

nlmixr: an open-source package for pharmacometric modelling in R

Rik Schoemaker¹, Matt Fidler², Yuan Xiong³, Justin Wilkins¹, Mirjam Trame, Christian Laveille⁴, Wenping Wang²
¹Occams, The Netherlands, ²Novartis Pharmaceuticals, USA, ³Certara Strategic Consulting, USA, ⁴Calvagone, France

Introduction

nlmixr is an open-source R package under development, available on GitHub¹. It builds on RxODE², an R package for simulation of nonlinear mixed effect models using ordinary differential equations (ODEs), providing an efficient and versatile way to specify pharmacometric models and dosing scenarios, with rapid execution due to compilation in C. By combining the RxODE core with population-type estimation routines, a versatile pharmacometric ecosystem entirely contained within R becomes feasible. Currently, estimation routines comprise the nlme³ package in R, a Stochastic Approximation Expectation Maximization (SAEM) algorithm⁴, and a proof-of-concept First-Order Conditional Estimation with Interaction (FOCE-I) implementation⁵, as well as adaptive Gaussian quadrature for odd-type data. Both closed-form and ODE model definitions are included in nlmixr.

Methods

Richly sampled profiles were simulated for 4 different dose levels (10, 30, 60 and 120 mg) of 30 subjects each as single dose (over 72h), multiple dose (4 daily doses), single and multiple dose combined, and steady state dosing, for a range of test models: 1- and 2-compartment disposition, with and without 1st order absorption, with either linear or Michaelis-Menten (MM) clearance (MM without steady state dosing). This provided a total of 42 test cases. All inter-individual variabilities (IIVs) were set at 30%, residual error at 20% and overlapping PK parameters were the same for all models. A similar set of models was previously used to compare NONMEM and Monolix⁶. Additionally, a sparse data estimation situation was investigated where 500 datasets of 600 subjects each (150 per dose) were generated consisting of 4 random time point samples in 24 hours per subject, using a first-order absorption, 1-compartment disposition, linear elimination model. NONMEM^{®5} with FOCE-I was used as a comparator to test the various nlmixr estimation routines.

