

nlmeVPC: Visual Model Diagnosis for the Nonlinear Mixed Effect Model

by Eun-Hwa Kang, Myungji Ko and Eun-Kyung Lee

Abstract A nonlinear mixed effect model is helpful when data are repeatedly measured in the same unit or correlated between units. Such models are widely used in medicine, disease mechanics, pharmacology, ecology, social science, psychology, etc. After fitting the nonlinear mixed effect model, model diagnostics are essential for obtaining reliable results in the model fitting process. The visual predictive check (VPC) has recently been highlighted as a visual diagnostic tool for pharmacometric models. This method can also apply to general nonlinear mixed effect models. However, functions for VPCs in existing R packages are specialized in pharmacometric model diagnosis, and it is not easy to use to diagnose general nonlinear mixed effect models. In this paper, we propose nlmeVPC, an R package for the visual diagnosis of various nonlinear mixed effect models. The nlmeVPC package allows for more diverse model diagnostics, including visual diagnostic tools that extend the concept of VPCs along with the capabilities of existing R packages.



Introduction

After fitting a model, diagnosing the fitted model is essential to obtain reliable results (Nguyen et al., 2017). For linear models, the residuals are usually used to determine the goodness of fit of the fitted model. However, it is not easy to diagnose the fitted model using residuals for nonlinear mixed-effect models, mainly due to random effects as well as nonlinearity. Therefore, various diagnostic tools for this type of model have been developed. The nonlinear mixed effect model is advantageous when data are repeatedly measured in the same unit, or the relationship between the dependent and independent variables is nonlinear. It is widely used in various fields, including medicine, disease mechanics, pharmacology, ecology, pharmacometrics, social science, and psychology (Pinheiro and Bates, 2000; Davidian, 2017). Recently, among the various diagnostic tools applicable to nonlinear mixed models, simulation-based diagnostic methods have been developed in the field of pharmacology (Karlsson and Savic, 2007). The visual predictive check (VPC; Karlsson and Holford, 2008) is the critical diagnostic tool that visually demonstrates the adequacy of the fitted model. It allows the diagnosis of fixed and random effects in mixed models (Karlsson and Holford, 2008) in the original data space. Currently, it is a widely used method for diagnosing population pharmacometric models: Heus et al. (2022) used the VPC method to evaluate their population pharmacokinetic models of vancomycin, Mudde et al. (2022) checked their final population PK model for each regimen of antituberculosis drug using the VPC method, and Otto et al. (2021) compares the predictive performance of parent-metabolite models of (S)-ketamine with the VPC method. This VPC method can be used for general nonlinear mixed effect models, including hierarchical models.

The **psn** (Lindbom et al., 2004) package and **xpose4** (Keizer et al., 2013) package provide various diagnostic methods for pharmacometric models, including the VPC plot. These packages are developed only for the pharmacometricians, and the NONMEM software (Bauer, 2011) should be used for fitting the model and generating inputs of functions in these two packages. However, NONMEM is licensed software, and it is mainly focused on the analysis of population pharmacometric models. Therefore, it is not easy for nonpharmacometrician to use NONMEM and to draw the VPC plot through **psn** and **xpose4**. Recently, the **vpc** (Keizer, 2021) package and **nlmixr2** (Fidler et al., 2019) package have been developed to draw VPC plots in R without results from NONMEM. However, **vpc** package provides the function for the original VPC plot only. **nlmixr2** is developed initially for fitting general dynamic models, focusing on the fitting process. To check the fitted model, **nlmixr2** provides a function to use graphical diagnostics in **xpose4** with the **nlmixr2** model object. Also, **nlmixr2** uses the VPC plot in the **vpc** package with the **nlmixr2** model object. Therefore, both packages only provide a function to draw the basic VPC plot, and the other newly developed simulation-based methods, including extensions of VPC, are not provided.

In this paper, we developed a new R package, **nlmeVPC**, to provide various visual checking methods for the nonlinear mixed models in one platform. This package includes various state-of-the-art model diagnostic methods based on the visual comparison between the original and simulated data. Most methods compare the statistics calculated from the observed data to the statistics from the simulated data. The percentiles, for example, the 10th, 50th, and 90th percentiles, are widely used statistics. We compare the similarities between the statistics from the observed data and those from the simulated data in two different spaces: the data space and the model space (Wickham et al., 2015). The original data construct the data space. Usually, the data space is represented by the independent and

dependent variables, such as time and concentration, in the pharmacokinetic data. On the other hand, the model space is represented by the data obtained from the fitted model, for example, residuals and summary values from the fitted model. From this viewpoint, we categorize the well-known visual diagnostic tools into two approaches. One method is compared in the original data space, and the other is in the model space. In the method with the data space, we developed functions for the original VPC (VPCgraph), the additive quantile regression VPC (aqrVPC), and the bootstrap VPC (bootVPC). In addition, we proposed a new VPC method: the average shifted VPC (asVPC). In the method with the model space, the coverage plot (coverageplot) and the quantified VPC (quantVPC) are included. For a more detailed diagnosis, we developed a detailed coverage plot (coverageDetailplot). In nlmeVPC, the ggplot2 package (Wickham, 2016) is used to create all plots.

Nonlinear mixed effect model

The general nonlinear mixed effect model is defined as follow:

$$\begin{aligned} y_{ij} &= f(x_{ij}, \theta, \eta_i, \epsilon_{ij}) \\ \eta_i &\sim N(0, \Omega) \\ \epsilon_{ij} &\sim N(0, \Sigma) \end{aligned} \quad (1)$$

where y_{ij} is the dependent variable of the j^{th} observation of the i^{th} individual, x_{ij} is the independent variable, f is the nonlinear function of x_{ij} , θ is the population parameter, η_i represents the variability of the individual i , and ϵ_{ij} represents the random error. From the data, θ , Ω , and Σ are estimated. For the simulated data, the fitted model $y_{ij} = f(x_{ij}, \hat{\theta}, \eta_i, \epsilon_{ij})$, $\eta_i \sim N(0, \hat{\Omega})$, $\epsilon_{ij} \sim N(0, \hat{\Sigma})$ are used.

Validation in the data space

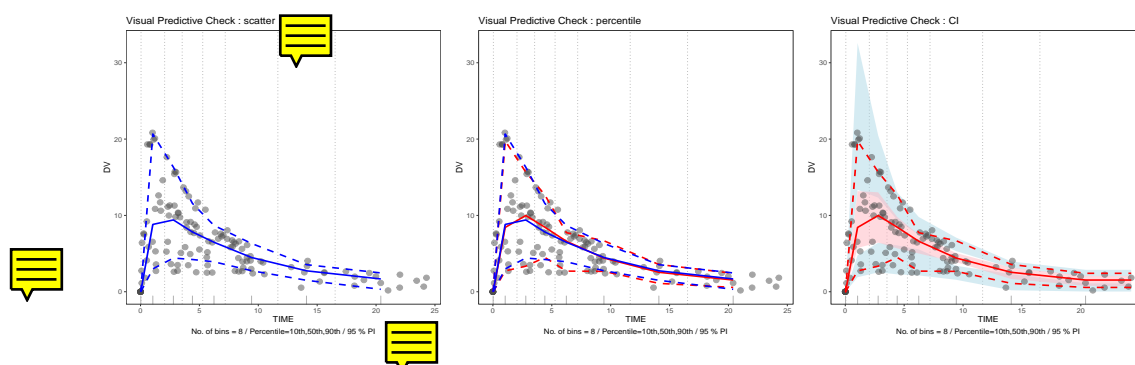
The VPC is the most popular model validation method in the pharmacometrics area. It was developed to diagnose population pharmacokinetic/pharmacodynamic models visually. The main idea of the VPC is the comparison between the distribution of observations and the distribution of predictions, where the distribution of predictions is obtained from simulated data drawn from the fitted model. If the fitted model represents the observed data well, these two distributions tend to be similar. Both distributions can be represented in the original data space that consists X axis as the independent variable and Y axis as the dependent variable. It allows us to compare the observed data with the fitted model in the original data space. In nlmeVPC, we include an original VPC plot, an additive quantile regression VPC (Jansen et al., 2018), and a bootstrap VPC (Post et al., 2008). We also proposed a new approach to draw the VPC: the average shifted VPC.

