Simulation of clinical trial predictions with model uncertainty using NMsim

Sanaya Shroff, PhD, Philip Delff, PhD

Vertex Pharmaceuticals Incorporated, Boston, MA

Introduction

Clinical Trial Simulation (CTS) is an application of pharmacokinetic (PK) or pharmacokinetic/pharmacodynamic (PK/PD) modeling to assist in the design of clinical trials. NMsim [1,2] can aid in effective CTS and trial design by simple, R-based, simultaneous multi-model simulation in NONMEM. With only paths to control stream(s) and a simulation data set, NMsim can incorporate model parameter uncertainty through several different methods, from a covariance step or bootstrap results, allowing calculations of confidence intervals or covariate effect evaluation (e.g. forest plots). We aim to show these features through examples that are highly relevant in pharmacometrics.

Objectives

Using simple adjustments to NMsim function calls we will demonstrate how to:

- Incorporate model parameter uncertainty from bootstrap results for an example typical subject
- Incorporate model parameter uncertainty from a covariance step for an example typical subject
- Conduct a full CTS using external methods like simpar [3]
- Summarize and compare simulation results from different methods

The seamless workflow of NMsim allows code to be readily applicable to most NONMEM models, the user interface entirely within R.

Table 1: Comparison of simulation methods using NMsim

METHOD	PROS	CONS	TIME
bootstrap	- covariance step not required - all param correlations	- requires bootstrap prior to NMsim - must choose which runs to accept - NONMEM run for each bootstrap trial	- slow - computationally intensive
mvrnorm	- only NMsim call needed - all param correlations	- requires covariance step - NONMEM run for each sampled trial - normally-distributed OMEGA/SIGMA	- slow
NWPRI	- only runs NONMEM once - correct param distributions*	- requires covariance step - only blocked param correlations^	- very fast
simpar	- currently only way to run a full CTS - correct param distributions	- requires setup using simpar - requires covariance step - only blocked param correlations^	- slow - requires facility with simpar

* NONMEM's NWPRI method uses normal distributions for THETAs and inverse-Wishart distributions for OMEGA/SIGMAs, but currently only works on single block OMEGA/SIGMA structures. Otherwise predictions are unreliable

^ simulations reproduce parameter correlations for given blocked OMEGA/SIGMA structures, rather than all parameter correlations

Methods

Simulate parameter uncertainty from bootstrap

In the example below, 1000 bootstrapped models were previously estimated with psn and are then simulated for 1 typical patient at 5 dose levels using NMsim.

```
#read in bootstrapped models
bootstrap.mods <-
    list.files(path=paste0("models/bs_",mod,"/m1"),
               pattern=".mod$",full.names = T,recursive = T)

#NMsim function call
bootstrap.sim <- NMsim(file.mod=bootstrap.mods,
                       data=dt.sim,
                       name.sim="ACOP2024_bootstrap",
                       method.sim=NMsim_default,
                       table.vars="PRED IPRED Y KA V2 V3 CL Q",
                       typical=T,
                       sge=T    #run sims in parallel on cluster
                       )

#read in simulation results
bootstrap.res <- NMreadSim(bootstrap.sim,wait=T)
```

Simulate parameter uncertainty from a covariance step

Normally-distributed parameters (mvrnorm)

In the example below, 1000 sampled models were simulated for 1 typical patient at 5 dose levels using NMsim.

```
mvrnorm.sim <-
  NMsim(file.mod=file.mod,
        data=dt.sim,
        name.sim="ACOP2024_mvrnorm",
        method.sim=NMsim_VarCov,
        table.vars="PRED IPRED Y KA V2 V3 CL Q",
        nsims=1000,
        typical=T,
        sge=T)
mvrnorm.res <- NMreadSim(mvrnorm.sim,wait=T)
```

Normally-distributed THETAs, Inverse-Wishart OMEGA and SIGMA (NWPRI)

In the example below, 1000 sampled models were simulated simultaneously for 1 typical patient at 5 dose levels using NMsim.

```
pars <- NMreadExt(file.mod,return="pars",as.fun="data.table")


nwpri.sim <-
  NMsim(file.mod=file.mod,
        data=dt.sim,
        name.sim="ACOP2024_nwpri",
        method.sim=NMsim_NWPRI,
        table.vars="PRED IPRED Y KA V2 V3 CL Q",
        subproblems=1000,
        sge=T)
nwpri.res <- NMreadSim(nwpri.sim,wait=T)
```

