BIO Integration

Original Article

Embedding R inside the PhysPK Bio-simulation Software for Pharmacokinetics Population Analysis

Sergio Sánchez-Herrero[1,2,*], Fernando Carbonero Martínez[2], Jenifer Serna[2], Marina Cuquerella-Gilabert[2], Almudena Rueda-Ferreiro[2], Angel A. Juan[3] and Laura Calvet[4]

Abstract

Background: PhysPK stands as a flexible and robust bio-simulation and modeling software designed for analysis of population pharmacokinetics (PK) and pharmacodynamics (PD) systems. PhysPK equips users with standard diagnostic plots for pre- and post-analysis to delineate PK and PD within population-based frameworks. Furthermore, PhysPK facilitates the establishment of mathematical models that elucidate the intricate interplay between exposure, safety, and efficacy.
Methods: Enhancing simulation modeling capabilities necessitates seamless integration between commercial discrete-event PK and PD simulation tools and external software. This synergy can be amplified by incorporating open-source solutions, like R, which boasts a rich array of comprehensive packages tailored for diverse tasks, including data analysis (ggplot2), scientific computation (stats), application development (shiny), back-end web development (dplyr), and machine learning (CARAT). The integration of R within PhysPK holds the potential to efficiently interpret and analyze PK/PD output and routines using R packages.
Results: This article presents a tutorial that highlights the incorporation of R code within PhysPK and the rendering of R scripts within the PhysPK monitor. The tutorial utilizes a two-compartment model for comparison against the analysis developed by Hosseini et al. in 2018 within the context of the gPKPDSim application and WinNonlin® software. The illustrative example that is provided and discussed demonstrate estimated and simulated plots, revealing negligible differences in the significance for C_L and C_{Ld} (6.89 ± 0.2 and 45.5 ± 17.4 [reference], and 7.06 ± 0.32 and 49.04 ± 9.2 [PhysPK], respectively), as well as volumes V_1 and V_2 (49.15 ± 3.8 and 34.61 ± 5.2 [reference], and 48.8 ± 3.66, and 33.2 ± 3.95 [PhysPK], respectively).
Conclusions: Our study underscores the potential of integrating open-source software, replete with an array of innovative packages, to elevate predictive capabilities and streamline analyses in PK methods. This integration ushers in new avenues for an advanced intelligent simulation modeling within the realm of PK, thus holding significant promise for the advancement of drug research and development.

Keywords

Biosimulation, modeling, pharmacokinetics, PhysPK, R.

[1]Department of Computer Science, Multimedia and Telecommunication, Universitat Oberta de Catalunya, Avenida Tibidabo 39-43, 08018, Barcelona, Spain

[2]Empresarios Agrupados Internacional S.A., Simulation Department, Magallanes 3, 28015, Madrid, Spain

[3]Department of Applied Statistics & Operations Research, Universitat Politecnica de Valencia, Camino de vera S/N, 46022, Valencia, Spain

[4]Telecommunications and Systems Engineering Department, Universitat Autònoma de Barcelona, Carrer Emprius, 2, 08202 Sabadell, Catalonia, Spain

*Correspondence to: Sergio Sánchez-Herrero, E-mail: ssanchezherre@uoc.edu

Received: May 8 2023
Revised: July 12 2023
Accepted: August 23 2023
Published Online: September 13 2023

Available at: https://bio-integration.org/

Introduction

Population-based approaches and mathematical modeling are key elements for pharmacologists and pharmaceutical companies to characterize the pharmacokinetics (PK) and pharmacodynamics (PD) of a drug product. By analyzing the relationship between exposure, safety, and efficacy, these techniques provide critical insight for drug development. It is imperative that the pharmaceutical industry makes these tools accessible to a wider audience to ensure transparency and reproducibility of the modeling process. Additionally, PK and PD descriptions are essential for regulatory approval. The United States Food and Drug Administration (FDA) and the European Medicines Agency (EMA) require comprehensive PK and PD data to evaluate the safety and efficacy of a new drug. Modeling and simulation (M&S) techniques have emerged as powerful tools to provide faster and cost-effective answers regarding a new drug's safety and efficacy, making them increasingly popular in the pharmaceutical industry [1–4].

Although many PK and PD software tools are available for generating diagnostic tables and plots, the built-in tools are often inflexible and inefficient. Therefore, it might be challenging to incorporate created displays into presentations, reports, and manuscripts [5, 6]. Computer communities, however, have proposed alternative solutions, such as linking PK and PD

doi: 10.15212/bioi-2023-0008

compuscript

platforms with external programming languages (Python, Julia, R, or MATLAB). In so doing, the simulation modeling process becomes more dynamic and flexible, thus allowing for more mathematical and algorithmic capabilities [7].

Among these languages, R [8] stands out as an open-source, platform-independent, and general-purpose programming language. R offers numerous complex and comprehensive packages for data analysis, scientific computing, application development, back-end web development, and machine learning. The popularity of R has skyrocketed in recent years due to its machine learning library stack. R has become one of the leading technologies for building models and data mining in many industries and developing new methods for researchers [9–12]. In the context of PK and PD models, R has the capability to automatically detect incomplete or erroneous data and implement customized rules that ensure the accuracy and consistency of information across different projects and processes. Leveraging the power of R can offer a multitude of benefits for data processing and manipulation, experimentation, optimization, result analysis, and visualization [13, 14]. Using R for PK and PD modeling offers limitless advantages in these areas. With its ability to handle complex data structures and its vast array of statistical and graphical packages, R can efficiently manage large datasets, explore and analyze data, and generate visual representations of results. Additionally, the flexible and modular programming language of R allows for rapid prototyping and model development, making R an ideal tool for iterative and exploratory research. Overall, R provides a versatile and powerful platform for pharmacologists and pharmaceutical companies to optimize PK and PD modeling workflow and generate robust results.

PhysPK® (https://www.physpk.com/) is a flexible bio-simulation software in the PK field for characterizing drug products through population-based proposed inside an integrated development environment (IDE) powered by EcosimPro® platform. PhysPK is a first-class simulation tool for modeling continuous-discrete PK and PD systems. The object-oriented approach prevents re-coding simulation applications every time a new project is initiated [15–17]. PhysPK offers a versatile and multidisciplinary tool for analyzing and simulating scenarios in non-compartmental analysis, PK and PD, physiologically-based pharmacokinetics (PBPK), and quantitative systems pharmacology (QSP). In addition, PhyPK is a multi-paradigm numerical computing environment and programming language IDE that supports tools for modeling, simulation, visualization graphs, and statistics. These data can be visualised in the monitor in a quick, editable, and easy way. To enrich PhysPK with R packages, the PhysPK flexibility tool can be configured to run external scripts. Thus, individual and population-based PK and PD M&S methods in PhysPK can be enhanced with R packages for stats, graphs, and reports to improve PhysPK outputs for optimization, estimation, or simulation.

Integrating R scripts and solutions inside PK and PD software can be useful to carry out complete PK population analysis and descriptions, especially because R packages are continually evolving to provide innovative new methods and solutions for a range of areas, including pharmacology and PK [18, 19]; however, there is currently no PK software that integrates R scripts inside PK population analysis in one step [20]. To address this gap in functionality, we propose using the new functionality of PhysPK to embed R inside the program to render R scripts. This flexible feature of PhysPK can be used for any output from the bio-simulation software, with default scripts available for common statistical processes, such as spaghetti plots, goodness-of-fit plots, parameter distribution profiles, or visual predictive checks. Although a basic understanding of the R programming language is required for adding new scripts, embedding R packages can simplify and improve the interpretation, visualization, and analysis of PK and PD output, as demonstrated by the plots, graphs, and statistics presented herein.

This tutorial has two goals: (i) describe the integration between PhysPK and R; and (ii) provide an illustrative example by implementing the validation of a population analysis developed by Hosseini et al. [21]. The remainder of this paper is structured as follows: Section 2 presents the methods used, including descriptions of the dataset and model for simulation and data fitting re-estimation, software, initial configuration of PhysPK, R scripts configuration, and model validation; Section 3 describes the results obtained; Section 4 provides a comprehensive discussion; and Section 5 draws some conclusions and points out a few lines of future research.

Methods

Dataset and model for simulation and data fitting re-estimation

To examine the functionality of embedding an R script within bio-simulation software, we implemented the two-compartment model developed by Hosseini et al. [21] (case 1). The dataset used in the analysis was provided by Hosseini et al. [21]. The study design involved a single dose (10 or 100 mg/kg) intravenous (IV) administration with a 15-min administration rate and 35 days of treatment. Each group was comprised of three subjects. A standard two-compartment antibody PK model was used to estimate the PK parameters.

The workflow connection is illustrated in **Figure 1** and will be briefly described below. The current population estimation module of PhysPK provides a first-order (FO) estimation method, a FO conditional estimation method (FOCE), and a FO conditional estimation method with interaction (FOCE-I) [22] to obtain the parameters of a population model. The process consists of three phases:

1. Initial condition estimation. The two-stage (TS) method [22] is applied to obtain an initial solution of the mean population parameters and their distribution.
2. Population parameter estimation. The population parameters are estimated using the FO, FOCE, or FOCE-I estimation methods, starting with values from the TS stage or the user's manual, as desired.
3. Post-process. The specific parameters for each individual are estimated through the population parameter distribution computed in the second stage. Also, there is the option of being provided by the user.

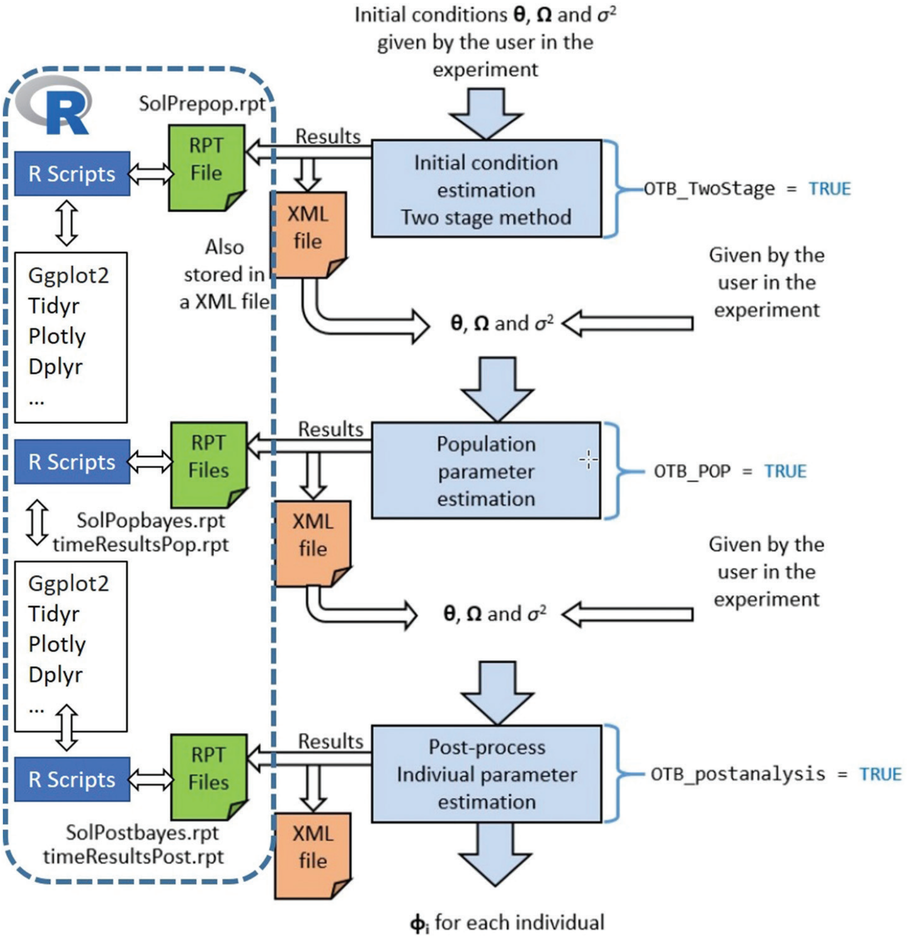


Figure 1 Workflow connection for the study implemented in PhysPK population analysis. R scripts are embedded for the automatic graphical display of population data from the study results.

