Supplementary material

S1: The "true" model

Covariate effects and interindividual variability (IIV) between subjects was included using the model parameters as follows:

$$PAR_i = TVPAR \cdot PARCOV_i \cdot exp(\eta_{PAR,i})$$

where PAR_i is the value of PAR (k_a, CL, V_2, V_3 or Q) in individual i, $TVPAR$ is the typical value of PAR in the population, and $\eta_{PAR,i}$ is IIV in PAR for individual i, defined as being normally distributed with mean 0 and variance ω^2_{PAR}. $PARCOV_i$ was defined as follows for the effects of body weight and sex:

$$PARCOV_i = \left(\frac{WT_i}{70}\right)^{\theta_{WT,PAR}}$$

for body weight; and

$$PARCOV_i = \begin{cases} 1 & \text{for male subjects, and} \\ 1 + \theta_{SEX,PAR} & \text{for female subjects.} \end{cases}$$

Here, WT_i is body weight for individual i, $\theta_{WT,PAR}$ is the power coefficient describing the effect of weight on parameter PAR, and $\theta_{SEX,PAR}$ is a factor describing the proportional effect of female sex on PAR.

Table S1: Structural parameter values in the model used for simulation

Parameter	Value	IIV [%CV]
Clearance (CL, L/h)	0.121	0.0625 [25%]
Central volume of distribution (V_2, L)	1.939	0.0625 [25%]
Peripheral volume of distribution (V_3, L)	5.650	0.0625 [25%]
Intercompartmental clearance (Q, L/h)	0.282	0.0900 [30%]
First-order absorption rate constant (k_a)	1.050	0.0900 [30%]
Effect of body weight on CL (power model)	0.750	–
Effect of sex (female) on V_2 (proportional model)	0.200	–

IIV=interindividual variability; %CV=percentage coefficient of variation.

Parameter values used for simulation are provided in Table S1, and the IIV variance-covariance matrix ω is defined as below:

$$\omega = \begin{bmatrix} \omega_{CL}^2 & & & & \\ \omega_{CL,V_2} & \omega_{V_2}^2 & & & \\ \omega_{CL,V_3} & \omega_{V_2,V_3} & \omega_{V_3}^2 & & \\ \omega_{CL,Q} & \omega_{V_2,Q} & \omega_{V_3,Q} & \omega_Q^2 & \\ \omega_{CL,k_a} & \omega_{V_2,k_a} & \omega_{V_3,k_a} & \omega_{Q,k_a} & \omega_{k_a}^2 \end{bmatrix} = \begin{bmatrix} 0.06250 & & & & \\ 0.03125 & 0.06250 & & & \\ 0.01563 & 0.03125 & 0.06250 & & \\ 0.00750 & 0.00750 & 0.00750 & 0.09000 & \\ 0.00000 & 0.00000 & 0.00000 & 0.00000 & 0.09000 \end{bmatrix}$$

Correlations were included between model parameters in order to emulate a real-world PK as closely as possible. Proportional residual error was applied to simulated observations, with a value of 7.5%.

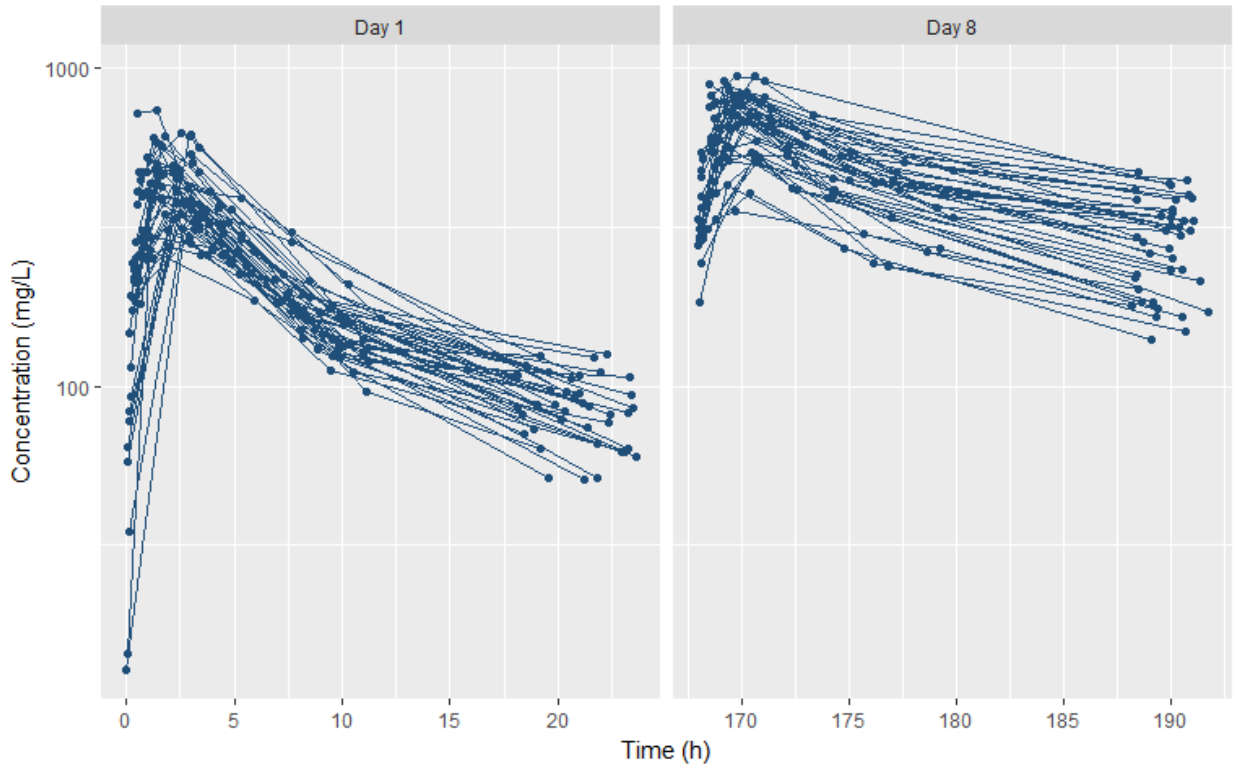
For each simulated patient, concentrations were simulated for the following time windows, with daily doses of 1200 mg:

- 0-1 h after first dose;
- 1-3 h after first dose;
- 3-6 h after first dose;
- 6-12 h after first dose;
- 168-169 h after first dose;

- 169-171 h after first dose;
- 171-180 h after first dose; and
- 180-192 h after first dose.

