R and nlmixr as a gateway between statistics and pharmacometrics

In pharmacometrics, we try to understand how the dose of a drug and measurable attributes of patients influence their drug concentration-time profiles, and to relate drug concentration-time profiles and patient attributes to the time profile of drug effects. Using this information, we can use tools like "nlmixr" to try to identify the right dose for the right patient. Our discipline is a fusion of pharmacology, physiology, and quantitative sciences, such as statistics.

Statistical methods commonly practiced in the analysis of clinical studies often utilize linear mixed models to investigate trial outcomes. The characterization of longitudinal concentration and drug effect in pharmacometrics extends this statistical paradigm to include nonlinear models, which are often based on ordinary differential equations (ODEs). These ODEs are required to describe complex systems affecting concentrations and effects of drugs that often cannot be described by simple linear models.

Nonlinear models can be fitted using the statistical language R[1] with its included "nlme" package. The nlme package allows the expression of nonlinear functions in R and fitting these models in a nonlinear mixed effect (NLME) modeling statistical framework. Only nonlinear equations with closed-form solutions can be used, however, and if more than one dose is administered to a patient, these equations quickly become extremely complex.

The nlmixr package[2] extends nlme by providing the ability to add flexible individual-based dosing regimens, and to describe the models using both ODEs and solved systems when these are available. The nlmixr package has made these ODEs easy to express by using Leibniz notation (see below). To run the model, one specifies the ODE/solved system and provides initial estimates for the model as described in the nlmixr tutorial.[2] The model then can be solved using the nlme algorithm, or, more optimally, using more advanced algorithms that have been shown to provide more accurate parameter estimates like first-order conditional estimation with interaction (FOCEI)[3] and stochastic approximation expectation maximization (SAEM).[3] Statisticians who are familiar with nlme and are becoming familiar with population pharmacokinetics (PKs) and pharmacodynamics (PDs) may thus find it relatively straightforward to transition from nlme to more robust methodologies and more complex models and data structures.

An example of nlmixr applied to a simple PK system serves to illustrate the uncomplicated syntax and how a transition to more complex and robust NLME methodologies might be accomplished without leaving the familiar R environment. Theophylline is used to inhibit phosphodiesterase and is used as treatment in many respiratory diseases. The analysis of the theophylline dataset of Dr. Upton[4] is one of the most common introductory applications of NLME modeling, and is a common introductory dataset for learning nlme (it is included in the base R distribution as the Theoph and reformatted and supplied in nlmixr as theo_sd and extrapolated to multiple doses with theo_md). If a single dose is given at time zero, the equations can be expressed as the closed-form solution to a one-compartment oral absorption system, and analyzed using nlme. In the syntax of nlmixr, fitting of the theophylline PK can be specified as:

```
one.compartment <- function() {
    ini({
        tka <- 0.45 # Log Ka
        tcl <- 1 # Log Cl
        tv <- 3.45    # Log V
        eta.ka ~ 0.6
        eta.cl ~ 0.3
        eta.v ~ 0.1
        add.sd <- 0.7
    })

    model({
        ka <- exp(tka + eta.ka)
        cl <- exp(tcl + eta.cl)
        v <- exp(tv + eta.v)
        linCmt() ~ add(add.sd)
    })
}
fit.nlme <- nlmixr(one.compartment, theo_sd, "nlme")
```

Briefly, the syntax for a nlmixr model contains a `model({})` block that specifies the equations using a solved solution with `linCmt()`. This model block also describes both the relationship between population parameters and individual parameters (like specifying that it is lognormally distributed, "`cl = exp(tcl + eta.cl)`") and the relationship between the end point and unexplained variability (as in additive or constant unexplained variability, "`cp ~ add(add.sd)`"). The initial estimates are provided in the `ini({})` block.

Note that when estimating the model with the "nlme" method, nlmixr() is just a wrapper that converts nlmixr function syntax to code for nlme().

In PK analyses, concentration data is often associated with multiple doses. In the case of a linear compartmental model, such as the one used above, concentrations from multiple doses can be coded by applying the principle of superposition. However, coding a model for multiple-dose PK in this way can be burdensome and error-prone, especially in the presence of complexities like missing doses, dose changes, time-varying covariates, and noncompliance to the dosing schedule. In nlmixr, the need for handling issues like these is removed from users. The syntax for fitting a multiple dose theophylline PK is identical to the previous example, except that extra doses were added as well as extra simulated observations:

```
fit.nlme <- nlmixr(one.compartment, theo_
md, "nlme")
```

In pharmacology, the binding of drug molecules with a target (e.g., a receptor on the cell membrane) triggers series of biological changes such as secondary messengers, signal transduction, signal transcription, DNA changes, and protein production. These time-sequenced biological changes can often be described conveniently by a set of ODEs. In population PK/PD analysis, it is thus crucial to be able to use ODEs for describing models.