This estimation block generates results files in *.rpt* or *.txt* format. The modeler can obtain different results of the population parameter estimation in these files. These files could be connected automatically with R scripts to develop complete results descriptions to show estimation results graphically.

Figure 2 shows a schematic diagram of the two-compartment PK model that we developed. By separating the single compartment into two distinct containers, known as the "central" and the "peripheral," we were able to account for distribution parameters unlike the one-compartment model. The central compartment represents plasma and highly perfused tissues, such as the kidneys and liver. The peripheral compartment represents the tissue space. The contrast agent leaves the plasma space at a rate represented by K_{12} (the volume transfer constant) and returns represented by K_{21} (the efflux constant).

This example provides open source, real observations, and data to validate our approach. The PK parameters were estimated using the PhysPK parameter estimation module. The PK model is defined by simulation component relationships

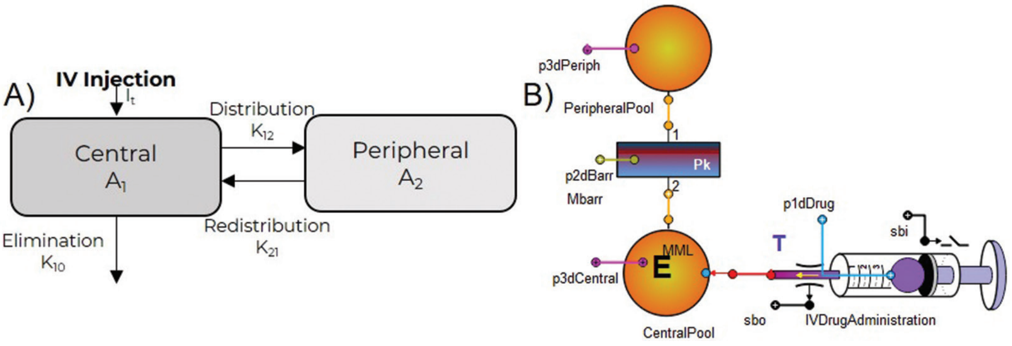


Figure 2 A schematic diagram of the PK model developed. The model includes a two-compartment PK model with specific and non-specific clearance. A) Two-compartment PK model single-dose intra-venous (IV) administration graph. B) Two-compartment PK model single-dose intra-venous (IV) administration PhysPK graph.

and mathematical equations of the mechanism. The mathematical equations for the components involved are shown in equations 1-5:

$$\frac{dA_1}{dt}=X_0-K_{10}\cdot X_1-\frac{V_m\cdot X_1}{K_m+X_1}-K_{12}\cdot(X_2-X_1) \tag{1}$$

$$\frac{dA_2}{dt}=K_{21}\cdot(X_1-X_2) \tag{2}$$

$$X_0=\frac{D_0}{I_t} \tag{3}$$

$$X_1=\frac{A_C}{V_1} \tag{4}$$

$$X_2=\frac{A_P}{V_2}, \tag{5}$$

where A_1 and A_2 characterize the amount of drug in the central and peripheral compartments, respectively. The units for drug concentrations are microgram/milliliter (μg/ml). D_0 represents the administration dose. I_t symbolizes the infusion rate time (0.6 mg/min for the 10 mg/kg IV single dose and 6.6 mg/min for the 100 mg/kg IV single dose). X_0 represents the initial drug concentration after administration. X_1 and X_2 denote the drug concentration in the central and peripheral compartments, respectively. Parameters V_1 and V_2 represent the volume of central and peripheral compartments, respectively. K_{10} represents the clearance from the central compartment, while K_{12} and K_{21} denote the distribution clearance between the central and peripheral compartments, respectively. Kinetic parameters for non-linear elimination are characterized by K_m (the Michaelis constant) and V_m (characterizes the maximum velocity achieved by the system at maximum [saturating] substrate concentrations).

The iterative two-stage (ITS) method is an approach to estimate an initial condition of population parameters for each individual subject without considering population knowledge. This approach is followed by an estimation of population parameters using the FOCE method, starting with the values obtained from ITS. With the help of the PhysPK parameter estimation module, end-users can estimate model parameter values to provide the best fit of model simulation to data with these estimation methods [23]. The initial values for the estimation methods are provided in **Table 1**. Finally, a two-compartment analysis is performed with PhysPK to develop precision plots.

Software

PhysPK v.2.4.1® is a software platform based on first-principle modeling of complex systems with continuous and discrete time equations. PhysPK uses the multi-object-oriented modelling (MOOM) paradigm and is coded using the EcosimPro language. The EcosimPro 6.4.0® program is a first-class modeling and simulation software for modeling multidisciplinary continuous-discrete systems and any kind of system based on differential-algebraic equations (DAE) and discrete events. EcosimPro models can be converted to algorithmic code (C++) and compiled through the EcosimPro platform for execution [24].

The PhysPK approach involves creating a PK model using mathematical equations in a PK simulation component that describes various processes through DAE. These equations are used to ensure mass conservation, as well as model processes, such as absorption, distribution, metabolism, and excretion for each chemical compound within the relevant spatial regions of the component. The aim is to create a comprehensive model that can accurately simulate the pharmacokinetic behavior of the drug within the body [25].

R is a versatile, cross-platform, free, and open-source programming environment that offers powerful packages for data management, statistical computing, and graphical production capabilities. Version 4.2.1 of the R programming language was used in the current study for data pre-processing and package implementation. The following packages were utilized: ggplot2 [26]; readxl [27]; rstudioapi; and this.path [28]. Unless otherwise specified, default parameters were used for each programming function.

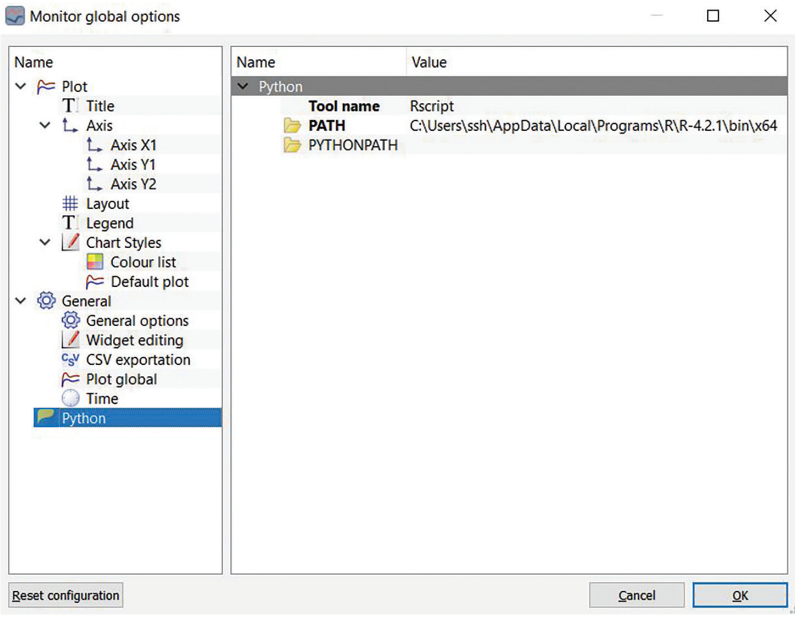
Initial configuration of PhysPK

After PhysPK is installed, PhysPK undergoes automatic validation by comparing the simulation of internal models and studies against reference models to ensure reusability. Additionally, advanced verification tools are included. A configuration is required to run R scripts within the PhysPK monitor, allowing one to select the version of R to use. Paths in the edit tab and global options need to be configured (**Figure 3**). Note that the version of R chosen will affect the libraries that can be used. The configuration process for linking PhysPK with R scripts requires four additional steps (**Figure 4**): *1)* Define the path to the R script.exe folder and specify the tool name to be used in the script. It is crucial to

Table 1 Initial Conditions for the Two-Compartment Model

Variable	Estimated	Initial	Units	Min	Max
V_1	TRUE	40	Milliliter/kilogram	10	200
V_2	TRUE	40	Milliliter/kilogram	10	200
C_L	TRUE	5	Milliliter/day/kilogram	1	30
C_{Ld}	TRUE	10	Milliliter/day/kilogram	1	100
V_m	FALSE	0	Microgram/day/kilogram	0	1200
K_m	FALSE	5	Microgram/milliliter	0.01	100

V_1: central volume; V_2: peripheral volume; C_L: central clearance; C_{Ld}: peripheral clearance; *Vm*: maximum velocity achieved by the system at maximum (saturating) substrate concentrations; *Km*: the Michaelis constant (kinetic parameters).



Name	Value
Tool name	Rscript
PATH	C:\Users\ssh\AppData\Local\Programs\R\R-4.2.1\bin\x64
PYTHONPATH	

Figure 3 Screenshot showing the monitor global options inside of PhysPK settings to embed R scripts.

ensure that the R script functions correctly because it will not display any errors if it fails. Refer to **Figure 4** for detailed instructions on configuring the monitor. *II)* Design a script file to execute within the render options located in the same folder as the PhysPK experiment. *III)* Ensure that the output variables from PhysPK are saved in *.csv* or a similar format to be managed inside the R script. Specify the location where the R script will generate the output. *IV)* Define the variables from PhysPK that will be stored in the *.csv* file. For more information on how to use PhysPK and this new feature, refer to the user manual [29], which includes detailed examples.

