

**TUTORIAL**

# gPKPDviz: A flexible R shiny tool for pharmacokinetic/pharmacodynamic simulations using mrgsolve

Tong Lu<sup>1</sup>  | Victor Poon<sup>1</sup>  | Logan Brooks<sup>1</sup>  | Erick Velasquez<sup>1</sup>  |  
Eric Anderson<sup>2</sup>  | Kyle Baron<sup>2</sup>  | Jin Y. Jin<sup>1</sup>  | Matts Kågedal<sup>1</sup> 

<sup>1</sup>Department of Clinical Pharmacology, Genentech, Inc., South San Francisco, California, USA

<sup>2</sup>Metrum Research Group, Tariffville, Connecticut, USA

**Correspondence**

Tong Lu, Department of Clinical Pharmacology, Genentech, Inc, 1 DNA Way, South San Francisco, CA 94080, USA.

Email: [lu.tong@gene.com](mailto:lu.tong@gene.com)

**Abstract**

GPKPDviz is a Shiny application (app) dedicated to real-time simulation, visualization, and assessment of the pharmacokinetic/pharmacodynamic (PK/PD) models. Within the app, gPKPDviz is capable of generating virtual populations and complex dosing and sampling scenarios, which, together with the streamlined workflow, is designed to efficiently assess the impact of covariates and dosing regimens on PK/PD end points. The actual population data from clinical trials can be loaded into the app for simulation if desired. The app-generated dosing regimens include single or multiple dosing, and more complex regimens, such as loading doses or intermittent dosing. When necessary, the dosing regimens can be defined externally and loaded to the app for simulation. Using mrgsolve as the simulation engine, gPKPDviz is typically used for population simulation, however, with a slight modification of the mrgsolve model, gPKPDviz is capable of performing individual simulations with individual post hoc parameters, individual dosing logs, and individual sampling timepoints through an external dataset. A built-in text editor has a debugging feature for the mrgsolve model, providing the same error messages as model compilation in R. GPKPDviz has had stringent validation by comparing simulation results between the app and using mrgsolve in R. GPKPDviz is a member of the suite of Modeling and Simulation Shiny apps developed at Genentech to facilitate the typical modeling work in Clinical Pharmacology. For broader access to the Pharmacometric community, gPKPDviz has been published as an open-source application in GitHub under the terms of GNU General Public License.

**INTRODUCTION**

Simulations based on available pharmacokinetic/pharmacodynamic (PK/PD) models are critical to explore PK/PD relationships, support clinical trial designs, and

facilitate dose and regimen selections. Graphical user interfaces (GUIs) that facilitate these simulations can greatly enhance the communication and influence of modeling and simulation (M&S) in multi-functional team settings. Such GUIs also allow M&S tools to be

This is an open access article under the terms of the [Creative Commons Attribution-NonCommercial-NoDerivs](https://creativecommons.org/licenses/by-nc-nd/4.0/) License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2023 Genentech, Inc. *CPT: Pharmacometrics & Systems Pharmacology* published by Wiley Periodicals LLC on behalf of American Society for Clinical Pharmacology and Therapeutics.

made accessible to a wider audience with limited programming skills.

There are few MATLAB-based open-source GUIs that are capable of PK/PD simulations, including gPKPDsim<sup>1</sup> (for any model implemented in SimBiology), A4S<sup>2</sup> (for the predefined model library), MatVPC<sup>3</sup> (for any model implemented in MATLAB; focus on visual predictive checks), and ATLAS mPBPK<sup>4</sup> (for minimal physiologically-based PD model). GPKPDsim and ATLAS mPBPK also have data fitting capabilities. Simulx-GUI<sup>5</sup> was recently released for the advanced clinical trial simulations as part of the Monolix Suite. Berkeley Madonna<sup>6,7</sup> software is capable of real-time simulation and visualization of PK/PD models, however, the implementation of interindividual variability and complex dosing scenarios are not straightforward. In addition, both tools require a commercial license.

Shiny is an open-source R package that builds interactive user interfaces straight from R.<sup>8</sup> Shiny applications (will be referred to as “Shiny apps” hereinafter) can be deployed over the web for local hosting or a hosting service with broader access. Shiny apps are being increasingly used by pharmacometricians and clinical pharmacologists<sup>9–11</sup> due to their flexible interfaces and the rich ecosystem of R packages. A suite of M&S Shiny apps, called R-Shiny Exploratory Analysis Platform in Clinical Pharmacology<sup>12</sup> has been developed by Genentech for internal use to facilitate the typical modeling work performed in clinical pharmacology, including the PK/PD simulation application in this manuscript.

To perform simulations in R for the ordinary differential equation (ODE)-based PK/PD models, the open-source R package with an ODE solver must be used. Packages developed in the PK/PD field include mrgsolve,<sup>13,14</sup> PKPDsim,<sup>15,16</sup> and RxODE/rxode2.<sup>17–19</sup> The deSolve<sup>20</sup> and diffeqr<sup>21</sup> packages have more broader applications, covering ODE, partial differential equation, and delayed differential equation, etc. In this regard, mrgsolve was intentionally designed to be similar to NONMEM syntax, in addition, it has an efficient simulation engine and is easy to integrate with Shiny.

Combining Shiny and mrgsolve, we developed gPKPDviz (Genentech PK/PD visualizer), which allows real-time simulation and visualization of PK/PD models. It comes packaged with a library of mrgsolve models for the Roche/Genentech molecules on the internal server, and some template models and the public available models in the open-source version (currently limited, but to be expanded). GPKPDviz has gone through stringent validation by comparing outputs from the app to simulation results performed in R (Supplementary *Rmd S1*). For broader access to the pharmacometrics

community, gPKPDviz was published as an open-source application in GitHub under the terms of GNU General Public License version 3.<sup>22</sup>

ModVizPop<sup>9</sup> is another open-source Shiny app for PK/PD simulations using mrgsolve as the simulation engine. Different from some model-specific Shiny apps,<sup>23–25</sup> ModVizPop and gPKPDviz are both considered as the Shiny “Platform,” in the way they were created to work with any ODE-based models in mrgsolve format. ModVizPop is unable to simulate virtual populations within the app, and it relies on the external dataset to read in covariate information. ModVizPop also needs to load external dataset for complex dosing and sampling schedules, as the in-app simulation capability is limited. In comparison, gPKPDviz is capable of simulating virtual populations and complex dosing and sampling schedules within the app, and, when needed, it can also load external datasets to take on whatever complexities for covariate, dosing, and sampling schedules. Given the flexible settings and the streamlined workflow, gPKPDviz was designed to effectively evaluate the impact of covariates and dosing regimens on PK (exposure) or PD (response) end points. GPKPDviz can efficiently address clinical pharmacology questions such as, “what is the impact of a dose delay or alternative dosing regimen, knowing the target PK threshold?” or “from the PK point of view, is there a benefit of fixed dosing over body size-based dosing for heavier patients?”

Recently, e-Campsis<sup>26,27</sup> was released by calvagine as a Shiny “Platform” for population PK/PD simulations using R packages campsis<sup>28</sup> and campsismod,<sup>29</sup> which provides a powerful front end to run simulations with mrgsolve or RxODE/rxode2. It comes with the free or pro version, and both required the comprehension of campsis model syntax. The free version has limited functionality. The covariate distribution can be flexible, as long as it is defined in R in the covariate field. E-Campsis does not allow loading external datasets for simulation, thus it has limited flexibility to handle complex dosing scenarios or individual simulations. Once the simulation is fully configured, e-Campsis generates the complete R code for the users to reproduce or refine the simulation offline; such functionality is not available for gPKPDviz or ModVizPop.

This tutorial for gPKPDviz starts with the description the app workflow and the key features of each tab, followed by two comprehensive clinical applications, including dose delay or alternative dosing regimen assessment for pertuzumab, and body size-based dosing regimen assessment for polatuzumab vedotin. The animated demonstrations are provided in the GitHub repository using pertuzumab simulations as examples.<sup>22</sup>

## APP DESCRIPTION

### App workflow

GPKPDviz is comprised of five tabs: (1) Model, (2) Population, (3) Dosing, (4) Simulate, and (5) Results. A typical workflow of use of the app is shown in [Figure 1](#). However, users can repeatedly update certain tabs to simulate multiple scenarios for comparison purposes, such as updating the Dosing and Simulate tabs for simulation of different dosing regimens or updating the Model and Simulate tabs for simulation of different parameter values. It is also feasible to run the Population and Simulate tabs iteratively to assess the impact of population distribution on simulations.

### Mrgsolve model

Mrgsolve is an R package actively maintained by Metrum Research Group.<sup>13,14</sup> The mrgsolve model is the backbone for all the simulations in gPKPDviz. It has similar annotations and code blocks as NONMEM, but is more flexible and efficient for simulation. The model specification file consists of R and C++ code that is parsed, compiled, and dynamically loaded into the R session. The commonly used code blocks for mrgsolve include \$PROB (notes about the model), \$PREAMBLE (initialize the C++ variables), \$CMT (declare model compartments), \$PARAM (declare model parameters), \$OMEGA, \$SIGMA, \$MAIN (alias \$PK), \$ODE (alias \$DES), \$TABLE (alias \$ERROR), and \$CAPTURE (declare output variables).

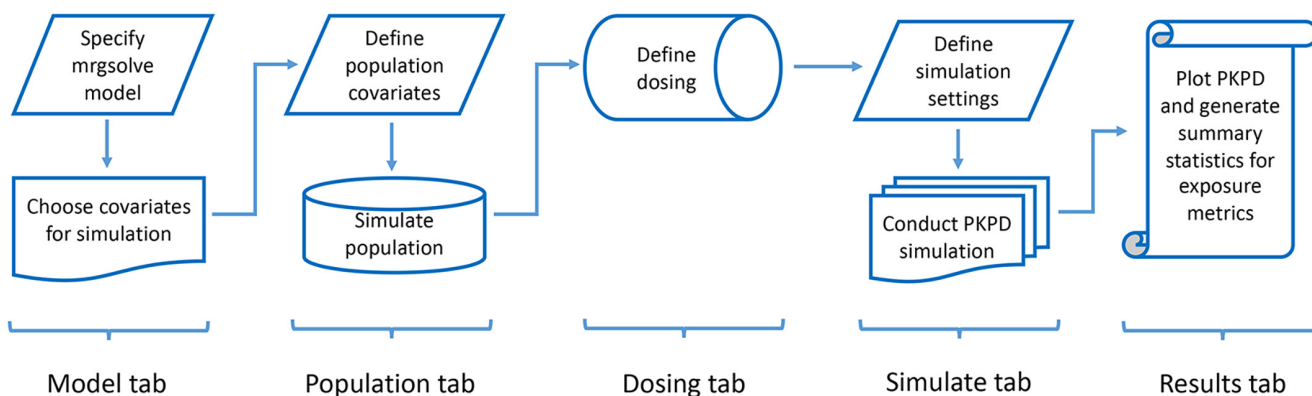
For details about the mrgsolve package, readers can refer to the mrgsolve user guide.<sup>13</sup> It is worth emphasizing here that any variable defined in the \$PARAM block can only be updated by the individual values outside of the model, but not re-defined within the model by other blocks. Similar to NONMEM, mrgsolve has the \$PRED

block and \$PKMODEL block (equivalent to ADVAN 1–4 in NONMEM). Any mrgsolve models that translated from NONMEM code should be validated against NONMEM simulations, for both typical and individual predictions.<sup>30,31</sup>

### Model tab

The first step in the app workflow is to specify an mrgsolve model in the “Model” tab, either by loading an external model or choosing one from the built-in library. Once the model is loaded, users are required to select the variables from the drop-down menu to generate the population data (see the “Population” tab). To make certain variables available in the drop-down menu, they need to be first defined in the \$PARAM block of the model. For each selected variable, the user can specify the data type as either continuous (default) or categorical. Such information is required in the “Population” tab for virtual simulation, and the “Result” tab for stratifications. The user also needs to specify the time unit from the drop-down list, as either day or hour, according to the time unit of the model. It is critical to specify the appropriate model time unit, which otherwise will lead to wrong simulation outcomes. The loaded model code can be manually edited without affecting the source code. This is particularly useful to allow quick exploration of different parameter values. The model will be compiled after the initial loading and re-compiled whenever changes are made. The built-in text editor has a debugging feature that provides the same error message as the mrgsolve code compilation in R, facilitating efficient model development.