Visual Predictive Check

The visual predictive check (VPC; Karlsson and Holford, 2008) is based on the principle that if the fitted model adequately describes the observed data, the distribution of the simulated data from the fitted model should be similar to the distribution of the observed data. There are several ways to compare the similarities between the distributions. In the VPC approach, profiles of quantiles are used. Two profiles are mainly used to compare the distributions of observations and predictions. One profile is from the upper bound of the prediction intervals, and the other is from the lower bound. These prediction intervals are calculated from the simulated data. 90% prediction intervals are usually used. For the small and sparse samples, 80% prediction interval is also used. The lower and upper bounds of 80% prediction interval are the 10th and 90th percentiles of the simulated data. The left side of Figure 1 shows the "scatter" type of the VPC plot. Dots indicate the observed data. Two dashed blue lines represent profiles of the 10th and 90th percentiles of the simulated data, and the solid blue line represents profiles of the 50th percentile. If the fitted model represents the observed data well, most observed data are between profiles of 10th and 90th percentiles.

The middle of Figure 1 is the "percentile" type of the VPC plot. In this plot, profiles of percentiles from the observed data are compared to profiles of percentiles from the simulated data. Two dashed red lines represent profiles of the 10th and 90th percentiles of the observed data, and the solid red line represents profiles of the 50th percentile of the observed data. If the fitted model represents the observed data well, two profiles in each percentile - one from the original data and the other from the simulated data - are similar. The right of Figure 1 is the "CI" type of the VPC plot. Sky blue areas represent the 95% confidence areas of the 10th and 90th percentiles, and pink areas represent the 95% confidence areas of the 50th percentile. These confidence areas were calculated from the simulated

Figure 1: The visual predictive check plot. The solid line represents the 50th percentile of the observed data, and dashed lines represent the 10th and 90th percentiles of the observed data. Sky blue and pink areas represent the 95% confidence areas of the 10th, 50th and 90th percentile lines.



data. After calculating percentiles in each simulated data, we find 95% confidence intervals for each percentile and represent the areas. In this plot, it is necessary to verify that the profiles of the original data are in confidence areas of each profile from the simulated data in each percentile. If each percentile line of the observed data is in the corresponding confidence area, this can be evidence that the fitted model represents the observed data quite well. Otherwise, the fitted model needs to be improved. The "CI" type of the VPC plot is the most widely used type in pharmacometrics.

The percentiles of the dependent variable are calculated in each bin. Because the main reason to implement bins is the lack of data for quantile calculation, we need to maintain enough data in each bin to estimate the corresponding percentiles effectively. No binning means that the number of bins is equal to the number of different independent variable values. In this case, the VPC plot shows all details of the relationship between the independent and dependent variables. However, the results become too irregular to characterize. As the number of bins decreases, the irregularity is lessened, and the detailed information is also reduced. Therefore, the selection of the best number of bins is crucial. If we increase the number of bins to obtain a more refined VPC plot, the variability in the resulting graphics can significantly increase because we have only a few data points in each bin. Decreasing the number of bins to obtain reasonable sample sizes produces quantiles that are too rough. The way to configure the bins is also essential. Lavielle and Bleakley (2011) proposed a procedure to find the optimal bin configuration automatically, including the optimal number of bins and the optimal bin cutoffs. VPCgraph provides the automatic binning with optK and makeCOVbin; here, optK finds the optimal number of bins, and makeCOVbin finds the optimal cutoffs of bins using Lavielle and Bleakley's method.

Additive quantile regression VPC

To overcome the difficulties of making bins as well as determining the number of bins, Jansen et al. (2018) used additive quantile regression to calculate the quantiles of the observed and simulated data. This regression method makes it possible to diagnose models without binning, which is especially useful when the data are insufficient, irregular, or inappropriate to configure the bins. To fit the additive quantile regression, we used the rqss function in the quantreg package and developed the aqrVPC function to draw the VPC type plot with additive quantile regression. The right side of Figure 2 shows the additive quantile regression VPC plot. The solid and dashed lines represent the 10th, 50th, and 90th additive quantile regression lines of the observed data, and the pink and sky blue areas represent the confidence areas of the additive quantile regression lines of the simulated data. Lines and areas in the additive quantile regression VPC plot are much smoother than those in the original VPC plot.

Bootstrap VPC

The bootstrap VPC (Post et al., 2008) compares the distribution of the simulated data to the distribution of the bootstrap samples that draw from the observed data. This plot reflects the uncertainty of the observed data and allows for more objective comparisons with the predicted median.

Figure 3 shows the bootstrap VPC plot using bootVPC. The solid and dashed blue lines represent the 10th, 50th, and 90th percentiles of the simulated data. The solid red line represents the 50th percentile line, and the pink areas represent the 95% confidence areas of the 50th percentile line, calculated from the bootstrap samples of the observed data. If the solid blue line and the solid red line are similar, the

Figure 2: The visual predictive check plot and Dots indicate the observed data. The solid line represents the 50th percentile of the observed data, and dashed lines represent the 10th and 90th percentiles of the observed data. Sky blue and pink areas represent the 95% confidence areas of the 10th, 50th and 90th percentile lines.

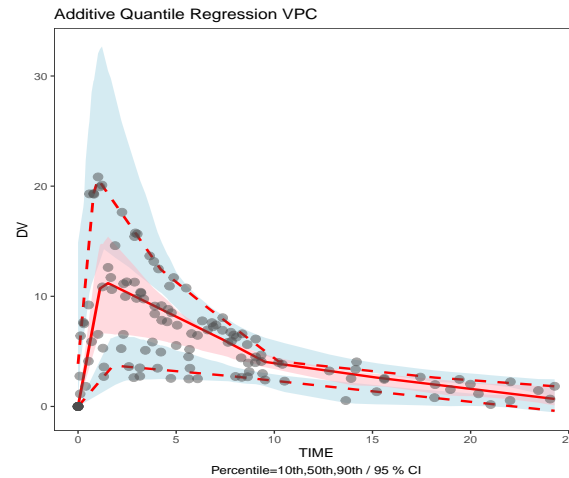
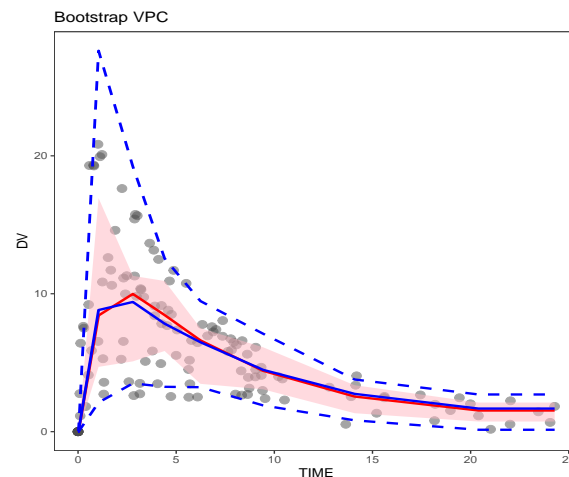


Figure 3: The bootstrap VPC plot. Dots indicate the observed data. The solid and dashed blue lines represent the 10th, 50th, and 90th percentiles of the simulated data. The solid red line represents the 50th percentile line, and the pink areas represent the 95% confidence areas of the 50th percentile line, calculated from the bootstrap samples of the observed data.



solid blue line is in the pink area, and the pink area is located between two dashed blue lines, then this is evidence that the fitted model explains the data well.