The theophylline PK model can be equivalently written in ODEs by changing the model block to:

```
model({
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v <- exp(tv + eta.v)
    d/dt(depot) = -ka * depot
    d/dt(center) = ka * depot - cl/v * center
    cp = center / v
    cp ~ add(add.sd)
})
```

The simplified Laplacian approximation use by nlme() is reasonably accurate and adequate for many nlme applications. One critical assumption for this approximation to work, however, is that residuals do not change a great deal with model predictions, which is not often the case with PK/PD data. In a typical PK context, assay precision changes with the drug concentration, which in turn requires a proportional residual error. With a proportional error model, it has been demonstrated[3] that the simplified Laplacian approximation is no longer adequate. More advanced approaches have been developed over the years, such as a high precision approximation of the Likelihood called the FOCEi (in the PK/PD literature) and SAEM. Simulation studies have demonstrated the superiority of these more recent, higher-precision approaches over the built-in nlme() algorithm.

The nlmixr() incorporates these high precision approaches, and syntax for fitting the theophylline PK model with either the SAEM or FOCEi algorithm is almost identical to the previous examples:

```
fit.saem <- nlmixr(one.compartment, theo_
md, "saem")fit.focei <- nlmixr(one.com-
partment, theo_md, "focei")
```

Another advantage to the use of nlmixr is the rich R ecosystem in which it is based. The nlmixr output can be used with many other R packages, facilitating end-to-end development of a NLME model from data creation to re-simulation. For example, exploratory graphical analysis can be completed with R's ggplot2/tidyverse[5] or with helper functions included in the R package xgxr.[6] Goodness-of-fit plots can be generated using xpose.nlmixr[7] or by the ggPMX[8] package. Visual predictive checks and simulations can be generated easily from the nlmixr object with RxODE[9] and the vpc[10] package.

The barriers to interested statisticians engaging with NLME model-based data analysis have traditionally been high—the gold standard tools used for this are somewhat arcane, use their own terminology and require a great deal of time and experience to learn, inhibiting the ability of pharmacometricians and statisticians to communicate and collaborate. Nlmixr operates in a shared, familiar programming environment, and democratizes pharmacometrics, reducing the requirements of successful cooperation to a common core: a shared understanding of models and drug development.

CONFLICTS OF INTEREST

FUNDING INFORMATION

Matthew Fidler[1]
Richard Hooijmaijers[2]
Rik Schoemaker[3]

Justin J. Wilkins[3]
Yuan Xiong[4]
Wenping Wang[1]

[1]*Novartis Pharmaceuticals, Inc, Fort Worth, Texas, USA*
[2]*Leiden experts on Advanced Pharmacokinetics and Pharmacodynamics (LAP&P, Leiden, The Netherlands*
[3]*Occams, Amstelveen, The Netherlands*
[4]*Johnson and Johnson, Janssen Research & Development, Spring House, Pennsylvania, USA*

Correspondence

Matthew Fidler, Novartis Pharmaceuticals, Inc., Fort Worth, TX, USA.
Email: matt.fidler@novartis.com

REFERENCES

1. Ihaka R, Gentleman R. R: a language for data analysis and graphics. *J Comput Graph Stat*. 1996;5:299–314.
2. Fidler M, Wilkins JJ, Hooijmaijers R, et al. Nonlinear mixed-effects model development and simulation using nlmixr and related R open-source packages. *CPT Pharmacometrics Syst Pharmacol*. Available from: http://www.ncbi.nlm.nih.gov/pubmed/31207186
3. Schoemaker R, Xiong Y, Wilkins J, Laveille C, Wang W. nlmixr: an open-source package for pharmacometric modelling in R. PAGE 26 (2017) Abstract 7102 [www.page-meeting.org/?abstract=7102].
4. Pinheiro J, Bates DM. *Mixed-effects Models in S and S-PLUS*. New York, NY: Springer (Appendix A.29); 2000.
5. Wickham H, Averick M, Bryan J, et al. Welcome to the tidyverse. *J Open Source Soft*. 2019;4(43):1686.
6. Stein A, Margolskee A, Khanshan F, et al. xgxr: Exploratory Graphics for Pharmacometrics. Version 1.0.9 (2020).
7. Wilkins J, Fidler M, Wang W, et al. xpose.nlmixr: Graphical Diagnostics for Pharmacometric Models: Extension to 'nlmixr' version 0.1.5 (2020).
8. Gassem A, Bieth B, Baltcheva I, et al. ggPMX: 'ggplot2' Based Tool to Facilitate Diagnostic Plots for NLME Models. Version 1.2 (2020).
9. 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.
10. Keizer R. vpc: Create Visual Predictive Checks. Version 1.2.1. (2020).