Sampling times were allocated at random within these time windows to generate a dataset with 360 observations (8 per subject). Body weights were generated at random for each subjects with a mean of 70 kg and a standard deviation of 0.15. Sex was assigned at random (50% male, 50% female).

Figure S1: Simulated concentration plotted against time, by day

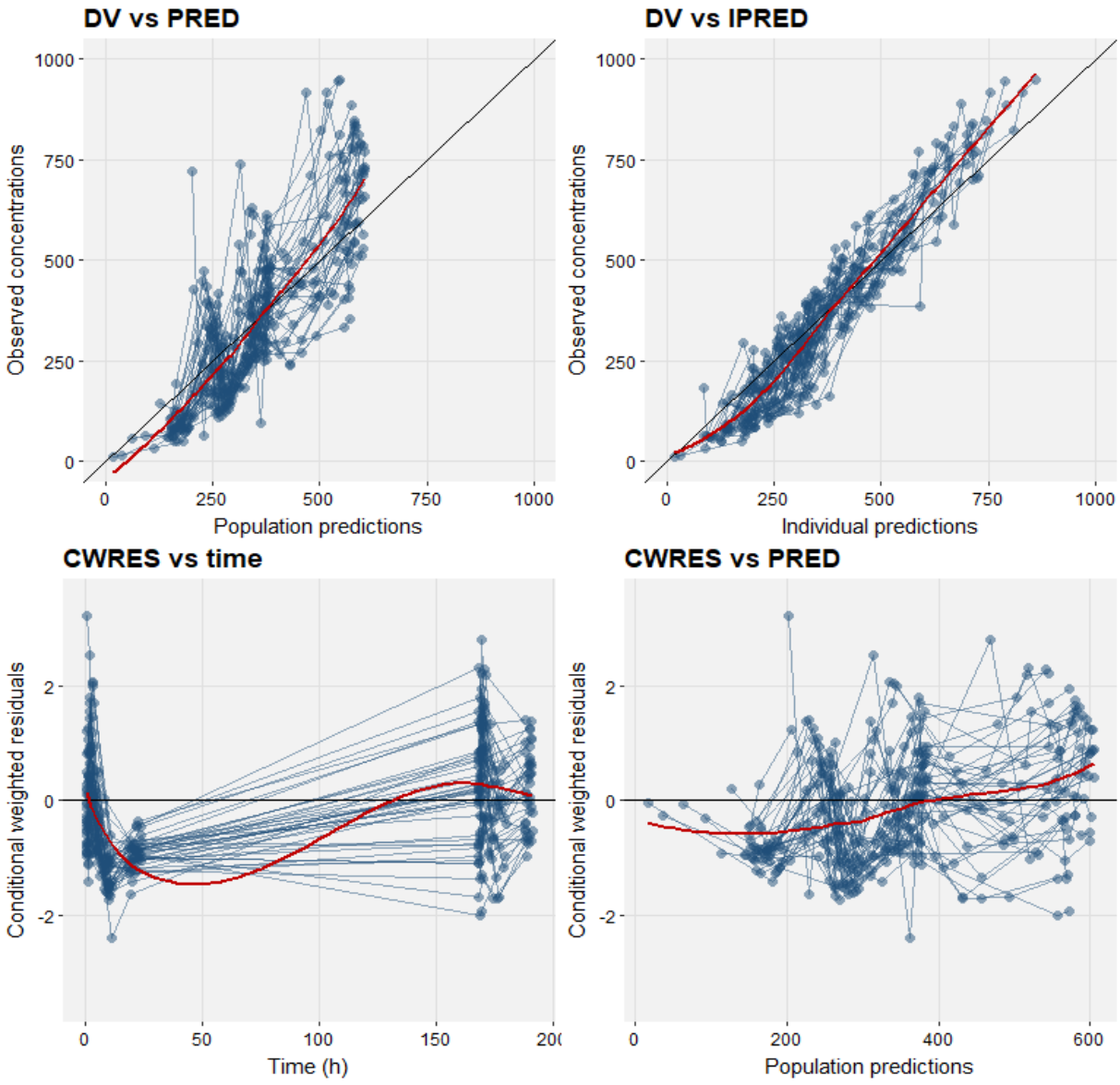


Points are observations. Lines represent individuals.

