experiments have been studied by Nussberger et al.<sup>8</sup>, we constructed an SBPDproject representation of those experiments in order to allow to easily handle multiple experiment simulations and associated analysis. The PK simulation for the different doses (40, 80, 160 and 640mg) from the SBPD project leads to **Figure 7**, using the PK values in Table 1 from Hong et al.<sup>2</sup>



**Figure 7:** Time-course of mean aliskiren plasma concentration after multiple dosing for 8 days. The symbols represent the data extracted from Nussberger et al.<sup>8</sup> and the line is the model-predicted profile. Doses are 40, 80, 160 and 640mg.

By incorporating the PD equation into the textual representation of the PK model and performing the same steps, as outlined before, we obtained the response profiles of AR after multiple dosing shown in **Figure 8**, where we also compare simulations to measurement data. For the parameterization we used the model parameter values, published in Hong et al.<sup>2</sup>



**Figure 8: Time-course of mean plasma active renin (AR) concentration in nine healthy male subjects after acute (day 1) and sustained (day 8) inhibition of the renin-angiotensin system by aliskiren. The symbols represent the data extracted from Nussberger et al.<sup>8</sup> and the lines are model-predicted profiles. Treatments are placebo, 40, 80, 160 and 640mg.**

Interestingly, while the model is able to fit most of the PD measurements, there are discrepancies between model and data on day one for the highest dose (640mg) and on day 8 for the lower doses (40 and 80mg). The model simulations, originally shown in Hong et al.<sup>2</sup> did not show these differences. Although we made sure that our model implementation represents exactly the model equations that were published in the original paper, we were not able to explain these differences.

Unfortunately, this is a very common phenomenon when it comes to models that are published in the scientific literature, which can often be attributed to typos or omissions in the model equations. Efforts, such as BioModels Database (<http://www.biomodels.net>), and MIASE/SED-ML<sup>9,10</sup> become increasingly important, ensuring a public access to curated implementations of published models, making sure that the model is able to reproduce the results shown in the original publications and providing a formal definition of how to reproduce these results. Several scientific journals have started to support this idea by requiring the authors to submit also their model implementations as supplementary information, leading to a easier reuse of previous work.

## CONCLUSIONS

Modeling of biological systems is an intensively interdisciplinary process. The use of systems and physiology model is gaining more acceptance due to their intrinsic mechanistic nature and flexibility. There is need to develop tools that marry mechanistic models with clinical observations to address intrinsic uncertainty and variability. Tasks to be performed are first the generation of quantitative data by investigators, and second the model based quantitative analysis by setting up biological and mathematical hypotheses about the considered processes, estimation of model parameters from the measured data, and computational analysis of the resulting models. The presented SBPOP platform provides a modeling and simulation framework to users in a powerful, open, and user extensible environment. The toolbox features a wide variety of specialized analysis tools adding to the already large number of MATLAB in-built functions. Furthermore, the use a high level scripting language like MATLAB allows the user to quickly and efficiently add new functionality to the toolbox.

As an example, the implementation on the SBPOP platform of a semi mechanistic pharmacokinetic PK/PD model<sup>2</sup> to evaluate the effects of aliskiren on the renin-angiotensin system (RAS) in humans validates the flexibility of the toolbox. It provided easy to perform and meaningful analysis allowing the users to test, improve and validate any biological system models, ultimately allowing the simulation of virtual clinical trials.

### REFERENCES

1. Kitano H., Systems Biology: A Brief Overview. *Science*; **295**, 1662-1664 (2002).
2. Hong Y., Dingemans J. & Mager DE., Pharmacokinetic/ Pharmacodynamic Modeling of Renin Biomarkers in Subjects Treated With the Renin Inhibitor Aliskiren. *Clinical Pharmacology & Therapeutics*; **84** (1), 136–143 (2008).
3. Schmidt H. & Jirstrand M., Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics*; **22** (4): 514-515 (2006).
4. Serban, R. & Hindmarsh, A.C., CVODES: the sensitivity-enabled ode solver in SUNDIALS. *In Proceedings of IDETC/CIE 2005*; Long Beach, CA (2005).
5. Beal S.L. & Scheiner L.B., NONMEM User's Guide, parts I-VII. *NONMEM Project Group*; San Francisco (1992).
6. Spiegelhalter D., Thomas A. & Best N., Winbugs Version 1.3 User Manual. *Medical research Council Biostatistics Unit*, Cambridge (2005).
7. Lavielle M., Monolix Users Manual [computer program]. Version 1.1. Orsay, France: *Laboratoire de Mathématiques*, U. Paris-Sud (2005).
8. Nussberger J., Wuerzner G., Jensen C. & Brunner H.R., Angiotensin II suppression in humans by the orally active renin inhibitor aliskiren (SPP100): comparison with enalapril. *Hypertension*; **39**, E1–E8 (2002).
9. Köhn D., Adams R., Bergman F., Goryanin I., Hucka M., Kolpakov F., Le Novère N., Moraru I., Sahle S. & Schmidt H., Describing Simulation Experiments using the SED-ML. *Poster, 17th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) and 8th European Conference on Computational Biology (ECCB)* (2009 to appear).
10. Köhn D. & Le Novère N., SED-ML - An XML Format for the Implementation of the MIASE Guidelines. *Computational Methods in Systems Biology* ed. by M. Heiner and A.M. Uhrmacher, Springer-Verlag Berlin Heidelberg. LNBI; 176-190 (2008).

### ACKNOWLEDGMENTS

The authors would like to thank Phil Lowe for discussions about the validity of the implemented PK/PD model.

### CONTACT INFORMATION

Comments and questions are valued and encouraged. Correspondence to::

Bruno Bieth  
Novartis Pharmaceuticals  
Modeling and Simulation  
Novartis Campus  
CH-4056 Basel  
Work Phone: +41.61.32.45823  
bruno.bieth@novartis.com

Brand and product names are trademarks of their respective companies.