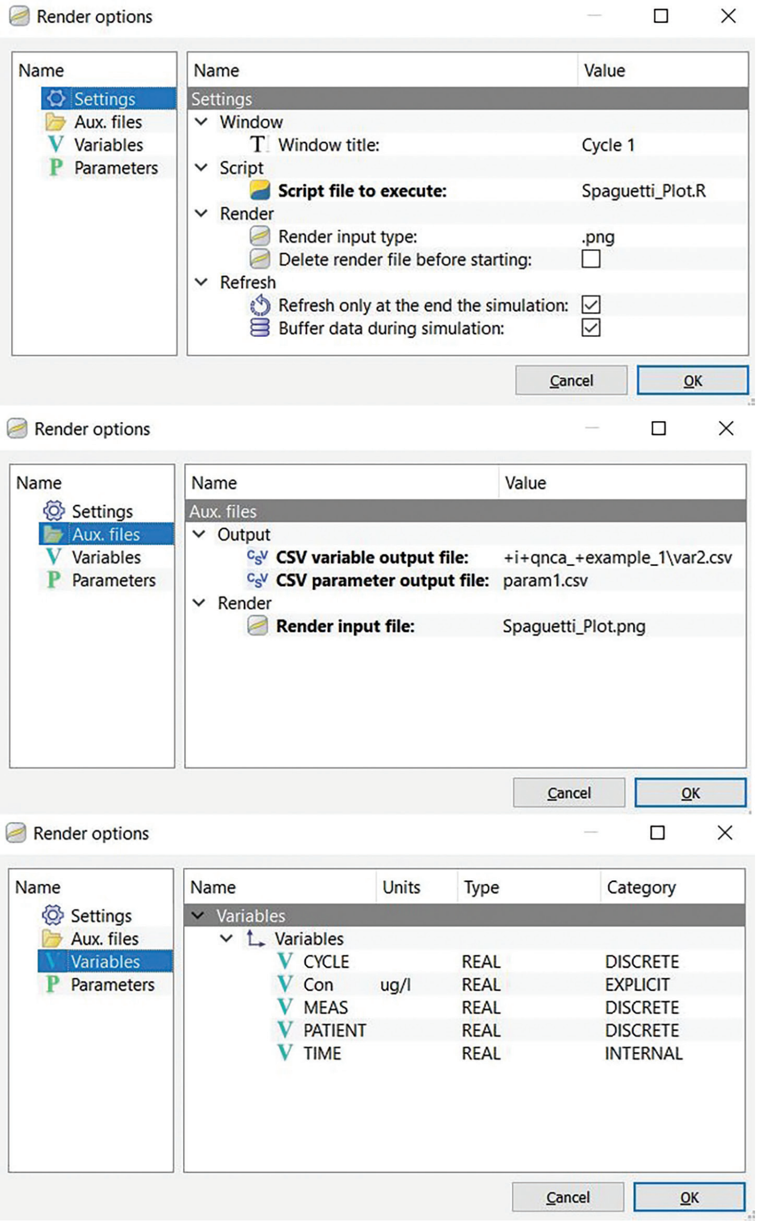
R scripts configuration

R scripts run simultaneously with PhysPK models and are used to associate variables or parameters with the mathematical model from a *.csv* file, where the variables are saved. The *.csv* file should contain the complete history of each selected variable in the form of a table with the variable names in the header to be used by R scripts. The R script file must be located in the same folder as the PhysPK experiment. As a result, all scripts must include mandatory lines to connect with PhysPK parameters, as shown below. The this.path library is used for path handling operations and to locate the script file in the PhysPK experiment folder. After defining the file script location, the main code of the script should connect with PhysPK outputs and generate stats, graphs, plots, and tables. The final lines are used to save an image created by R and display the image in the PhysPK monitor. It is also possible to develop *.xml* files to show additional content in the PhysPK monitor. Below is the entire R script code mandatory for any R script.

```
R>library(this.path)
R># Get the directory from the path of the current file.
R>cur_dir2 = dirname(this.path())
R># Set the working directory.
R>setwd(cur_dir2)
R># Check that the working directory has been set as desired.
R>getwd()
R>png("desirefilename.png", width = 10, height = 5, units=’in’,res=300)
R>dev.control("enable")

..... R code for stats, graphs, plots, tables, etc.....

R>dev.copy(png, file = "desirefilename.png")
R>dev.off()
R>dev.off()
```



Render options

Name
Settings
Aux. files
Variables
Parameters

Name	Value
Settings	
Window	
Window title:	Cycle 1
Script	
Script file to execute:	Spaguetti_Plot.R
Render	
Render input type:	.png
Delete render file before starting:	☐
Refresh	
Refresh only at the end the simulation:	☑
Buffer data during simulation:	☑

Render options

Name
Settings
Aux. files
Variables
Parameters

Name	Value
Aux. files	
Output	
CSV variable output file:	+i+qnca_+example_1\var2.csv
CSV parameter output file:	param1.csv
Render	
Render input file:	Spaguetti_Plot.png

Render options

Name
Settings
Aux. files
Variables
Parameters

Name	Units	Type	Category
Variables			
Variables			
CYCLE		REAL	DISCRETE
Con	ug/l	REAL	EXPLICIT
MEAS		REAL	DISCRETE
PATIENT		REAL	DISCRETE
TIME		REAL	INTERNAL

Figure 4 Screenshots showing the configuration inside of PhysPK settings to embed R scripts.

Model validation

For the model optimization step, 40 maximum-likelihood expectation-maximization (MLEM) iterations are used. The two-compartment model is described using log-normal models; a combined model is applied to the residual error.

A population dataset of 1000 subjects is generated using Monte Carlo simulation in PhysPK. A linear two-compartment with intravenous administration, Michaelis-Menten peripheral distribution, and FO elimination is applied. Common values and variability of model parameters are provided in **Table 2** for 10- and 100-mg doses. The model is characterized using log-normal distributions and an additive plus proportional error model. Concentrations are simulated for each subject at 1, 2, 3, 4, 6, 12, 24, and 36 h following a single intravenous dose of 10 or 100 mg administered for 15 min.

Table 2 Common Values and Variability of Model Parameters for the Simulated Data

	Simulated Data Two-compartment Model
Dose	10 mg/ 100mg
Population	1000 subjects
Population Parameters	K_a = 0.042*h* – 1 V_c = 40*i* V_p = 40*i* C_{lt} = 5*i*/*h* C_{ld} = 10*i*/*h*
Intersubject Variability	10% on Clt and VC
Residual Error	Var = (0.01 + 0.1·IPRED)^2

The predictions of two-compartment concentrations via Monte Carlo simulation are com- pared to the results reported by Hosseini et al. [21]. Additionally, the predictions of two-compartment parameters are evaluated and compared. The parameter values are estimated using maximum *a posteriori* (MAP) Bayesian forecasting, which is commonly used in therapeutic drug monitoring (TDM). To assess the model's prediction performance, we calculated the prediction error (PE; Eq. 6) and the mean prediction error (MPE; Eq. 7).

$$PE(\%)=\frac{PRED-OBS}{OBS}\cdot 100 \tag{6}$$

$$MPE=\frac{1}{n}\cdot\sum PE \tag{7}$$

The overall predictability of the model is evaluated with respect to bias and precision using the conventional metrics of average-fold error (AFE; Eq. 8) and absolute average-fold error (AAFE; Eq. 9), respectively.

$$AFE=10^{\frac{\sum log\frac{PRED}{OBS}}{n}} \tag{8}$$

$$AAFE=10^{\frac{\sum\left|log\frac{PRED}{OBS}\right|}{n}} \tag{9}$$

If the model predictions satisfy the criteria for the AFE and AAFE between 0.8- and 1.25-fold, the predictive performance is considered to be satisfactory [30, 31].

In addition, a visual predicted check (VPC) is carried out to assess the PK model performance based on simulations. To ensure consistency with the Monte Carlo simulations, the observed concentrations were normalized by dose and expressed as the standard dose used in the simulations for each respective dose group. The prediction intervals (PIs) were calculated based on a virtual population of 1000 patients, with both 90% and 50% PIs taken into account. If the observed concentrations are distributed within the 90% PIs, the model prediction capability is deemed to be adequate [32]. Then, VPC plots are generated with R based on the output of the simulations performed in PhysPK.

Observations of the parameter distribution may reveal the presence of subpopulations that can be identified by a binomial distribution. In R, it is possible to create a fitted density estimate distribution and a histogram of estimate frequencies for the Bayesian estimate distribution parameters for each model. These types of plots enhance the informational content by displaying statistics, such as the estimated parameter mean, median, and range. The distribution and frequency plots were generated based on Monte Carlo analysis (**Table 2**). An example of an R script code used to obtain model validation results is provided below.

```
R># Load libraries
R>library(ggplot2)
R>library(xpose)
R>library(dplyr)
R>library(readxl)
R>library(rstudioapi)
R>library(this.path)
R>library(gridExtra)
R>library(writexl)
R># Get the directory from the path of the current file.
R>cur_dir2 = dirname(this.path())
R># Set the working directory.
R>setwd(cur_dir2)
R># Check that the working directory has been set as desired.
R>getwd()
R># Read tabular data into R
R>data <- read.csv("Observed_Data.csv", header = TRUE, sep = ";", dec = "."
    )
R># Read tabular data into R
R>Predicted <- read.table("timeResultsPop.txt", fill = TRUE , header =
    FALSE)
R>Predicted<-  Predicted[-(1:6),]
R>names(Predicted) <- Predicted[1,]
R>Predicted <- Predicted[-1,]
R>#PredvsObs <- cbind(Predicted,data)
R>Predicted$PRED <- as.numeric(Predicted$PRED)
R>data$LN <- as.numeric(data$LN)
R>PE <- ((Predicted$PRED-data$LN)/data$LN)*100
R>Predicted_Error <- data.frame(Predicted$Individual,Predicted$time_sample,
    data$LN, R>Predicted$PRED, data$CONC, data$DOSE, data$CYCLE, Predicted$
    RES,Predicted$WRES, PE)
R>n<-max(Predicted$Individual)
R>n <- as.numeric(n)
R>PredvsObs<- Predicted$PRED/data$LN
R>PredvsObs<-log(x=PredvsObs, base=2)
R>PredvsObs <- as.numeric(PredvsObs)
R>PredvsObs<- data.frame(t(sapply(PredvsObs,c)))
R>PredvsObs<-sum(PredvsObs[, ])
```

```
R>AFE <- 10^((1/n)*PredvsObs)
R>PredvsObs_AFFE<- Predicted$PRED/data$LN
R>PredvsObs_AFFE<-log(x=PredvsObs_AFFE, base=2)
R>#PredvsObs_AFFE<-abs(PredvsObs_AFFE)
R>PredvsObs_AFFE <- as.numeric(PredvsObs_AFFE)
R>PredvsObs_AFFE<- data.frame(t(sapply(PredvsObs_AFFE,c)))
R>PredvsObs_AFFE<-abs(sum(PredvsObs_AFFE[, ]))
R>AAFE <- 10^((1/n)*PredvsObs_AFFE)
R>PE_table <- data.frame(AFE =  c(1),
                         AAFE =  c(2))
R>PE_table[PE_table == 1] <- AFE
R>PE_table[PE_table == 2] <- AAFE
R>Predicted_Error<- cbind(Predicted_Error, PE_table)
R>MPE <-png("Predicted_Error.png", height =
        25*nrow(Predicted_Error),width=100*ncol(Predicted_Error))
R>dev.control("enable")
R>grid.table(Predicted_Error)
R>write.csv(Predicted_Error,"Predicted_Error.csv", row.names = FALSE)
R># Read tabular data into R
R>results <- read.csv("Predicted_Error.csv", header = TRUE, sep = ",", dec
    = ".")
R>png("Predicted_Error.png", height = 50*nrow(results), width = 200*ncol(
    results))
R>dev.control("enable")
R>grid.table(results)
R>dev.off()
R>dev.off()
```

Results

Dependent variable versus time profiles

Although PhysPK graphs, plots, and statistics are valuable tools for characterizing and projecting PK for different dosing scenarios (**Figure 5**), Kamath [33] demonstrated the potential of R in analyzing PK data from a monoclonal antibody administered as a single dose. Using real data from Hosseini et al. [21], individual PhysPK plots are generated prior to prediction, with log-concentration (μg/ml) plotted on the x-axis and time in hours on the y-axis. The TS method [23] is then applied to obtain an initial solution for the mean population parameters and the distributions. We simulate

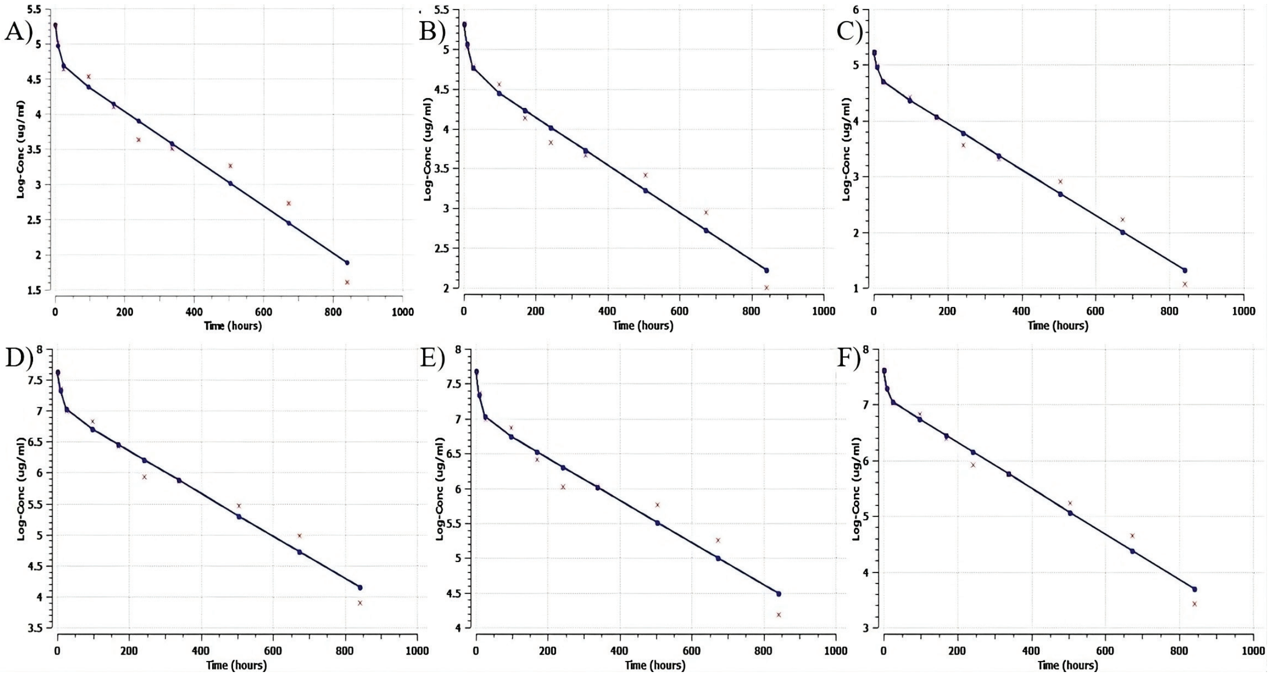


Figure 5 A-F are PhysPK individual's plots Pre prediction. Log-Con μg/ml for x axe and time in hours for y axe to obtain an initial solution of mean population parameters and their distributions.

and predict multi-dose PK and population variability by fitting the data to a two-compartment PK model. These findings have significant implications for drug development and clinical use.