S2: Intermediate models and output

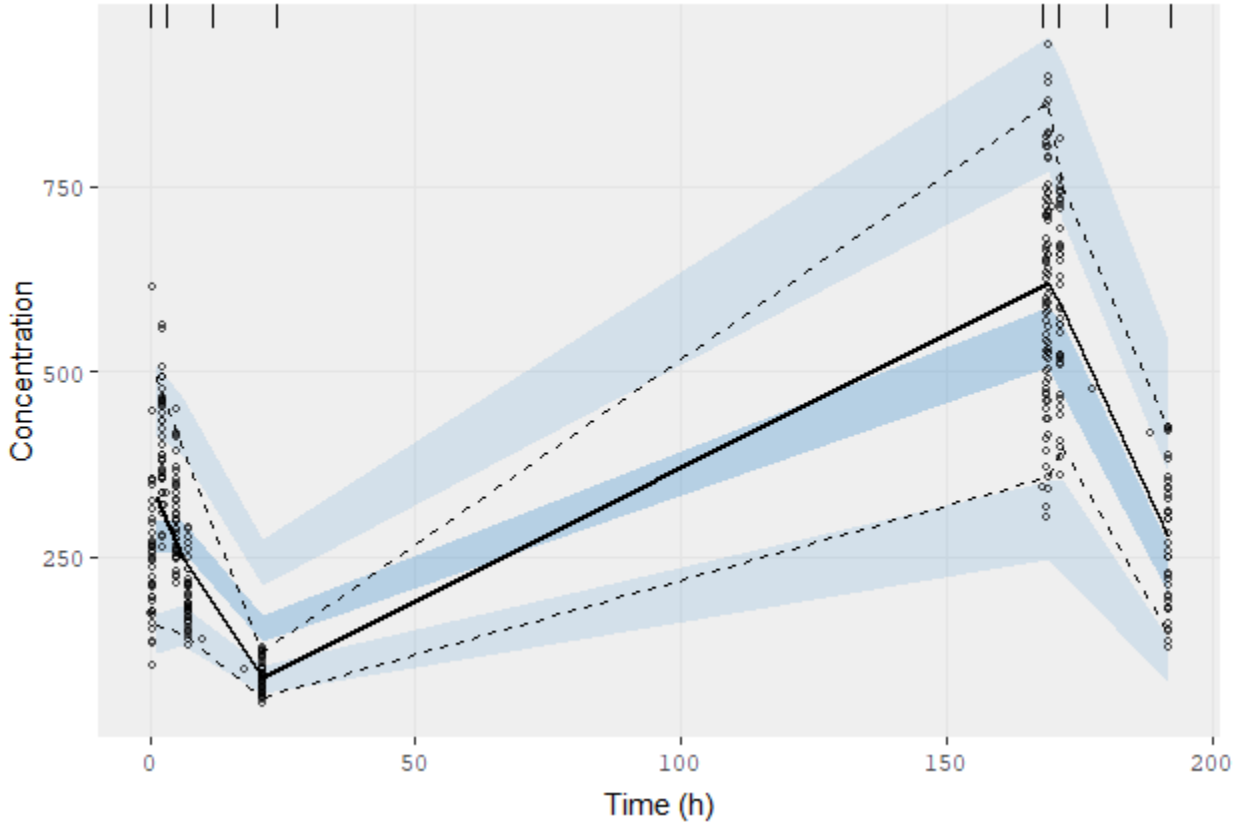
Diagnostic plots for the one-compartment model are shown in Figure S2. A visual predictive check is provided in Figure S3.

Figure S2: Basic goodness-of-fit plots for the one-compartment model



Blue points are observations. Blue lines represent individuals. Red lines are loess smooths through the data. Black lines are lines of identity.

Figure S3: Visual predictive check for the one-compartment model



Black points are original observations. Solid black line is the observed median, by bin. Dashed black lines represent observed 5th and 95th percentiles, by bin. Blue shaded areas represent 90% confidence intervals around simulated 5th, 50th and 95th percentiles. Binning at 0, 3, 12, 24, 168, 171, 180 and 192 hours. N=400 iterations.

Two-compartment model with additive residual error:

```
model.2cpt.ode <- function() {
  ini({
    tka <- log(1.05)
    tcl <- log(0.121)
    tv2 <- log(1.939)
    tv3 <- log(5.65)
    tq  <- log(0.282)
    eta.ka ~ 0.1
    eta.cl ~ 0.1
    eta.v2 ~ 0.1
    eta.v3 ~ 0.1
```

```
    eta.q ~ 0.1
    add.err <- 75
  })
  model({
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v2 <- exp(tv2 + eta.v2)
    v3 <- exp(tv3 + eta.v3)
    q  <- exp(tq + eta.q)
    d/dt(depot) = -ka * depot
    d/dt(center) = ka * depot - cl / v2 * center + q/v3 * periph - q/v2 * center
    d/dt(periph) = q/v2 * center - q/v3 * periph
    cp = center / v2
    cp ~ add(add.err)
  })
}

> fit.2cpt.ode.saem
-- nlmixr SAEM(ODE); OBJF not calculated fit -----------------------------------
          OBJF      AIC      BIC Log-likelihood Condition Number
FOCEi 2975.608 3659.243 3701.991      -1818.622         8.200009

-- Time (sec; fit.2cpt.ode.saem$time): -----------------------------------------
   saem  setup optimize covariance table
 673.63 89.835        0          0  0.25

-- Population Parameters (fit.2cpt.ode.saem$parFixed or $parFixedDf): ----------
         Est.     SE %RSE Back-transformed(95%CI) BSV(CV%) Shrink(SD)%
tka     0.104 0.0659 63.1      1.11 (0.976, 1.26)     39.5      7.66%
tcl      -2.1 0.0486 2.31    0.123 (0.111, 0.135)     30.9    -0.132%
tv2     0.657 0.0382 5.81       1.93 (1.79, 2.08)     21.0      11.6%
tv3      1.68 0.0254 1.51        5.35 (5.1, 5.63)     9.18      83.2%
tq      -1.31  0.052 3.97     0.27 (0.244, 0.299)     27.7      13.7%
add.err  19.8                                19.8

  Covariance Type (fit.2cpt.ode.saem$covMethod): fim
  No correlations in between subject variability (BSV) matrix
  Full BSV covariance (fit.2cpt.ode.saem$omega)
    or correlation (fit.2cpt.ode.saem$omegaR; diagonals=SDs)
  Distribution stats (mean/skewness/kurtosis/p-value) available in $shrink

-- Fit Data (object fit.2cpt.ode.saem is a modified tibble): -------------------
# A tibble: 360 x 30
  ID     TIME    DV  EVID  PRED    RES    WRES IPRED   IRES  IWRES CPRED   CRES
  <fct> <dbl> <dbl> <int> <dbl>  <dbl>   <dbl> <dbl>  <dbl>  <dbl> <dbl>  <dbl>
1 1     0.302  175.     0  172.   3.18  0.0213  168.   6.60  0.333  172.   2.46
2 1     2.91   320.     0  399. -79.0   -0.529   339. -18.9   -0.955  399. -79.7
3 1     3.14   336.     0  387. -50.8   -0.341   328.   8.27  0.418  388. -51.4
# ... with 357 more rows, and 18 more variables: CWRES <dbl>, eta.ka <dbl>,
#   eta.cl <dbl>, eta.v2 <dbl>, eta.v3 <dbl>, eta.q <dbl>, ka <dbl>, cl <dbl>,
#   v2 <dbl>, v3 <dbl>, q <dbl>, cp <dbl>, depot <dbl>, center <dbl>,
#   periph <dbl>, EPRED <dbl>, ERES <dbl>, NPDE <dbl>
```

