Facilitating the Pharmacometrics Work-Flow with the MItools R Package

William Knebel*[1], Timothy T. Bergsma[1], Jeannine Fisher[1], George Georgalis[1], Leonid Gibiansky[1,2],
William R. Gillespie[1], Matthew M. Riggs[1], and Marc R. Gastonguay[1]
[1]Metrum Institute, Tariffville, CT 06081 USA, [2] Current Address: QuantPharm LLC, North Potomac, MD, USA

BACKGROUND

- A pharmacometrician's typical work-flow requires access to a variety of software tools for different stages (data preparation, modeling, evaluation, simulation and reporting) of the strategic modeling and simulation process.
- This process requires reliable, reproducible and traceable inter-operability between software tools.
- No single tool capable of managing and performing these tasks is publicly available.
- We have developed an R software (version 2.4.0+; www.r-project.org) [1] package, MItools, to facilitate the pharmacometrics work-flow.

METHODS

- All necessary system (operating system, R, Sun Grid Engine) software tools for the computing grid were installed via a set of shell scripts under version control (subversion).
- The R package was developed on a single workstation running Mac OSX or Windows XP and a computer grid, with each node running NetBSD 3.1 (64bit) on AMD64 processors and the GNU Fortran Compiler, G95 or GCC-3.3.3 (GNU Project, http://www.GNU.org/).
- NMQual 6.1.1 (Metrum Institute) was used to install, qualify, and track all code patches/options for NONMEM® Version V and VI (ICON Development Solutions).

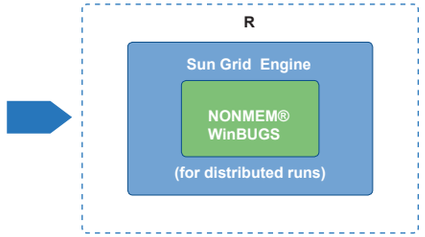
MItools PACKAGE REQUIREMENTS

- Platform independent
- Controlled, reproducible environment for:
  - Data acquisition and exploration
  - Initial model building and diagnostics
  - Model checking and evaluation
  - Modeling and simulation based inferences
  - Preparation of figures and tables for reporting
- Modular nature allowing for the easy addition of new/revised functions
- Allow for the sourcing/execution of user-defined R scripts/functions
- Customization of R-scripts would be required for certain aspects of every project
- ASCII text controlled/scripted interface
  - Minimal bandwidth requirements for remote operations
  - Easy implementation of version control (open-source tools, i.e. subversion)
  - No third-party X-windowing/graphical requirements for remote access

MItools FUNCTIONALITY

- MItools consists of 50 + functions
- The software flow control for a typical pharmacometric analysis is shown in Figure 1.
- An outline of the pharmacometric workflow and the applicable MItools functions is shown in Figure 2.
- An example R script for a pharmacometric analysis is shown in Figure 3.

Figure 1. Software Flow Control



- csv dataset
- INFN routine (typical code for $PRED and PREDPP models)
- ASCII model code
  - NONMEM® control stream
  - BUGS model script
  - R script

- NONMEM® ASCII output
- NONMEM® Runlog (csv file)
- ASCII tables
- Plots as pdf or png files (user customized R script)
- WinBUGS output

