Simulating with Parameter Uncertainty

September 27, 2012

Bill Knebel
Tim Bergsma

1 Purpose

This script shows how to conduct a simulation that considers uncertainty in the parameter estimates.

2 Data

Here we load metrumrg and read in the data to be used for simulations.

Listing 1:

```
> library(metrumrg)
> data <- read.csv("../data/derived/phase1.csv")
> head(data)
```

```
  C ID TIME SEQ EVID  AMT    DV SUBJ HOUR TAFD  TAD LDOS MDV HEIGHT WEIGHT SEX
1 C  1 0.00   0    0    .     0    1 0.00 0.00    .    .   0    174   74.2   0
2 .  1 0.00   1    1 1000     .    1 0.00 0.00    0 1000   1    174   74.2   0
3 .  1 0.25   0    0    . 0.363    1 0.25 0.25 0.25 1000   0    174   74.2   0
4 .  1 0.50   0    0    . 0.914    1 0.50 0.50  0.5 1000   0    174   74.2   0
5 .  1 1.00   0    0    .  1.12    1 1.00 1.00    1 1000   0    174   74.2   0
6 .  1 2.00   0    0    .  2.28    1 2.00 2.00    2 1000   0    174   74.2   0
   AGE DOSE FED SMK DS CRCN predose zerodv
1 29.1 1000   1   0  0 83.5       1      0
2 29.1 1000   1   0  0 83.5       0      0
3 29.1 1000   1   0  0 83.5       0      0
4 29.1 1000   1   0  0 83.5       0      0
5 29.1 1000   1   0  0 83.5       0      0
6 29.1 1000   1   0  0 83.5       0      0
```

We use NONMEM output from a simple two compartment model to generate parameters. We use 1005.lst and 1005.cov output from NM7 to populate a call to metrumrg::simpar().

Listing 2:

```
> cov <- read.table("../nonmem/1005/1005.cov", skip=1, header=T)
> head(cov)
```

```
    NAME     THETA1      THETA2       THETA3      THETA4      THETA5
1 THETA1  0.8759870  0.79288000  1.06070e-03  0.06301820  -1.7590100
2 THETA2  0.7928800  4.74648000  6.68044e-03  0.89631100   5.2850000
3 THETA3  0.0010607  0.00668044  2.75847e-05  0.00222162  -0.0305675
4 THETA4  0.0630182  0.89631100  2.22162e-03  0.28693100   0.1902840
5 THETA5 -1.7590100  5.28500000 -3.05675e-02  0.19028400 566.1060000
6 THETA6 -0.0421234 -0.02569100 -1.05405e-04 -0.01064250   0.7806910
        THETA6       THETA7   SIGMA.1.1. SIGMA.2.1.   SIGMA.2.2.  OMEGA.1.1.
1 -0.042123400 -0.180383000 -5.30926e-04          0  2.09474e-02 6.20306e-03
2 -0.025691000  0.066978500 -3.13015e-03          0  1.92041e-02 5.76993e-03
3 -0.000105405 -0.000138306 -1.02726e-05          0  5.91125e-05 3.21603e-06
4 -0.010642500  0.015518300 -6.29448e-04          0  2.54028e-03 4.29694e-03
5  0.780691000 -0.602816000  4.56339e-02          0 -4.26533e-01 2.73809e-01
6  0.013065500  0.000798277  1.21261e-04          0 -1.07146e-03 1.64777e-03
    OMEGA.2.1.   OMEGA.2.2.   OMEGA.3.1.   OMEGA.3.2.   OMEGA.3.3.
1 -1.83821e-04 -4.40561e-03 -5.44206e-03 -2.59575e-03 -3.35118e-03
2 -2.19650e-02 -2.44862e-02 -1.96039e-02 -1.12215e-02  4.77949e-03
3 -6.50573e-05 -7.81727e-05 -6.76632e-05 -2.75979e-05  2.83104e-05
4 -6.21366e-03 -7.78974e-03 -4.55505e-03 -2.25175e-03  3.07354e-03
5  1.60570e-01  2.81746e-02 -4.85574e-03  7.48252e-02 -3.43426e-02
6  3.02269e-04  6.01990e-04 -5.29743e-04 -5.31172e-05 -3.35780e-04
```

We are interested in theta covariance, so we remove extra columns and rows.