Users have the option to define a seed in the Model tab to make the simulations reproducible. Once defined, the same seed will be used across the app to control all the randomization processes, including the virtual population simulation and sampling with replacement (see the “Population tab” section), and the simulations (see the “Simulate tab” section).



**FIGURE 1** Typical workflow of gPKPDviz. PK/PD, pharmacokinetic/pharmacodynamic.

## Population tab

The population data are defined based on the variables selected in the Model tab. The user can either simulate the virtual population within the app (i.e., covariates simulated as per certain distribution or proportion), or load the actual covariate data from an external dataset (in .csv format). The population data is automatically summarized in tables and histogram plots for verifications and visualization purposes.

The selected variables in the Model tab typically represent the model covariates for the virtual population generation. Continuous covariates are simulated based on a truncated multivariate normal distribution with user-specified parameters (mean, standard deviation, bounds, and correlations). Categorical covariates are simulated based on binomial distributions with user-defined proportions. The correlation between continuous and categorical variables is achieved by allowing different distributions for each category. The continuous covariates can be further categorized into bins for stratification of simulated output. Of note, only the categorical variables and categorized continuous variables are used for stratifications.

The selected variables can also include any variables that serve as the data input for simulation, (such as the post hoc parameters for individual simulation) as long as they are defined in \$PARAM block of the model.

Compared to the app-generated virtual population, the user-loaded actual population data have certain advantages. First, it naturally reflects the actual covariate correlations and covariate distributions in the clinical data. Second, it can handle time-varying covariates and inter-occasion variability. Finally, it could incorporate the post hoc parameters for individual simulations. The user-loaded population data could be further enlarged by sampling with replacement.

## Dosing tab

Dosing regimens for the virtual populations are defined in the “Dosing” tab. Users have the option to generate the dosing regimen within the app or load the dosing data through an external dataset. The app can generate up to three unique dosing interventions. Each intervention is configured individually by the dose amount, number of doses, dosing interval, dosing compartment, and route of administration, which can all be combined in a sequential or simultaneous manner when needed. The app provides two options of defining doses: flat dosing and covariate-based dosing (e.g., body weight-based dosing).

Within the app, it is straightforward to define the regular dosing regimens using only one dosing intervention

(e.g., single dose, or repeated dose with equal interval). The irregular dosing regimens, such as loading dose, step-up dosing, and intermittent dosing, can be defined by combining several interventions in a sequential manner. A “Dosing Sequence Table” would be automatically created to reflect the dosing regimens.

Although less frequently used, when combining interventions in a simultaneous manner, it can handle models with more than one dosing compartment, with each compartment defined by a unique intervention. The models that fall into this category include, for example, a PK/PD model for combination therapy where the PD is driven by the PK from each treatment, or a PK model with concurrent absorption process (or different absorption sites) where the PK is driven by each absorption process (or site). Combining interventions in a simultaneous manner also allows the users to explore concurrent intravenous and subcutaneous dosing scenarios for compounds that are eligible for both dosing routes.

Despite its convenience, the app-generated dosing regimen has several limitations. First, it can only handle up to three unique interventions (although in most cases it is enough). Second, it does not allow individualized dosing regimens (i.e., the app-generated regimen is always applied to the entire population). The user-uploaded dosing dataset, on the other hand, can take on whatever complexity is required. The individual dosing regimen, coupling with the post hoc parameters and individual sampling time, can be used for the simulation of individual profiles.

## Simulate tab

The next step is to conduct simulations in the “Simulate” tab, using the loaded mrgsolve model, the previously defined population data, the dosing information, and the sampling times. Depending on the loaded mrgsolve model, either population simulation (random sampling from OMEGA and SIGMA metrics) or individual simulation (reading from individual post hoc parameters) would be performed.

The simulation can be conducted according to the app-generated time grid with equal interval (i.e., the fine time grid), user-defined timepoints with irregular interval, or a combination of both. When the app finds records with actual sampling times (evid=0) in the dosing dataset loaded in the dosing tab, the simulations can follow the actual sampling times, alone or augmented with the fine time grid. If necessary, the typical profile could be simulated, using the “zero\_re” function on the back-end.<sup>13</sup>

Mrgsolve uses the C++ translation of DLSODA solver from ODEPACK.<sup>32</sup> The ‘simulate’ tab allows for the adjustment of key solver settings,<sup>13</sup> including hmax

(maximum step size), maxsteps (maximum steps between adjacent simulation time points), atol (absolute tolerance), and rtol (relative tolerance). Whereas the default values are sufficient for most cases, any modifications should be based on a deep understanding of the system dynamics.

As described in the app workflow, the user can conduct batches of simulations that differ only in dosing regimen, which can be done iteratively between “Dosing” and “Simulate” tabs. It is also possible to conduct and compare simulations with different populations or even different model parameters.

## Results tab

The simulated data is composed of the generic variables (i.e., model unspecific) and the model-specific variables. The generic variables include dosing and sampling information (SIM\_TYPE, REC\_TYPE), subject identifier (USUBJID, ID), simulation identifier (SIM\_ID), time, event identification (EVID), dosing amount (AMT), and cumulative dose count. The model-specific variables include state variables (defined in \$CMT or \$PKMODEL), output variables (defined in \$CAPTURE), and model covariates (including binned covariate derivatives). The state variables and output variables are populated automatically from the loaded mrgsolve model. To include a covariate in the simulated data, it must be defined in \$PARAM and selected in the drop-down menu. A snapshot of the simulated data is shown in [Figure S1](#).

SIM\_TYPE defines the data source, including “Simulation,” “User Specified,” and “User Overlay.” “Simulation” data source corresponds to the three record types in REC\_TYPE, including the “Population” type for covariate data (at time 0 for baseline covariates and non-zero times for time-varying covariates), the “Dose” type for dosing records (at the dosing times), and the “Observation” type for sampling records (at the sampling times). EVID of 2, 1, and 0 are assigned to the “Population,” “Dose,” and “Observation” types, respectively. The definition of evid in the app is the same as NONMEM. To ensure the proper order of the simulated data from different record types, the tiny time shifts are automatically added to the “Dose” (1E-10) and “Observation” (1E-11) types.

Of note, only the simulated fine time grid with equal interval belongs to the “Observation” type of “Simulation” data source; the user-defined sampling times with irregular interval or the actual sampling time loaded from external data belongs to the “Observation” type of “User Specified” data source. The overlaid observed data, if any,

belong to the “Observation” type of “User Overlay” data source.

Filtering of the simulated data can be applied prior to plotting or the calculation of summary stats. Typically, users would filter “Observation” type in REC\_TYPE, because only the simulations at the sampling times are of interest for plotting or summary stats. The output variables declared in \$CAPTURE and the state variables defined in \$CMT or \$PKMODEL will be automatically available for plotting or summary stats. The plots can be shown side-by-side when there are two or more variables selected for the same SIM\_ID (e.g., PK and PD outputs), or different SIM\_IDs for the same variable (e.g., PK output across regimens). The users also have the option to overlay the outputs from different SIM\_IDs in the same plot with different colors. Further stratifications can be made by categorical covariates or categorized continuous covariates, or by subject ID to the extreme. Users have the option to further customize the plot, such as median with 90% interval in log scale. To enhance efficiency in updating the plots, it is preferred to make several changes before hitting the “Update Results” button, so that all the changes can be rendered at once.

One important goal of the simulation is to compare the exposure metrics across dosing regimens or covariate categories. The exposure metrics provided by the app include area under the curve (AUC), concentration at the last time point (LAST, or  $C_{\text{trough}}$ ), maximum concentration (MAX), and minimum concentration (MIN). AUC is calculated by PK/PDmisc using the trapezoidal method.<sup>33</sup> MAX, MIN, and LAST are captured directly from the simulation output. Of note, the “LAST” and “MIN” can be different in the case of oral or subcutaneous dosing. The summary statistics and box plots are generated to enable comparisons across scenarios. The options for summary statistics include mean, median, geometric mean, standard deviation, 90% confidence interval (CI), coefficient of variation (CV%), and geometric coefficient of variation (geoCV%). A reference line (e.g.,  $IC_{90}$  from preclinical studies) can be applied to the exposure metric (e.g.,  $C_{\text{trough}}$ ), thus the percentage above the reference level can be derived and included to the summary stats table. Another feature of the app is to check for agreement between simulation and observations, by allowing the user to upload observed data (e.g., from on-going studies).

Finally, users can export the entire simulation as a zip file, including the model code, seed number (if defined), population data with the associated user-specified parameters (if simulated within the app), dosing table, simulated data, plots, and summary stats. The zip file contains all the information needed for reproducing the simulations.

The downloaded data can be further analyzed by other programs, if desired.

## CASE EXAMPLES

### Case 1: Impact of dose delays and alternative dosing regimens on pertuzumab PK following intravenous infusion

#### Background

The first case demonstrated how to use the app to conduct simulations of various dosing regimens using pertuzumab as an example. The impact of dose delays and alternative dosing regimens on pertuzumab PK was previously published by Liu et al.<sup>34</sup>

Perjeta (generic name pertuzumab) as an intravenous (i.v.) infusion was initially approved in 2012<sup>35</sup> in combination with trastuzumab, to treat patients with HER2-positive breast cancer across the neoadjuvant, adjuvant, and metastatic treatment settings. The approved dose is 420 mg every 3-week cycle following an initial loading dose of 840 mg (i.e., 840 mg/420 mg every 3 weeks). A re-loading dose of 840 mg is required if the dose interval is 6 weeks or longer. This approved regimen will be hereafter referred to as the base case. The dose-delay simulations have the same regimen as the base case, except that the dosing interval between the third and fourth doses is extended to 4, 6, or 9 weeks (i.e., DL4W, DL6W, and DL9W). Two alternative-dosing regimens were tested, that is, 840 mg/420 mg every 4 weeks and 840 mg every 6 weeks (i.e., ALTQ4W and ALTQ6W).

