Simulation from ODE-based population PK/PD and systems pharmacology models in R with mrgsolve

Kyle T. Baron and Marc R. Gastonguay

Metrum Research Group LLC; Tariffville, CT USA

METRUM RESEARCH GROUP

ABSTRACT

Objective

To develop computer software facilitating large-scale simulation from hierarchical, ordinary differential equation (ODE) based models typically employed in drug development.

Methods

Specific software requirements were specified to enable a modern and efficient simulation platform. A C++ interface between R and the ODEPACK solver DLSODA was developed using Rcpp. C++ classes were developed to abstract solver setup, data sets and records, and PK dosing events. S4 classes and methods were created to represent the model in R as an updatable object. The modeler creates a model specification file consisting of R and C++ code that is parsed, compiled, and dynamically loaded into the R session. Input data are passed in and simulated data are returned as R objects, so disk access is never required during the simulation cycle after compiling.

Results

The following software requirements were met:

- NMTRAN-like input data sets
- Bolus, infusion, compartment on/off and reset functionality
- Bioavailability, ALAG, SS, II, ADDL, MTIME
- Multivariate normal random effects simulated using RcppArmadillo
- Compatible with parameter estimation and design packages in R (nlme, saemix, PopED, PFIM)
- Integration with data summary (dplyr) and plotting (ggplot, lattice) packages
- Parallelization with existing R infrastructure (mclapply) or Sun Grid Engine (qapply)
- Compatible with output from many different model estimation platforms
- Easily integrated with Shiny to create model-visualization applications

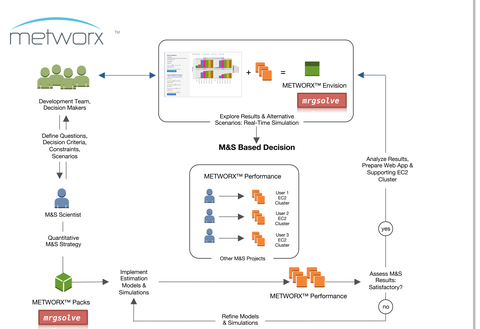
Benchmark

Simulation of 4- and 26-week trials of 1000 patients with daily oral dosing as an indirect response model with 1-compartment PK took 1.35 and 9.01 seconds (average, 10 replicates), respectively, on MacBookPro 2.4GHz

Conclusions

mrgsolve is a powerful and efficient tool for simulation from ODE-based PK/PD and systems pharmacology models. The resulting computational efficiency facilitates model exploration and application, both during model development and decision-making phases of a drug development program.

M&S Based Decision Support w/EC2, Interactive App, & Packs Content



REFERENCES

- A. C. Hindmarsh, "ODEPACK, A Systematized Collection of ODE Solvers," in Scientific Computing, R. S. Stepleman et al. (eds.), North-Holland, Amsterdam, 1983, pp.55-64.
- L. Rong, H. Dahari, R. M. Ribeiro, and A. S. Perelson. Rapid emergence of protease inhibitor resistance in hepatitis c virus. Sci Transl Med, 2(30):30ra32, May 2010.

OBTAINING mrgsolve

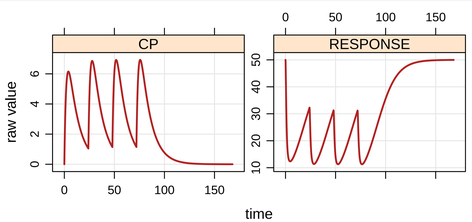
http://www.metrumrg.com/ *click on* **Open Source Tools**

MODEL specification file [saved in pkpd.cpp]

```
$PROB
A simple model to get started.
Date: September 24, 2015

$PARAM  KA=0.5, CL=1, VC=10
KIN=100, KOUT=2, IC50=2, FBIO=0.92, TLAG=0

$CMT GUT CENT RESPONSE

$SET end=36, delta=0.5

$GLOBAL
#define CP  (CENT/VCi)
#define INH  (CP/(CP+IC50))

$MAIN
RESPONSE_0 = KIN/KOUT;

double CLi = CL*exp(ETA(1));
double VCi = VC*exp(ETA(2));
double KAi = KA*exp(ETA(3));

_ALAG(1) = TLAG; _F(1) = FBIO;

$OMEGA 0 0
$OMEGA 0
$SIGMA 0

$ODE
dxdt_GUT = -KA*GUT;
dxdt_CENT = KA*GUT - (CLi/VCi)*CENT;
dxdt_RESPONSE = KIN*(1-INH) - KOUT*RESPONSE;

$CAPTURE CP

$TABLE table(GOAL) = int(RESPONSE<25);
```