Average shifted VPC

Even though binning solves the most irregularity problem, the VPC plot still has a precision problem with sparse data. In this paper, we propose a new approach to draw the adapted VPC plot from the average shifted histograms (Scott, 1985). A histogram is a widely used method to display the density of a single continuous variable. However, histograms can look quite different with different bin widths. This requires computing an optimal bin width. To overcome the problem with the choice of bin width and to obtain smoother estimates, Scott (1985) proposed the averaged shifted histogram (ASH). The ASH idea is to make K different histograms with different starting points and to combine them with the weighted average. K is a predetermined fixed number. For each histogram, the starting point is shifted by d/K , where d is the width of the bin. We extend this idea to the VPC graph and propose the average shifted visual predictive check (asVPC) plot.

The binning of the VPC varies from the traditional binning of a histogram. It is usually determined with a fixed number of observations instead of a fixed bin width. To apply the ASH idea to the VPC, we divide our data into $K * B$ bins along with the independent variable, where each bin has a different

width but the same number of observations. Here, B is the number of the original bins in the VPC plot, and K is the number of the shifted bins averaged in the asVPC plot.

To draw the VPC plots, the following information is needed from each bin:

- (A) the median of the independent variable and percentiles of the dependent variable of the observed data.
- (B) the median of the independent variable and percentiles of the dependent variable of the simulated data.
- (C) 95% confidence interval of percentiles (usually 10^{th} , 50^{th} , and 90^{th} percentiles) of the dependent variable, calculated from the simulated data.

These values are also needed to produce confidence areas and percentile lines in the asVPC plot. Furthermore, additional steps are needed for asVPC. To find the median of the independent variable and percentiles of the dependent variable using the ASH algorithm, the following procedures are needed:

1. Divide the independent variable into $N = K * B$ bins.
2. For $i = 1, \dots, N$,
 - (a) If $i < K$, combine $bin_1, \dots, bin_{i+K-1}$
 else if $K \leq i \leq N - K$, combine $bin_{i-K+1}, \dots, bin_{i+K-1}$
 else if $N - K < i$, combine $bin_{i-K+1}, \dots, bin_N$
 - (b) Calculate the median of the independent variable and the weighted percentiles of the dependent variable in the combined bin
3. Collect the medians of the independent variable and the weighted percentiles of the dependent variable from 2, and connect them to the lines.

We can implement (A) and (B) by applying these procedures separately to the observed and simulated data. Additionally, (C) can be implemented using these procedures for each simulated dataset. First, we find the weighted percentiles, combine the results from each simulated dataset, and then calculate the 95% confidence intervals of each percentile. Using these three quantities (A), (B), and (C), we can draw the VPC type plot with the ASH approach, producing the asVPC plot.

Determining the weights

In the asVPC plot, the observations in each bin are combined using weights. Typically the data near the center of the integrated bin have higher weights, and the data far from the center have smaller weights. This idea is used in the ASH algorithm as well as the density estimation literature. We suggest two different ways to apply weights for the asVPC calculation.

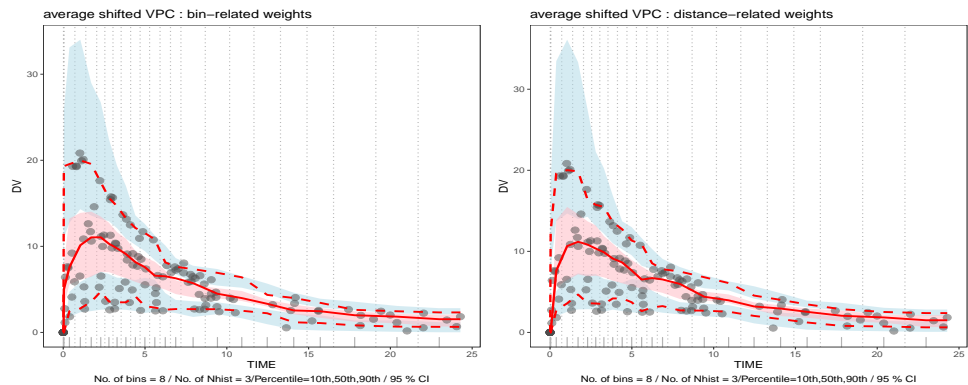
- Bin-related weight: Use $\frac{1}{K}, \dots, \frac{i-1}{K}, \dots, \frac{K-1}{K}, 1, \frac{K-1}{K}, \dots, \frac{i-1}{K}, \dots, \frac{1}{K}$ for each bin in the combined bin. This approach is the same as the original ASH approach.
- Distance-related weight: Use the reciprocal of the distance from the center of the independent variable in each combined bin so that the points near the center have higher weights and the points far from the center have lower weights.

Figure 4 shows the results from the asVPC function using bin-related weights and distance-related weights. The solid and dashed lines represent the average shifted quantile lines of the observed data, and the pink and sky blue areas represent the confidence areas of the simulated data. The lines in the asVPC plot are smoother than those in the original VPC plot, and the confidence areas in the asVPC plot are thinner than those in the original VPC plot.

Validation in the model space

To validate the fitted model in the model space, we need to define the proper statistics to represent it, calculate the statistics, and describe them in the model space. In **nlmeVPC**, we use the same statistics, quantiles, as the VPC plot in the data space, and compare them in two ways - numerically and visually. The numerical predictive check and the coverage plot are commonly used methods in pharmacometrics. However, there is a limitation to detecting the illness of the fitted model. These methods combine the results of all ranges of the independent variable, and it is helpful to see the overall fitness. However, it is not easy to detect the detailed discrepancy between the fitted model and the observed data. To overcome this limitation, we developed a new plot, the detailed coverage plot, to compensate for the shortness of the coverage plot.

Figure 4: The average shifted VPC plot. Dots indicate the observed data. The solid line represents the 50th quantiles of the observed data, and dashed lines represent the 10th and 90th percentiles of the observed data. Sky blue and pink areas represent the 95% confidence areas of the 10th, 50th, and 90th percentiles.



Numerical predictive check

Based on the simulated data, the predicted percentiles are computed. The VPC plot visually compares the observed data to the simulated percentiles in the data space. On the other hand, the numerical predictive check (NPC; Wang and Zhang, 2012; Karlsson and Savic, 2007) numerically compares the observed data to the simulated data. For a given level of prediction (for example, 90%), the predicted interval is calculated using the simulated data, and the number of observed data below the predicted interval is counted. The expected number of points below the predicted interval (for example, 5% of observations) is also calculated and compared to the observed number. If these two numbers are similar, the suggested model is suitable (Maharaj et al., 2019). The same procedures are performed for the number of observations above the predicted interval.