Figure 2. Pharmacometrics Work Flow

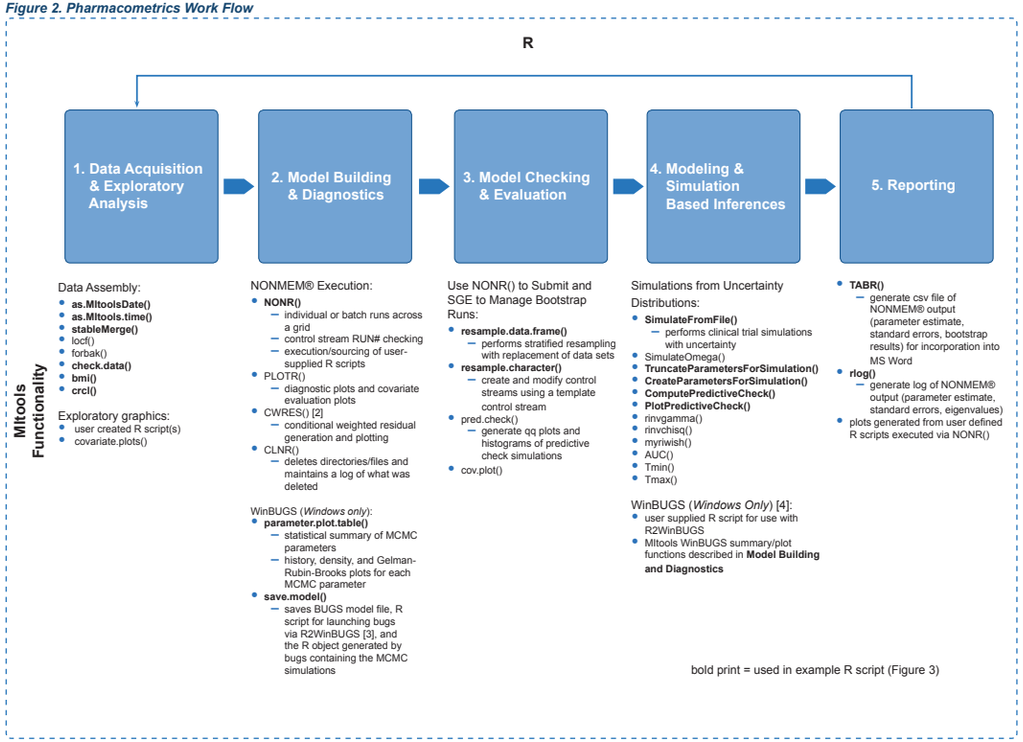


Figure 3. Example R Script

1. Data Acquisition and Exploratory Analysis

```
library(MItools)
library(lattice)

# read in raw data files, format for NONMEM analysis, and explore graphically
dataPRE<-read.csv("/data/PK", header=T, sep=",",quote="", comment.char="", flush=T);
dataPD<-read.csv("/data/PD", header=T, sep=",",quote="", comment.char="", flush=T);
dataCOV<-read.csv("/data/COV", header=T, sep=",",quote="", comment.char="", flush=T);

dim(dataPK)# 133 5
dim(dataPD)# 240 5
dim(dataCOV)# 10 6

names(dataPK) # ID DATE TIME CONC DOSE
names(dataPD) # ID DATE TIME PDMEAS BASE
names(dataCOV) # ID WT AGE HCR SEX

# calculate Creatinine Clearance and BMI
dataCOV$BMI <- bmi(dataCOV$WT, dataCOV$HT)
dataCOV$CRCL <- crcl(dataCOV$AGE, dataCOV$WT, dataCOV$SEX, dataCOV$SCR)

# convert DATE and TIME into NMTRAN formatted DATE and TIME

# merge covariate data into PK data
PKDS <- stable.merge(dataPK, dataCOV) , c("ID","WT","AGE","BMI","CRCL");

# generate pdf files containing exploratory plots of covariates and
# concentration-time plots
pdf("/data/explore/cov.pdf")

par(mfcol=c(2,2))

cov <- c("AGE", "WT", "CRCL","BMI")
pairs(PKDS.cov[,cov],
      panel=function(x,y){
        points(x,y);
        lines(lowess(x,y))
        }
)
xyplot(CONC~TIME|ID, data=PKDS, xlab="Time", ylab="Conc.")
dev.off();
```