EXAMPLE: Basic PK/PD model simulation

```
library(mrgsolve)

mod <- mread("pkpd")

out <- mod %>%
  ev(amt=100, ii=24, addl=3) %>%
  mrgsim(delta=0.1, end=168, digits=4)

head(out)

. Model:  pkpd

.      ID time   GUT   CENT RESPONSE     CP GOAL
. [1,]  1  0.0  0.00  0.000    50.00 0.0000    0
. [2,]  1  0.0  0.00  0.000    50.00 0.0000    0
. [3,]  1  0.1 87.51  4.464    49.08 0.4464    0
. [4,]  1  0.2 83.25  8.667    47.00 0.8667    0
. [5,]  1  0.3 79.19 12.620    44.39 1.2620    0
. [6,]  1  0.4 75.32 16.340    41.59 1.6340    0

plot(out,CP+RESPONSE~., col="firebrick")
```



EXAMPLE: Simulate population pharmacokinetic data

```
library(mrgsolve)
```

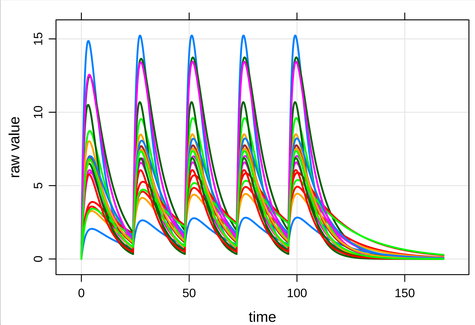
Read the model specification file and update the model object

- We also update the OMEGA matrix with omat

```
mod <- mread("pkpd") %>%
  omat(list(cmat(0.1, 0.8, 0.3),dmat(0.5)))

mod

.
.
. -------- mrgsolve model object (unix) --------
.   Project: /Users/kyleb/dev/m4solve/acop6
.   model:          pkpd
.   source:         pkpd.cpp
.   compile date:   10/01 10:05
.   dll loaded:     TRUE
.
.   Time:           start: 0 end: 36 delta: 0.5
.   >               add: <none>
.   >               tscale: 1
.
.   Compartments:   GUT CENT RESPONSE [3]
.   Parameters:     KA CL VC KIN KOUT IC50
.   >               FBIO TLAG [8]
.   Omega:          2x2,1x1
.   Sigma:          1x1
.
.   Solver:         atol: 1e-08 rtol: 1e-08
.   >               maxsteps: 2000 hmin: 0 hmax: 0

omat(mod)

. $...
.           [,1]      [,2]
. 1:  0.1000000 0.1385641
. 2:  0.1385641 0.3000000
.
. $...
.     [,1]
. 3:   0.5
```

Simulate with events object: dose 100 units Q 24 hours x 5 doses

```
set.seed(20202)
out <- mod %>% param(TLAG=0) %>%
  ev(amt=100, ii=24, addl=4) %>%
  idata_set(data.frame(ID=1:20)) %>%
  mrgsim(delta=0.1, end=168,Req="CP")

plot(out)
```



EXAMPLE: Sensitivity analysis on mechanistic, two-strain viral dynamic model

```
library(mrgsolve)

mod <- mread("viral2")
```