NumericalCheck provides information summarizing the results of the NPC for various prediction levels. PI is the prediction level $\times 100$ and n is the number of observations. “Expected points below PI” and “Expected points above PI” are respectively the expected numbers below and above the PI; “points below PI” and “points above PI” respectively represent the numbers of points below and above the PI; “95%CIBelowFrom(%)” and “95%CIBelowTo(%)” represent the 95% confidence interval of “points below PI(%)”; and “95%CIAboveFrom(%)” and “95%CIAboveTo(%)” represent the 95% confidence interval of “points above PI(%)”. If “points below PI(%)” is in the 95% confidence intervals of “points below PI(%)” and is similar to “Expected points below PI”, and if “points above PI(%)” is in the 95% confidence intervals of “points above PI(%)” and is similar to “Expected points above PI”, this is the evidence that the fitted model explains the data well.

```
> NumericalCheck(origdata,simdata,pred.level=c(0,0.2,0.4,0.6,0.8,0.9),N_xbin=8)$NPC
```

	PI	n	Expected points below PI	points below PI	points below PI(%)
1	0	132	66.0	57	43.18182
2	20	132	52.8	46	34.84848
3	40	132	39.6	37	28.03030
4	60	132	26.4	27	20.45455
5	80	132	13.2	17	12.87879
6	90	132	6.6	7	5.30303
	95%CIBelowFrom(%)	95%CIBelowTo(%)	Expected points above PI	points above PI	
1	35.6060606	56.81818	66.0	63	
2	26.8750000	48.88258	52.8	49	
3	20.4545455	36.76136	39.6	35	
4	11.3636364	26.15530	26.4	28	
5	3.7878788	16.66667	13.2	12	
6	0.3598485	10.64394	6.6	5	
	points above PI(%)	95%CIAboveFrom(%)	95%CIAboveTo(%)		
1	47.727273	34.090909	55.303030		
2	37.121212	25.359848	46.609848		
3	26.515152	16.666667	36.003788		
4	21.212121	10.965909	25.000000		
5	9.090909	2.632576	15.549242		
6	3.787879	1.117424	8.333333		

Coverage plot



Various values are calculated for comparison in the NPC. However, it is too complicated to check the validity of the fitted model with these many numbers. The coverage plot (Karlsson and Holford, 2008) is developed to help visually check the fitted model with the NPC result. In each level of the predicted interval, the ratios between the expected number of points (Exp) outside the prediction interval and the observed number of data (Obs) outside the prediction interval are calculated. These ratios on the upper and lower sides of the prediction interval are calculated separately. For example, when the prediction level is 90, a 90% prediction interval is used, and 10% of the observations are expected to locate outside this prediction interval. To be more precise, 5% of observations are expected to be above the upper limit, and 5% of observations are expected to be below the lower limit.

The coverage plot with the NPC result can diagnose a model using multiple prediction intervals. The X-axis represents the prediction level, and the Y-axis represents Obs/Exp. The closer Obs/Exp is to 1, the more appropriate the model is. Furthermore, the confidence intervals of Obs/Exp values are obtained with the nonparametric method using simulated data and then expressed together in the plot for a more objective comparison. This plot can provide more information than the VPC plot, which interprets only a couple of quantiles - usually the 10%, 50%, and 90% percentiles.

The `xpose4` package (Keizer et al., 2013) provides a coverage plot. However, to draw the coverage plot using `xpose4`, PsN (Lindbom et al., 2004) software is needed to calculate the NPC result. Therefore, we developed `coverageplot` to draw the coverage plot using the results from `NumericalCheck`.

The left side of Figure 5 shows the coverage plot. The X-axis shows the level of the prediction interval. The Y-axis shows the ratio between the observed number of data and the expected number of data of the lower and upper parts in each level of the prediction interval. The white line is the reference line, and the gray area represents the confidence areas of the ratios in each prediction level. If the solid lines are near the white line or in the gray area, we can conclude that the suggested model is suitable.

Plot of the detailed coverage

Unlike the VPC plot, which represents the data space, the information in the observed data space does not come together in the coverage plot, which makes it difficult to determine whether the model is overestimated, underestimated, or adequate in the specific region of the data space. To overcome this limitation of the coverage plot, we propose a new method called the detailed coverage plot. The percentages of observations above the prediction interval are calculated in each bin of the independent variable. Additionally, the percentages of observations below the prediction interval are calculated. The white dots in the plot represent the expected percentages. If the percentages of upper and lower observations are near the white dots, we can conclude that the suggested model is suitable for the specific prediction interval.

The right side of Figure 5 is the result of `coverageDetailedplot` when the prediction level is 80%. The white dots represent the expected percentages of the lower and upper the prediction intervals, 10%, and 90%, respectively. The upper and lower percentages of observation in each time bin are shown in darker gray. The left bin (before 0.045 hours) shows all light gray in the coverage detailed plot, and it is quite different patterns from the expected one. However, it is mainly due to the characteristics of this example data. All observations in this bin are 0. It makes the lower and upper bound of the prediction interval all 0, and the lower and upper percentages become 0.

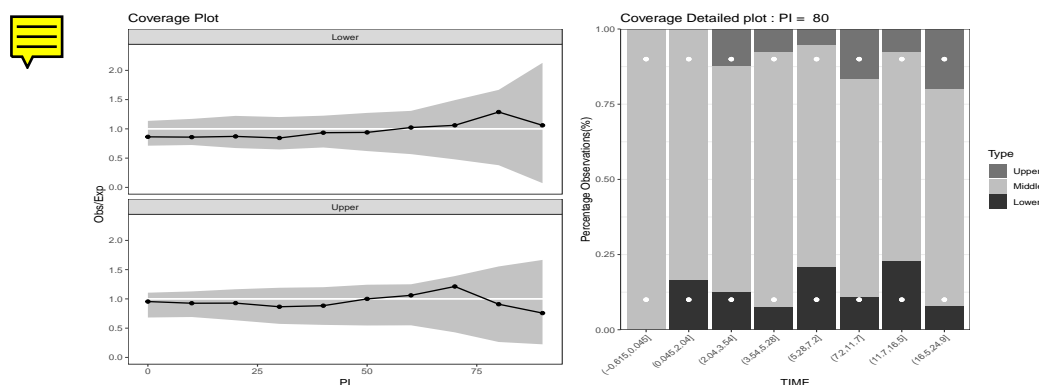
Quantified VPC

Post et al. (2008) proposed the quantified VPC (QVPC). It expanded the existing VPC, including information about missing data. The QVPC plot visually represents actual and unavailable observations around the predicted medians through the form of percent, regardless of the observed data's density or shape. Here, "unavailable observations" refer to all kinds of missing values and unobserved data that occur for various reasons, including deviations and unexpected omissions. The median is a value in the middle of the range of observations.

If the model is appropriate, observations at each time bin are allocated randomly around the predicted median of the model. In the QVPC plot, white dots represent the model's predicted median in each bin. If the borderlines above and below are close to the white dots, we can conclude that the fitted model describes the observed data well.

Figure 6 shows the result of `quantVPC`. The darker gray areas represent the percentages below the median. The lighter gray areas represent the percentage above. The brightest gray areas represent the percent unavailable in each time bin. The white dots represent the ideal location where the above and

Figure 5: The coverage plot and the coverage detailed plot for the 80% prediction interval. In the coverage plot, the X-axis is the level of the prediction interval. The Y-axis is the ratio between the number of observed data and the number of expected data of the lower and upper parts in each level of the prediction interval. The white line is the reference line, and the gray area represents the confidence area of the ratios. If the solid lines are near the white line, we can conclude that the suggested model is suitable. In the detailed coverage plot, the white dots represent the expected percentages of lower and upper prediction intervals of 10%, and 90%, respectively. The upper and lower percentages of observation in each time bin are darker gray.



