Simulating with Parameter Uncertainty

February 4, 2014

Bill Knebel
Tim Bergsma

1 Purpose

This script shows how to conduct a simulation that considers uncertainty in the parameter estimates. See also http://www.page-meeting.org/page/page2006/P2006III_11.pdf.

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 HEIGHT WEIGHT SEX  AGE DOSE FED SMK
1 C  1 0.00   0    0    .     0    1 0.00    174   74.2   0 29.1 1000   1   0
2 .  1 0.00   1    1 1000     .    1 0.00    174   74.2   0 29.1 1000   1   0
3 .  1 0.25   0    0    . 0.363    1 0.25    174   74.2   0 29.1 1000   1   0
4 .  1 0.50   0    0    . 0.914    1 0.50    174   74.2   0 29.1 1000   1   0
5 .  1 1.00   0    0    .  1.12    1 1.00    174   74.2   0 29.1 1000   1   0
6 .  1 2.00   0    0    .  2.28    1 2.00    174   74.2   0 29.1 1000   1   0
  DS CRCN TAFD  TAD LDOS MDV predose zerodv
1  0 83.5 0.00    .    .   0       1      0
2  0 83.5 0.00    0 1000   1       0      0
3  0 83.5 0.25 0.25 1000   0       0      0
4  0 83.5 0.50  0.5 1000   0       0      0
5  0 83.5 1.00    1 1000   0       0      0
6  0 83.5 2.00    2 1000   0       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  0.85947800  0.7848260  1.05073e-03  0.06297000  -1.6425100
2 THETA2  0.78482600  4.7421000  6.67920e-03  0.89652600   5.3176400
3 THETA3  0.00105073  0.0066792  2.75922e-05  0.00222269  -0.0304355
4 THETA4  0.06297000  0.8965260  2.22269e-03  0.28707800   0.1958110
5 THETA5 -1.64251000  5.3176400 -3.04355e-02  0.19581100 563.8350000
6 THETA6 -0.04113180 -0.0252131 -1.04883e-04 -0.01065710   0.7701760
        THETA6       THETA7   SIGMA.1.1. SIGMA.2.1.   SIGMA.2.2.  OMEGA.1.1.
1 -0.041131800 -0.176199000 -5.18961e-04          0  2.06030e-02 6.09321e-03
2 -0.025213100  0.068704500 -3.12567e-03          0  1.90856e-02 5.73980e-03
3 -0.000104883 -0.000135683 -1.02658e-05          0  5.89818e-05 3.21218e-06
```

```
4 -0.010657100  0.015500000 -6.28838e-04          0  2.53788e-03 4.30468e-03
5  0.770176000 -0.633694000  4.55841e-02          0 -4.24311e-01 2.73913e-01
6  0.013008700  0.000572209  1.20518e-04          0 -1.04929e-03 1.65243e-03
    OMEGA.2.1.   OMEGA.2.2.   OMEGA.3.1.   OMEGA.3.2.   OMEGA.3.3.
1 -2.40140e-04 -4.36000e-03 -5.36432e-03 -2.57895e-03 -3.33572e-03
2 -2.19992e-02 -2.44779e-02 -1.95821e-02 -1.12216e-02  4.78859e-03
3 -6.51294e-05 -7.81938e-05 -6.76593e-05 -2.76252e-05  2.83097e-05
4 -6.21638e-03 -7.79632e-03 -4.55928e-03 -2.25358e-03  3.07430e-03
5  1.60551e-01  2.73617e-02 -5.50834e-03  7.46183e-02 -3.41613e-02
6  3.04227e-04  5.99259e-04 -5.34503e-04 -5.46264e-05 -3.36875e-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   SG1.1
1  7.565 19.23 0.06670 3.882 107.50 1.1020 1.340 0.1847 0.15400 0.13630 0.06894
2  6.531 20.18 0.06637 3.861 102.60 1.0680 2.325 0.2862 0.12000 0.16400 0.06099
3  8.257 21.93 0.06598 3.722  74.43 0.8294 2.140 0.1647 0.12770 0.11300 0.06041
4  6.394 19.65 0.06679 3.521  92.78 0.9400 2.011 0.1886 0.11460 0.08460 0.07700
5  7.266 20.13 0.07281 4.136 114.00 0.9471 1.937 0.1526 0.08448 0.13140 0.06269
6  8.205 21.46 0.07480 4.221 116.30 0.9340 1.544 0.2462 0.17640 0.08805 0.07274
7  8.495 23.50 0.07476 4.147  78.29 1.0610 1.906 0.2221 0.14440 0.09957 0.06160
8  7.988 21.95 0.07318 4.524  98.36 0.9228 1.700 0.2287 0.13820 0.06118 0.06692
9  8.268 19.21 0.07017 3.554  68.39 0.9785 1.814 0.1765 0.12310 0.08504 0.06092
10 8.144 20.51 0.06545 3.754 100.90 1.0090 1.511 0.2116 0.11940 0.09954 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:

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