Simulating with Parameter Uncertainty

September 7, 2011

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)

metrumrg 5.0
```

Listing 2:

```
> 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 3:

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

```
    NAME       THETA1     THETA2       THETA3      THETA4       THETA5
1 THETA1  0.669038000  0.3187200  1.58905e-04  0.03757190   2.59715000
2 THETA2  0.318720000  4.0841800  6.94170e-03  0.69266000   9.96862000
3 THETA3  0.000158905  0.0069417  3.02696e-05  0.00193254  -0.00604366
4 THETA4  0.037571900  0.6926600  1.93254e-03  0.26139800   1.58175000
5 THETA5  2.597150000  9.9686200 -6.04366e-03  1.58175000 283.39800000
6 THETA6 -0.055585600 -0.0248295 -1.00494e-04 -0.02667240  -0.03980440
       THETA6       THETA7   SIGMA.1.1.   OMEGA.1.1. OMEGA.2.1.   OMEGA.2.2.
1 -0.055585600 -0.133741000  1.02030e-03 -7.07190e-04          0 -6.46117e-04
2 -0.024829500  0.187881000 -8.79108e-03  9.36297e-03          0 -1.98732e-02
3 -0.000100494  0.000259341 -2.61526e-05 -8.69484e-06          0 -9.83597e-05
4 -0.026672400  0.044585600 -1.16815e-03  6.89103e-04          0 -4.78282e-03
5 -0.039804400 -0.677987000  1.53154e-02  2.13660e-01          0  3.21359e-02
6  0.021986700 -0.011466100 -9.43146e-05  2.71730e-03          0 -1.45631e-04
  OMEGA.3.1. OMEGA.3.2.   OMEGA.3.3.
1          0          0 -7.29033e-04
2          0          0 -8.34369e-03
3          0          0 -2.35296e-06
4          0          0  2.75930e-03
5          0          0  1.20400e-02
6          0          0 -6.06465e-04
```

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

Listing 4:

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

3 Parameters

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

Listing 5:

```
> 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   8.869 19.32 0.06426 4.117 106.8 0.8772 1.2390 0.1847 0.15400 0.13630
2  10.280 20.16 0.06251 3.439 110.1 0.7905 1.3400 0.2862 0.12000 0.16400
3   9.403 22.91 0.06295 3.583 130.1 1.0810 1.6990 0.1647 0.12770 0.11300
4  10.180 19.99 0.06534 3.444 117.1 1.1330 0.9176 0.1886 0.11460 0.08460
5   9.529 19.84 0.07000 3.896 102.1 0.7982 1.7000 0.1526 0.08448 0.13140
6   8.845 21.08 0.07446 4.225 100.4 0.9269 1.7120 0.2462 0.17640 0.08805
7   9.405 24.17 0.07370 4.071 127.3 0.9100 1.4820 0.2221 0.14440 0.09957
8   9.414 22.03 0.06953 4.473 113.1 0.8243 1.6990 0.2287 0.13820 0.06118
9   8.829 20.76 0.06609 3.679 134.5 0.8774 1.6720 0.1765 0.12310 0.08504
10  8.733 20.77 0.06396 3.913 111.4 1.0090 1.4240 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 6:

```
> 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 7:

```
> 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 8:

```
> NONR72(
+       run=1:10,
+       command="/common/NONMEM/nm7_osxi/test/nm7_osxi.pl",
+       project="../nonmem/sim",
+       diag=FALSE,
+       checkrunno=FALSE,
+       grid=TRUE
+ )
```