Function	Description
VPCgraph	draw the original VPC plot
aqrVPC	draw the additive quantile regression VPC plot
bootVPC	draw the bootstrap VPC plot
asVPC	draw VPC using ASH method
NumericalCheck	calculate the numbers to check coverage in each prediction level
coverageplot	plot the result of NumericalCheck
coverageDetailplot	plot for checking specific prediction level
quantVPC	plot for the quantified VPC

Table 1: List of functions to check the validity of model with the observed data and the simulated data

the below percentages meet. In this example, there is no missing value.

The nlmeVPC package: structure and functionality

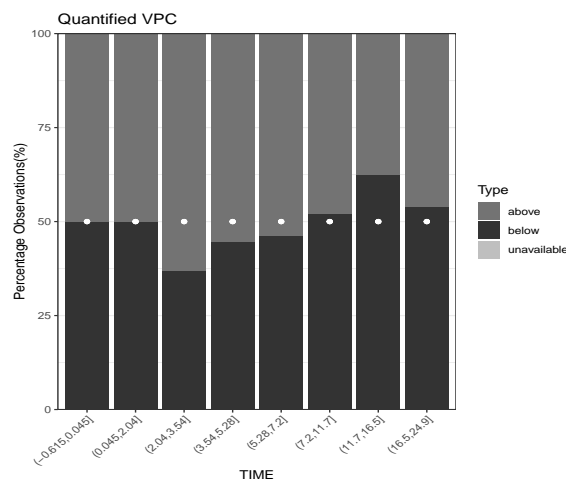
Table 1 shows the list of functions and Table 2 shows the list of arguments used in the functions of **nlmeVPC**. The following codes are for Figure 2 to Figure 6.

```
> library(nlmeVPC)
> data(origdata)
> data(simdata)
> optK(origdata$TIME)$K

[1] 8

> # Figure 1
> VPCgraph(origdata,simdata,N_xbin=8,type="scatter")
> + ggtitle("Visual Predictive Check : scatter")
> VPCgraph(origdata,simdata,N_xbin=8,type="percentile")
> + ggtitle("Visual Predictive Check : percentile")
> VPCgraph(origdata,simdata,N_xbin=8,type="CI")
> + ggtitle("Visual Predictive Check : CI")
> # Figure 2
> aqrVPC(origdata,simdata)
> # Figure 3
> bootVPC(origdata,simdata,N_xbin=8)
> # Figure 4
> asVPC(origdata,simdata,type="CI",N_xbin=8,N_hist=3,weight_method="bin")
```


Figure 6: The quantified VPC plot. The darker gray areas represent the percentages below the median, the lighter gray areas represent the percentage above, and the brightest gray areas represent the percent unavailable in each time bin. The white dots represent the ideal location where the above and the below percentages meet.



Argument	Description
orig_data	A data frame of original data with X and Y variable
sim_data	A matrix of simulated data with only Y values collected
type	Type of VPC graph; "CI", "percentile", or "scatter"
weight_method	The way to put weights in asVPC; "bin" or "distance"
N_xbin	Number of bins in X variable
probs	A numeric vector of probabilities
conf.level	Confidence level of the interval
X_name	Name of X variable in orig_data
Y_name	Name of Y variable in orig_data
subject_name	Name of subject variable in orig_data
MissingDV	Name of missing indicator variable in orig_data
DV_point	Draw point (X, Y) in the plot if TRUE; omit if FALSE
CIvpc_type	Type of CI area in VPC graph; "line" or "segment"
bin_grid	Draw grid lines for binning in X variable if TRUE; omit if FALSE
plot_caption	Put caption with additional information if TRUE; omit if FALSE
plot_flag	Draw plot if TRUE; generate data for drawing plot if FALSE
linesize	Size of line in the plot
pointsize	Size of point in the plot
captionsize	Size of caption
maxK	The maximum number of bins
Kmethod	The way to calculate the penalty in automatic binning; "cluster" or "kernel"
beta	Additional parameter for automatic binning, used in optK function
lambda	Additional parameter for automatic binning, used in optK function

Table 2: Summary of the arguments of functions in **nlmeVPC**.

```

> asVPC(origdata,simdata,type="CI",N_xbin=8,N_hist=3,weight_method="distance")
> # Numerical Predictive Check
> NumericalCheck(origdata,simdata,N_xbin=8,pred.level=c(0,0.2,0.4,0.6,0.8,0.9))$NPC
> # Figure 5
> coverageplot(origdata,simdata,N_xbin=8)
> coverageDetailplot(origdata,simdata,N_xbin=8,predL=0.8)
> # Figure 6
> quantVPC(origdata,simdata)

```

We use an example to show how to use functions in **nlmeVPC** and how they work.

Example

The origdata in **nlmeVPC** is from an experiment on the pharmacokinetics of theophylline. Twelve patients were given oral doses of theophylline, and blood concentrations were measured at 11 time points over the next 25 hours. Each patient had different time points. We consider the following first-order absorption one-compartment model:

$$y_{ij} = \frac{Amt_i * Ke_i * Ka_i}{Cl_i} \left(\exp(-Ke_i * TIME_{ij}) - \exp(-Ka_i * TIME_{ij}) \right) + \varepsilon_{ij}. \quad (2)$$