By connecting R with PhysPK, users can generate reliable graphical output using comprehensive graphical packages from the R programming language. PhysPK is compatible with any population dataset formatted according to regulatory requirements. Users can input data files in either *.txt* or *.csv* format and specify the file path to read the desired data file, while considering headers for typical names, such as subject identifier (ID), dependent variable (DV), missing dependent variable (MDV), and elapsed time (TIME). Users can also adapt these headers to their preferences in PhysPK. After the formatted data is read by PhysPK, users can run R scripts to generate population scatter and line plots (also called spaghetti plots) for each dose level detected in the dataset, as well as individual plots (**Figure 6**).

Plots generated by PhysPK thanks to R are widely diverse. Any plot combining PhysPK outputs and R features may be created, such as plots of the dependent variable versus time. The purpose of these graphs is to provide fit plots after estimation parameters on the PK profile shapes. Thus, this includes observing the density of sampling, the influence of the dosing history on the PK and PD profiles, non-linearity, potential inter-individual variability (IIV), and the number of points below the limit of quantitation from observed data. This feature for embedding R inside PhysPK may create

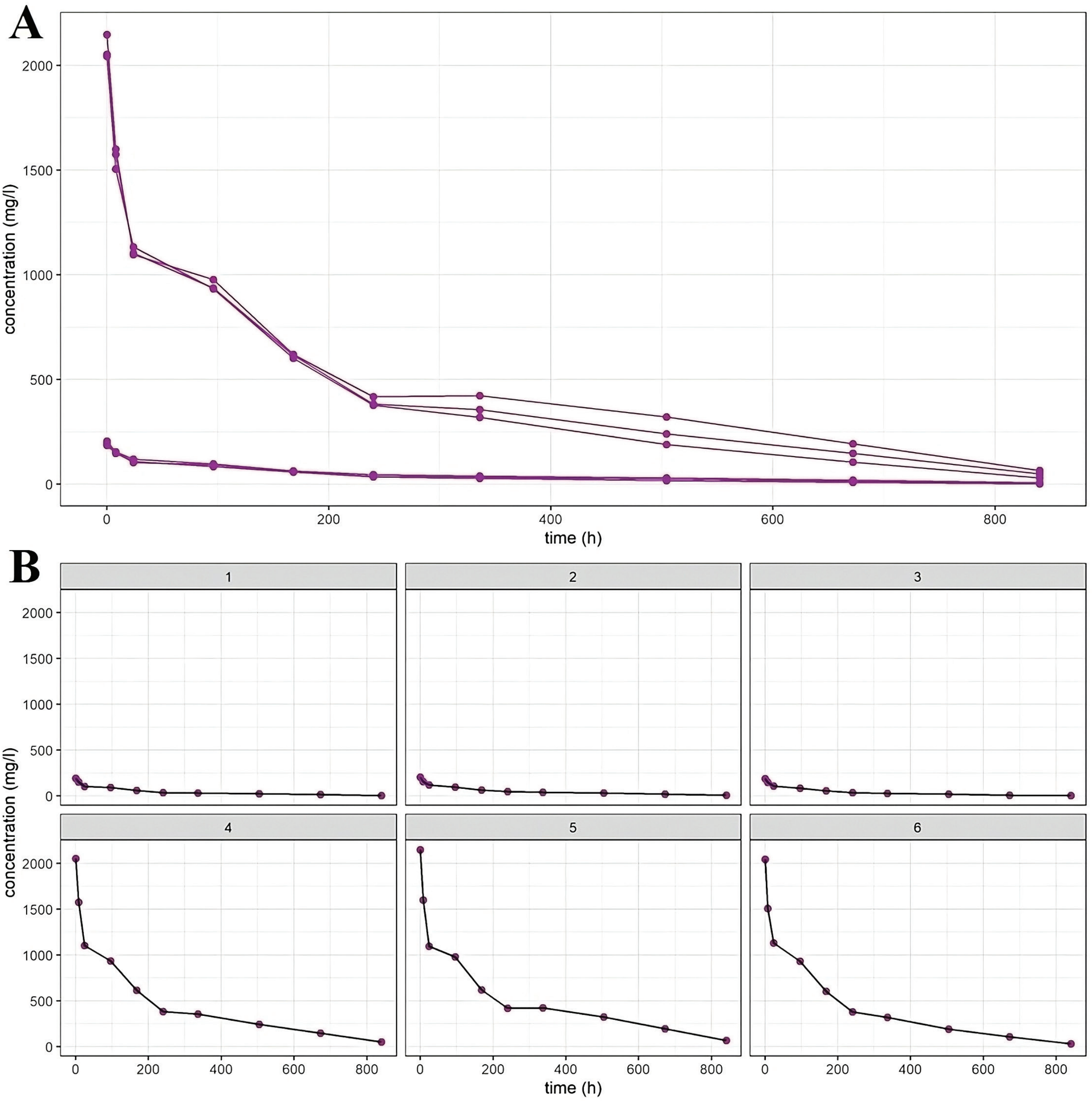


Figure 6 PhysPK screenshot from monitor to spaghetti plasma concentration-time graph generated for (A) population and, (B) individuals with R by dose according to intravenous administration.

plots of interest using the input data file of observed data (*.csv* format) in combination with PhysPK output from simulation, estimation, or optimization analysis.

After running the model with PhysPK, R scripts are used to generate graphics, including spaghetti plots for each dose level in the dataset for the entire population and individuals (**Figure 6**).

Graph modification must done in the scripts, and involves adjusting linear and semi-logarithmic scales, adding horizontal lines with numerical values and units, and indicating the number of points below the quantification limit. In addition, it is important to include additional relevant information, such as the minimum concentration value (*Cmin*) and the corresponding time (*Tmin*), the maximum concentration value (*Cmax*) and the corresponding time (*Tmax*), as well as a visual representation of the time and number of doses (**Figure 7**).

There is no limit to the number of points that can be simulated by PhysPK for use in R plots. Additional information, such as *Cmin*, *Tmin*, *Cmax*, and *Tmax*, can also be displayed.

With PhysPK simulated data, complex and comprehensive graphics and statistics can be quickly generated. The semi-logarithmic scale spaghetti plots reveal a biphasic distribution for PK profiles with FO absorption.

The quality of parameter estimation can be assessed by the standard error values of the fitted parameter estimates and the visual predictive checks, which are included in the fitting summary report (**Table 3**). The results of the pooled fitting are shown in **Figure 8**, and the parameter estimates for both pooled and group-specific fittings are listed in **Table 3** with relatively low standard errors. The estimated value of CL is 7.06 ± 0.32, which is consistent with the results in case study #1: two-compartment model fit for PK of a large molecule from "gPKPDSim: a SimBiology-based GUI application for PKPD modeling in drug development" [21], which was 6.89 mL/kg/day. The central and peripheral compartment volumes are estimated at 48.9 ± 3.66 and 33.2 ± 3.95 mL/Kg, respectively.

In addition, the model and parameter estimates are utilized together to project the PK for an alternative dosing regimen and investigate the potential PK variability. The parameter estimates obtained from the two-compartment model are saved for the purpose of projecting the PK profile of a 10 mg/kg (weekly dosing for 4 weeks) intravenous dosing regimen under various conditions (**Table 4**) to determine the impact of PK parameters on drug exposure. This determination was achieved using the simulation and population simulation functionalities.

Table 3 Fitting Combined Pooled Results [21] versus PhysPK

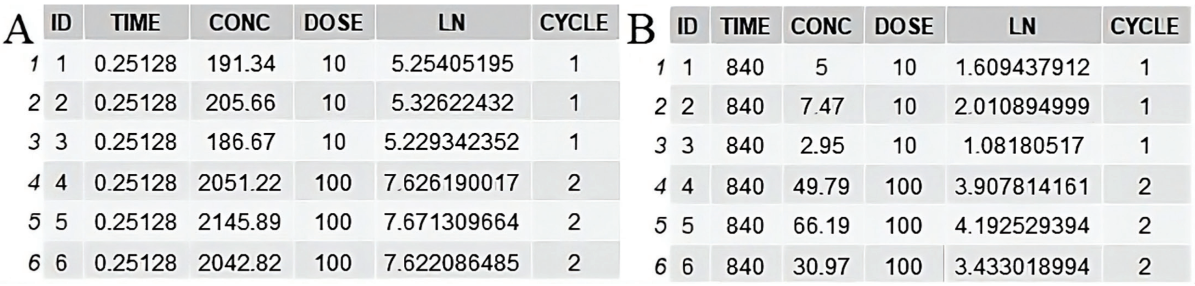
Name	Initial	Fit - Group	PhysPK	Units
V_1	40	49.15 ± 3.8	48.8 ± 3.66	ml/kg
V_2	40	34.61 ± 5.2	33.2 ± 3.95	ml/kg
C_l	5	6.89 ± 0.2	7.06 ± 0.32	ml/day/kg
C_{ld}	10	45.5 ± 17.4	49.04 ± 9.26	ml/day/kg

Figure 8 depicts the dependent variable plotted against time for various simulations. The function serves to present the PK and PD profile shapes. Additionally, the function enables the observation of sampling density, dosing history influence on PK and PD profiles, non-linearity, potential IIV, and the number of points below the limit of quantitation. R generates these profiles from the fitting simulation output *.txt* file obtained from PhysPK. The selection of the data file and format is automatically defined by the PhysPK fitting result. Nonetheless, users can create new R scripts for rendering with the PhysPK fitting results.

While the headers of the dataset must be adapted to the PhysPK files and R scripts provided, the flexibility of the PhysPK fitting results allows for the template to be adjusted to the dataset.

Goodness-of-fit

Goodness-of-fit (GOF) plots are an effective means of assessing model adequacy. R can generate a variety of GOF plots, including observations versus individual and population model predictions on both linear and logarithmic scales, standardized residual versus time, and standardized residual versus individual and population model predictions. Values are obtained from the PhysPK fitting results, but GOF plots and related information must be managed by R. For example, the title space of each plot could include the model name. GOF plots were used to compare the results from "gPKPDSim: a SimBiology-based GUI application for PKPD modeling in drug development" [21] (6.89 mL/kg/day) with the results from PhysPK. Visual inspection of



A

	ID	TIME	CONC	DOSE	LN	CYCLE
1	1	0.25128	191.34	10	5.25405195	1
2	2	0.25128	205.66	10	5.32622432	1
3	3	0.25128	186.67	10	5.229342352	1
4	4	0.25128	2051.22	100	7.626190017	2
5	5	0.25128	2145.89	100	7.671309664	2
6	6	0.25128	2042.82	100	7.622086485	2

B

	ID	TIME	CONC	DOSE	LN	CYCLE
1	1	840	5	10	1.609437912	1
2	2	840	7.47	10	2.010894999	1
3	3	840	2.95	10	1.08180517	1
4	4	840	49.79	100	3.907814161	2
5	5	840	66.19	100	4.192529394	2
6	6	840	30.97	100	3.433018994	2

Figure 7 PhysPK screenshot from monitor to maximum and minimum plasma concentration-time stats generated with R by dose according to intravenous administration. (A) *Cmax* concentrations; (B) *Cmin* concentrations.