Selected model specification blocks

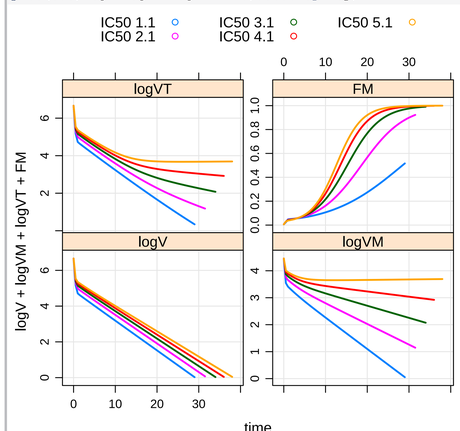
```
blocks(mod,GLOBAL,MAIN,ODE)

.
. Model file: viral2.cpp
.
. $GLOBAL
. #define a (s+rho*T_0*(1-((T_0+N)/Tmax))-d*T_0)
. #define b1 (beta*T_0 *((1-fit)/(1-mu-fit)))
. #define b2 (1+(rho*T_0)/(delta*Tmax))
. #define b (b1*b2)
. #define eps (expos/(IC50+expos))
. #define epsm (expos/(IC50*shift+expos))
.
. $MAIN
. T_0 = c*delta/((1-mu)*beta*p);
. V_0 = a/b;
. I_0 = (beta/delta)*V_0 * T_0;
. VM_0 = V_0*(mu)/(1-mu-fit);
. IM_0 = (beta/delta) * VM_0 * T_0;
. expos_0 = AUC;
. bool cure = false;
.
. $ODE
. dxdt_T = s+rho*T*(1-((T+I+IM+N)/Tmax))-d*T- beta*T*(V+VM);
. dxdt_I = beta*V*T - delta*I;
. dxdt_V = (1-mu)*(1-eps)*p*I - c*V;
. dxdt_IM = beta*VM*T - delta*IM;
. dxdt_VM = mu*(1-eps)*p*I + (1-epsm)*fit*p*IM - c*VM;
. dxdt_expos = 0;
```

Sensitivity analysis on IC50

```
out <- mod %>%
  Req(logVT,logV,logVM,FM) %>%
  param(AUC=75,step=12,fit=0.9) %>%
  knobs(IC50 = seq(1.1,5.1,1)) %>% limit(logV>0)
```

Plot: **V: wild type, VM: mutant, VT: total virus (V+VM), FM: Fraction Mutant**

```
plot(out,logV+logVM+logVT+FM~.,auto.key=key)
```



WORKFLOW & BENCHMARK: Simulation of population viral kinetics and SVR12

```
library(mrgsolve);library(dplyr)
library(tidyr);library(ggplot2)

mod <- mread("viralpop") %>%  update(end=168, delta=3.5)
```

Sustained viral response

The patient has a sustained viral response (SVR12) when the viral load is less than the quantitation limit 12 weeks after the end of therapy.

```
ql <- log10(22.5)  ## copies/mL
```

Create template data set to simulate 24 week trial of 1000 subjects at 4 AUC levels, 9 mutant combinations

```
data <- expand.ev(ID=1:1000,
                  amt=c(10,40,70,100),
                  shift=c(6,10,14),
                  fit=c(0.2,0.6,0.9)) %>% mutate(AUC=amt)

data <- data %>%
  mutate(time=84,amt=0,evid=4) %>%
  bind_rows(data) %>%
  arrange(ID,time)

filter(data, ID %in% c(1,2)) %>% as.data.frame

.   ID amt shift fit evid cmt time AUC
. 1  1  10     6 0.2    1   1    0  10
. 2  1   0     6 0.2    4   1   84  10
. 3  2  10     6 0.2    1   1    0  10
. 4  2   0     6 0.2    4   1   84  10
```

Simulate

```
set.seed(2020442)
system.time({
out <- mod %>% obsonly %>%
  carry.out(AUC,fit,shift) %>%
  data_set(data) %>%
  mrgsim(atol=1E-100,seed=1010134,Req="logVT")
})

.    user  system elapsed
.  11.494   0.129  11.694

Nrow(out)

. [1] "1,764,000"
```

Summarize

```
sum <- out %>% as.tbl %>%
  filter(time==168) %>%
  group_by(shift,fit,AUC) %>%
  summarise(SVR12 = mean(logVT<ql)) %>%
  ungroup %>% as.data.frame %>%
  mutate(shift = nfact(shift, "shift="))
```

© Metrum Research Group LLC 2015