Two-compartment model with proportional residual error:

```
model.2cptp.ode <- function() {
  ini({
    tka <- log(1.05)
    tcl <- log(0.121)
    tv2 <- log(1.939)
    tv3 <- log(5.65)
    tq  <- log(0.282)
    eta.ka ~ 0.1
    eta.cl ~ 0.1
    eta.v2 ~ 0.1
    eta.v3 ~ 0.1
    eta.q ~ 0.1
    prop.err <- 0.075
  })
  model({
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v2 <- exp(tv2 + eta.v2)
    v3 <- exp(tv3 + eta.v3)
    q  <- exp(tq + eta.q)
    d/dt(depot) = -ka * depot
    d/dt(center) = ka * depot - cl / v2 * center + q/v3 * periph - q/v2 * center
    d/dt(periph) = q/v2 * center - q/v3 * periph
    cp = center / v2
    cp ~ prop(prop.err)
  })
}
> fit.2cptp.ode.saem
-- nlmixr SAEM(ODE); OBJF calculated from FOCEi approximation fit --------------
         OBJF      AIC      BIC Log-likelihood Condition Number
FOCEi 2899.71 3583.346 3626.093      -1780.673         3.792522

-- Time (sec; fit.2cptp.ode.saem$time): ----------------------------------------
   saem setup optimize covariance table
 451.64 0.491        0          0  0.18

-- Population Parameters (fit.2cptp.ode.saem$parFixed or $parFixedDf): ---------
           Est.     SE %RSE Back-transformed(95%CI) BSV(CV%) Shrink(SD)%
tka       0.087 0.0619 71.1      1.09 (0.966, 1.23)    37.7%      5.15%
tcl        -2.1 0.0464 2.21    0.122 (0.112, 0.134)    29.9%    -0.455%
tv2       0.612 0.0344 5.63       1.84 (1.72, 1.97)    19.7%      9.69%
tv3         1.6 0.0322 2.01        4.98 (4.67, 5.3)    18.6%      13.2%
tq        -1.22 0.0505 4.12    0.294 (0.267, 0.325)    31.4%      3.85%
prop.err 0.0481                                0.0481

  Covariance Type (fit.2cptp.ode.saem$covMethod): fim
  No correlations in between subject variability (BSV) matrix
  Full BSV covariance (fit.2cptp.ode.saem$omega)
    or correlation (fit.2cptp.ode.saem$omegaR; diagonals=SDs)
  Distribution stats (mean/skewness/kurtosis/p-value) available in $shrink

-- Fit Data (object fit.2cptp.ode.saem is a modified tibble): ------------------
```

```
# A tibble: 360 x 29
  ID     TIME    DV  PRED    RES    WRES IPRED    IRES   IWRES CPRED   CRES
* <fct> <dbl> <dbl> <dbl>  <dbl>   <dbl> <dbl>   <dbl>   <dbl> <dbl>  <dbl>
1 1     0.302  175.  176.  -1.51 -0.0574  174.   0.809  0.0966  177.  -2.34
2 1     2.91   320.  401. -81.2  -0.598   337. -17.8   -1.10    402. -82.7
3 1     3.14   336.  388. -52.0  -0.408   326.  10.7    0.682   389. -53.3
# ... with 357 more rows, and 18 more variables: CWRES <dbl>, eta.ka <dbl>,
#   eta.cl <dbl>, eta.v2 <dbl>, eta.v3 <dbl>, eta.q <dbl>, ka <dbl>, cl <dbl>,
#   v2 <dbl>, v3 <dbl>, q <dbl>, cp <dbl>, depot <dbl>, center <dbl>,
#   periph <dbl>, EPRED <dbl>, ERES <dbl>, NPDE <dbl>
```

