MIfuns Sample Script

Simulating with Parameter Uncertainty

October 18, 2010

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 MIfuns and read in the data to be used for simulations.

Listing 1:

```
> library(MIfuns)

MIfuns 4.1.0
```

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      1
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 MIfuns::simpar().

Listing 3:

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

    NAME       THETA1      THETA2       THETA3      THETA4       THETA5
1 THETA1  0.665158000  0.31249200  1.65973e-04  0.02989100   2.13169000
2 THETA2  0.312492000  4.08110000  6.94328e-03  0.69166700   9.76609000
3 THETA3  0.000165973  0.00694328  3.02940e-05  0.00193292  -0.00658463
4 THETA4  0.029891000  0.69166700  1.93292e-03  0.26105200   1.50038000
5 THETA5  2.131690000  9.76609000 -6.58463e-03  1.50038000 283.10500000
6 THETA6 -0.046470200 -0.02244780 -9.69369e-05 -0.02437590   0.05291770
```

```
        THETA6       THETA7   SIGMA.1.1.   OMEGA.1.1. OMEGA.2.1.   OMEGA.2.2.
1 -4.64702e-02 -0.146935000  9.41749e-04 -1.56849e-04          0 -9.04248e-04
2 -2.24478e-02  0.186394000 -8.82373e-03  9.40159e-03          0 -2.00309e-02
3 -9.69369e-05  0.000253729 -2.62223e-05 -8.61550e-06          0 -9.88614e-05
4 -2.43759e-02  0.043642100 -1.18030e-03  6.64550e-04          0 -4.82235e-03
5  5.29177e-02 -0.671658000  1.53099e-02  2.17642e-01          0  3.31492e-02
6  1.86049e-02 -0.009628420 -7.39197e-05  2.54053e-03          0 -1.02414e-04
  OMEGA.3.1. OMEGA.3.2.   OMEGA.3.3.
1          0          0 -9.34269e-04
2          0          0 -8.34612e-03
3          0          0 -2.33533e-06
4          0          0  2.75895e-03
5          0          0  1.11018e-02
6          0          0 -5.50259e-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 OM.1.1.1 OM.2.1.1 OM.3.1.1
1  8.858 19.33 0.06423 4.091 106.8 0.9002 1.1870   0.1847  0.15400  0.13630
2 10.270 20.15 0.06250 3.433 110.1 0.8190 1.2940   0.2862  0.12000  0.16400
3  9.371 22.89 0.06297 3.585 130.1 1.0860 1.7050   0.1647  0.12770  0.11300
4 10.160 19.98 0.06527 3.399 117.1 1.1520 0.8838   0.1886  0.11460  0.08460
5  9.540 19.84 0.07016 3.908 102.1 0.8257 1.6340   0.1526  0.08448  0.13140
6  8.855 21.08 0.07458 4.227 100.4 0.9416 1.6640   0.2462  0.17640  0.08805
7  9.377 24.16 0.07357 4.054 127.3 0.9219 1.4800   0.2221  0.14440  0.09957
8  9.408 22.03 0.06965 4.473 113.1 0.8532 1.6320   0.2287  0.13820  0.06118
9  8.784 20.74 0.06608 3.686 134.4 0.8937 1.6620   0.1765  0.12310  0.08504
```

```
10  8.719 20.77 0.06393 3.896 111.3 1.0180 1.4060   0.2116  0.11940  0.09954
   SG.1.1.1
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.nmcontrol(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.nmcontrol(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:

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