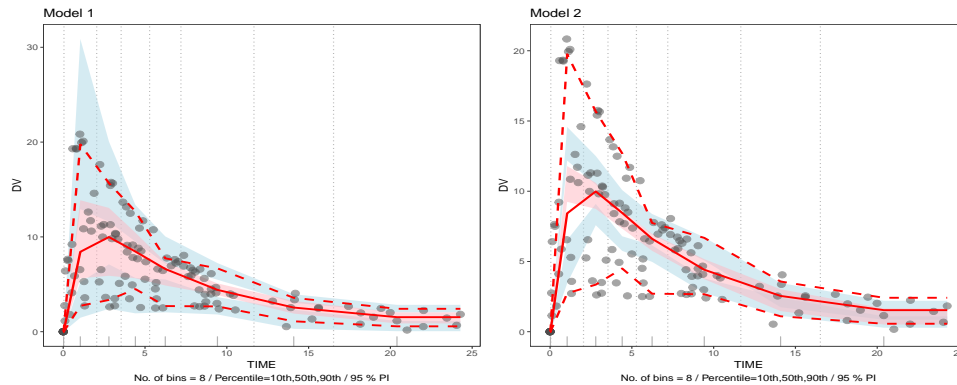
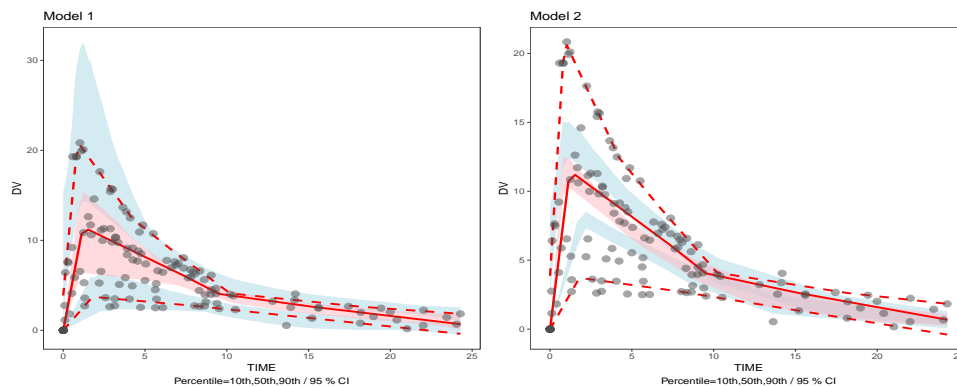
In this model, y_{ij} is the theophylline concentration at $TIME_{ij}$ after an initial dose of Amt_i . The pharmacokinetic parameters are the absorption rate constant Ka , the elimination rate constant Ke , and the clearance Cl . In this example, two different models are fitted and diagnosed using functions in the **nlmeVPC** package. In Model 1, Ka and Cl are considered as random effects. In Model 2, Ke is considered as random effect, and Ka and Cl are considered only as a fixed effect. The **nlme** (Pinheiro et al., 2022) and **nlraa** (Miguez, 2022) packages are used to fit the nonlinear mixed models and to generate the simulated data from the fitted model.

```

> library(nlme)
> library(nlraa)
> library(nlmeVPC)
> data(origdata)
> origdataT <- groupedData(DV~TIME|ID,origdata)
> # Model 1
> T.nlme <- nlme(DV ~ exp(lKe+lKa-lCl)*AMT*
+               (exp(-exp(lKe)*TIME) - exp(-exp(lKa)*TIME))/
+               (exp(lKa)-exp(lKe)), data=origdataT,
+               fixed=lKa+lKe+lCl~1,
+               random=lKa+lCl~1,
+               start=c(lKe=-2,lKa=1.5,lCl=-3))
> set.seed(123456)
> sim.T <- simulate_nlme(T.nlme,nsim=100,psim=3,level=1,value="data.frame")
> simdata.T <- matrix(sim.T$sim.y,ncol=100)
> # Model 2
> F.nlme <- nlme(DV ~ exp(lKe+lKa-lCl)*AMT*
+               (exp(-exp(lKe)*TIME) - exp(-exp(lKa)*TIME))/
+               (exp(lKa)-exp(lKe)), data=origdataT,
+               fixed=lKa+lKe+lCl~1,
+               random=lKe~1,
+               start=c(lKe=-2,lKa=1.5,lCl=-3))
> sim.F <- simulate_nlme(F.nlme,nsim=100,psim=3,level=1,value="data.frame")
> simdata.F <- matrix(sim.F$sim.y,ncol=100)

> # Figure 7
> VPCgraph(origdata,simdata.T,type="CI",N_xbin=8) + ggtitle("Model 1")
> VPCgraph(origdata,simdata.F,type="CI",N_xbin=8) + ggtitle("Model 2")
> # Figure 8
> aqrVPC(origdata,simdata.T) + ggtitle("Model 1")
> aqrVPC(origdata,simdata.F) + ggtitle("Model 2")
> # Figure 9
> asVPC(origdata,simdata.T,type="CI",weight_method="distance",N_xbin=8) + ggtitle("Model 1")
> asVPC(origdata,simdata.F,type="CI",weight_method="distance",N_xbin=8) + ggtitle("Model 2")
> # Figure 10
> bootVPC(origdata,simdata.T,N_xbin=8) + ggtitle("Model 1")

```

Figure 7: The VPC plots of Model 1 and Model 2.**Figure 8:** The additive quantile regression VPC plots for Model 1 and Model 2.

```

> bootVPC(origdata,simdata.F,N_xbin=8) + ggtitle("Model 2")
> # Figure 11
> coverageplot(origdata,simdata.T,conf.level=0.9,N_xbin=8) + ggtitle("Model 1")
> coverageplot(origdata,simdata.F,conf.level=0.9,N_xbin=8) + ggtitle("Model 2")
> # Figure 12
> coverageDetailplot(origdata,simdata.T,predL=0.5,N_xbin=8) + ggtitle("Model 1")
> coverageDetailplot(origdata,simdata.F,predL=0.5,N_xbin=8) + ggtitle("Model 2")
> # Figure 13
> coverageDetailplot(origdata,simdata.T,predL=0.8,N_xbin=8) + ggtitle("Model 1")
> coverageDetailplot(origdata,simdata.F,predL=0.8,N_xbin=8) + ggtitle("Model 2")
> # Figure 14
> quantVPC(origdata,simdata.T,N_xbin=8) + ggtitle("Model 1")
> quantVPC(origdata,simdata.F,N_xbin=8) + ggtitle("Model 2")

```

Figure 7 shows the results of the VPCgraph for Model 1 and Model 2. The solid line represents the 50th quantiles of the observed data, and the dashed lines represent the 10th and 90th percentiles of the observed data. In Model 1, all quantile lines (the solid and dashed lines) are in the confidence area. However, the 10th and 90th percentiles in Model 2 are mostly outside the confidence area, especially in 0 to 5 hours.

Figure 8 shows the results of aqrVPC, and Figure 9 shows the results of asVPC for Model 1 and Model 2. The additive quantile regression VPCs and the average shifted VPCs show similar patterns to the original VPCs. Model 1 shows all quantile lines in the confidence area, and the 10th and 90th percentiles in Model 2 are mostly outside the confidence area.

Figure 10 shows the results of bootVPC for Model 1 and Model 2. The solid and dashed blue lines show the 10th, 50th, and 90th percentiles of the simulated data. The solid red line represents the 50th percentile line of the observed data, and the pink areas represent the 95% confidence areas of the 50th percentile line, calculated from the bootstrap samples of the observed data. The solid blue line is in the pink area, and the two solid red and blue lines are almost similar in both models. The dashed blue lines in Model 1 cover most of the observed data. However, the dashed blue lines in Model 2 do not cover the observed data. Many observed data are outside these dashed blue lines in Model 2, especially in 0 to 10 hours.

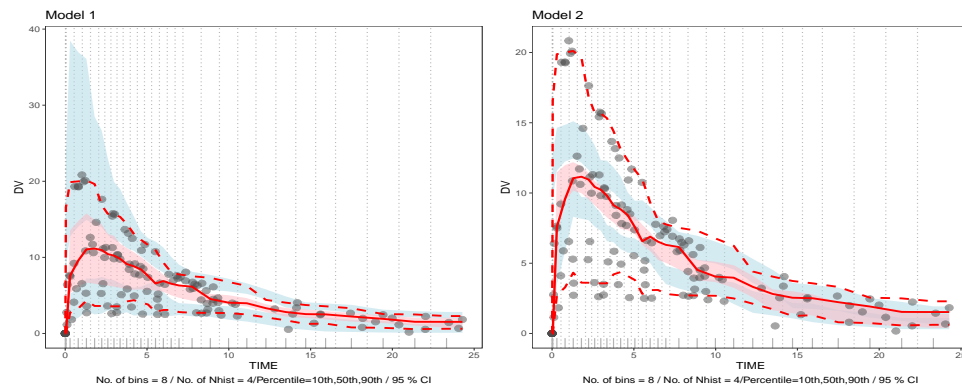
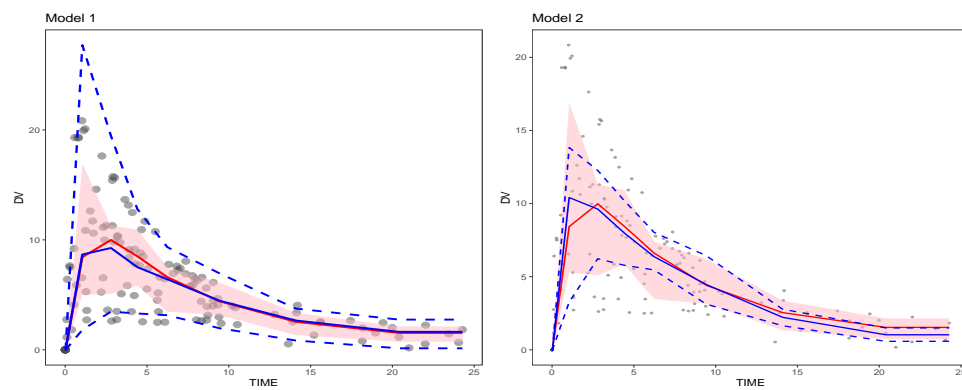
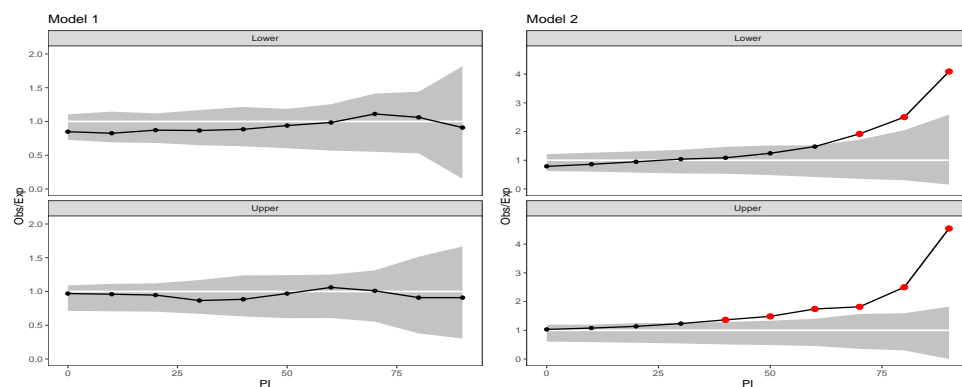
Figure 9: The average shifted VPC plots for Model 1 and Model 2.**Figure 10:** The bootstrap VPC plots for Model 1 and Model 2.**Figure 11:** The coverage plots for Model 1 and Model 2.