Three-compartment model with proportional residual error:

```
model.3cpt.ode <- function() {
  ini({
    tka <- log(1.42)
    tcl <- log(0.044)
    tv2  <- log(2)
    tv3  <- log(10)
    tv4  <- log(10)
    tq2   <- log(0.5)
    tq3   <- log(0.5)
    eta.ka ~ 0.1
    eta.cl ~ 0.1
    eta.v2 ~ 0.1
    eta.v3 ~ 0.1
    eta.v4 ~ 0.1
    eta.q2 ~ 0.1
    eta.q3 ~ 0.1
    prop.err <- 0.075
  })
  model({
    ka <- exp(tka + eta.ka)
    cl <- exp(tcl + eta.cl)
    v2 <- exp(tv2 + eta.v2)
    v3 <- exp(tv3 + eta.v3)
    v4 <- exp(tv4 + eta.v4)
    q2  <- exp(tq2 + eta.q2)
    q3  <- exp(tq3 + eta.q3)
    d/dt(depot) = -ka * depot
    d/dt(center) = ka * depot - cl / v2 * center + q2/v3 * periph1 - q2/v2 * center + q3/v4 *
periph2 - q3/v2 * center
    d/dt(periph1) = q2/v2 * center - q2/v3 * periph1
    d/dt(periph2) = q3/v2 * center - q3/v4 * periph2
    cp = center / v2
    cp ~ prop(prop.err)
  })
}
> fit.3cpt.ode.saem
-- nlmixr SAEM(ODE); OBJF not calculated fit -----------------------------------
          OBJF      AIC      BIC Log-likelihood Condition Number
FOCEi 2911.829 3603.465 3661.757      -1786.733         73.30701

-- Time (sec; fit.3cpt.ode.saem$time): -----------------------------------------
   saem   setup optimize covariance table
 589.77 101.516        0          0  0.23

-- Population Parameters (fit.3cpt.ode.saem$parFixed or $parFixedDf): ----------
           Est.     SE %RSE Back-transformed(95%CI) BSV(CV%) Shrink(SD)%
tka       0.081 0.0566 69.8      1.08 (0.971, 1.21)     35.8      6.24%
tcl       -2.42 0.0575 2.38  0.089 (0.0795, 0.0996)     35.3      8.57%
tv2       0.605 0.0332 5.49       1.83 (1.72, 1.95)     20.3      11.1%
tv3        2.76   0.22    8       15.7 (10.2, 24.2)     142.      45.2%
tv4        1.38 0.0272 1.98       3.96 (3.75, 4.17)     11.7      54.4%
tq2       -2.85 0.0745 2.61   0.0578 (0.05, 0.0669)     43.7      31.3%
tq3       -1.32 0.0593 4.49      0.267 (0.238, 0.3)     36.9      5.43%
```

```
prop.err 0.0469                                0.0469

  Covariance Type (fit.3cpt.ode.saem$covMethod): fim
  No correlations in between subject variability (BSV) matrix
  Full BSV covariance (fit.3cpt.ode.saem$omega)
    or correlation (fit.3cpt.ode.saem$omegaR; diagonals=SDs)
  Distribution stats (mean/skewness/kurtosis/p-value) available in $shrink

-- Fit Data (object fit.3cpt.ode.saem is a modified tibble): --------------------
# A tibble: 360 x 35
  ID     TIME    DV  EVID  PRED    RES    WRES IPRED    IRES   IWRES CPRED
  <fct> <dbl> <dbl> <int> <dbl>  <dbl>   <dbl> <dbl>   <dbl>   <dbl> <dbl>
1 1     0.302  175.     0  177.  -1.80 -0.0755  174.   0.587  0.0718  177.
2 1     2.91   320.     0  404. -84.4  -0.678   341. -21.1   -1.32    403.
3 1     3.14   336.     0  391. -55.2  -0.473   329.   7.61   0.494   390.
# ... with 357 more rows, and 24 more variables: CRES <dbl>, CWRES <dbl>,
#   eta.ka <dbl>, eta.cl <dbl>, eta.v2 <dbl>, eta.v3 <dbl>, eta.v4 <dbl>,
#   eta.q2 <dbl>, eta.q3 <dbl>, ka <dbl>, cl <dbl>, v2 <dbl>, v3 <dbl>,
#   v4 <dbl>, q2 <dbl>, q3 <dbl>, cp <dbl>, depot <dbl>, center <dbl>,
#   periph1 <dbl>, periph2 <dbl>, EPRED <dbl>, ERES <dbl>, NPDE <dbl>
```

Two-compartment model with proportional residual error and weight on CL:

```
model.2cpt.ode.wtcl <- function() {
  ini({
    tka <- log(1.14)
    tcl <- log(0.0190)
    tv2  <- log(2.12)
    tv3  <- log(20.4)
    tq   <- log(0.383)
    wteff <- 0.35
    eta.ka ~ 1
    eta.cl ~ 1
    eta.v2 ~ 1
    eta.v3 ~ 1
    eta.q ~ 1
    prop.err <- 0.075
  })
  model({
    ka = exp(tka + eta.ka)
    cl = exp(tcl + wteff*lnWT + eta.cl)
    v2 = exp(tv2 + eta.v2)
    v3 = exp(tv3 + eta.v3)
    q  = exp(tq + eta.q)
    d/dt(depot) = -ka * depot
    d/dt(center) = ka * depot - cl / v2 * center + q/v3 * periph - q/v2 * center
    d/dt(periph) = q/v2 * center - q/v3 * periph
    cp = center / v2
    cp ~ prop(prop.err)
  })
}
> fit.2cpt.ode.wtcl.saem
-- nlmixr SAEM(ODE); OBJF not calculated fit -----------------------------------
        OBJF      AIC      BIC Log-likelihood Condition Number
FOCEi 2876.8 3562.436 3609.069      -1769.218         48.85183

-- Time (sec; fit.2cpt.ode.wtcl.saem$time): ------------------------------------
   saem  setup optimize covariance table
 409.92 69.635        0          0  0.23

-- Population Parameters (fit.2cpt.ode.wtcl.saem$parFixed or $parFixedDf): -----
           Est.     SE %RSE Back-transformed(95%CI) BSV(CV%) Shrink(SD)%
tka       0.129  0.056 43.4       1.14 (1.02, 1.27)     33.7      5.49%
tcl       -2.03 0.0402 1.98    0.132 (0.122, 0.142)     22.3     0.342%
tv2       0.645  0.033 5.12       1.91 (1.79, 2.03)     20.9      8.80%
tv3        1.61 0.0328 2.04       5.02 (4.71, 5.35)     19.2      15.6%
tq        -1.26 0.0502 3.99    0.284 (0.258, 0.314)     32.4      3.62%
wteff     0.875  0.224 25.6        2.4 (1.55, 3.72)
prop.err 0.0483                             0.0483

  Covariance Type (fit.2cpt.ode.wtcl.saem$covMethod): fim
  No correlations in between subject variability (BSV) matrix
  Full BSV covariance (fit.2cpt.ode.wtcl.saem$omega)
    or correlation (fit.2cpt.ode.wtcl.saem$omegaR; diagonals=SDs)
  Distribution stats (mean/skewness/kurtosis/p-value) available in $shrink
```

```
-- Fit Data (object fit.2cpt.ode.wtcl.saem is a modified tibble): --------------
# A tibble: 360 x 31
  ID     TIME    DV  EVID   lnWT  PRED    RES   WRES IPRED    IRES   IWRES CPRED
  <fct> <dbl> <dbl> <int>  <dbl> <dbl>  <dbl>  <dbl> <dbl>   <dbl>   <dbl> <dbl>
1 1     0.302  175.     0 -0.297  177.  -2.48 -0.103  174.   0.657  0.0780  178.
2 1     2.91   320.     0 -0.297  406. -86.4  -0.684  336. -16.5   -1.02    404.
3 1     3.14   336.     0 -0.297  394. -58.2  -0.488  324.  11.7    0.748   392.
# ... with 357 more rows, and 19 more variables: CRES <dbl>, CWRES <dbl>,
#   eta.ka <dbl>, eta.cl <dbl>, eta.v2 <dbl>, eta.v3 <dbl>, eta.q <dbl>,
#   ka <dbl>, cl <dbl>, v2 <dbl>, v3 <dbl>, q <dbl>, cp <dbl>, depot <dbl>,
#   center <dbl>, periph <dbl>, EPRED <dbl>, ERES <dbl>, NPDE <dbl>
```