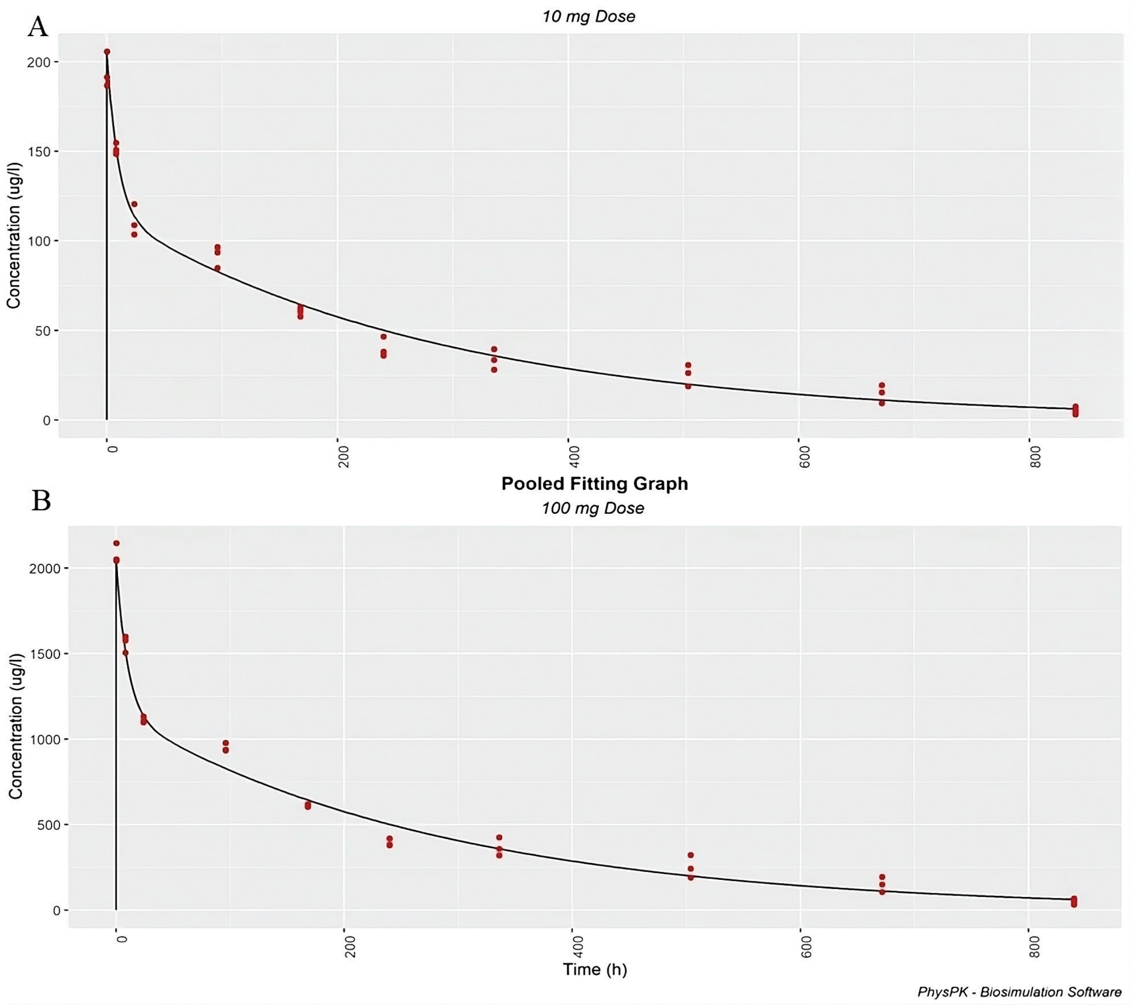


Figure 8 PhysPK screenshot from monitor to pooled fit graphs generated with R by dose according to intravenous administration, (A) 10 mg. and (B) 100 mg.

Table 4 Estimated AUC Values for the Period of 0-28 days of a 10 μg day/mL Intravenous Dose Weekly for 4 weeks under Different Conditions

	Scenario	Reference AUC (0-28)	PhysPK AUC (0-28)
Simulation	CL1 = 7.06 (PhysPK result of fitting)	4162.31	4159
Simulation	CL2 = 0.5 × Cl1 = 3.53	5759.83	5690
Population Simulation	CL3 = 7.06 + 10% CV	Median = 4137.45	Median = 4065
	V1 = 48.9 + 10% CV	5-95% = 3052-5097	5-95% = 3052-5097

the precision plot (observation vs. individual and population model prediction) indicates that the predictions obtained by PhysPK are acceptable compared to the reference (bottom panels in **Figure 9**).

Assessment predictions

To evaluate the accuracy of model-based predictions, PhysPK automatically calculates bias and precision metrics (AFE and AAFE respectively,). PhysPK considers predictions to be appropriate when the AFE and AAFE values fall between 0.8 and 1.4. The precision metrics results are saved in a *.csv* file, which is generated by R for each intravenous dose administered. **Figure 10** includes the population residual error (RES), weighted residual (WRES), percentage of prediction error (PE%), AFE, and AAFE.

Parameter distribution profiles

PhysPK estimates model parameters using a normal or log-normal distribution. The distribution of parameters is crucial because observing the distribution can reveal the presence of subpopulations. PhysPK generates individual Bayesian estimate distribution reports for each model parameter. These reports include a fitted density of estimate distribution and a histogram of estimate frequencies created using R, which are displayed in the PhysPK monitor (**Figure 11**). Visual inspection of the plots provides information on the