Pertuzumab PK following i.v. infusion was best described by a two-compartment model with linear elimination. Baseline serum albumin (ALBU) and lean body weight (LBW) were identified as covariates on clearance (CL), while LBW as covariates on central ( $V_1$ ) and peripheral volume of distribution ( $V_2$ ).<sup>36</sup> The mrgsolve model of pertuzumab PK was implemented (Perjeta\_valid.cpp; Supplementary Model S1) and validated against the NONMEM simulations (shown in HTML S1 from R Markdown).

Preclinical xenograft efficacy models indicated that the target steady-state  $C_{\text{trough}}$  for pertuzumab is 20  $\mu\text{g}/\text{mL}$  for maximum tumor growth suppression.<sup>37</sup> The approved regimen for HER2-positive breast cancer (i.e., the base case) demonstrated favorable efficacy when  $C_{\text{trough}}$  greater than 20  $\mu\text{g}/\text{mL}$  was achieved in 90% of patients.<sup>38</sup> Thus, the percentage of the simulated patients with  $C_{\text{trough}}$  above 20  $\mu\text{g}/\text{mL}$  was used as the criteria for the assessment of various regimens.

#### Model tab

After loading the model (Perjeta\_valid.cpp), a pop-up window appeared to define the “Model Covariates” (select “ALBU” and “LBW”), “Covariate Types” (select “Continuous”), and the “Model Time Unit” (select “day”). A randomization seed was provided to ensure reproducibility. A “Model Summary” was automatically created for verification, and the complete mrgsolve code was also displayed (Figure 2).

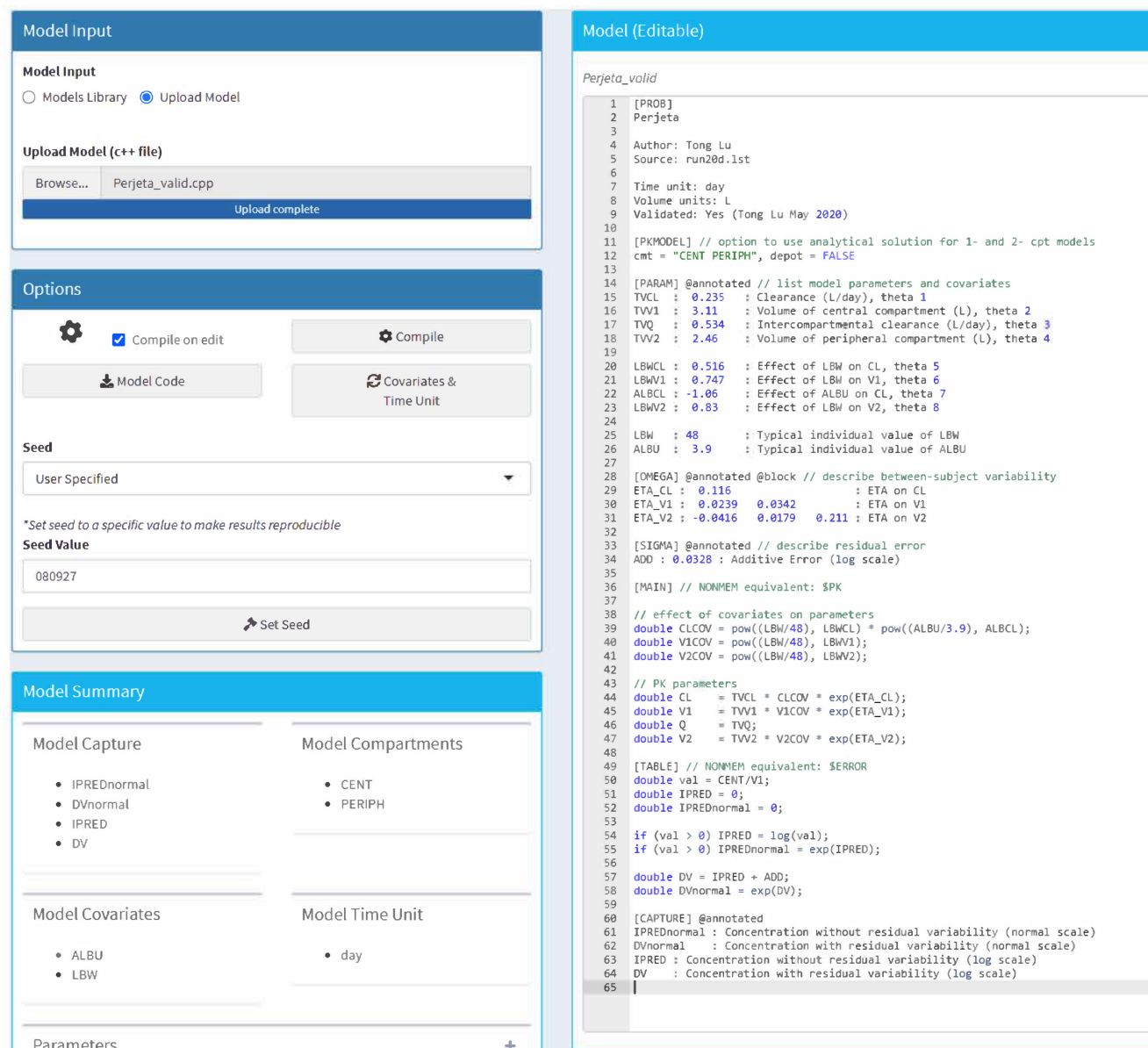
#### Population tab

In the paper by Liu et al.,<sup>34</sup> LBW and ALBU were simulated by gPKPDviz assuming a truncated normal distribution. In this analysis, the dataset containing actual values of LBW and ALBU (population.data.csv;  $n=477$ ) were loaded for simulation (Figure S2). Data from representative individuals were included in the supplementary file as Data S1. The requirements for the format of external data sets are shown under “Data Upload Criteria” after clicking “Upload Dataset.” One thousand virtual patients were further generated by sampling with replacement (Figure 3), which was summarized with a table and histogram plot. By default, the population data for the first 10 individuals is shown under the “Data” sub-tab. The data distribution as histogram plots can be revealed under the “Baseline Values” sub-tab. The histogram plots for the 477 and the 1000 patients confirmed the data similarity (data not shown). LBW and ALBU were categorized into four bins for plotting and summarization purposes, and were appended to the original data (Figure 3).

#### Dosing tab

The dosing regimen for all the cases (i.e., base case [BASE], delay-4-week [DL4W], delay-6-week [DL6W], delay-9-week [DL9W], alternative-q4-week [ALTQ4W], and alternative-q6-week [ALTQ6W]) were defined as external datasets for loading (Data S1). The doses were given as i.v. infusions over 1-h for 840 mg or 30-min for 420 mg, for a total of eight cycles.

It is also feasible to define the base case and alternative regimens within the app. Shown in Figure S3 is an example of defining the base case, where two dose interventions describing loading and maintenance phases separately were combined sequentially. However, the app could not generate the delayed regimens, given that more than three interventions would be needed.



**Model Input**

Model Input

Models Library  Upload Model

Upload Model (c++ file)

Browse... Perjeta\_valid.cpp

Upload complete

**Options**

Compile on edit

**Seed**

User Specified

\*Set seed to a specific value to make results reproducible

**Seed Value**

080927

**Model Summary**

Model Capture	Model Compartments
<ul style="list-style-type: none"> <li>• IPREDnormal</li> <li>• DVnormal</li> <li>• IPRED</li> <li>• DV</li> </ul>	<ul style="list-style-type: none"> <li>• CENT</li> <li>• PERIPH</li> </ul>
Model Covariates	Model Time Unit
<ul style="list-style-type: none"> <li>• ALBU</li> <li>• LBW</li> </ul>	<ul style="list-style-type: none"> <li>• day</li> </ul>

Parameters +

**Model (Editable)**

```
Perjeta_valid
1 [PROB]
2 Perjeta
3
4 Author: Tong Lu
5 Source: run20d.lst
6
7 Time unit: day
8 Volume units: L
9 Validated: Yes (Tong Lu May 2020)
10
11 [PKMODEL] // option to use analytical solution for 1- and 2- cpt models
12 cmt = "CENT PERIPH", depot = FALSE
13
14 [PARAM] @annotated // list model parameters and covariates
15 TVCL : 0.235 : Clearance (L/day), theta 1
16 TVV1 : 3.11 : Volume of central compartment (L), theta 2
17 TVQ : 0.534 : Intercompartmental clearance (L/day), theta 3
18 TVV2 : 2.46 : Volume of peripheral compartment (L), theta 4
19
20 LBWCL : 0.516 : Effect of LBW on CL, theta 5
21 LBWV1 : 0.747 : Effect of LBW on V1, theta 6
22 ALBCL : -1.06 : Effect of ALBU on CL, theta 7
23 LBWV2 : 0.83 : Effect of LBW on V2, theta 8
24
25 LBW : 48 : Typical individual value of LBW
26 ALBU : 3.9 : Typical individual value of ALBU
27
28 [OMEGA] @annotated @block // describe between-subject variability
29 ETA_CL : 0.116 : ETA on CL
30 ETA_V1 : 0.0239 0.0342 : ETA on V1
31 ETA_V2 : -0.0416 0.0179 0.211 : ETA on V2
32
33 [SIGMA] @annotated // describe residual error
34 ADD : 0.0328 : Additive Error (log scale)
35
36 [MAIN] // NONMEM equivalent: $PK
37
38 // effect of covariates on parameters
39 double CLCOV = pow((LBW/48), LBWCL) * pow((ALBU/3.9), ALBCL);
40 double V1COV = pow((LBW/48), LBWV1);
41 double V2COV = pow((LBW/48), LBWV2);
42
43 // PK parameters
44 double CL = TVCL * CLCOV * exp(ETA_CL);
45 double V1 = TVV1 * V1COV * exp(ETA_V1);
46 double Q = TVQ;
47 double V2 = TVV2 * V2COV * exp(ETA_V2);
48
49 [TABLE] // NONMEM equivalent: $ERROR
50 double val = CENT/V1;
51 double IPRED = 0;
52 double IPREDnormal = 0;
53
54 if (val > 0) IPRED = log(val);
55 if (val > 0) IPREDnormal = exp(IPRED);
56
57 double DV = IPRED + ADD;
58 double DVnormal = exp(DV);
59
60 [CAPTURE] @annotated
61 IPREDnormal : Concentration without residual variability (normal scale)
62 DVnormal : Concentration with residual variability (normal scale)
63 IPRED : Concentration without residual variability (log scale)
64 DV : Concentration with residual variability (log scale)
65
```

**FIGURE 2** Model tab for pertuzumab population simulation in gPKPDviz. Select model covariates (albumin [ALBU] and lean body weight [LBW]); set model time unit (day); set random seed (080927) for replicable simulation.

## Simulate tab

For the base case, the user-specified sampling times relative to the most recent dose were defined in “Additional (Optional) inputs,” according to the Garg analysis,<sup>36</sup> as day 0, 0.021 (30-min), 0.0625 (90-min), and each day from day 1 to day 21. For the delayed regimens or alternative regimens, one sample right before the next dose was added automatically to the user-specified sampling times, which captured the  $C_{\text{trough}}$  for every regimen without needing to manually revise the sampling times. To simulate only by user-specified sampling times, the “Sampling Interval” and “Sampling Duration” were set to zero. The snapshot for defining sampling times for the base case is shown

in Figure S4. As explained in the App overflow section, the simulated data at the user-specified sampling times will correspond to the “Observation” type with “User Specified” data source.

