Response to Reviewer 4

* Manuscript

REVIEWER'S COMMENT:

In my opinion, the manuscript improved a lot with the revision. Both the introductory section and the example application have become cleaner and easier to follow. I would still recommend a through language check; I noticed a few typos here and there.

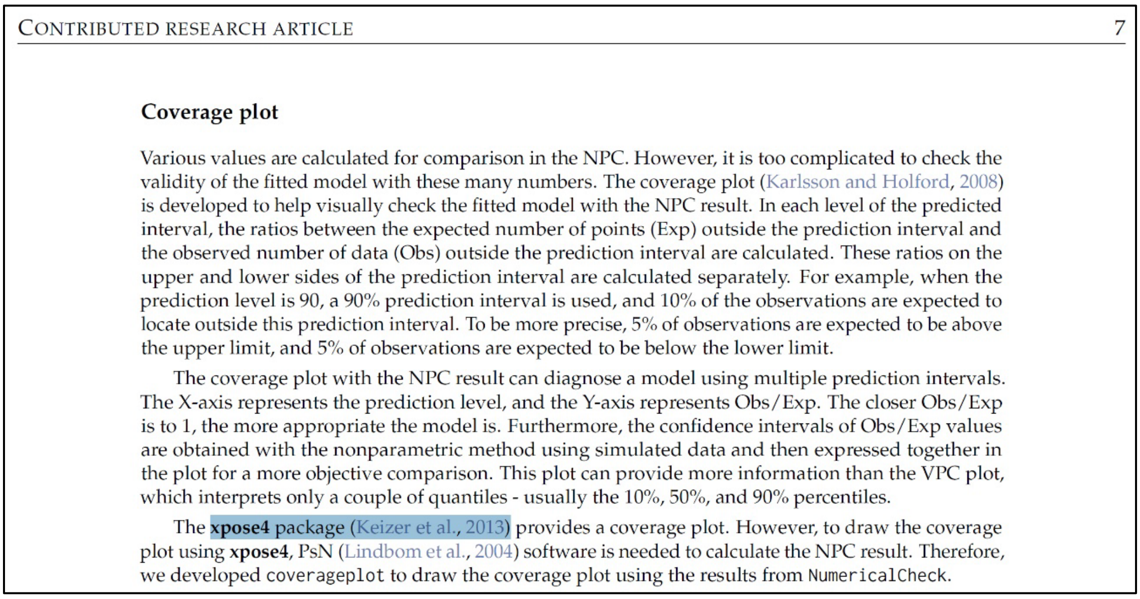
Additional comments:

- p7, paragraph 3: A reference is missing for xpose4.

AUTHORS' COMMENT:

Thank you for reviewing our paper in detail. We read the comments carefully and followed them for our revision.

- **We add the reference of xpose4 on p7, paragraph3.**



CONTRIBUTED RESEARCH ARTICLE 7

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 usage of ₩pkg{} and ₩CRANpkg{} is still not consistent with the Instructions for Authors: "The first citation of a CRAN package should use ₩CRANpkg ... Further citations use ₩pkg and need not be followed by a citation." (see, e.g., the paragraph referenced above)

- **We change all ₩CRANpkg{} to ₩pkg{} except for the first citation of a cran package.**

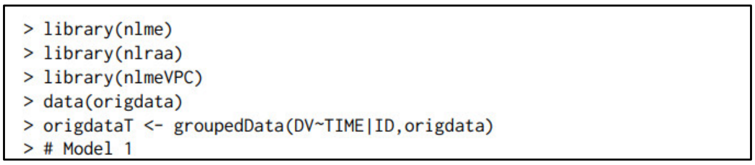
REVIEWER’S COMMENT:

- Example section:

  - I still don't understand why we need library(tidyverse), which loads many unused packages.

AUTHORS’ COMMENT:

- **We remove library(tidyverse) in the example.**



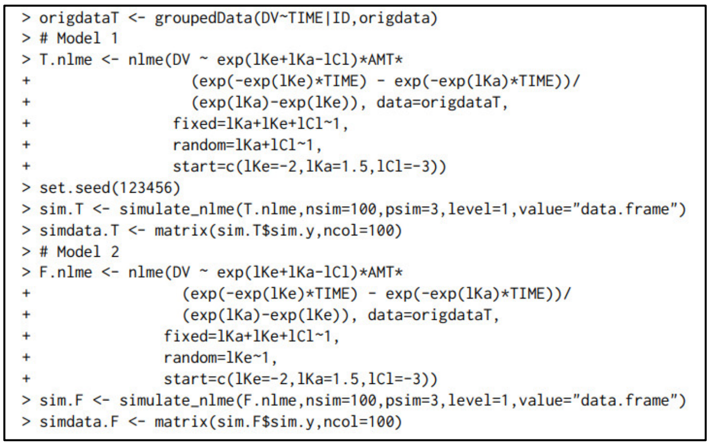
```
> library(nlme)
> library(nlraa)
> library(nlmeVPC)
> data(origdata)
> origdataT <- groupedData(DV~TIME|ID,origdata)
> # Model 1
```

REVIEWER’S COMMENT:

- Minor issue but using <- instead of = for assignments is usually considered cleaner and more 'R-like'.