Two-compartment model with proportional residual error and sex on V_2:

```
model.2cpt.ode.sexv2 <- function() {
  ini({
    tka <- log(1.14)
    tcl <- log(0.0190)
    tv2  <- log(2.12)
    tv3  <- log(20.4)
    tq   <- log(0.383)
    sexeff <- -0.2
    eta.ka ~ 1
    eta.cl ~ 1
    eta.v2 ~ 1
    eta.v3 ~ 1
    eta.q ~ 1
    prop.err <- 0.075
  })
  model({
    ka = exp(tka + eta.ka)
    cl = exp(tcl + eta.cl)
    v2 = exp(tv2 + sexeff*(SEX) + eta.v2)
    v3 = exp(tv3 + eta.v3)
    q  = exp(tq + eta.q)
    d/dt(depot) = -ka * depot
    d/dt(center) = ka * depot - cl / v2 * center + q/v3 * periph - q/v2 * center
    d/dt(periph) = q/v2 * center - q/v3 * periph
    cp = center / v2
    cp ~ prop(prop.err)
  })
}
> fit.2cpt.ode.sexv2.saem
-- nlmixr SAEM(ODE); OBJF not calculated fit -----------------------------------
          OBJF      AIC      BIC Log-likelihood Condition Number
FOCEi 2894.738 3580.374 3627.007      -1778.187         735.6578

-- Time (sec; fit.2cpt.ode.sexv2.saem$time): -----------------------------------
   saem  setup optimize covariance table
 627.42 68.036        0          0  0.14

-- Population Parameters (fit.2cpt.ode.sexv2.saem$parFixed or $parFixedDf): ----
           Est.      SE  %RSE Back-transformed(95%CI) BSV(CV%) Shrink(SD)%
tka       0.106  0.0556  52.3      1.11 (0.997, 1.24)     34.1      5.42%
tcl        -2.1  0.0466  2.21    0.122 (0.111, 0.134)     29.9    -0.448%
tv2       0.718  0.0345  4.81       2.05 (1.92, 2.19)     19.6      16.2%
tv3        1.61  0.0519  3.22       5.01 (4.53, 5.55)     18.8      13.0%
tq        -1.24 0.00209 0.169     0.289 (0.287, 0.29)     32.3      4.04%
sexeff   -0.158  0.0324  20.5     0.854 (0.801, 0.91)
prop.err 0.0479                                0.0479

  Covariance Type (fit.2cpt.ode.sexv2.saem$covMethod): fim
  No correlations in between subject variability (BSV) matrix
  Full BSV covariance (fit.2cpt.ode.sexv2.saem$omega)
    or correlation (fit.2cpt.ode.sexv2.saem$omegaR; diagonals=SDs)
  Distribution stats (mean/skewness/kurtosis/p-value) available in $shrink
```

```
-- Fit Data (object fit.2cpt.ode.sexv2.saem is a modified tibble): -------------
# A tibble: 360 x 31
  ID     TIME    DV  EVID   SEX  PRED   RES   WRES IPRED    IRES   IWRES CPRED
  <fct> <dbl> <dbl> <int> <dbl> <dbl> <dbl>  <dbl> <dbl>   <dbl>   <dbl> <dbl>
1 1     0.302  175.     0     0  162.  13.1  0.658  174.   0.671  0.0804  162.
2 1     2.91   320.     0     0  378. -58.8 -0.540  338. -18.0   -1.12    384.
3 1     3.14   336.     0     0  368. -31.7 -0.308  326.  10.4    0.669   373.
# ... with 357 more rows, and 19 more variables: CRES <dbl>, CWRES <dbl>,
#   eta.ka <dbl>, eta.cl <dbl>, eta.v2 <dbl>, eta.v3 <dbl>, eta.q <dbl>,
#   ka <dbl>, cl <dbl>, v2 <dbl>, v3 <dbl>, q <dbl>, cp <dbl>, depot <dbl>,
#   center <dbl>, periph <dbl>, EPRED <dbl>, ERES <dbl>, NPDE <dbl>
```