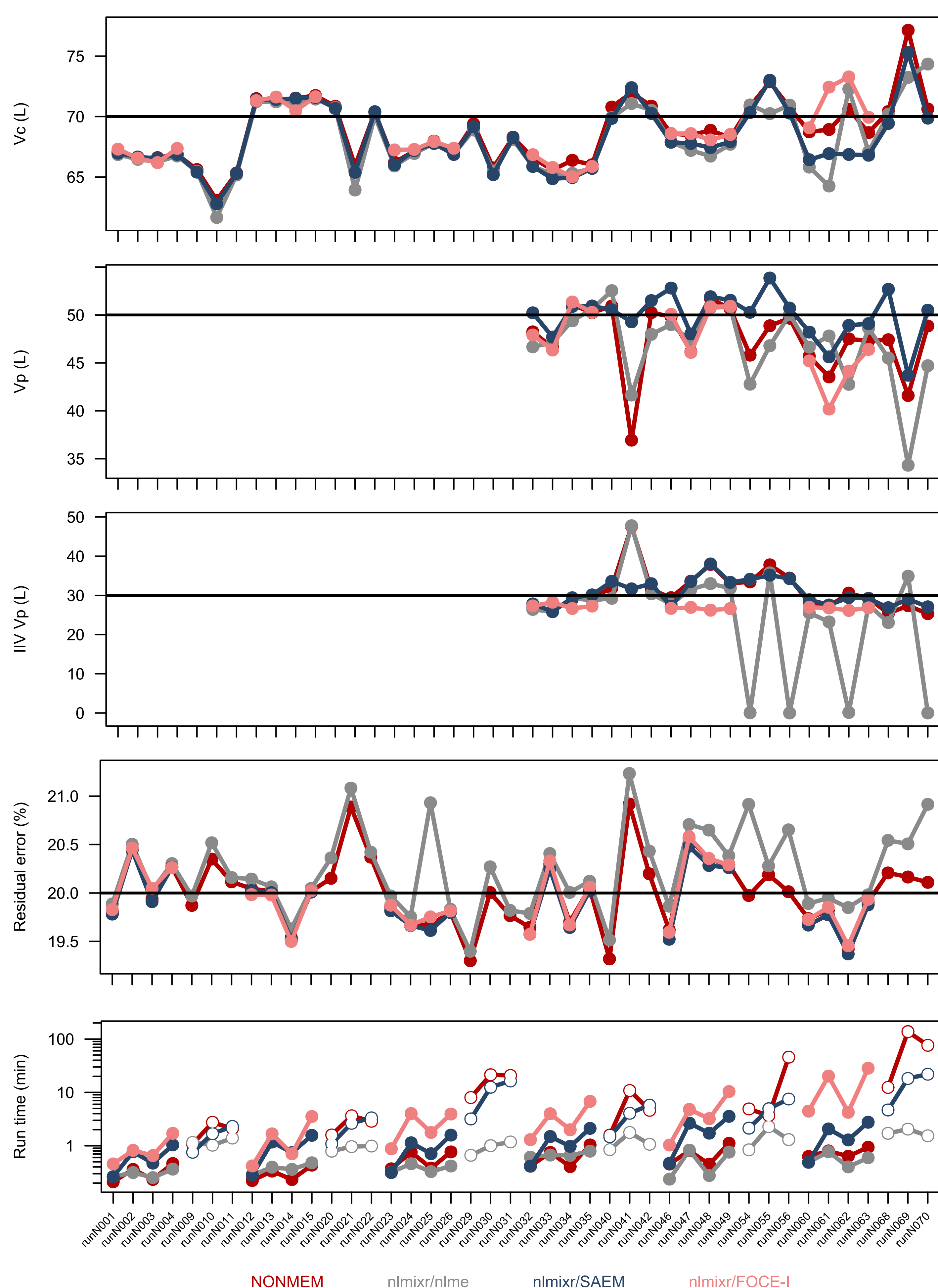


Figure 1. Rich sampling data results: theta estimates for Vc and Vp, IIV estimates for Vp, residual error, and log run times for closed-form solution models (closed marks) and ode solution models (open markers), comparing NONMEM (red lines), nlmixr/nlme (grey lines), nlmixr/SAEM (blue lines), and nlmixr/FOCE-I (pink lines). Horizontal line: value used for simulation.

Results

For the richly sampled profiles, theta parameter estimates and residual error estimates were comparable across all estimation methods. IIV estimates were regularly estimated close to 0% with nlmixr/nlme, whereas NONMEM, nlmixr/SAEM, and nlmixr/FOCE-I provided estimates closer to the original simulation values (see Figure 1). Estimation run times were shorter for nlmixr/nlme when compared to NONMEM, but considerably longer for nlmixr/SAEM, and especially for nlmixr/FOCE-I.

The sparse data analyses indicated a good correlation between NONMEM Ka estimates and both nlmixr/nlme and nlmixr/SAEM estimates, but for nlmixr/FOCE-I, estimates seemed to be uncorrelated (see Figure 2). IIV for Ka was estimated close to zero for 91.1% of the analyses for nlmixr/nlme, for 2.2% for NONMEM, and for 0.0% of the cases for both nlmixr/SAEM, and nlmixr/FOCE-I. These results suggest that at this stage, the nlmixr/SAEM algorithm in particular, is a viable alternative to NONMEM-based parameter estimation.