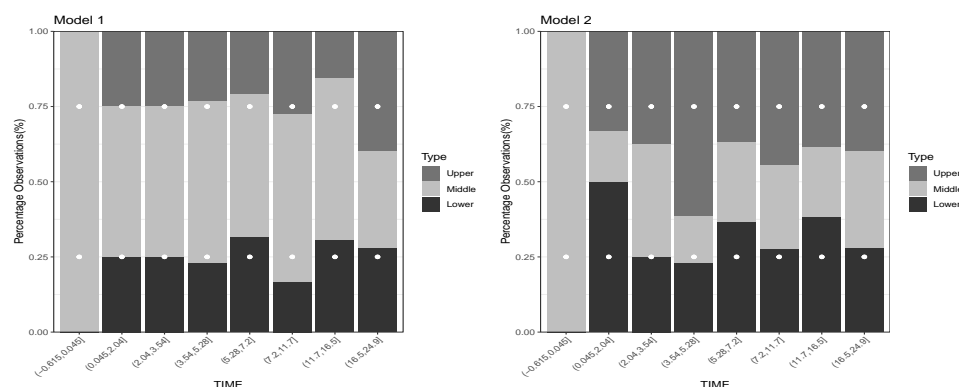
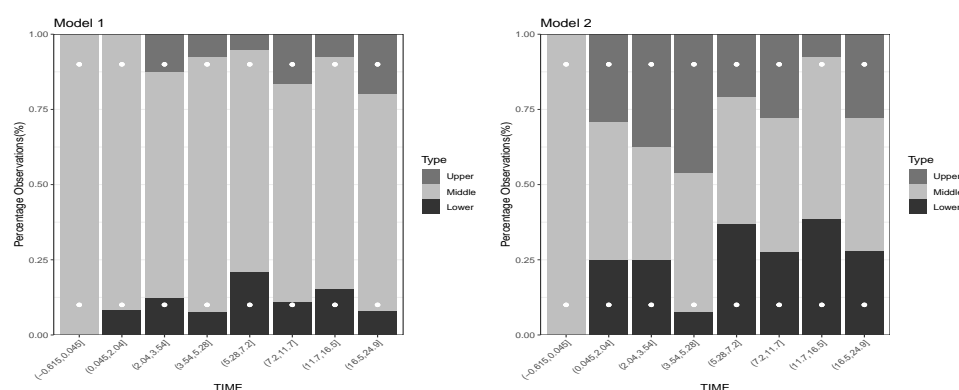
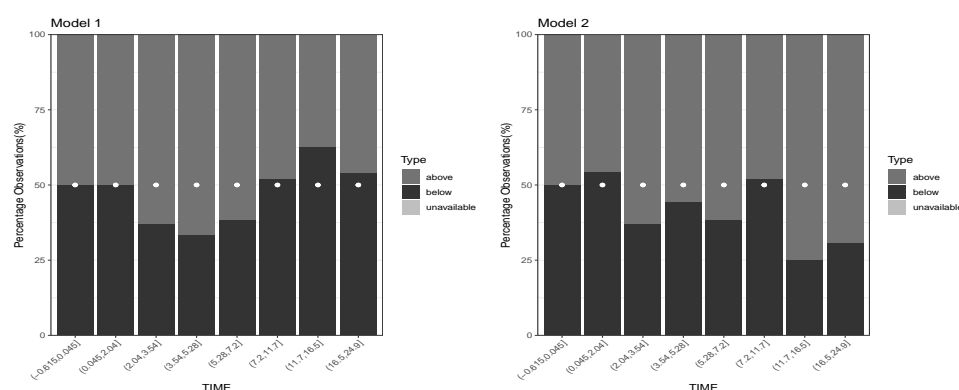
Figure 12: The detailed coverage plots for Model 1 and Model 2 when PI=50%.**Figure 13:** The detailed coverage plots for Model 1 and Model 2 when PI=80%.

Figure 11 shows the coverageplot results for Model 1 and Model 2. The lines are in the gray area and close to the white line in Model 1. However, the lines in Model 2 are not in the gray area, especially when the PI value is large. Figures 12 and 13 show the results of coverageDetailplot for Model 1 and Model 2 when PIs are 50% and 80%. The upper and lower percentages in both figures are close to the white points in Model 1. On the other hand, the upper percentages of the most time bins are far from the white points in Model 2, especially the time bin (5.28,7.2] when PI = 50%. When PI = 80%, most upper and lower percentages are far from the white points.

Figure 14 shows the results of quantVPC for Model 1 and Model 2. In Model 2, the right tail area (after 11 hours) looks quite different from the expected pattern. The above percentages are much larger than the below percentages.

The results from Figure 7 through Figure 14 show that Model 1 explains the origdata quite well. However, Model 2 shows different patterns than Model 1 in most figures. We can conclude that Model 1 is better than Model 2, and treating Ka and Cl as random effects is better.

Figure 14: The quantified VPC plots for Model 1 and Model 2.

Summary

This paper introduces the **nlmeVPC** package. The VPC and its extensions are useful for validating the nonlinear mixed effect model. The **nlmeVPC** package provides various visual **validation** tools for the nonlinear mixed effect model in two different approaches: validation in data space and model space. Both approaches are valuable. Validation in data space can compare the fitted model with the original data, and validation in model space provides detailed comparisons in various ways. In the **nlmeVPC** package, we also provide new approaches - `asVPC` and `coverageDetailplot`. Here, `asVPC` provides a more precise VPC plot, and `coverageDetailplot` provides the detailed feature of the fitted model that is not captured in the coverage plot. Even though the coverage plot does not show any problem with the fitted model, the detailed coverage plot can reveal the problem in a specific region (Figures 10, 11, and 12).

Acknowledgements

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (2018R1A2B6001251).