The final model fitted using FOCE with interaction shows a higher OFV (after removing IIV on V_3 and Q for reasons of stability), but parameter estimates are close to the "true" values and those obtained from SAEM:

```
> fit.2cpt.ode.wtcl.sexv2.foce
-- nlmixr FOCEi (outer: nlminb) fit ---------------------------------------------
          OBJF      AIC      BIC Log-likelihood Condition Number
FOCEi 3019.554 3703.189 3745.937      -1840.595         43.57929

-- Time (sec; fit.2cpt.ode.wtcl.sexv2.foce$time): -------------------------------
         setup optimize covariance table
elapsed 73.741  128.545    128.545   0.2

-- Population Parameters ($parFixed or $parFixedDf): ----------------------------
           Est.     SE %RSE Back-transformed(95%CI) BSV(CV%) Shrink(SD)%
tka       0.196 0.0615 31.4       1.22 (1.08, 1.37)     47.1      6.10%
tcl       -2.02 0.0389 1.93    0.132 (0.123, 0.143)     21.7      1.94%
tv2       0.789 0.0754 9.56         2.2 (1.9, 2.55)     29.4      5.36%
tv3        1.55 0.0394 2.53       4.73 (4.38, 5.11)
tq        -1.34 0.0542 4.04    0.262 (0.235, 0.291)
wteff      0.83  0.117 14.1       2.29 (1.82, 2.88)
sexeff   -0.158  0.102 64.4       0.854 (0.7, 1.04)
prop.err 0.0809                             0.0809

  Covariance Type (fit.2cpt.ode.wtcl.sexv2.foce$covMethod): r,s
  No correlations in between subject variability (BSV) matrix
  Full BSV covariance (fit.2cpt.ode.wtcl.sexv2.foce$omega)
    or correlation (fit.2cpt.ode.wtcl.sexv2.foce$omegaR; diagonals=SDs)
  Distribution stats (mean/skewness/kurtosis/p-value) available in $shrink
  Minimization message (fit.2cpt.ode.wtcl.sexv2.foce$message):
    false convergence (8)
  In an ODE system, false convergence may mean "useless" evaluations were performed.
  See https://stackoverflow.com/questions/40039114/r-nlminb-what-does-false-convergenc
e-actually-mean

-- Fit Data (object fit.2cpt.ode.wtcl.sexv2.foce is a modified tibble): --------
# A tibble: 360 x 30
  ID     TIME    DV  EVID   SEX   lnWT  PRED   RES    WRES IPRED   IRES   IWRES
  <fct> <dbl> <dbl> <int> <dbl>  <dbl> <dbl> <dbl>   <dbl> <dbl>  <dbl>   <dbl>
1 1     0.302  175.     0     0 -0.297  163.  11.6  0.149   174.   1.14  0.0808
2 1     2.91   320.     0     0 -0.297  378. -58.2 -0.139   331. -11.8  -0.440
3 1     3.14   336.     0     0 -0.297  369. -32.4 -0.0816  323.  12.8   0.491
# ... with 357 more rows, and 18 more variables: CPRED <dbl>, CRES <dbl>,
#   CWRES <dbl>, eta.ka <dbl>, eta.cl <dbl>, eta.v2 <dbl>, rx1c <dbl>,
#   ka <dbl>, cl <dbl>, v2 <dbl>, v3 <dbl>, q <dbl>, depot <dbl>,
#   central <dbl>, peripheral <dbl>, EPRED <dbl>, ERES <dbl>, NPDE <dbl>
```

S3: nlmixr tuning options and support functions

Table S2 shows the most useful control parameters for the nlme algorithm. These are not usually required but can be useful. An example illustrating usage is shown below:

```
nlmixr(fit, data,
       est = "nlme",
       control = nlmeControl(maxIter=100,
                             niterEm=50))
```

Additional information may be obtained from the R online help (`?nlmeControl`).

Table S2: Commonly-used control options for the nlme algorithm in nlmixr

Parameter	Purpose
`maxIter`	Maximum number of iterations (default 50)
`pnlsMaxIter`	Maximum iterations for PNLS optimization (default 7)
`msMaxIter`	Maximum number of iterations for NLM optimization (default 50)
`minScale`	Minimum scaling factor for PNLS (default 0.001)
`tolerance`	Convergence tolerance (default $1e^{-6}$)
`niterEm`	Iterations for EM step (default 25)
`pnlsTol`	Tolerance for PNLS step (default $1e^{-3}$)
`msTol`	Tolerance for MS step (default $1e^{-7}$)
`returnObject`	Return object after unsuccessful convergence (default FALSE)
`msVerbose`	Show trace details for NLM (default FALSE)
`apVar`	Calculate approximate covariance matrix (default TRUE)
`.relStep`	Relative step size for numerical derivatives
`minAbsParApVar`	Minimum absolute parameter value in the approximate variance calculation (default 0.05)
`opt`	Optimizer (`'nlminb'` [default] or `'nlm'`)
`natural`	Use natural parameterization for general positive definite matrices (`pdSymm`) in `reStruct` (default TRUE)
`sigma`	Fixed residual error. Calculate if `NULL` (default) or 0

Table S3 shows the most useful control parameters for the SAEM algorithm. These are most commonly used for tuning the number of iterations used and the number of Markov chains. An example illustrating usage is shown below:

```
nlmixr(fit, data,
       est = "saem",
       control = saemControl(seed=740727,
                             n.burn=250,
                             n.em=250))
```

Additional information may be obtained from the R online help (`?saemControl`).

Table S3: Commonly-used control options for the SAEM algorithm in nlmixr

Parameter	Purpose
`seed`	Random seed (default 99)
`n.burn`	Number of iterations in the stochastic approximation (burn-in) step (default 200)
`n.em`	Number of iterations in the expectation maximization step (default 300)
`nmc`	Number of Markov chains (default 3)
`nu`	Numbers of transitions of kernels used in the Metropolis-Hastings algorithm. Default is `c(2,2,2)`, representing 40 for each transition initially (each is multiplied by 20)
`atol`	Absolute convergence tolerance (default $1e^{-8}$)
`rtol`	Relative convergence tolerance (default $1e^{-6}$)
`stiff`	Flag for stiff ODE systems (uses LSODA, default TRUE)
`transit_abs`	Flag for transit absorption model (default TRUE)
`print`	Iterations to be completed before printing an update to the console (default 1)

Table S4 shows the most useful control parameters for the FOCEi family of algorithms. These are most commonly used for tuning the outer optimization method and the ODE/convergence criterion (based on significant digits [sigdig], though this is not equivalent to NONMEM's SIGDIG). An example illustrating usage is shown below:

```
nlmixr(fit, data,
       est = "focei",
       control = foceiControl(outerOpt="bobyqa", sigdig=3))
```

Additional information may be obtained from the R online help (`?foceiControl`).