Listing 3:

```
> cov<- cov[1:7,c(2:8)]
```

3 Parameters

Now we generate 10 sets of population parameters based on the 1005.lst results.

Listing 4:

```
> set.seed(10)
> PKparms <- simpar(
+     nsim=10,
+     theta=c(8.58,21.6, 0.0684, 3.78, 107, 0.999, 1.67),
+     covar=cov,
+     omega=list(0.196, 0.129, 0.107),
+     odf=c(40,40,40),
+     sigma=list(0.0671),
+     sdf=c(200)
+ )
> PKparms
```

```
     TH.1  TH.2    TH.3  TH.4   TH.5   TH.6  TH.7  OM1.1   OM2.2   OM3.3
1   9.472 24.04 0.06312 3.508 106.50 1.0140 1.589 0.1847 0.15400 0.13630
2  10.740 22.97 0.06794 3.814 111.40 0.8452 1.296 0.2862 0.12000 0.16400
3   9.001 21.22 0.06626 3.966 139.60 1.0780 1.512 0.1647 0.12770 0.11300
4  10.680 23.59 0.07180 3.906 121.20 1.1480 1.009 0.1886 0.11460 0.08460
5  10.030 23.02 0.06999 3.573  99.98 0.9437 1.762 0.1526 0.08448 0.13140
6   8.964 21.73 0.06877 3.343  97.72 1.0600 1.806 0.2462 0.17640 0.08805
7   8.630 19.70 0.06611 3.377 135.80 0.9626 1.348 0.2221 0.14440 0.09957
8   9.213 21.23 0.06096 3.082 115.70 1.0420 1.752 0.2287 0.13820 0.06118
9   8.974 23.95 0.06980 4.110 145.70 0.9455 1.782 0.1765 0.12310 0.08504
10  8.939 22.72 0.06436 3.704 113.10 1.0610 1.581 0.2116 0.11940 0.09954
     SG1.1
1  0.06894
2  0.06099
3  0.06041
4  0.07700
```

```
5  0.06269
6  0.07274
7  0.06160
8  0.06692
9  0.06092
10 0.06269
```

4 Control Streams

We read in a control stream and clean out extra xml markup.

Listing 5:

```
> ctl <- as.nmctl(readLines("../nonmem/ctl/1005.ctl"))
> ctl[] <- lapply(ctl,function(rec)sub("<.*","",rec))
```

Now we iterate across the rows of PKparms, writing out a separate ctl for each.

Listing 6:

```
> dir.create('../nonmem/sim')
> set <- lapply(
+       rownames(PKparms),
+       function(row,params,ctl){
+               params <- as.character(PKparms[row,])
+               ctl$prob <- sub(1005,row,ctl$prob)
+               ctl$theta <- params[1:7]
+               ctl$omega <- params[8:10]
+               ctl$sigma <- params[11]
+               names(ctl)[names(ctl)=='estimation'] <- 'simulation'
+               ctl$simulation <- paste(
+                       '(',
+                       as.numeric(row) + 7995,
```

```
+                       'NEW) (',
+                       as.numeric(row) + 8996,
+                       'UNIFORM) ONLYSIMULATION'
+               )
+               ctl$cov <- NULL
+               ctl$table <- NULL
+               ctl$table <- NULL
+               ctl$table <- 'ID TIME DV WT SEX LDOS NOPRINT NOAPPEND FILE=sim.tab'
+               write.nmctl(ctl,file=file.path('../nonmem/sim',paste(sep='.',row,'ctl')))
+               return(ctl)
+       },
+       params=PKparms,
+       ctl=ctl
+ )
```

5 Simulation

Finally, we run NONMEM simulations using NONR.

Listing 7:

```
> NONR72(
+       run=1:10,
+       command="/opt/NONMEM/nm72/nmqual/autolog.pl",
+       project="../nonmem/sim",
+       diag=FALSE,
+       checkrunno=FALSE,
+       grid=TRUE
+ )
```