2. Model Building and Diagnostics

```
# run one (1.ctl) and two (2.ctl) compt model and generate diagnostic plots and evaluate covariate relationships

NONR(ProjectDir="/data/PK", b=c(1,2), loqtrans=F, NHome="am4am64c.pl", checkrunno=F,
     boot=0, diag=T, eplog="/data/R/PKPD.lk", dvname="Drug& ng/mL", cmplt=T,
     cont.cov=c("AGE","CRCL","BMI"), cat.cov=c("SEX"), par.list=c("CL","V1","V2",
     eta.list=c("ETA1","ETA2","ETA3"))

# run WinBUGS via R and R2WinBUGS (windows only)
ProjDir = "C:/Data/Bayes"

model.name = "linear1b" # root names of model file
example.dir = "linear" # subdirectory containing model file

setwd(ProjDir)
library(R2WinBUGS)
library(coda)

# create WinBUGS data set
bugs.data = list(
x = c(1,2,3,4,5,6,7,8,9,10),
y = c(5.19,6.54,9.19,8.09,7.6,7.08,6.74,9.3,8.98,11.5)
)

# create initial estimates
fit = lm(y~x,bugs.data)
ms = summary(fit)$coefficients[,2]
sigma = summary(fit)$sigma

bugs.init = function() list(
a = rnorm(1,coef(fit)[1],5*ms[1]),
b = rnorm(1,coef(fit)[2],5*ms[2]),
sigma = exp(rnorm(1,log(sigma),log(sigma)))
)

# specify what variables to monitor and plot
parameters = c("a","b","sigma","ypred")
parameters.to.plot =c("a","b","sigma")

# run WinBUGS and save scripts, data and results
n.chains = 3
n.iter = 1000
n.burnin = 100
n.thin = 1

bugs.fit = bugs(data=bugs.data,inits=bugs.init,
parameters.to.save=parameters,model.
file=paste(getwd(),"/",example.dir,"/",model.name,".txt",sep=""),
n.chains=n.chains,n.iter=n.iter,n.thin=n.thin,n.burnin=n.burnin, clearWD=T)

save.model(bugs.fit,model.name,example.dir)

sims.array = bugs.fit$sims.array
posterior = array(as.vector(sims.array),dim=c(prod(dim(sims.array)[1:2]),dim(sims.array)[3]),
dimnames=list(NULL,dimnames(sims.array)[3]))

# plot posterior distributions

pdf(file = paste(example.dir,"/",model.name,"/",model.name,"_plots.pdf",sep=""),
width=6,height=6)

xl = sims.array[,,unlist(sapply(c(paste("^",parameters.to.plot,"$",sep="")),

paste("^",parameters.to.plot,"\\[",sep="")),grep,x=dimnames(sims.array)[[3]])]

ptable = parameter.plot.table(xl)
write.csv(signif(ptable,3),paste(example.dir,"/",model.name,"/",model.name,"_summary.csv",sep=""));

pred = posterior[,grep("ypred\\[",dimnames(posterior)[[2]])]
xl = data.frame(x=bugs.data$x,y=bugs.data$y)
xltype =rep("observed",nrow(xl))
x2 = rbind(xl,xl,xl)
x2$y = as.vector(t(apply(pred,2,quantile,probs=c(0.05,0.5,0.95))))
x2$type = rep(c("5%ile","median","95%ile"),a=nrow(xl))
x1 = rbind(x1,x2)

xyplot(y~x,xl,groups=type,panel=panel.superpose,
type=c("p","l"),"gr",lty=c(1,3,1,0),cex1=c(2,2,4,1),
pch=c(24,NA,NA,19),cex=1.5,jw=1,
scales=list(cex=1),xlab=list(xlab="x",cex=1.2),
ylab=list(ylab="y",cex=1.2))
dev.off();
```

3. Model Checking and Evaluation

```
# create bootstrap datasets, control streams, and run NONMEM

data <- read.table("/data/derived/PKDs.csv", header=T, sep=",",quote="", comment.char="", flush=T);

ctl <- scan(file = "/data/PK/1.ctl", what = "", comment.char = "", allowEscapes = T,
            sep = "\n", quiet = T)

resample(data, key="ID",
         rekey=TRUE,
         name=1:1000,
         out="./",
         strat2p=list=c(dataSEX, dataSCRCL < 30)
         seed= 2357
)
resample(
         ctl,
         name=1:1000,
         patterns=c(
                  "^$DA 1",
                  "PKDs.csv",
                  "2.MSF"
         ),
         replacement=c(
                  "$DA 1",
                  "*.csv",
                  "*.MSF"
         ),
         out="./",
         suffix=".ctl"
)

NONR(ProjectDir="/data/PK/boot", b=c(1:1000), loqtrans=F, NHome="am4am64c.pl",
     boot=1, checkrunno=F, diag=F, eplog=NULL, dvname="Drug& ng/mL", cmplt=F)

rlog(ProjectDir="/data/PK/boot", b=c(1:1000), boot=1, runlog=2)
```