The process of defining the Dosing tab and Simulation tab can be repeated for all cases, with corresponding SIM\_IDs (BASE, DL4W, DL6W, DL9W, ALTQ4W, and ALTQ6W).

## Result tab

The structure of the simulated data is shown in Figure S1 for the BASE case. The variables that are unique to the

**Data Input Type**

Population Data

Simulate Dataset  Upload Dataset

---

**Upload Data Set**

Choose CSV File

Browse... population.data.csv

Upload complete

---

**Data Upload Criteria:**

- Comma separated (.csv) file
- Necessary columns: **USUBJID**, **time**, **ALBU**, **LBW**
- All columns not in the list of **Necessary** columns are dropped
- **ID** column will be auto-generated based on **USUBJID**
- If your covariate data is not time-varying, create a **time** column with all values of 0 (zero)

---

**Population Data**

Number of subjects: 1,000

Baseline Across Time

Variable	n obs	mean	sd	cv	min	max	median
ALBU	1000	3.85	0.53	13.75	2.1	5.3	3.9
LBW	1000	51.72	9.61	18.58	33.16	82.25	48.77

Correlation Matrix

	ALBU	LBW
ALBU	1.00	0.01
LBW	0.01	1.00

Data Baseline Values Values Across Time

Show 10 entries Search:

USUBJID	ID	time	ALBU	LBW	ALBU (Bins)	LBW (Bins)
1	1	0	3.8	62.2	[3.5,3.9]	[58,82.2]
2	2	0	4.6	44.1	[4.2,5.3]	[33.2,44.6]
3	3	0	3.8	62.2	[3.5,3.9]	[58,82.2]
4	4	0	2.7	64.1	[2.1,3.5]	[58,82.2]
5	5	0	4.2	41.7	[3.9,4.2]	[33.2,44.6]
6	6	0	3	59.5	[2.1,3.5]	[58,82.2]
7	7	0	3.4	61.6	[2.1,3.5]	[58,82.2]
8	8	0	4.1	42.8	[3.9,4.2]	[33.2,44.6]
9	9	0	3.2	41.9	[2.1,3.5]	[33.2,44.6]
10	10	0	3.4	63.4	[2.1,3.5]	[58,82.2]

Showing 1 to 10 of 1,000 entries Previous 1 2 3 4 5 ... 100 Next

Download Data

---

**Sample From Initial Population Data**

Number of Subjects in Initial Data: 477

Number Subjects to Sample

1000

Sample

\*With replacement used when Number Subjects to Sample > Number Subjects in Initial Data

---

**Create Bin Columns for Continuous Covars**

Select Covariate  ALBU  LBW

Method  Define Number of Bins  Define Break Points

Number of Bins

4

Create

\*These columns will be used subsequently for grouping/visualizations

\*If covariate is time-varying, subjects will get binned based on their first observation

**FIGURE 3** Population tab for pertuzumab population simulation in gPKPDviz: sampling with replacement. First, upload covariate values from the external dataset (population.data.csv;  $n = 477$ ); second, sample with replacement to generate 1000 virtual patients from the 477 actual patients; third, generate binned covariates for albumin (ALBU) and lean body weight (LBW).

pertuzumab case are ALBU, LBW (model covariates), CENT, PERIPH (state variables),  $IPRED_{normal}$ ,  $DV_{normal}$ ,  $IPRED$ ,  $DV$  (output variables), and  $ALBU_{bin}$  and  $LBW_{bin}$  (binned derivatives for covariates).

The simulation exercises shown below were to reproduce the key results from Liu et al.<sup>34</sup>

### 1. Pertuzumab base case PK

First, filter BASE under “SIM\_ID” variable to extract the base case simulations. By clicking “Update Results,” the plots under default settings were created, which was the pooled individual PK predictions over time (i.e.,  $IPRED_{normal}$ ), colored by SIM\_TYPE (Figure S5). It was generated by default due to the fact that  $IPRED_{normal}$  was the first output variable listed in \$CAPTURE.

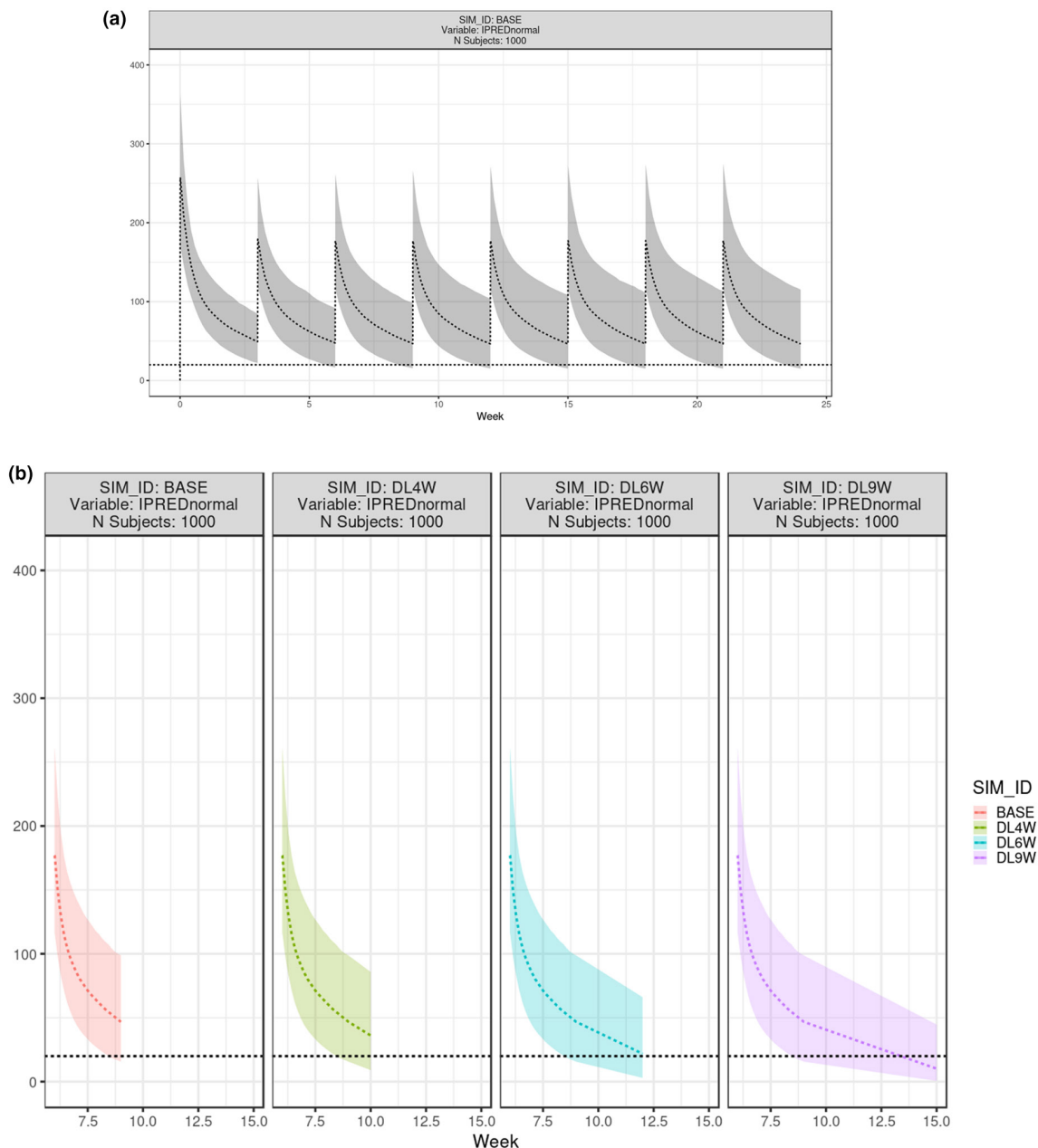
Several steps were taken to customize the plot (Figure S6A). First, for data source, filter “User Specified” under SIM\_TYPE (or filter “Observation” under REC\_TYPE); second, for plot display, select “Percentiles” for “User Specified” data; third, for reference line, enter 20 as reference value; fourth, for time scale, adjust “Time Unit” to Week; fifth, for color, set “Color by” to “None” (i.e., gray color); and sixth, for Y-Axis, set “Y-Axis Max”

to 400. After clicking “Update Results”, the updated plot was shown as median and 90% interval in the time scale of week (Figure 4a; refer to figure 1 from Liu et al.<sup>34</sup>).

To derive the summary stats for AUC,  $C_{max}$ , and  $C_{trough}$  in the first four cycles (Table 1; refer to table 1 from Liu et al.<sup>34</sup>), first extract the cycle one to four data (filters 1, 2, 3, and 4 under “Cumulative Dose Count” variable), choose the exposure metrics (tick AUC, LAST, MAX in “Stats Display”), choose the summary stats of interests (tick median/5th percentile/95th percentile in “Stats Type”), and then summarize by “Cumulative Dose Count” (Figure S6B). The 20  $\mu\text{g}/\text{mL}$  threshold was defined under the “Summary Stats” sub-tab for  $C_{trough}$  (“LAST” Threshold), in order to derive the percentage above threshold for the simulated  $C_{trough}$  values in the summary stats table.

Using the BASE case simulation, the impact of ALBU on the steady-state PK at cycle four can be easily explored by the stratified covariate  $ALBU_{bin}$  (i.e., summarized by  $ALBU_{bin}$ ). By providing 20  $\mu\text{g}/\text{mL}$  as threshold for  $C_{trough}$ , the percentage above threshold for each category was also included in the summary stats table (Figure S7, Table S1). The animated demonstrations for the BASE case simulation stratified by  $ALBU_{bin}$  were provided on GitHub.<sup>22</sup>





**FIGURE 4** Result tab for pertuzumab population simulation in gPKPDviz: PK profiles. (a) Simulation of individual prediction (IPRED) for pertuzumab base case, shown as median and 90% interval from 1000 virtual patients for 8 cycles; X axis is in week, and Y axis is in  $\mu\text{g/mL}$ ; (b) Effect of dose delay on pertuzumab PK at cycle 3, shown as median and 90% interval of IPRED from 1000 virtual patients. X axis is in week, and Y axis is in  $\mu\text{g/mL}$ ; Plots were laid out side-by-side for base case (BASE) and the dose delay cases (DL4W, DL6W, and DL9W). PK, pharmacokinetic; SIM, simulated.