Table S4: Commonly-used control options for the FOCEi family of algorithms (fo, foi, foce and focei) in nlmixr

Parameter	Purpose
outerOpt	Outer Optimization Routine (the default is "nlmimb", but it can be others like "bobyqa", or "lbfgsb3")
Sigdig	Controls tolerances of estimation and ODE solving routines. Not the same as NONMEM sigidig parameter but with similar meaning (the default is 4)
scaleC	Custom scaling for focei optimization If left blank, these are scaled based on the parsed parameter type and initial estimates. This can be manually adjusted to provide a better estimation when scaling causes problems.
covMethod	Covariance method "r,s" uses sandwich matrix, "r" uses hessian matrix and "s" uses cross-product matrix. "" does not calculate covariance/standard errors
maxOuterIterations	Maximum number of outer iterations before stopping overall parameter estimation
maxInnerIterations	Maximum number of inner iterations before stopping individual eta estimation.

Table S5 provides a summary of the most useful fit object properties provided by nlmixr (accessed using `fit$property`, where `property` is the property of interest).

Table S5: nlmixr fit object properties (accessed by fit$property)

Property	Purpose
`$par.hist`	Parameter history, if available
`$par.hist.stacked`	Stacked parameter history, if available
`$parFixed`	Fixed-effect parameter table
`$eta`	Individual random effects
`$omega`	Between-subject covariance matrix
`$omegaR`	Between-subject correlation matrix
`$covMethod`	Covariance method
`$model.name`	Model name
`$data.name`	Name of input data frame
`$shrink`	Distribution statistics, including shrinkage
$time	Time used for steps in fitting process

Table S6 shows the parameter combinations recognized by the linCmt() function when assigning a solved-system PK model.

Table S6: Parameter combinations recognized by linCmt() to infer solved systems

1-Compartment PK	2-compartment PK	3-compartment PK
CL, V	CL, Vc, Q, Vp	CL, Vc, Q, Vp1, Vp2
Kel, V	Kel, k12, k21, V	Kel, k12, k21, k13, k31, V
A, alpha	A, alpha, B, beta	A, alpha, B, beta, C, gamma

Parameter aliases are also supported, e.g. V=Vc=V1 for 1-compartment model. Case does not matter. Parameter aliases are context dependent; the first can be volume = Vc (can start with V2); second numbered volume = Vp. All NONMEM style parameters are supported. Where ka is present, first-order oral absorption is inferred; compartment 1 is then depot.

S4: shinyMixR functions

Table S7 provides a summary of key shinyMixR functions.

Table S7: shinyMixR functions

Function	Purpose
`create_proj()`	Create folder structure for a shinyMixR project
`run_nmx()`	Run an nlmixr model fit
`overview()`	Create overview of all models in a project
`tree_overview()`	Create collapsible tree overview for visualizing relationships between models
`gof_plot()`	Create general goodness-of-fit plots
`fit_plot()`	Create individual plots
`get_proj()`	Get project information with available models and high-level results

S5: Required R packages

Package [version]	Details	Recommended source
Core packages		
nlmixr [1.1.0-3]	Fidler ML, Xiong Y, Schoemaker R, Wilkins JJ, Trame M, Post TM, Wang W (2018). nlmixr: Nonlinear Mixed Effects Models in Population Pharmacokinetics and Pharmacodynamics.	github.com/ nlmixrdevelopment/nlmixr
RxODE [0.9.0-2]	Fidler ML, Hallow M, Wang W (2018). RxODE: Facilities for Simulating from ODE-Based Models.	github.com/ nlmixrdevelopment/RxODE
shinyMixR [0.1.4]	Hooijmaijers R, Post TM (2018). shinyMixR: Shiny dashboard interface for nlmixr	github.com/ RichardHooijmaijers/ shinyMixR
Support packages		
*lbfgsb3c** [2018-2.13]	Fidler ML, Nash JC, Zhu C, Byrd R, Nocedal J, Morales JL (2018). Limited Memory BFGS Minimizer with Bounds on Parameters with optim() 'C' Interface.	github.com/ nlmixrdevelopment/lbfgsb3c
n1qn1 [6.0.1-3]	Fidler M, Wang W, Lemarechal C, Bonnans J, Gilbert JC, Sagastizabal C, Campbell SL, Chancelier JP, Nikoukhah R, Eddelbuettel D, Jofret B, INRIA (2018). Port of the 'Scilab' 'n1qn1' and 'qnbd' Modules for (Un)constrained BFGS Optimization.	github.com/ nlmixrdevelopment/n1qn1
PreciseSums [0.3]	Fidler ML, Hettinger R, Shewchuk J, Taylor J, Smith N (2018). PreciseSums: Accurate Floating Point Sums and Products.	github.com/ nlmixrdevelopment/ PreciseSums
*SnakeCharmR*** [1.0.7]	Sieira A [forked from rPython by Gil Bellosta CJ] (2017). SnakeCharmR: R and Python Integration.	github.com/ nlmixrdevelopment/ SnakeCharmR
vpc [1.1.0]	Keizer R (2018). vpc: Create Visual Predictive Checks.	github.com/ ronkeizer/vpc
xpose [0.4.4]	Guiastrennec B, Hooker AC, Olofsson A, Ueckert S, Keizer R, Karlsson MO	github.com/ UUPharmacometrics/xpose

	(2018). xpose: Diagnostics for Pharmacometric Models.	
xpose.nlmixr [0.1.4]	Wilkins JJ, Fidler ML, Guiastrennec B, Hooker AC, Olofsson A, Ueckert S, Keizer R, Karlsson MO (2018). xpose.nlmixr: Graphical Diagnostics for Pharmacometric Models: Extension to 'nlmixr'.	github.com/ nlmixrdevelopment/ xpose.nlmixr

** Forked from github.com/cran/lbfgsb3*
*** Forked from github.com/asieira/SnakeCharmR*

Additional direct support packages are required by nlmixr and are available on the Comprehensive R Archive Network (CRAN): brew, cli, collapsibleTree, crayon, devtools, digest, dparser, DT, fastGHQuad, ggplot2, gridExtra, inline, lbfgs, magrittr, memoise, minqa, mvnfast, numDeriv, R.utils, Rcpp, RcppArmadillo, rex, shiny, shinyAce, shinyBS, shinydashboard, shinyFiles, stringi, tidyr, whisker, and all dependencies. It is recommended to use the most recent available versions.