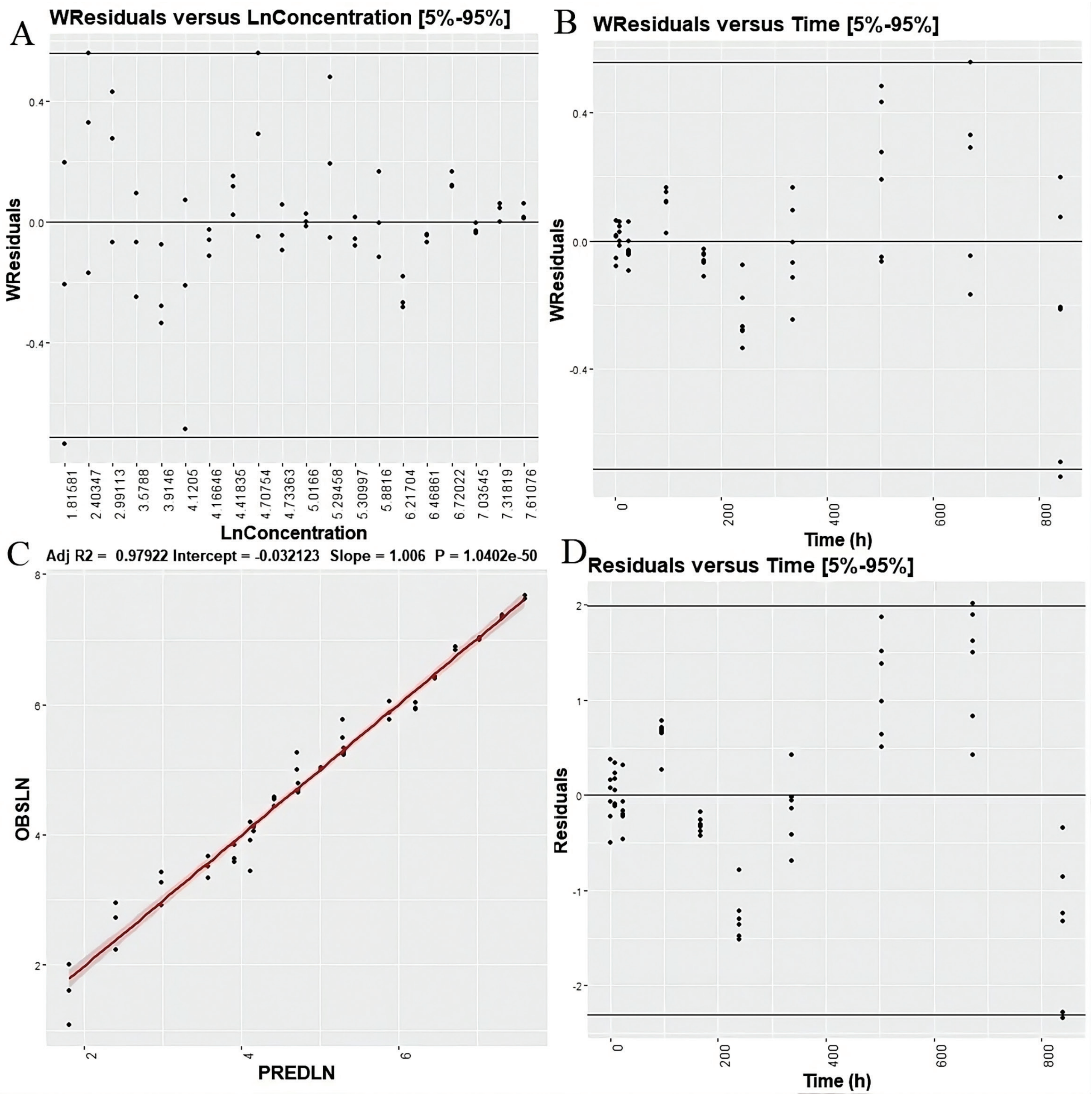


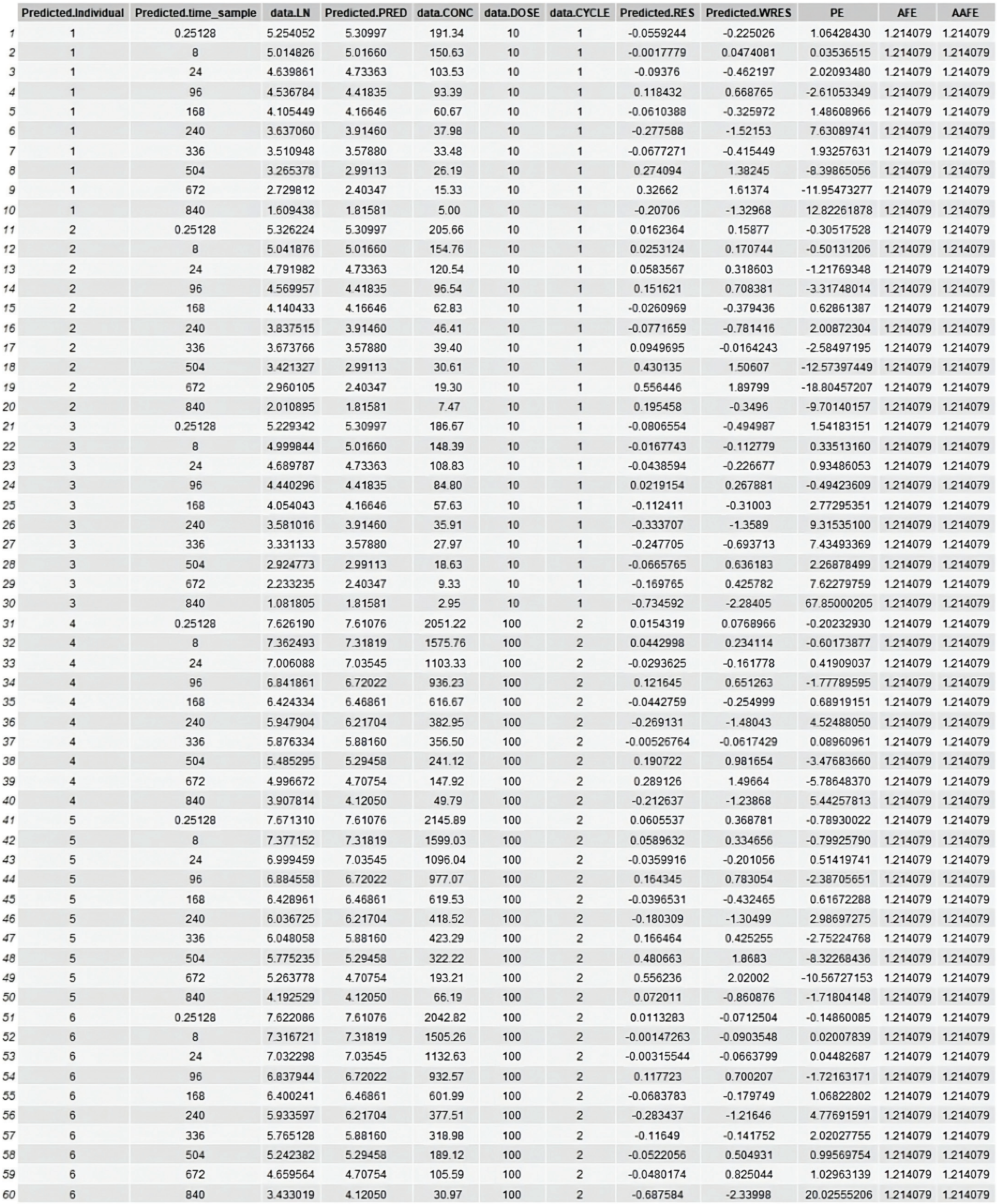
Figure 9 PhysPK screenshot from monitor to evaluate model adequacy generated with R by dose according to intravenous administration. (A) WResiduals versus LnConcentration, (B) WResiduals versus Time, (C) Goodness-of-fit plot and (D) Residuals versus Time.

variability around each parameter with the shape of the density curves resembling a normal distribution, which could have been a better choice for this model.

The log-normal distribution was used for all the parameters in this work to evaluate the distribution and density. The parameter distribution function of PhysPK was applied for this purpose, and the resulting plots were visually inspected both inside PhysPK and in R within PhysPK. These plots provide information on the variability around each parameter, and suggest that a log-normal distribution is an appropriate choice for this model.

Visual predictive check

To assess the effectiveness of a model's predicted performance, VPC charts can be employed. VPC plots utilize a Monte Carlo simulation that considers the model structure, final parameter estimates, and analysis dataset. Unlike the results of the model estimation run, VPC plots are designed to enable comparisons between the distributions of the observed data from the analysis dataset and the simulated data (usually a selected set of percentiles). PhysPK provides percentile calculations on the simulated



	Predicted.Individual	Predicted.time_sample	data.LN	Predicted.PRED	data.CONC	data.DOSE	data.CYCLE	Predicted.RES	Predicted.WRES	PE	AFE	AAFE
1	1	0.25128	5.254052	5.30997	191.34	10	1	-0.0559244	-0.225026	1.06428430	1.214079	1.214079
2	1	8	5.014826	5.01660	150.63	10	1	-0.0017779	0.0474081	0.03536515	1.214079	1.214079
3	1	24	4.639861	4.73363	103.53	10	1	-0.09376	-0.462197	2.02093480	1.214079	1.214079
4	1	96	4.536784	4.41835	93.39	10	1	0.118432	0.668765	-2.61053349	1.214079	1.214079
5	1	168	4.105449	4.16646	60.67	10	1	-0.0610388	-0.325972	1.48608966	1.214079	1.214079
6	1	240	3.637060	3.91460	37.98	10	1	-0.277588	-1.52153	7.63089741	1.214079	1.214079
7	1	336	3.510948	3.57880	33.48	10	1	-0.0677271	-0.415449	1.93257631	1.214079	1.214079
8	1	504	3.265378	2.99113	26.19	10	1	0.274094	1.38245	-8.39865056	1.214079	1.214079
9	1	672	2.729812	2.40347	15.33	10	1	0.32662	1.61374	-11.95473277	1.214079	1.214079
10	1	840	1.609438	1.81581	5.00	10	1	-0.20706	-1.32968	12.82261878	1.214079	1.214079
11	2	0.25128	5.326224	5.30997	205.66	10	1	0.0162364	0.15877	-0.30517528	1.214079	1.214079
12	2	8	5.041876	5.01660	154.76	10	1	0.0253124	0.170744	-0.50131206	1.214079	1.214079
13	2	24	4.791982	4.73363	120.54	10	1	0.0583567	0.318603	-1.21769348	1.214079	1.214079
14	2	96	4.569957	4.41835	96.54	10	1	0.151621	0.708381	-3.31748014	1.214079	1.214079
15	2	168	4.140433	4.16646	62.83	10	1	-0.0260969	-0.379436	0.62861387	1.214079	1.214079
16	2	240	3.837515	3.91460	46.41	10	1	-0.0771659	-0.781416	2.00872304	1.214079	1.214079
17	2	336	3.673766	3.57880	39.40	10	1	0.0949695	-0.0164243	-2.58497195	1.214079	1.214079
18	2	504	3.421327	2.99113	30.61	10	1	0.430135	1.50607	-12.57397449	1.214079	1.214079
19	2	672	2.960105	2.40347	19.30	10	1	0.556446	1.89799	-18.80457207	1.214079	1.214079
20	2	840	2.010895	1.81581	7.47	10	1	0.195458	-0.3496	-9.70140157	1.214079	1.214079
21	3	0.25128	5.229342	5.30997	186.67	10	1	-0.0806554	-0.494987	1.54183151	1.214079	1.214079
22	3	8	4.999844	5.01660	148.39	10	1	-0.0167743	-0.112779	0.33513160	1.214079	1.214079
23	3	24	4.689787	4.73363	108.83	10	1	-0.0438594	-0.226677	0.93486053	1.214079	1.214079
24	3	96	4.440296	4.41835	84.80	10	1	0.0219154	0.267881	-0.49423609	1.214079	1.214079
25	3	168	4.054043	4.16646	57.63	10	1	-0.112411	-0.31003	2.77295351	1.214079	1.214079
26	3	240	3.581016	3.91460	35.91	10	1	-0.333707	-1.3589	9.31535100	1.214079	1.214079
27	3	336	3.331133	3.57880	27.97	10	1	-0.247705	-0.693713	7.43493369	1.214079	1.214079
28	3	504	2.924773	2.99113	18.63	10	1	-0.0665765	0.636183	2.26878499	1.214079	1.214079
29	3	672	2.233235	2.40347	9.33	10	1	-0.169765	0.425782	7.62279759	1.214079	1.214079
30	3	840	1.081805	1.81581	2.95	10	1	-0.734592	-2.28405	67.85000205	1.214079	1.214079
31	4	0.25128	7.626190	7.61076	2051.22	100	2	0.0154319	0.0768966	-0.20232930	1.214079	1.214079
32	4	8	7.362493	7.31819	1575.76	100	2	0.0442998	0.234114	-0.60173877	1.214079	1.214079
33	4	24	7.006088	7.03545	1103.33	100	2	-0.0293625	-0.161778	0.41909037	1.214079	1.214079
34	4	96	6.841861	6.72022	936.23	100	2	0.121645	0.651263	-1.77789595	1.214079	1.214079
35	4	168	6.424334	6.46861	616.67	100	2	-0.0442759	-0.254999	0.68919151	1.214079	1.214079
36	4	240	5.947904	6.21704	382.95	100	2	-0.269131	-1.48043	4.52488050	1.214079	1.214079
37	4	336	5.876334	5.88160	356.50	100	2	-0.00526764	-0.0617429	0.08960961	1.214079	1.214079
38	4	504	5.485295	5.29458	241.12	100	2	0.190722	0.981654	-3.47683660	1.214079	1.214079
39	4	672	4.996672	4.70754	147.92	100	2	0.289126	1.49664	-5.78648370	1.214079	1.214079
40	4	840	3.907814	4.12050	49.79	100	2	-0.212637	-1.23868	5.44257813	1.214079	1.214079
41	5	0.25128	7.671310	7.61076	2145.89	100	2	0.0605537	0.368781	-0.78930022	1.214079	1.214079
42	5	8	7.377152	7.31819	1599.03	100	2	0.0589632	0.334656	-0.79925790	1.214079	1.214079
43	5	24	6.999459	7.03545	1096.04	100	2	-0.0359916	-0.201056	0.51419741	1.214079	1.214079
44	5	96	6.884558	6.72022	977.07	100	2	0.164345	0.783054	-2.38705651	1.214079	1.214079
45	5	168	6.428961	6.46861	619.53	100	2	-0.0396531	-0.432465	0.61672288	1.214079	1.214079
46	5	240	6.036725	6.21704	418.52	100	2	-0.180309	-1.30499	2.98697275	1.214079	1.214079
47	5	336	6.048058	5.88160	423.29	100	2	0.166464	0.425255	-2.75224768	1.214079	1.214079
48	5	504	5.775235	5.29458	322.22	100	2	0.480663	1.8683	-8.32268436	1.214079	1.214079
49	5	672	5.263778	4.70754	193.21	100	2	0.556236	2.02002	-10.56727153	1.214079	1.214079
50	5	840	4.192529	4.12050	66.19	100	2	0.072011	-0.860876	-1.71804148	1.214079	1.214079
51	6	0.25128	7.622086	7.61076	2042.82	100	2	0.0113283	-0.0712504	-0.14860085	1.214079	1.214079
52	6	8	7.316721	7.31819	1505.26	100	2	-0.00147263	-0.0903548	0.02007839	1.214079	1.214079
53	6	24	7.032298	7.03545	1132.63	100	2	-0.00315544	-0.0663799	0.04482687	1.214079	1.214079
54	6	96	6.837944	6.72022	932.57	100	2	0.117723	0.700207	-1.72163171	1.214079	1.214079
55	6	168	6.400241	6.46861	601.99	100	2	-0.0683783	-0.179749	1.06822802	1.214079	1.214079
56	6	240	5.933597	6.21704	377.51	100	2	-0.283437	-1.21646	4.77691591	1.214079	1.214079
57	6	336	5.765128	5.88160	318.98	100	2	-0.11649	-0.141752	2.02027755	1.214079	1.214079
58	6	504	5.242382	5.29458	189.12	100	2	-0.0522056	0.504931	0.99569754	1.214079	1.214079
59	6	672	4.659564	4.70754	105.59	100	2	-0.0480174	0.825044	1.02963139	1.214079	1.214079
60	6	840	3.433019	4.12050	30.97	100	2	-0.687584	-2.33998	20.02555206	1.214079	1.214079

Figure 10 PhysPK screenshot from monitor to pooled Residual Error, AFE, and AFEE. Table generated with R by dose according to intravenous administration.

data, as well as the median for the observations and other statistical outputs. While the process to perform these simulations, determine the percentiles, and produce visualizations can be time-consuming, PhysPK facilitates this process with an R connection (**Figure 12**). VPC plots can be generated in both linear and semi-logarithmic scales for each of the different outputs available in the data using R scripts.