AUTHORS’ COMMENT:

- **We change “=” to “<-“   in the example.**



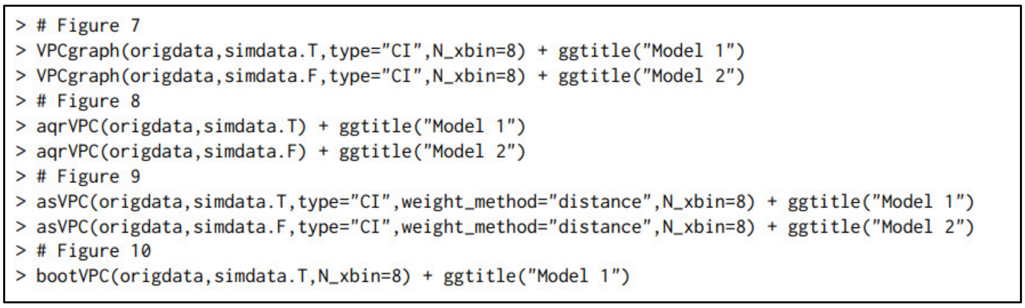
```
> origdataT <- groupedData(DV~TIME|ID,origdata)
> # Model 1
> T.nlme <- nlme(DV ~ exp(lKe+lKa-lCl)*AMT*
+                  (exp(-exp(lKe)*TIME) - exp(-exp(lKa)*TIME))/
+                  (exp(lKa)-exp(lKe)), data=origdataT,
+                fixed=lKa+lKe+lCl~1,
+                random=lKa+lCl~1,
+                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)
```

REVIEWER’S COMMENT:

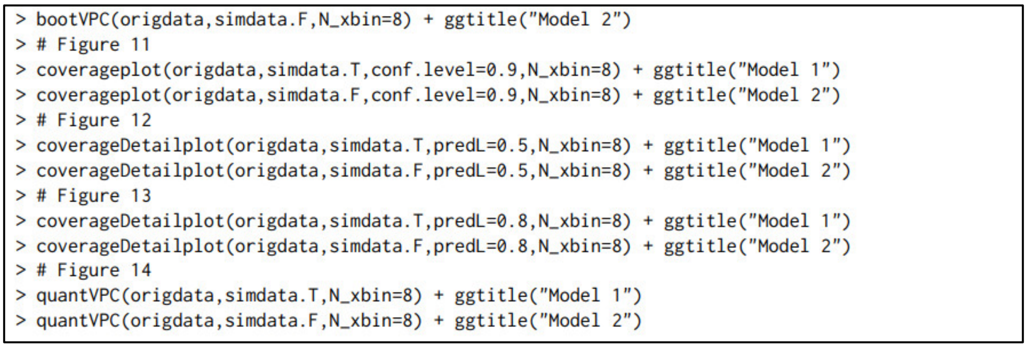
- The code block included in the section only fits the models and simulates from the resulting objects. It would be helpful to include the function calls that correspond to the plots shown in Figures 7 to 14.

AUTHORS’ COMMENT:

- **We add all codes for Figures 7 to 14.**



```
> # 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”)
```



```
> 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”)
```

* Code review

REVIEWER'S COMMENT:

The code and documentation of nlmeVPC v2.5 have also improved since the last version. Because I still see a few issues, I would recommend a general code review to remove computational bottlenecks and improve readability before releasing the next version.

AUTHORS' COMMENT:

We followed the following reviewer's comment, revised our package, and uploaded nlmeVPC v2.6 on CRAN (https://cran.r-project.org/web/packages/nlmeVPC/index.html).

REVIEWER'S COMMENT:

- I found several cases of length(table(X)) to count the unique elements of a vector: This is very inefficient.

AUTHORS' COMMENT:

- **We change "length(table(X))" to "length(unique(X)) ".**

REVIEWER'S COMMENT:

- I found lots of examples of NULL and NA assignments to variables without any apparent use. What's the purpose of these? _Some_ examples: Misc.R (l23), asVPC.R (l101-103), NumericalCheck.R (l68-72, l79-82), aqrVPC.R (l67-69) etc.

AUTHORS' COMMENT:

- **We mainly add NULL and NA assignments to variables to remove "no visible binding for global variable … " notes from the cran check. In nlmeVPC v2.6, we remove all NULL and NA assignments and add "global.R" with utils::globalVariables( ).**

REVIEWER'S COMMENT:

- There are still examples of filling arrays in loops without preassignment. _Some_ examples in NumericalCheck.R (l142, l159, l162)

AUTHORS' COMMENT:

- **We remove filling arrays in loops without pre-assignment in NumercalCheck.R. First, we define keepAll, keepAll2, and CIkeep before the loop and assign values to the specific location of keepAll, keepAll2, and CIkeep.**