4. Modeling and Simulation Based Inferences

```
# Perform simulations with uncertainty

DirName <- "/data/sim/"
ThetaMean  <-c(11.8,8)
ThetaCovar <- matrix(c(0.232,0.449,0.449,12.8),2,2)
OmegaModelist <-matrix(c(0.0572,0.011,0.011,0.0615),2,2)
OmegaDflist <-20
SigmaModelist <- 0.0454
SigmaDflist <- 200

set.seed(123)

parameters <- CreateParametersForSimulation(nsim=29,
                 ThetaMean=ThetaMean,ThetaCovar=ThetaCovar,
                 OmegaModelist=OmegaModelist,OmegaDflist=OmegaDflist,
                 SigmaModelist=SigmaModelist,SigmaDflist=SigmaDflist)

bounds <- data.frame(par =c(1,2),lower =c(5,30),upper=c(20,150))
parametersTruncated <- TruncateParametersForSimulation(parameters, bounds)
write.table(parametersTruncated, file=paste(DirName,"Example1Par.csv",sep=""),
            quote = F,sep=",",row.names = F,col.names = F)

SimulateFromFile(DirName,CtlFileName="Example1.ctl",
                 Parameters=parametersTruncated,
                 OutputFileName="Example1Output.tab",
                 ntheta=2,nomegaList=2,nsigmaList=1,
                 NONMEMcommand="perl batch.p sim",
                 SimTabFileName="sim.tab")

# Read results of simulations and generate plots
SimulatedData <- read.table(paste(DirName,"Example1Output.tab",sep=""))
names(SimulatedData) <- c("ID","TIME","DV")
NonmemData <- read.table(paste(DirName,"Example1Data.CSV",sep=""),sep=",",header=T)

PCData <- ComputePredictiveCheck(NonmemData,SimulatedData,
          SimpleFun =c("min","max","mean","median"),
          CombFun = c("Tmin","Tmax","AUC"),
          StatFun = c("min","max","mean","median","Q25","Q75"),
          IDname="ID",TimeName="TIME",DVname="DV",Cname="C",ignore="C")

PlotPredictiveCheck(PCData,
                    Parameters=c("min","max","mean","median","Tmin","Tmax","AUC"),
                    Summaries = c("min","max","mean","median","Q25","Q75"),
                    plot.hist=T,plot.qq=T,DrugName = "DV",plotType="pdf",
                    FileTemplate="Example1PlotAll")

PlotPredictiveCheck(PCData, Parameters=c("AUC"),
                    Summaries = c("mean","median","Q25","Q75"),
                    plot.hist=T,plot.qq=T,DrugName = "DV",plotType="pdf",
                    FileTemplate="Example1PlotAUC")
```

5. Reporting

```
# generate csv file of parameter estimates, standard errors, and bootstrap results for incorporation into report

TABR(ProjectDir="/data/PK/boot", i=1, boot=1, bootres="/data/PK/boot/PKbootres.csv",
     ntoim=c(1:6), ntoi1=7:93,5,all", ntoim2=97:5,all", outname="/Fullfit.csv")

# read in the final model table file and generate some additional diagnostic plots conditioned on STDV variable

datares <- read.table(file="/data/PK/1.TAB",sep="",skip=1,header=T)

pdf("/data/PK/Figures/AddtlDiagRun1.pdf")

xyplot(PRED+DV~TIME|STDV, data=datares, xlab="Time", ylab="Drug& Conc (ng/mL)",
       main="Concentration vs Time by Study")

dev.off();
```

CONCLUSIONS

- The MItools package allows pharmacometricians to work in one common scriptable environment (R software) for data preparation, modeling, model evaluation, modeling/simulation based inferences, and reporting.
- The R packaging system allows for easy distribution of this set of pharmacometric tools and has the added benefit of exposing the user to the R software, a powerful tool for data formatting, data mining/exploration, and statistical analysis.
- This work proves the concept that a production-capable modeling and simulation infrastructure can be implemented entirely using open-source software (with the exception of NONMEM® and the Windows OS).
- The combination of MItools plus an open-source versioning tool (e.g. subversion) results in the ability to track and reproduce an analysis from start to finish, critical goals of 21CFRPart11 compliance.

FUTURE FUNCTIONALITY

- Additional functions for interfacing with WinBUGS/OPENBUGS model analysis, evaluation, and simulation
- Generation of a common log detailing the results of a NONMEM®, WinBUGS/OPENBUGS, and/or R based analysis
- Qualification of MItools R Package is in progress

REFERENCES

1. R Development Core Team (2008). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.
2. Hooker A, Staatz CE, Karlsson MO. Conditional weighted residuals, an improved model diagnostic for the FO/FOCE methods. PAGE 15 (2006) Abstr 1001
3. Sturtz, S., Ligges, U., Gelman, A. (2005): R2WinBUGS: A Package for Running WinBUGS from R. Journal of Statistical Software 12(3), 1-16.
4. D J Spiegelhalter and A Thomas and N G Best (1999). WinBUGS Version 1.2 User Manual MRC Biostatistics Unit.

©2008 Metrum Institute copies of this poster are available at www.metruminstitute.org/publications

NONMEM® is licensed and distributed by ICON Development Solutions