## 2. Effect of dose delay on pertuzumab PK

To assess the impact of the extended dose interval (i.e., dose delay) on the  $C_{\text{trough}}$  at cycle 3, BASE, DL4W, DL6W, DL9W were filtered under “SIM\_ID,” and “3” was filtered under “Cumulative Dose Count.” Once summarized by SIM\_ID, the PK profiles at cycle three were plotted across regimens (Figure 4b). Similarly, the summary stats table was generated to compare across regimens (Table 1, refer

to table 3 from Liu et al.<sup>34</sup>). The animated demonstration for comparing BASE and DL6W was provided on GitHub.<sup>22</sup>

## 3. Effect of alternative dosing on pertuzumab PK

To assess the impact of the alternative dosing regimen (ALT4W and ALT6W), the PK profiles following eight cycles of dosing were compared to the BASE case in Figure S8 (refer to figure 3 in Liu et al.<sup>34</sup>). The summary stats for

**TABLE 1** Result tab of gPKPDviz for pertuzumab population simulation: summary statistics.

Summarized by	% Above LAST Threshold (20)	AUC stats	LAST stats	MAX stats
<i>Summary Stats Table (for base case in the first four cycles)</i>				
SIM_ID BASE	96.9	Median: 1990	Median: 49.3	Median: 254
Cumulative dose count 1		5th Percentile: 1350	5th Percentile: 23.3	5th Percentile: 176
Variable: IPREDnormal		95th Percentile: 2830	95th Percentile: 83.4	95th Percentile: 366
N subjects: 1000				
SIM_ID BASE	93.7	Median: 1700	Median: 47.6	Median: 178
cumulative dose count 2		5th Percentile: 982	5th Percentile: 17.8	5th Percentile: 122
Variable: IPREDnormal		95th Percentile: 2710	95th Percentile: 90.4	95th Percentile: 259
N subjects: 1000				
SIM_ID BASE	92.5	Median: 1680	Median: 47.1	Median: 177
cumulative dose count 3		5th Percentile: 949	5th Percentile: 16.2	5th Percentile: 120
Variable: IPREDnormal		95th Percentile: 2860	95th Percentile: 97.7	95th Percentile: 261
N subjects: 1000				
SIM_ID BASE	92.2	Median: 1680	Median: 46.7	Median: 177
cumulative dose count 4		5th Percentile: 931	5th Percentile: 15.9	5th Percentile: 119
Variable: IPREDnormal		95th Percentile: 2940	95th Percentile: 101	95th Percentile: 264
N subjects: 1000				
<i>Summary Stats Table (for base case and dose delay cases at cycle 3)</i>				
SIM_ID BASE	92.5	Median: 1680	Median: 47.1	Median: 177
Variable: IPREDnormal		5th Percentile: 949	5th Percentile: 16.2	5th Percentile: 120
N subjects: 1000		95th Percentile: 2860	95th Percentile: 97.7	95th Percentile: 261
SIM_ID DL4W	80.7	Median: 1980	Median: 36	Median: 177
Variable: IPREDnormal		5th Percentile: 1040	5th Percentile: 9.89	5th Percentile: 120
N subjects: 1000		95th Percentile: 3480	95th Percentile: 83.2	95th Percentile: 261
SIM_ID DL6W	52.8	Median: 2410	Median: 21.3	Median: 177
Variable: IPREDnormal		5th Percentile: 1170	5th Percentile: 3.46	5th Percentile: 120
N subjects: 1000		95th Percentile: 4480	95th Percentile: 62	95th Percentile: 261
SIM_ID DL9W	24.5	Median: 2890	Median: 9.78	Median: 177
Variable: IPREDnormal		5th Percentile: 1320	5th Percentile: 0.655	5th Percentile: 120
N subjects: 1000		95th Percentile: 5700	95th Percentile: 42.2	95th Percentile: 261

Note: Summary statistics include the median, 5th and 95th percentile for AUC,  $C_{max}$  (MAX), and  $C_{trough}$  (LAST) from the 1000 virtual patients, and the percent of patients with  $C_{trough}$  above 20  $\mu\text{g}/\text{mL}$  threshold; The dose delay cases have the same regimen as the base case, except that the dosing interval between the 3rd and 4th dose is extended to 4-, 6-, or 9-weeks (i.e., DL4W, DL6W, and DL9W).

Abbreviations: AUC, area under the curve;  $C_{max}$ , maximum plasma concentration;  $C_{trough}$ , trough plasma concentration; LAST, concentration at the last time point; MAX, maximum concentration.

cycle one or four were derived and compared across regimens in Table S2 (i.e., summarized by both SIM\_ID and cumulative dose count) (refer to table 4 in Liu et al.<sup>34</sup>).

## Individual simulation and sensitivity analysis

The simulation exercises shown below were to illustrate additional features of the app, including individual simulation and sensitivity analysis:

### 1. Individual simulation

To illustrate the features of individual simulation, the dataset containing post hoc estimates was compiled

for the population tab (population.data.indpara.csv; representative individuals in Data S1), and the dataset containing individual dosing and sampling records was compiled for the dosing tab (SamplingDosing.ind.csv; representative individuals in Data S1). The mrgsolve code for population simulation was modified for individual simulations (Perjeta.ind.cpp; Model S1), including (1) replace \$PARAM block with a list of post hoc parameters (use the population estimates as placeholder); (2) remove \$OMEGA block, and (3) redefine the PK parameter in \$MAIN block with the post hoc parameters. During the simulation process, the post-hoc parameters in the \$PARAM block were updated seamlessly with the individual estimates from the population data (population.data.indpara.csv), for the variables with common names (i.e.,  $CL_{ind}$ ,  $V_{1ind}$ ,  $Q_{ind}$ , and  $V_{2ind}$ ).

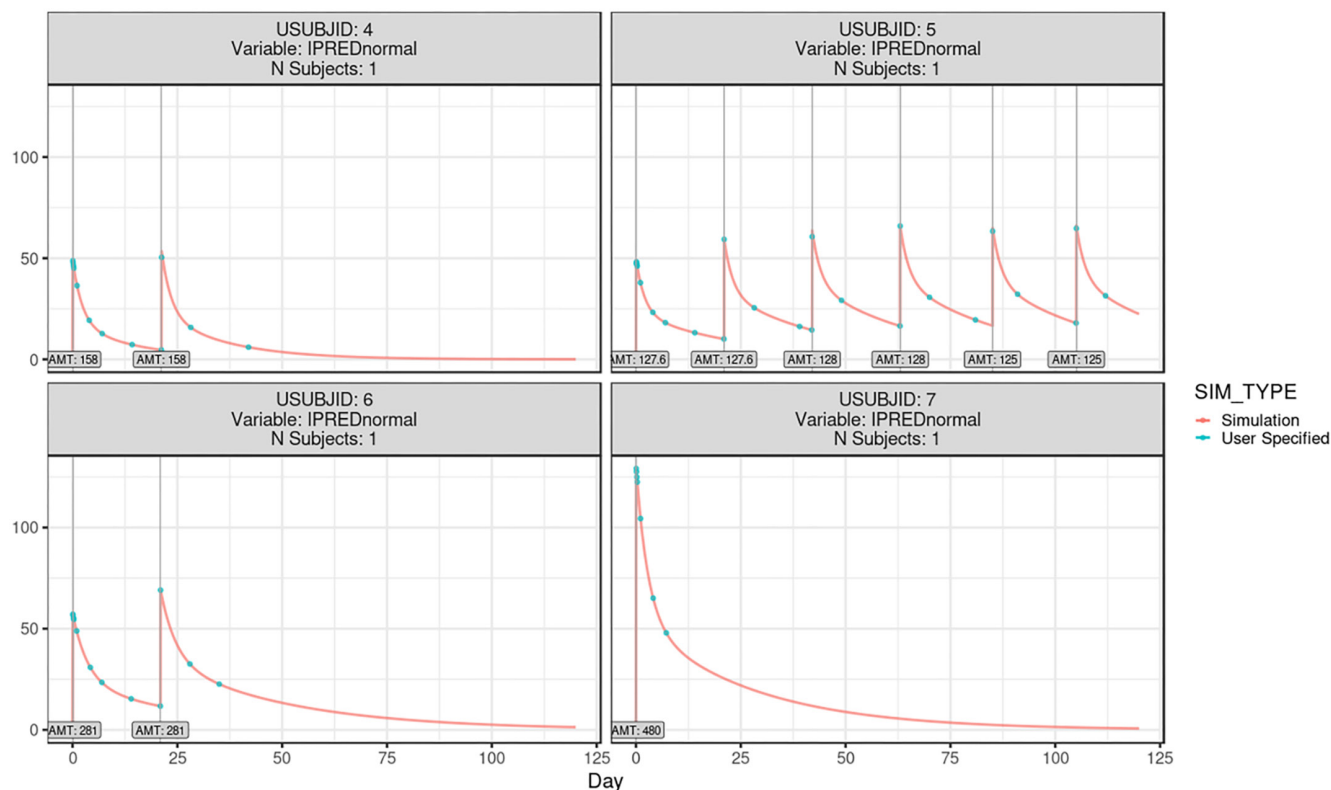
After loading `SamplingDosing.ind.csv` in the dosing tab, the following message popped up: “Non-dosing rows found in dosing data set. These records will be used as additional observation time points in the simulated data.” In addition, the fine time grid was also simulated at every half hour for 120 days, overlaying the simulations at the actual observation time points (colored by `SIM_TYPE`). Of note, the simulations from fine time grid and actual sampling times have `SIM_TYPE` as “Simulation” and “User Specific,” respectively. The individual simulations were displayed when further stratifying the plots by subject ID (i.e., summarized by `USUBJID`), shown in [Figure 5](#) for the representative subjects. To label the dose time and amount on the individual plots, the box for “Show dose events” was ticked. The animated demonstration for individual simulation was provided on [GitHub](#).<sup>22</sup>

## 2. Sensitivity analysis

Sensitivity analysis here refers to how sensitive the model prediction is to the change of a structural parameter, whereas all the other parameters are fixed. It is meant to evaluate the contribution/importance of a structural parameter to the model overall, through graphical visualization.<sup>6,39</sup> The population prediction is usually used for

this purpose, without accounting for the interindividual variability. The following steps are needed to perform the sensitivity analysis in the app for all the parameters at once. The structural parameters being tested included  $CL$ ,  $V_1$ , inter-compartmental clearance, and  $V_2$ .

- Dataset: create a dataset containing a list of individuals with varying parameter values, sorted by `PARAM` (set to 1 when  $CL$  varies and 2 when  $V_1$  varies, etc.) and `RANK` (set to 1 for the lowest value tested and 5 as the highest, etc., increased gradually by a 2-fold magnitude; `population.data.sensitivity.allpara.csv`; Data [S1](#)).
- Model file: modify `Perjeta_valid.cpp` by adding `PARAM` and `RANK` to `$PARAM` block (`Perjeta.sensitivity.cpp`; Model [S1](#)).
- Model tab: select `PARAM`, `RANK`, and all the structural parameters from the drop-down list; set “Model Time Unit” to day; set `PARAM` and `RANK` as categorical covariate.
- Population tab: load the dataset created in step a.
- Dosing tab: load the dosing regimen from the external dataset (e.g., `Base.csv`).
- Simulate tab: simulate the population prediction by ticking the box for “PRED Simulation”, using the user-specified sampling time defined in “Simulate tab” of Case 1.