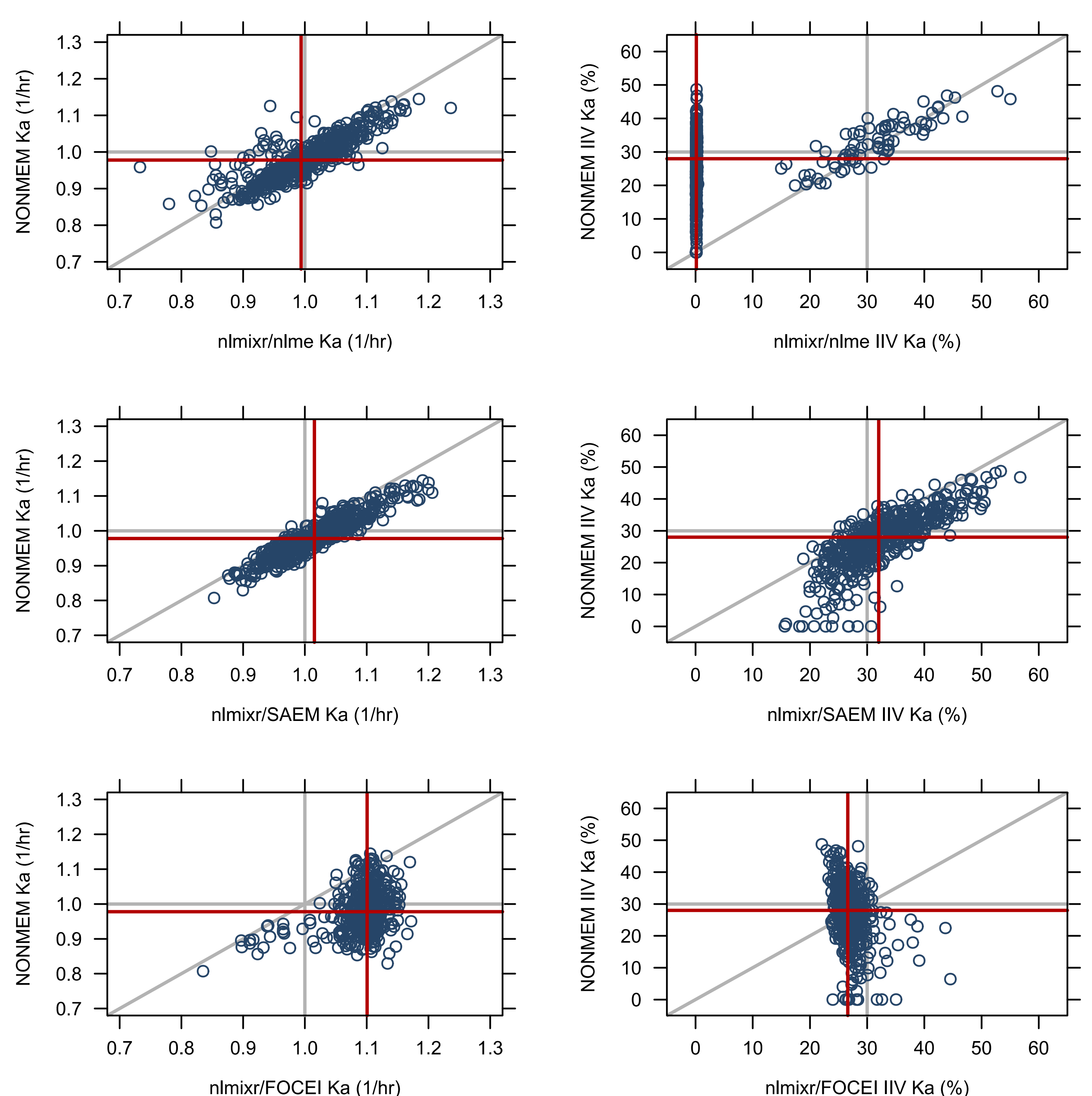


Figure 2. Sparse sampling data results: theta (left), IIV estimates (right) for Ka comparing NONMEM with nlmixr/nlme (top), nlmixr/SAEM (middle), and nlmixr/FOCE-I (bottom). Horizontal and vertical red lines: median estimate across 500 replications. Horizontal and vertical grey lines: value used for simulation.

Conclusions

These findings provide further evidence that nlmixr may provide a viable open-source parameter estimation alternative for fitting nonlinear mixed effects pharmacometric models within the R environment.

Example nlmixr/SAEM code

```
library(nlmixr)
datr <- read.csv("BOLUS_1CPT.csv", header=TRUE)
datr$EVID <- ifelse(datr$EVID==1,101,datr$EVID)
odel <- "d/dt(centr) = -(CL/V)*centr;"
mypar <- function(lcl, lv) {
  CL <- exp(lcl)
  V <- exp(lv)
}
m1 = RxODE(odel, modelName="m1")
PRED = function() centr / V
saem_fit <- gen_saem(user_fn(model=m1, PKpars=mypar, pred=PRED)
model = list(saem_mod=saem_fit, res.mod=2)
inits = list(theta=c(4, 70), omega=c(0.1,0.1), bres=0.2)
cfg = configsaem(model, datr, inits)
fit = saem_fit(cfg)
```

References

- <https://github.com/nlmixrdevelopment/nlmixr>
- Wang W *et al.* A Tutorial on RxODE. CPT:PSP (2016) 5, 3–10.
- Pinheiro J *et al.* (2016). nlme: Linear and Nonlinear Mixed Effects Models [R package]
- Kuhn E and Lavielle M. M. Comput Stat Data An, 49:1020–1038, 2005.
- Beal SL *et al.* 1989–2011. NONMEM Users Guides. Icon Development Solutions, USA.
- Laveille C *et al.* PAGE 17 (2008) Abstr 1356 [www.pagemeeting.org/?abstract=1356]