A VPC plot was generated using a 1000-subject Monte Carlo simulation based on the model structure and final

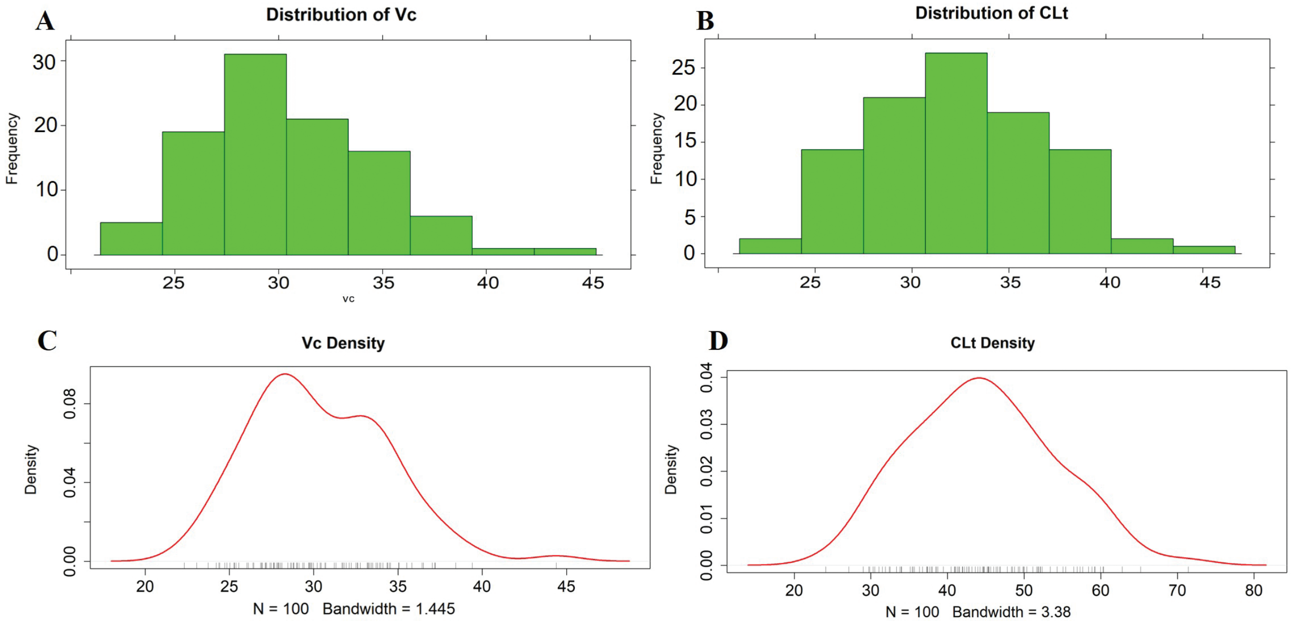


Figure 11 PhysPK screenshot from monitor to show distribution and density from Montecarlo simulation parameters (A) Distribution of VC, (B) Distribution of CLt, (C) Density of Vc and (D) Density of CL.

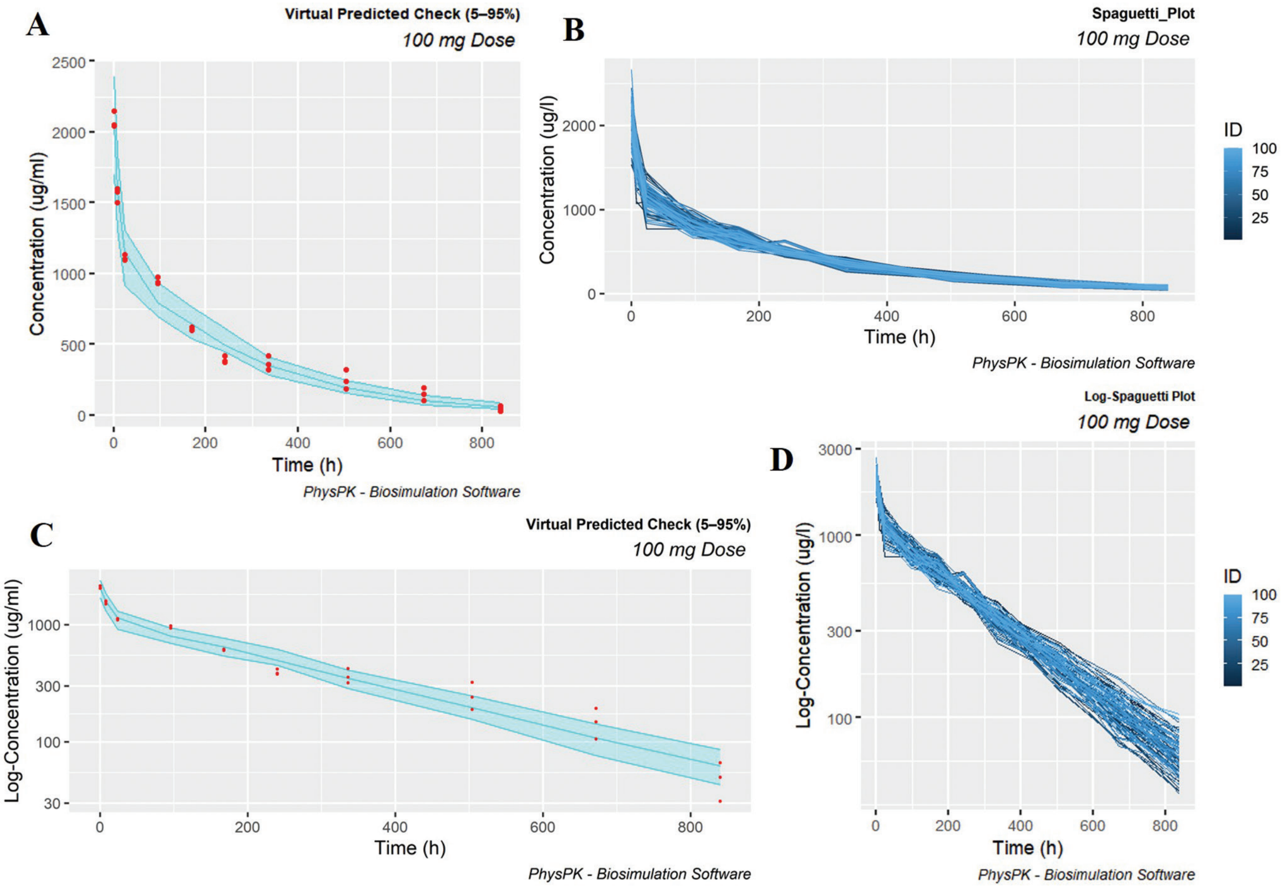


Figure 12 PhysPK screenshot from monitor to Virtual Predicted Check (VPC) generated with R by dose according to intravenous administration. A population dataset of 1000 subjects was generated using Monte Carlo simulation method for 10% CV in CL and V1 in linear (A-B) and semi-logarithmic (C-D) scales.

MLEM estimates. The VPC function of PhysPK produces a *.txt* file that contains the simulated data and the data from the MLEM estimation step, which includes the observations. The lower percentile was set to 5% and the upper percentile to 95%, creating a 90% prediction interval envelope. The VPC plot provides a description of the central tendency and variability of the simulated data, which is highly representative of the observations.

Discussion

PhysPK is a robust bio-modeling software, especially valuable for conducting PK, PD, physiologically-based pharmacokinetic (PBPK), and quantitative systems pharmacology (QSP) simulations, and related applications applicable in both academic and industrial settings. PhysPK software is a multi-libraries M&S tool for optimization, estimation, and validation of PK/PD/PBPK parameters for individuals and populations. PhysPK software is based on first principles modelling of complex systems with continuous and discrete equations using the MOOM paradigm. PhysPK language is designed to model systems formulated through DAE and discrete events using a non-algorithmic code (a causal simulation language) and ultimately converted to an algorithmic code in C++ language. This modeling language supports strong model reusability, as well as connectivity with external hardware using object linking and embedding for process control (OPC), also allowing out-of-the-box execution [17]. Although the PhysPK custom and default graphs, plots, and statistics are generally effective for characterizing PK and projecting PK for other dosing scenarios, PhysPK may have limited capabilities in generating advanced statistical graphs and producing post-script file reports, especially when compared to programming language software, such as R. Nevertheless, to enhance PhysPK's output, high-quality R scripts can be integrated into the PhysPK monitor to execute R functions for statistics, graphs, and reports. As an alternative approach, users can utilize the output files generated by PhysPK user-friendly models after each simulation run and independently create desired plots using R scripts. This integration enables faster and more cost-effective assessments of the efficacy, automation, and safety of new drugs.

Post-processing analysis has proven highly beneficial for other modeling programs because post-processing analysis enables modelers to concentrate on evaluating simulation results rather than creating graphics. Additionally, the creation of diagnostic plots for each model rely heavily on the skill level of the user. Furthermore, optimizing population PK/PD models can be a time-consuming process that is influenced by factors, such as the number of observations and parameters, processing power, and the software algorithm used.

However, embedding R scripts in PhysPK saves time by automating repetitive tasks and reducing the likelihood of errors that can arise from manual data manipulation. The ability of the software to generate complex and detailed results automatically after simulation or optimization methods provides additional benefits to users. This seamless integration enhances the overall efficiency and reliability of the modeling process, making it a valuable tool for researchers and practitioners in the PK/PD field.

Compared to other bio-simulation software, PhysPK takes a different approach by embedding R scripts directly within the software, thus offering a more streamlined and efficient process. While other software may use R packages, such as nlmixr, saemix, RxODE, mrgsolve, mapbayr, nonmem2R, or Rsmlx, which are dedicated to population PK/PD modeling, the other software often requires two separate steps: generating the PK model; and running R scripts for simulations or estimations. This approach can be tedious to develop, manage, or understanding PK models because the approach involves switching between different platforms and dealing with PK equations in R code [34].

In contrast, PhysPK integrates R scripts into its interface, automating repetitive tasks and reducing the risk of errors associated with manual data manipulation. The ability of the software to generate complex and detailed results automatically after simulation or optimization methods provides added convenience to users. In addition, PhysPK Excel or a Web application demonstrates how stochastic simulation in PhysPK can be used to explore a series of dosing regimens. This seamless integration enhances the overall efficiency and reliability of the modeling process, making PhysPK a valuable tool for researchers and practitioners in the PK/PD field.

The main advantage of using PhysPK with R is the ability to automatically update when input changes. As models are performed in PhysPK, the R computational time will not be utilized for stochastic simulations, which can take minutes or more to update the plot, thus negating the reactive benefits of Shiny. The results are saved from PhysPK simulations, reducing the results to machine instructions at run time. In contrast, compiled languages save the code directly to an executable file of machine instructions, often resulting in speed benefits.

Indeed, embedding R within PhysPK offers several advantages that set R apart from other software options in the market. By integrating R scripts directly into PhysPK, users can benefit from a user-friendly interface, population model, estimation, and validation modules. This seamless integration allows for a more streamlined workflow, making it easier to manage and validate custom and reusable PK/PD models.

Incorporated within the PhysPK software as part of the open source nature, there are various software options and applications. It is important to note that this innovative feature extends beyond the realm of R software. In fact, the integration of other open source software, such as Python, Matlab, and Julia, into the PhysPK monitor is showcased similar to what has been demonstrated with R.

With this integration, users can harness the strengths of PhysPK and R, taking advantage of the user-friendly aspects of PhysPK while leveraging the advanced capabilities of R for statistical analysis and visualization in PK/PD processes, including dependent variable versus time plots, GOF plots, *post hoc* fits, a posterior mean estimate distribution of model parameters, and VPC plots. These graphics are specifically designed to enhance end-user productivity by automatically generating diagnostic plots. This synergy empowers users to conduct sophisticated PK and PD studies in a more efficient, accurate, and reproducible manner.

The R connection with PhysPK leverages the information contained in the output files within R to summarize and display both graphical and pertinent numerical model evaluation criteria on the same interface. Default scripts generated by the new version of PhysPK allow users without R experience to easily navigate the analysis. Additionally, advanced users can create their own scripts by calling the functions distributed in the PhysPK monitor.

Through a program-related settings file, users have the freedom to customize graphics according to their specific needs. This flexibility enables graphs to be utilized for routine model evaluation, as well as for publications or reports. Graph files also include embedded information, such as the analysis project's date, time, and other relevant details, providing better organization for users.