**FIGURE 5** Individual simulation of pertuzumab PK in gPKPDviz for the representative individuals. The individual simulations were based on the individual dosing history, post hoc PK parameters, and the actual sampling time (green dots) augmented with the fine time grid (red curve); the vertical lines represented the dose time and amount; X axis is in day, and Y axis is in µg/mL. IPRED, individual prediction; PK, pharmacokinetic.

g. Results tab: summarize the plots by PARA, and color by RANK; untick “Share Y-Axis” under “Axis”.

The resulting sensitivity plots were shown in [Figure 6](#) for the base case. The users can conduct the sensitivity analysis more efficiently in R using the `mrgsim.sa` package, which is showcased in the `mrgsolve` gallery.<sup>40</sup>

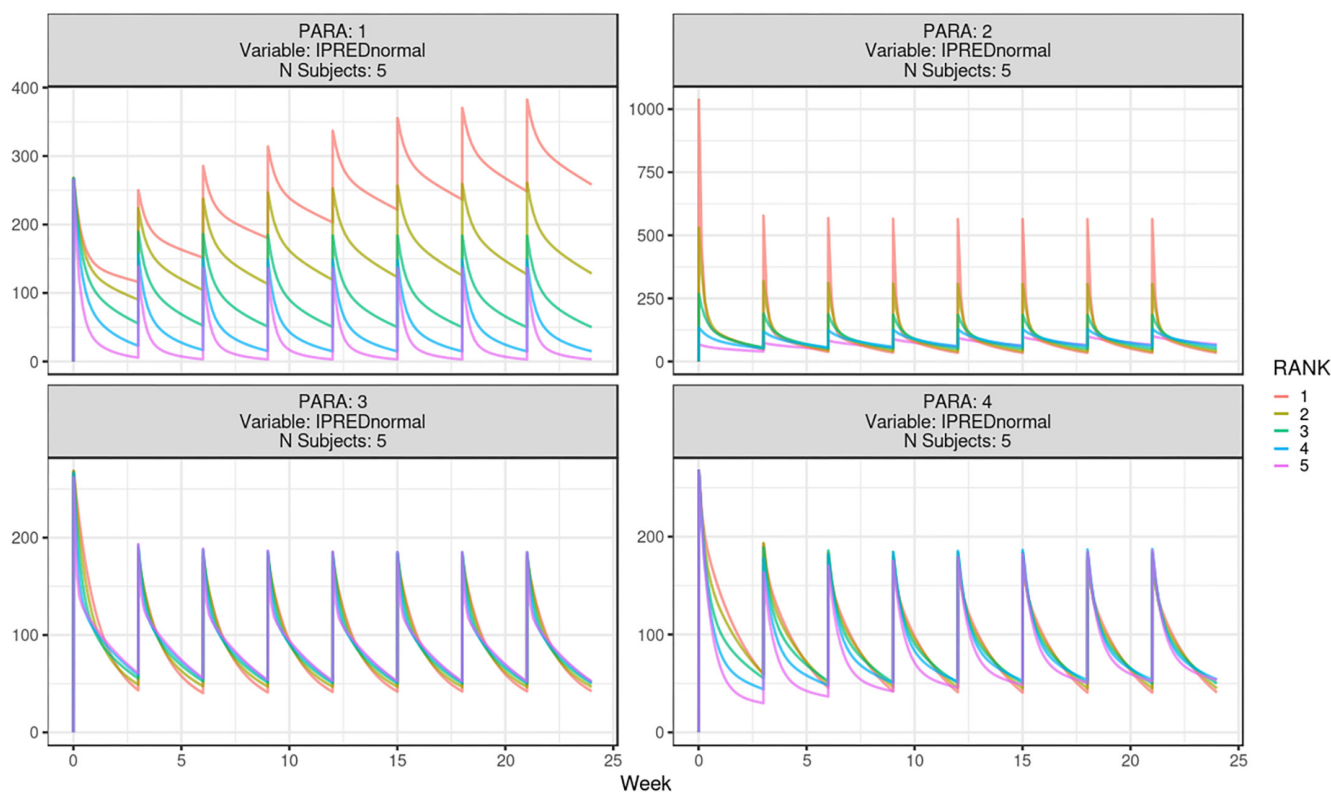
## Case 2: Body weight-based dosing versus fixed dosing of polatuzumab vedotin for heavier patients

The second case was to demonstrate how to use `gPKPDviz` to efficiently assess body weight-based dosing versus flat dosing for a heavier patient population, using polatuzumab vedotin as an example.

Polivy (generic name polatuzumab vedotin; pola) is a CD79b-directed antibody-drug conjugate with activity against dividing B cells. The small molecule, Monomethyl auristatin E (MMAE), is an anti-mitotic agent covalently attached to the antibody via a cleavable linker. Pola was approved as the combination therapies to treat adult patients with previously untreated or treated diffuse large B-cell lymphoma.<sup>41,42</sup> The recommended dosing

regimen is 1.8 mg/kg (i.v. infusion over 90 min) every 3 weeks for six cycles. A two-analyte integrated population PK (PopPK) model<sup>43</sup> for pola (antibody-conjugated MMAE [acMMAE] and unconjugated MMAE) has been developed based on data from 460 patients in four clinical studies. The `mrgsolve` code for pola PopPK model was developed and validated against the NONMEM output (`Pkcase_Polatuzumab.valid.cpp`; included in the published model library). It was modified for the individual simulation purpose (`Pola.ind.cpp`; Model S1).

Given that pola dosing is based on body weight, there is a risk of overdosing for patients with heavier body weights ( $\geq 100$  kg) considering that acMMAE clearance increases less than proportionally to body weight<sup>43</sup>; a flat dose of 180 mg could be a better regimen for this subpopulation. To compare PK between the two regimens, the 460 patients were stratified into two subgroups, that is, 100–146 kg versus 38–100 kg within the app, and their cycle six exposures were simulated following either six cycles of bodyweight-based dosing at 1.8 mg/kg or flat dosing at 180 mg. The simulations were conducted based on the user-loaded post hoc estimates, app-generated dosing regimens, and the fine time grid for simulation (every 2 h sample for 18 weeks). [Figure 7](#) illustrates the implementation of covariate-based



**FIGURE 6** Sensitivity analysis of pertuzumab PK in `gPKPDviz`. Sensitivity simulation of individual prediction (IPRED) for pertuzumab base case; PARA 1, 2, 3, 4 represent CL,  $V_1$ ,  $Q$ , and  $V_2$ , respectively; for each parameter, set RANK to 1, 2, 3, 4, 5, with 1 as the lowest value tested and 5 as the highest value (increase gradually by a two-fold magnitude); X axis is in week, and Y axis is in  $\mu\text{g}/\text{mL}$ . CL, clearance;  $Q$ , inter-compartmental clearance;  $V_1$ , central volume of distribution;  $V_2$ , peripheral volume of distribution.

Data Input Type

**Dose Schedule**

Build Using Interface  Upload Schedule

---

Construct Dose Interventions (up to three unique)

Intervention A    Intervention B    Intervention C

**Number of Doses**

**Dosing Amount**

**Dosing Interval**

**Event ID (EVID)**

**Compartment**

**Infusion Duration**

**Amount in Proportion to Covariate**

**Interval Unit**

**Admin**

**Unit**

Intervention Sequence

**First Intervention**

Intervention:

Start time:

\*First intervention starts at time = 0

**Second Intervention**

Intervention:

---

Applied Dosing Sequence Table

Intervention	Amount	Amount Proportion	Admin.	Start Time (Hour)	Dose Interval (Hour)	# Doses	Event ID	Comp.
A	32.5326	covBWT	Infusion (Duration: 1.5 hours)	0	504	6	1	CENT

**FIGURE 7** Implementation of body weight-based dosing in gPKPDviz for polatuzumab. The polatuzumab was dosed at 1.8 mg/kg (corresponding to 32.5326  $\mu\text{g}/\text{kg}$  acMMAE), by intravenous.v. infusion of 1.5 h every 3 weeks for six cycles. acMMAE, antibody-conjugated monomethyl auristatin E.

dosing in gPKPDviz. Comparisons were made between the two regimens for the heavier bodyweight subgroup, and also between the two subgroups following the bodyweight-based dosing (published previously<sup>44</sup>).

For the heavier bodyweight subgroup, the cycle six PK profiles for acMMAE and MMAE were shown in [Figure S9](#) following bodyweight-based or flat dosing; the geometric mean ratio of cycle six AUC for bodyweight-based versus flat dosing were 1.13 and 1.17, respectively, for acMMAE and MMAE (derived from [Table S3](#)). The magnitude of increase ( $\leq 17\%$ ) for bodyweight-based dosing was considered not of clinical relevance. Similarly, following bodyweight-based dosing, the geometric mean ratio for higher versus lower bodyweight subgroups were 1.08 and 1.27, respectively, for acMMAE and MMAE ([Figure S10](#), [Table S4](#)). The magnitude of increase for heavier subgroup was relatively higher for MMAE (27%) but was still small compared with the interindividual variability as indicated by coefficient of variation of the geometric mean (geoCV%: 40.5~53%; [Table S4](#)). The comparison outcomes for  $C_{\text{max}}$  were very similar to those for AUC (results not shown). Overall, dose capping of 180 mg for patients with bodyweight greater than or equal to 100 kg is not warranted based on the available data, supporting bodyweight-based dosing for pola.<sup>44</sup>

## Model validation

To validate the mrgsolve models used in the app, we compared the simulations from the mrgsolve model to the corresponding NONMEM model. The original NONMEM dataset was fed into both models to check the discrepancies for population prediction (PRED), individual prediction (IPRED), and individual prediction with residual error (DV).

Any noticeable discrepancies at the PRED level would be indicative of misspecification in the structural component of the mrgsolve model. To assess the discrepancies at the IPRED and DV levels, population simulation with 1000 replicates was conducted. For each individual, the median, 5th, and 95th of the 1000 simulated profiles were plotted and compared between the two models. The 5th and 95th percentiles from mrgsolve might deviate slightly from NONMEM purely due to the randomness, even for a correctly implemented mrgsolve model. To reduce the impact from randomness, the VPC-like plots (with CIs around the prediction intervals<sup>45</sup>) were generated for IPRED and DV using the nominal time. To complete this, first extract a subset of the dataset; ideally, it would be a group of patients ( $n > 50$ ) taking the same dose level with similar sampling and dosing schedules. Second, for each replicate, derive the median,

5th, and 95th percentiles (i.e., prediction intervals) for the patient subset at each nominal timepoint. Third, calculate the median, 5th, and 95th percentiles (i.e., CIs) around each prediction interval.

The `mrgsolve` code for pertuzumab population PK model (`Perjeta_valid.cpp`) is validated. See the html output (Supplementary *Rmd S2*) generated from R markdown.

## App validation

The `gPKPDviz` app itself was also validated by confirming that the key results from `mrgsolve` simulations outside of the app by R matched the output from the app, when applying the identical seed. The simulation of pertuzumab base case was used for the validation. Specifically, it is 420 mg every 3-week cycle following an initial loading dose of 840 mg, for a total of eight cycles. The user-defined sampling times relative to the most recent dose were defined according to the Garg analysis<sup>36</sup> as day 0, 0.021 (30-min), 0.0625 (90-min), and each day from day 1 to day 21.