Bibliography

- R. J. Bauer. Nonmem users guide introduction to nonmem 7.2.0. *ICON Development Solutions Ellicott City, MD*, 2011. URL <https://www.iconplc.com/innovation/nonmem/>. [p1]
- M. Davidian. *Nonlinear models for repeated measurement data*. Routledge, 2017. [p1]
- M. Fidler, J. J. Wilkins, R. Hooijmaijers, T. M. Post, R. Schoemaker, M. N. Trame, Y. Xiong, and W. Wang. Nonlinear mixed-effects model development and simulation using nlmixr and related r open-source packages. *CPT: pharmacometrics & systems pharmacology*, 8(9):621–633, 2019. doi: 10.1002/psp4.12445. URL <https://doi.org/10.1002/psp4.12445>. [p1]
- A. Heus, D. W. Uster, V. Grootaert, N. Vermeulen, A. Somers, D. H. In't Veld, S. G. Wicha, and P. A. De Cock. Model-informed precision dosing of vancomycin via continuous infusion: a clinical fit-for-purpose evaluation of published pk models. *International Journal of Antimicrobial Agents*, 59(5):106579, 2022. doi: 10.1016/j.ijantimicag.2022.106579. URL <https://doi.org/10.1016/j.ijantimicag.2022.106579>. [p1]
- K. M. Jansen, K. Patel, K. Nieforth, and C. M. Kirkpatrick. A regression approach to visual predictive checks for population pharmacometric models. *CPT: pharmacometrics & systems pharmacology*, 7(10): 678–686, 2018. doi: 10.1002/psp4.12319. URL <https://doi.org/10.1002/psp4.12319>. [p2, 3]
- M. Karlsson and R. Savic. Diagnosing model diagnostics. *Clinical Pharmacology & Therapeutics*, 82(1): 17–20, 2007. doi: 10.1038/sj.clpt.6100241. URL <https://doi.org/10.1038/sj.clpt.6100241>. [p1, 6]
- M. O. Karlsson and N. Holford. A tutorial on visual predictive checks. In *17th meeting of the Population Approach Group in Europe, Marseille, France, page Abstr*, volume 1434, 2008. URL <https://www.page-meeting.org/?abstract=1434>. [p1, 2, 7]
- R. Keizer. `vpc`: Create visual predictive checks. r package version 1.2.2, 2021. URL <https://cran.r-project.org/web/packages/vpc/index.html>. [p1]
- R. J. Keizer, M. Karlsson, and A. Hooker. Modeling and simulation workbench for nonmem: tutorial on pirana, psn, and xpose. *CPT: pharmacometrics & systems pharmacology*, 2(6):1–9, 2013. doi: 10.1038/psp.2013.24. URL <https://doi.org/10.1038/psp.2013.24>. [p1, 7]
- M. Lavielle and K. Bleakley. Automatic data binning for improved visual diagnosis of pharmacometric models. *Journal of pharmacokinetics and pharmacodynamics*, 38(6):861–871, 2011. doi: 10.1007/s10928-011-9223-3. URL <https://doi.org/10.1007/s10928-011-9223-3>. [p3]
- L. Lindbom, J. Ribbing, and E. N. Jonsson. Perl-speaks-nonmem (psn)—a perl module for nonmem related programming. *Computer methods and programs in biomedicine*, 75(2):85–94, 2004. doi: 10.1016/j.cmpb.2003.11.003. URL <https://doi.org/10.1016/j.cmpb.2003.11.003>. [p1, 7]

- A. R. Maharaj, H. Wu, C. P. Hornik, and M. Cohen-Wolkowicz. Pitfalls of using numerical predictive checks for population physiologically-based pharmacokinetic model evaluation. *Journal of pharmacokinetics and pharmacodynamics*, 46(3):263–272, 2019. doi: 10.1007/s10928-019-09636-5. URL <https://doi.org/10.1007/s10928-019-09636-5>. [p6]
- F. Miguez. nlraa: nonlinear regression for agricultural applications. r package version 1.5, 2022. URL <https://cran.r-project.org/web/packages/nlraa/index.html>. [p10]
- S. E. Mudde, R. Ayoun Alsoud, A. van der Meijden, A. M. Upton, M. U. Lotlikar, U. S. Simonsson, H. I. Bax, and J. E. de Steenwinkel. Predictive modeling to study the treatment-shortening potential of novel tuberculosis drug regimens, toward bundling of preclinical data. *The Journal of Infectious Diseases*, 225(11):1876–1885, 2022. doi: 10.1093/infdis/jiab101. URL <https://doi.org/10.1093/infdis/jiab101>. [p1]
- T. Nguyen, M.-S. Mouksassi, N. Holford, N. Al-Huniti, I. Freedman, A. C. Hooker, J. John, M. O. Karlsson, D. Mould, J. Pérez Ruixo, et al. Model evaluation of continuous data pharmacometric models: metrics and graphics. *CPT: pharmacometrics & systems pharmacology*, 6(2):87–109, 2017. doi: 10.1002/psp4.12161. URL <https://doi.org/10.1002/psp4.12161>. [p1]
- M. Otto, K. Bergmann, G. Jacobs, and M. J. van Esdonk. Predictive performance of parent-metabolite population pharmacokinetic models of (s)-ketamine in healthy volunteers. *European journal of clinical pharmacology*, 77(8):1181–1192, 2021. doi: 10.1007/s00228-021-03104-1. URL <https://doi.org/10.1007/s00228-021-03104-1>. [p1]
- J. Pinheiro, D. Bates, S. DebRoy, D. Sarkar, S. Heisterkamp, B. Van Willigen, and R. Maintainer. nlme: Linear and nonlinear mixed effects models. r package version 3.1-159, 2022. URL <https://cran.r-project.org/web/packages/nlme/index.html>. [p10]
- J. C. Pinheiro and D. M. Bates. *Mixed-Effects Models in S and S-PLUS*. Springer New York, 2000. doi: 10.1007/978-1-4419-0318-1. URL <https://doi.org/10.1007/978-1-4419-0318-1>. [p1]
- T. M. Post, J. I. Freijer, B. A. Ploeger, and M. Danhof. Extensions to the visual predictive check to facilitate model performance evaluation. *Journal of pharmacokinetics and pharmacodynamics*, 35(2):185–202, 2008. doi: 10.1007/s10928-007-9081-1. URL <https://doi.org/10.1007/s10928-007-9081-1>. [p2, 3, 7]
- D. W. Scott. Average shifted histograms: Effective nonparametric density estimations in several dimensions. *Annals of Statistics*, 13(3):1024–1040, 1985. doi: 10.1214/aos/1176349654. URL <https://doi.org/10.1214/aos/1176349654>. [p4]
- D. D. Wang and S. Zhang. Standardized visual predictive check versus visual predictive check for model evaluation. *The Journal of Clinical Pharmacology*, 52(1):39–54, 2012. doi: 10.1177/0091270010390040. URL <https://doi.org/10.1177/0091270010390040>. [p6]
- H. Wickham. *ggplot2: Elegance graphics for Data Analysis*. Springer, 2016. [p2]
- H. Wickham, D. Cook, and H. Hofmann. Visualizing statistical models: Removing the blindfold. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 8(4):203–225, 2015. doi: 10.1002/sam.11271. URL <https://doi.org/10.1002/sam.11271>. [p1]

Eun-Hwa Kang
Department of Statistics
Ewha Womans University
Seoul, Korea

Myungji Ko
Department of Statistics
Ewha Womans University
Seoul, Korea

Eun-Kyung Lee
Department of Statistics
Ewha Womans University
Seoul, Korea
(0000-0003-0817-5000)
lee.eunk@ewha.ac.kr