PhysPK is available under license, and regular updates are released to continuously improve the software. These updates involve adding new features, enhancing existing functions, and fixing bugs. Future developments of PhysPK include the implementation of faster, easier, and more automatic non-compartmental analysis. Additionally, a side project is underway to integrate PhysPK within graphical user interfaces, further enhancing the user experience by simplifying model management. Regular updates and information about PhysPK can be found on the software website and LinkedIn profile.

Conclusions

M&S software offers numerous benefits for modeling and simulating complex stochastic systems and processes. Over the years, PK processes have extensively relied on such software; however, the capabilities can be significantly expanded by enabling bio-simulation models to interact with external statistical, graphics, optimization algorithms, and machine learning methods. For these reasons, in this study we explored the integration of R statistical packages with PhysPK further streamlining population PK/PD analysis, and simplifying and expediting the process. R statistical packages were utilized to produce, organize, interpret, and present selected and comprehensive population PK/PD results. Obtaining these results in a single step simplifies population analysis, making it faster and more efficient. By embedding R scripts within PhysPK results/monitor, the software gains the flexibility to be used for routine model evaluation, as well as for publications or reports. The experimental results were compared with other population PK/PD analyses, confirming the utility of this new PhysPK feature in PK analysis. This new feature allows for flexible model evaluation, supports routine use, and facilitates publication and reporting of results.

In contrast to other software, PhysPK integrates R scripts into its interface, automating repetitive tasks, reducing time, and reducing the risk of errors associated with manual data manipulation. In addition, by integrating bio-simulation software with programming languages, such as R, new opportunities arise for developing advanced intelligent simulation modeling. The simulation output of PhysPK serves as a platform that generates clean and structured data, which can be effectively utilized by various algorithmic approaches to facilitate decision-making processes. This seamless integration enhances the overall efficiency and reliability of the modeling process, making PhysPK a valuable tool for researchers and practitioners in the field of PK/PD.

This connection allows for future integration of machine learning methods with PhysPK. This innovative feature will offer the potential to enhance the overall performance of sampling-based non-linear model predictive controllers in PK. By combining first principles models with machine learning algorithms, it becomes possible to capture general trends in state variables for dynamics. Additionally, the inclusion of deep neural networks allows for compensation of additional errors in the predicted states. This hybrid modeling approach enables the use of black box machine learning procedures, which can be supported and trusted more reliably in conjunction with first principles methods, such as mechanistic, semi-empirical, phenomenologic, or white box models. The utilization of machine learning methods in PK can significantly decrease mathematical complexity and reduce the time and resources required for mathematical modeling of the relationships between drug exposure, safety, and efficacy. Machine learning predictions of drug PK can serve as input for PK/PD models, enabling more time-efficient analysis or training of machine learning algorithms using supervised or unsupervised learning techniques. This integration of machine learning streamlines the process and enhances the efficiency of PK analyses and drug development.

Acknowledgments

The authors would like to thank Empresarios Agrupados Internacional S.A. for supporting this article.

Author Contributions

Sergio Sánchez-Herrero and **Fernando Carbonero Martínez** developed the theoretical formalism and the methodology to perform the analytic calculations and performed the numerical analysis. The investigation and the case study population analysis and a proof of concept for the embedding feature was developed to prove this innovation by **Sergio Sánchez-Herrero** and **Fernando Carbonero Martínez** and verified and supported by **Jenifer Serna**, **Marina Cuquerella-Gilabert**, **Almudena Rueda-Ferreiro**, **Angel A. Juan**, and **Laura Calvet**. Findings of this work simulations was supervised by **Almudena Rueda-Ferreiro** and **Angel A. Juan**. Writing the original draft preparation was performed by **Sergio Sánchez-Herrero** and **Fernando Carbonero Martínez** and reviewing and editing were done by **Almudena Rueda-Ferreiro**, **Angel A. Juan**, and **Laura Calvet**. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Rajman I. PK/PD modelling and simulations: utility in drug development. Drug Discov Today 2008;13(7-8):341-46. [PMID: 18405847 DOI: https://doi.org/10.1016/j.drudis.2008.01.003]

[2] Guiastrennec B, Wollenberg L, Forrest A, Ait-Oudhia S. AMGET, an R-based postprocessing tool for ADAPT 5. CPT Pharmacometrics Systs Pharmacol 2013;2(7):1-10. [PMID: 23903464 DOI: https://doi.org/10.1038/psp.2013.36]

[3] Sun D, Gao W, Hu H, Zhou S. Why 90% of clinical drug development fails and how to improve it? Acta Pharmaceutica Sinica B 2022;12:3049-62. [PMID: 35865092 DOI: https://doi.org/10.1016/j.apsb.2022.02.002]

[4] Han JJ. FDA Modernization Act 2.0 allows for alternatives to animal testing. Artif Organs 2023;47:449-50. [PMID: 36762462 DOI: https://doi.org/10.1111/aor.14503]

[5] Bauer RJ, Guzy S, Chee Ng. A survey of population analysis methods and software for complex pharmacokinetic and pharmacodynamic models with examples. AAPS J 2007;9:E60-83. [PMID: 17408237 DOI: https://doi.org/10.1208/aapsj0901007]

[6] Dartois C, Brendel K, Comets E, Laffont CM, Laveille C, et al. Overview of model-building strategies in population PK/PD analyses: 2002–2004 lit- erature survey. Br J Clin Pharmacol 2007;64(5):603-12. [PMID: 17711538 DOI: https://doi.org/10.1111/j.1365-2125.2007.02975.x]

[7] Kouatchou J, Medema A. Basic comparison of Python, Julia, Matlab, IDL and Java. USA: Modeling Guru–National Aeronautics and Space Administration; 2018.

[8] R Core Team. R: A Language and Environment for Statistical Computing. Vienna, Austria: R Foundation for Statistical Computing; 2022. https://www.R-project.org/.

[9] Tippmann S. Programming tools: adventures with R. Nature 2015;517(7532):109-10. [PMID: 25557714 DOI: https://doi.org/10.1038/517109a]

[10] Muenchen RA. The popularity of data analysis software. Data Science Software Reviews; 2012. http://r4stats.com/popularity.

[11] Zhao Y. R and data mining: examples and case studies. Academic Press; 2012.

[12] Giorgi FM, Ceraolo C, Mercatelli D. The R language: an engine for bioinformatics and data science. Life 2022;12(5):648. [PMID: 35629316 DOI: https://doi.org/10.3390/life12050648]

[13] Raghav RS, Amudhavel J, Dhavachelvan P. A survey on tools used in big data platform. Adv Appl Math Sci 2017;17(1):213-29.

[14] Acharya C, Hooker AC, Türkyılmaz GY, Jönsson S, Karlsson MO. A diagnostic tool for population models using non-compartmental analysis: The *nca p pc* package for R. Comput Methods Programs Biomed 2016;127:83-93. [PMID: 27000291 DOI: https://doi.org/10.1016/j.cmpb.2016.01.013]

[15] Reig-Lopez J, Merino-Sanjuan M, Mangas-Sanjuan V, Prado-Velasco M. A multilevel object-oriented modelling methodology for physiologically-based pharmacokinetics (PBPK): evaluation with a semi-mechanistic pharmacokinetic model. Comput Methods Programs Biomed 2020;189:105322. [PMID: 31954235 DOI: https://doi.org/10.1016/j.cmpb.2020.105322]

[16] Prado-Velasco M, Borobia A, Carcas-Sansuan A. Predictive engines based on pharmacokinetics modelling for tacrolimus personalized dosage in paediatric renal transplant patients. Sci Rep 2020;10(1):7542. [PMID: 32371893 DOI: https://doi.org/10.1038/s41598-020-64189-9]

[17] Zazo H, Lagarejos E, Prado-Velasco M, Sánchez-Herrero S, Serna J, et al. Physiologically-based pharmacokinetic modelling and dosing evaluation of gentamicin in neonates using PhysPK. Front Pharmacol 2022;13:977372. [PMID: 36249803 DOI: https://doi.org/10.3389/fphar.2022.977372]

[18] Kiang TK, Sherwin CM, Spigarelli MG, Ensom MH. Fundamentals of population pharmacokinetic modelling: modelling and software. Clin Pharmacokinet 2012;51:515-25. [PMID: 22676301 DOI: https://doi.org/10.2165/11634080-000000000-00000]

[19] Sherwin CM, Kiang TK, Spigarelli MG, Ensom MH. Fundamentals of population pharmacokinetic modelling: validation methods. Clin Pharmacokinet 2012;51:573-90. [PMID: 22799590 DOI: https://doi.org/10.1007/BF03261932]

[20] Blanchard S. Cran task view: analysis of pharmacokinetic data. 2014. Available from: https://cran.r-project.org/web/views/Pharmacokinetics.html.

[21] Hosseini I, Gajjala A, Bumbaca Yadav D, Sukumaran S, Ra- manujan S, et al. gPKPDSim: a SimBiology®-based GUI application for PKPD modeling in drug development. J Pharmacokinet Pharmacodyn 2018;45(2):259-75. [PMID: 29302838 DOI: https://doi.org/10.1007/s10928-017-9562-9]

[22] Verbeke G, Fieuws S, Molenberghs G, Davidian M. The analysis of multivariate longitudinal data: a review. Stat Methods Med Res 2014;23(1):42-59. [PMID: 22523185 DOI: https://doi.org/10.1177/0962280212445834]

[23] Davidian M. Introduction to statistical population modeling and analysis for pharma- cokinetic data. Invited white paper for the International Workshop on Uncertainty and Variability in Physiologically Based Pharmacokinetic (PBPK) Models; 2006, Volume 89.

[24] Martí R, Rodríguez T, Pitarch JL, Sarabia D, de Prada C. Dynamic optimization by automatic differentiation using EcosimPro and CASADI. XXXV Jornadas de Automática 2014:354-61.

[25] Roa L, Prado M. Simulation languages. Wiley encyclopedia of biomedical engineering; 2006.

[26] Wickham H. ggplot2. Wiley interdisciplinary reviews: computational statistics 2011;3(2):180-5. [DOI: https://doi.org/10.1002/wics.147]

[27] Wickham H, Bryan J. readxl: Read Excel Files. 2023. Available from: https://readxl.tidyverse.org, https://github.com/tidyverse/readxl.

[28] Wickham H. R packages: organize, test, document, and share your code. O'Reilly Media, Inc.; 2015.

[29] Empresarios Agrupados. Ecosimpro, user manual; 2022.

[30] Puttrevu SK, Arora S, Polak S, Patel NK. Physi- ologically based pharmacokinetic modeling of transdermal selegiline and its metabolites for the evaluation of disposition differences between healthy and special populations. Pharmaceutics 2020;12(10):942. [PMID: 33008144 DOI: https://doi.org/10.3390/pharmaceutics12100942]

[31] Corral Alaejos A, Zarzuelo Castaneda A, Jiménez Cabrera S, Sánchez-Guijo F, Otero MH, et al. External evaluation of population pharmacokinetic models of imatinib in adults diagnosed with chronic myeloid leukaemia. Br J Clin Pharmacol 1924;88(4):1913-24. [PMID: 34705297 DOI: https://doi.org/10.1111/bcp.15122]

[32] Rubinstein RY, Kroese DP. Simulation and the Monte Carlo method. John Wiley & Sons; 2016.

[33] Kamath AV. Translational pharmacokinetics and pharmacodynamics of monoclonal antibodies. Drug Discov Today 2016;21:75-83. [PMID: 27978991 DOI: https://doi.org/10.1016/j.ddtec.2016.09.004]

[34] Wojciechowski J, Hopkins AM, Upton RN. Interactive pharmacometric applications using R and the shiny package. CPT Pharmacometrics Systa Pharmacol 2015;4(3):146-59. [PMID: 26225240 DOI: https://doi.org/10.1002/psp4.21]