The key simulations performed for validation purpose include the following:

Case 1: population simulation based on the simulated covariates (1000 virtual individuals), the simulated dosing regimen and the user-defined sampling times; set seed to for randomization control;

Case 2: population simulation based on the actual covariates (sampling with replacement from the 477 individuals to generate 1000 individuals), the simulated dosing regimen and the user-defined sampling time; set seed to for randomization control;

Case 3: case 2 with simulation of typical profile (PRED);

Case 4: case 2 filtered by cycle eight and stratified by  $ALBU_{bin}$ ;

Case 5: individual simulation of the actual patients ( $n=477$ ) based on the post hoc parameters, the individual dosing history, and the individual sampling time augmented with the fine time grid.

The key app features being validated directly or indirectly include the following:

- Population tab: truncated multi-normal distribution, sampling-with-replacement, binning of continuous covariates, randomization control.
- Dosing tab: regimen simulation with two interventions (840mg loading dose and 420mg maintenance dose), individual dosing history filtering for individual simulation.

- Simulate tab: population simulation, typical profile simulation, fine time grid simulation, simulation as per individual sampling time with or without the fine time grid, randomization control.
- Result tab: data assembly and filtering, median time-profile with 90% CI, stratifications, exposure metrics derivation, percent above threshold derivation, and summary statistics.

The detailed validation process is included in the Supplementary *Rmd S1* file generated from R markdown. The testing of time-varying covariates and correlation between continuous variables are not covered here because they did not apply to the pertuzumab case; those features were validated in-house by other case examples (results not shown).

## DISCUSSION

We have developed and showcased a powerful and flexible PK/PD model simulator (`gPKPDviz`), with an intuitive interface to cover most needs for the scenario simulations in clinical pharmacology. Although all the simulations done in this app could be achieved in R using `mrgsolve` package, `gPKPDviz` is an attractive tool for the real-time simulations in a project team setting, with fast turnaround time. Developing `mrgsolve` models is the prerequisite for using the app. Once the `mrgsolve` models are developed by the PK/PD scientists, it can be loaded onto the app for applications by team members who know PKs but have limited programming skills. This app can also serve as an exploratory tool to unveil the impact of various components of the model on the outcomes. Furthermore, it offers an interactive, open-source platform for teaching `mrgsolve` or pharmacometrics in general in an academic context.

We internally use `gPKPDviz` as a platform to host models for the selected Roche/Genentech molecules. As an open-source platform,<sup>22</sup> we offer a selection of basic model templates in the library including one, two, three compartment PopPK models and indirect response model, and some publicly available models including PopPK models (pertuzumab and polatuzumab) and PK/PD model (Neutropenia model for Paclitaxel<sup>46</sup>). More model codes will be added into the library.

In this manuscript, we selected two PopPK cases to demonstrate the actual clinical applications of the app. Although adding a PK/PD case would be beneficial, we chose to provide such model in the model library (i.e., neutropenia PK/PD model for paclitaxel), without further expanding the manuscript. The app is capable of describing

paclitaxel PK and neutrophil count profile side-by-side for various dosing regimens. Users can use the app to derive the nadir of neutrophil count profile and compare across regimens.

In gPKPDviz, the derivation of AUC (trapezoidal method) or MAX and MIN (capture from simulation output) are sensitive to the sampling time. One can use this feature to evaluate how different sampling intervals might impact the assessment of these parameters. Whereas increasing sampling intensity can improve accuracy, it will prolong run time and increase the chance of the termination due to memory limit. Alternatively, MAX or MIN can be estimated within the ODE system. However, such estimates could be affected by overshooting which is inherent to the ODE solver. In mrgsolve, the impact of overshoot can be properly managed by reducing the  $h_{max}$ ,<sup>47</sup> which is the trade-off of estimation accuracy and simulation time. Mrgsolve code is provided in the model library to showcase the ODE-based derivation of AUC, MAX (and  $T_{max}$ ), MIN (and  $T_{min}$ ) after each dose, using pertuzumab PopPK model as the template.

Currently, there are no tools available yet to fully automate the model translation process from NONMEM (or other model estimation software) to mrgsolve syntax, although in most cases the translations are straightforward. The \$NMXML and \$NMEXT blocks in mrgsolve allow to import the estimates of \$THETA, \$OMEGA, and \$SIGMA from the NONMEM run files (.xml file and .ext file) into the mrgsolve model,<sup>14</sup> which helps to simplify the translation process and reduce the translation error considerably. Among the code blocks, there are four C++ functions that mrgsolve manages: PREAMBLE, MAIN, ODE, and TABLE. There is a specific calling order for those functions, although they can be specified in any order in the model. Understanding the calling order and calling nature of the C++ functions is the key for proper model translation/development.<sup>13</sup> For instance, PREAMBLE is called only once at the first record of the data set, whereas \$MAIN is called repeatedly as per the data record. For the variables that need to be re-defined within the model, they can be declared either in the \$MAIN block (e.g., time after dose TAD<sup>48</sup>), or the \$PREAMBLE block (e.g., the variables for adaptive simulation<sup>49</sup>), depending on the nature of the variable.

Formal validation of the mrgsolve model is highly recommended before applying it for the decision making. The validation of the mrgsolve model is essential at the PRED level, considering most errors in translation are related to the structure model and the covariate model. The validation at IPRED and DV level is also preferred to verify the random effect model, especially for models with atypical random effects, such as

Box-Cox transformation. The topic regarding mrgsolve model validation was discussed in the mrgsolve blog by the package developer.<sup>30,31</sup> The gPKPDviz app itself was also extensively validated against the mrgsolve simulations outside of the app by R, which, together with the validated mrgsolve model, would ensure the accuracy of the app-generated outputs when being operated properly by the users. We opted for manual validation of the app, utilizing pre-defined case examples to assess its key functionalities. In contrast to automated testing methods, such as shinytest<sup>50</sup> (tests user interface) and testthat<sup>51</sup> (assesses R functionality), manual testing leverages the tester's contextual understanding and domain knowledge. Although manual testing requires less initial setup, it can be more time-consuming, especially for repetitive tasks, and does not offer the consistent reproducibility inherent in automated tests. Given the scope of enhancements to gPKPDviz is expected to be modest, we are inclined to continue with manual validation for the newly incorporated features. Ideally, utilizing a combination of manual and automated testing would ensure the most thorough validation of an app.

GPKPDviz is under continuous development in collaboration with Metrum Research Group. Although gPKPDviz was originally developed for simulating continuous endpoints, it can handle categorical end points well, except that, in the current version, the postprocessing is designed exclusively for continuous endpoints (i.e., exposure metrics derivation and summary statistics).

Additional features under consideration include simulating truncated log-normal distribution with correlation for continuous covariates, providing a last observation carried forward (locf) option for the simulation of time-varying covariates,<sup>13</sup> adding  $T_{max}$  to the summary table, handling longitudinal categorical endpoints in the graphical visualization, and report generation of key results in a certain format.<sup>10</sup> Some exposure metrics that are unique to the infectious diseases, including time above minimum inhibitory concentration (MIC),  $C_{max}/MIC$  ratio, and AUC/MIC ratio,<sup>52,53</sup> are also under consideration for future implementation. The population simulation feature involving repeated simulation with randomly sampled parameter sets<sup>54</sup> is not available in gPKPDviz. Considering the aim of replicated stimulation is mainly for VPCs, we plan to develop a stand-alone Shiny app for VPCs, by leveraging mrgsolve and the VPC packages, and dedicate the use of gPKPDviz for the scenario simulations.

Simulations in mrgsolve outpace NONMEM and native R significantly, particularly for ODE-based models, in part due to its seamless integration with C++. The simulation process in gPKPDviz is nearly as efficient as running mrgsolve in R, yet the most time and memory consuming step it to load the simulated data. For instance, on

our Shiny server, it took 15 s to simulate the pertuzumab PopPK model for 500 subjects with hourly sample for 27 weeks, but it costs around 2 min to load the simulated data with 2 million entries. The suggestion is to choose the sampling interval wisely to reduce the size of simulated data and thus the loading time. Given the overhead from Shiny's web framework and added data loading/processing needs, running mrgsolve within a Shiny app might demand more memory in general than run it directly in R. We recommend deploying gPKPDviz on a robust server to allocate ample session memory, particularly when simulating complex ODE-based model with intensive sampling, like every hour, spanning prolonged durations such as daily dose over a year.

Despite being powered by mrgsolve, there are certain limitations for gPKPDviz that are inherent to the app-based simulations. Besides the previously mentioned high memory demand and limited session memory allocation, there are other limitations. These includes limited flexibility (compared with mrgsolve in R), reliance on the web server, and, most importantly, the difficulty to be QCed because most Shiny apps, including gPKPDviz, lack the ability to output reproducible R code after running the simulations. Shinymeta package provides functionality to capture and export the underlying code of a Shiny app,<sup>55</sup> allowing users to reproduce the Shiny outcomes outside of the app. However, it would be less efficient to implement it at the late stage of the app development. Thus, the current gPKPDviz app is primarily intended for exploratory analyses and internal decision making, and may not be suitable when source code is required, for example, in regulatory submissions. It is at the user's discretion to determine the most appropriate situations for the application.

## ACCESSING THE APPLICATION

GPKPDviz can be accessed as an open-source application on GitHub.<sup>22</sup> To create a stable and reproducible environment, the application is distributed as a docker image that can be loaded in the host computer either through docker desktop or through docker commands in the terminal. Once the docker image is loaded into the host computer, the user can access the application using a local port (by default, it is programmed to be <http://localhost:3838/>).

## ACKNOWLEDGMENTS

The authors would like to thank Stephanie Liu for the collaboration of pertuzumab dosing regimen simulation and the valuable inputs on the manuscript. The authors would like to thank Angelica Quartino for initiating the development of gPKPDviz app, Jenny Nguyen for compiling

mrgsolve models during her internship in 2018, and all the internal evaluators for their assessment and feedback.

## FUNDING INFORMATION

The analysis was sponsored by Genentech, Inc., a member of the Roche group.

## CONFLICT OF INTEREST STATEMENT

T.L., V.P., L.B., J.Y.J., and M.K. are full-time employees of Genentech, Inc., and own Roche stock. E.V. is the Drug Development Training Program (DDTP) fellow in Department of Clinical Pharmacology, Genentech, Inc. E.A. and K.B. are employees of Metrum Research Group.

## ORCID

Tong Lu  <https://orcid.org/0000-0001-5899-6857>

Victor Poon  <https://orcid.org/0000-0002-4215-2198>

Logan Brooks  <https://orcid.org/0000-0001-9215-408X>

Erick Velasquez  <https://orcid.org/0000-0002-0874-4317>

Eric Anderson  <https://orcid.org/0009-0003-8191-2879>

Kyle Baron  <https://orcid.org/0000-0001-7252-5656>

Jin Y. Jin  <https://orcid.org/0000-0002-3627-0323>

Matts Kågedal  <https://orcid.org/0000-0002-1261-0354>

## REFERENCES

- Hosseini I, Gajjala A, Bumbaca Yadav D, et al. gPKPDSim: a SimBiology((R))-based GUI application for PKPD modeling in drug development. *J Pharmacokinet Pharmacodyn*. 2018;45:259-275.
- Germani M, Del Bene F, Rocchetti M, Van Der Graaf PH. A4S: a user-friendly graphical tool for pharmacokinetic and pharmacodynamic (PK/PD) simulation. *Comput Methods Programs Biomed*. 2013;110:203-214.
- Biliouris K, Lavielle M, Trame MN. MatVPC: a user-friendly MATLAB-based tool for the simulation and evaluation of systems pharmacology models. *CPT Pharmacometrics Syst Pharmacol*. 2015;4:547-557.
- Mavroudis PD, Ayyar VS, Jusko WJ. ATLASmPBPK: a MATLAB-based tool for modeling and simulation of minimal physiologically based Pharmacokinetic models. *CPT Pharmacometrics Syst Pharmacol*. 2019;8:557-566.
- Simulx. [cited 2023 July 23]. <https://simulx.lixoft.com/>
- Krause A, Lowe PJ. Visualization and communication of pharmacometric models with Berkeley madonna. *CPT Pharmacometrics Syst Pharmacol*. 2014;3:e116.
- Marcoline FV, Furth J, Nayak S, Grabe M, Macey RI. Berkeley Madonna version 10-a simulation package for solving mathematical models. *CPT Pharmacometrics Syst Pharmacol*. 2022;11:290-301.
- Shiny package. [cited 2023 July 23]. <https://cran.r-project.org/web/packages/shiny/>
- Vaddady P, Kandala B. ModVizPop: a shiny interface for empowering teams to perform interactive pharmacokinetic/pharmacodynamic simulations. *CPT Pharmacometrics Syst Pharmacol*. 2021;10:1323-1331.



10. Wojciechowski J, Hopkins AM, Upton RN. Interactive Pharmacometric applications using R and the shiny package. *CPT Pharmacometrics Syst Pharmacol*. 2015;4:e00021.
11. Nolain P et al. PopkinR: a suite of Shiny applications focused on the pharmacometrics workflow. 2018 [cited 2023 July 23]. [https://www.page-meeting.org/pdf\\_assets/7336-PAGE%20Poster%20PopkinR%20-%20Final%20version.pdf](https://www.page-meeting.org/pdf_assets/7336-PAGE%20Poster%20PopkinR%20-%20Final%20version.pdf)
12. Liu Q. REAP - R-Shiny Exploratory Analysis Platform in Clinical Pharmacology. 2018 [cited 2023 July 23]. [https://rinpharma.com/publication/rinpharma\\_26/](https://rinpharma.com/publication/rinpharma_26/)
13. Baron K. mrgsolve user guide. [cited 2023 July 23]. <https://mrgsolve.org/user-guide/>
14. Elmokadem A, Riggs MM, Baron KT. Quantitative systems pharmacology and physiologically-based pharmacokinetic modeling with mrgsolve: a hands-on tutorial. *CPT Pharmacometrics Syst Pharmacol*. 2019;8:883-893.
15. PKPDsim package. [cited 2023 July 23]. <https://github.com/InsightRX/PKPDsim>
16. Keizer R, Pastoor D, Savic R. New open source R libraries for simulation and visualization: “PKPDsim” and “vpc”. 2015 [cited 2023 July 23]. <https://www.page-meeting.org/?abstract=3636>
17. Wang W, Hallow KM, James DA. A tutorial on RxODE: simulating differential equation Pharmacometric models in R. *CPT Pharmacometrics Syst Pharmacol*. 2016;5:3-10.
18. rxode2 package. [cited 2023 July 23]. <https://github.com/nlmixr2/rxode2>
19. RxODE package. [cited 2023 July 23]. <https://nlmixrdevelopment.github.io/RxODE/>
20. desolve package. [cited 2023 July 23]. <https://cran.r-project.org/web/packages/deSolve/>
21. diffeqr package. [cited 2023 July 23]. <https://cran.r-project.org/web/packages/diffeqr/>
22. gPKPDviz. [cited 2023 Aug 14]. <https://github.com/Genentech/gPKPDviz/>
23. Interactive-appsdecision-making (Metrum Research Group). [cited 2023 July 23]. <https://www.metrumrg.com/interactive-apps-decision-making/>
24. Nima Chamyani HL, Centanni M. PKPD Simulation Toolkit. 2023 [cited 2023 July 23]. <https://nima-ch.shinyapps.io/PKPD/>
25. Kawuma AN, Benecke RM, Mouksassi S, Pillai G, Chirehwa MT. R Based Shiny App To Simulate PKPD Profiles Using TMDD Models. [cited 2023 July 23]. [https://www.page-meeting.org/pdf\\_assets/10697-abstract.pdf](https://www.page-meeting.org/pdf_assets/10697-abstract.pdf)
26. Nicolas Luyckx AL, van Schaick E, Hénin E, Blaizot S, Noguine P-A, Laveille C. e-Campsis: A Shiny PK/PD model simulator based on CAMPSIS. 2023 [cited 2023 July 23]. [https://www.page-meeting.org/pdf\\_assets/10493-abstract.pdf](https://www.page-meeting.org/pdf_assets/10493-abstract.pdf)
27. e-Campsis. [cited 2023 Aug 14]. <https://calvagone.github.io/ecampsis>
28. Campsis package. [cited 2023 July 23]. <https://github.com/Calvagone/campsis>
29. Campsismod package. [cited 2023 July 23]. <https://github.com/Calvagone/campsismod>
30. Baron K. Validation model translation from NONMEM. 2022 [cited 2023 July 23]. <https://mrgsolve.org/blog/posts/2022-05-validate-translation/>
31. Baron K. Update: Validate translation from NONMEM. 2023 [cited 2023 July 23]. <https://mrgsolve.org/blog/posts/2023-update-validation.html>
32. Baron K. About the lsoda differential equation solver used by mrgsolve. 2023. [https://mrgsolve.org/docs/reference/about\\_solver.html](https://mrgsolve.org/docs/reference/about_solver.html)
33. Pastoor D. PKPDmisc. 2020 [cited 2023 Oct 20]. <https://metrumresearchgroup.github.io/PKPDmisc/>
34. Liu SN, Lu T, Jin JY, et al. Impact of dose delays and alternative dosing regimens on Pertuzumab pharmacokinetics. *J Clin Pharmacol*. 2021;61:1096-1105.
35. Perjeta (pertuzumab)-highlights of prescribing information. [cited 2023 May 3rd]. [https://www.accessdata.fda.gov/drugsatfda\\_docs/label/2012/1254091bl.pdf](https://www.accessdata.fda.gov/drugsatfda_docs/label/2012/1254091bl.pdf)
36. Garg A, Quartino A, Li J, et al. Population pharmacokinetic and covariate analysis of pertuzumab, a HER2-targeted monoclonal antibody, and evaluation of a fixed, non-weight-based dose in patients with a variety of solid tumors. *Cancer Chemother Pharmacol*. 2014;74:819-829.
37. Malik M et al. Dose response studies of recombinant humanized monoclonal antibody 2C4 in tumor xenograft models. *Proc Am Assoc Cancer Res*. 2003;44:150.
38. Quartino AL, Li H, Jin JY, et al. Pharmacokinetic and exposure-response analyses of pertuzumab in combination with trastuzumab and docetaxel during neoadjuvant treatment of HER2+ early breast cancer. *Cancer Chemother Pharmacol*. 2017;79:353-361.
39. Park WS. Pharmacometric models simulation using NONMEM, Berkeley Madonna and R. *Transl Clin Pharmacol*. 2017;25:125-133.
40. mrglove gallery. [cited 2023 July 23]. <https://github.com/mrgsolve/gallery>
41. Sehn LH, Herrera AF, Flowers CR, et al. Polatuzumab Vedotin in relapsed or refractory diffuse large B-cell lymphoma. *J Clin Oncol*. 2020;38:155-165.
42. Tilly H, Morschhauser F, Sehn LH, et al. Polatuzumab Vedotin in previously untreated diffuse large B-cell lymphoma. *N Engl J Med*. 2022;386:351-363.
43. Lu D, Lu T, Gibiansky L, et al. Integrated two-analyte population pharmacokinetic model of Polatuzumab Vedotin in patients with non-Hodgkin lymphoma. *CPT Pharmacometrics Syst Pharmacol*. 2020;9:48-59.
44. Lu D, Lu T, Shi R, et al. Application of a two-analyte integrated population pharmacokinetic model to evaluate the impact of intrinsic and extrinsic factors on the pharmacokinetics of Polatuzumab Vedotin in patients with non-Hodgkin lymphoma. *Pharm Res*. 2020;37:252.
45. Holford N, Karlsson M. Model Evaluation. Visual Predictive Checks. 2008 [cited 2023 July 23]. [https://www.page-meeting.org/pdf\\_assets/8694-Karlsson\\_Holford\\_VPC\\_Tutorial\\_hires.pdf](https://www.page-meeting.org/pdf_assets/8694-Karlsson_Holford_VPC_Tutorial_hires.pdf)
46. Friberg LE, Henningson A, Maas H, Nguyen L, Karlsson MO. Model of chemotherapy-induced myelosuppression with parameter consistency across drugs. *J Clin Oncol*. 2002;20:4713-4721.
47. Baron K. DLSODA Overshoot. 2018 [cited 2023 Oct 20]. <https://github.com/metrumresearchgroup/mrgsolve/issues/369>
48. K. B. Time after dose. 2023 <https://mrgsolve.org/docs/articles/extra/time-after-dose.html>
49. K. B. adaptive simulation. 2019 [https://github.com/mrgsolve/gallery/blob/master/adaptive/adaptive\\_simple.md](https://github.com/mrgsolve/gallery/blob/master/adaptive/adaptive_simple.md)
50. Chang WC, Wickham H. shinytest: Test Shiny Apps. 2023 [cited 2023 Oct 20]. <https://github.com/rstudio/shinytest>
51. Wickham H. Testthat: get started with testing. *R J*. 2011;3:5-10.
52. Lepak AJ, Andes DR. Antifungal pharmacokinetics and pharmacodynamics. *Cold Spring Harb Perspect Med*. 2014;5:a019653.

53. Pereira LC, Fátima MA, Santos VV, Brandão CM, Alves IA, Azeredo FJ. Pharmacokinetic/pharmacodynamic modeling and application in antibacterial and antifungal pharmacotherapy: a narrative review. *Antibiotics*. 2022;11:986-1105.
54. Mould DR, Upton RN. Basic concepts in population modeling, simulation, and model-based drug development. *CPT Pharmacometrics Syst Pharmacol*. 2012;1:e6.
55. Cheng JSC. shinymeta 0.2.0.3. 2021 [cited 2023 Oct 20]. <https://rstudio.github.io/shinymeta/>

**How to cite this article:** Lu T, Poon V, Brooks L, et al. gPKPDviz: A flexible R shiny tool for pharmacokinetic/pharmacodynamic simulations using mrgsolve. *CPT Pharmacometrics Syst Pharmacol*. 2024;13:341-358. doi:[10.1002/psp4.13096](https://doi.org/10.1002/psp4.13096)

## SUPPORTING INFORMATION

Additional supporting information can be found online in the Supporting Information section at the end of this article.