Full clinical trial simulation using simpar then NMsim

In the example below, 1000 sampled parameter sets were simulated using simpar from the estimated variance-covariance matrix. Then using these parameter sets, either 20 or 200 new subjects each at 2 different dose levels were simulated using NMsim.

```
#determine parameter space with simpar
#sampleParsSimpar available in NMsim 0.1.4 (see Appendix)
ext.simpar <- sampleParsSimpar(file.mod=file.mod,nsim=1000)

#run NMsim (separate call for each dataset)
simpar.sim <-
  NMsim(file.mod=file.mod
        ,data=full.dt.sim
        ,ext=ext.simpar
        ,name.sim="ACOP2024_simpar"
        ,method.sim=NMsim_VarCov
        ,table.vars="PRED IPRED Y KA V2 V3 CL Q"
        ,sge=TRUE)

simpar.res <- NMreadSim(simpar,wait=T)
```

Results

Figure 1 describes the median and 95% confidence interval of simulated concentration-time profiles (n = 1000) for a typical subject at each dose level using three of the methods (bootstrap, mvrnorm, NWPRI) described above.

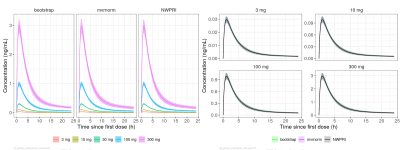

Figure 1: Typical subject (n=1) simulated 1000 times by different NMsim methods

Figure 2 describes the median and 95% confidence interval from two different clinical trial simulations containing 20 or 200 new subjects per dosing level, respectively. Across all simulated trials and individuals, the median simulated profiles without individual or residual variability (PRED) demonstrates the effect of parameter uncertainty only, while profiles including individual variability (IPRED), and or both individual and residual variability (Y) illustrate the combinatorial effect of multiple sources of variability.

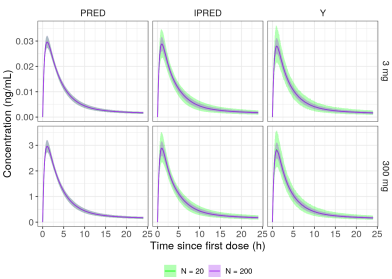

Figure 2: Full clinical trial simulations (n = 20 or 200 subjects per dose, 1000 times) using simpar and NMsim

Table 2 describes the median AUC and 95% confidence interval across all simulated trials and individuals for both CTS. As with the profiles, sources of variability are evident between the different metrics (PRED, IPRED, and Y).

Table 2: Comparison of AUC for two CTS simulated with NMsim

	AUC (ng*hr/mL) [95% CI]			
	N = 20		N = 200	
	3 mg	300 mg	3 mg	300 mg
PRED	0.174 [0.158, 0.191]	17.4 [15.8, 19.1]	0.175 [0.159, 0.191]	17.5 [15.9, 19.1]
IPRED	0.173 [0.137, 0.206]	17.2 [14, 20.8]	0.172 [0.153, 0.191]	17.2 [15.3, 19]
Y	0.173 [0.136, 0.207]	17.2 [14, 20.9]	0.172 [0.152, 0.19]	17.2 [15.3, 19]

When the number of subjects is large (N=200), individual and residual variability have little effect on simulated outcomes, and only the effect of parameter uncertainty is reflected in confidence intervals shown (Figure 2, Table 2). In contrast, with only 20 subjects per trial arm, both individual and residual variability are evident from the simulated outcomes (Figure 2), with individual variability dominating the variability in exposure (Table 2).

Conclusion

The features explored here provide a powerful tool to easily simulate with parameter uncertainty, e.g. simulations for forest plots, or to conduct clinical trial simulation.

Using NMsim, one can simply and effectively conduct simulation with parameter uncertainty based on both non-parametric bootstrap, and parametric methods such as NONMEM's native NWPRI. Externally-sampled parameter values can also easily be fed into NMsim and NMsim even includes automation of parameter sampling with simpar by supplying only the control stream path and number of desired samples.

We hope these examples will aid pharmacometricians in making simulation with uncertainty and clinical trial simulations more easily accessible by automating analyses, and can help support the design of clinical studies during the drug development process.

Supplementary Code and Additional Information:

Related posters at ACoP 2024:

- NMsim - Seamless NONMEM simulation platform in R (**T32**)
- Building automated pharmacometrics analysis workflows in R with NMsim (**T49**)
- Simulate modified NONMEM models using NMsim (**T19**)
- A model-based simulation workflow enables automated and accurate generation of clinical pharmacology summary statistics (**T103**)

References & Appendix

[1] 2024. NMsim: An R package that can simulate Nonmem models. https://philipdelff.github.io/NMsim

[2] 2024. NMsim: Seamless Nonmem Simulation Platform. https://cran.r-project.org/web/packages/NMsim

[3] 2023. simpar. https://mpa.metworx.com/docs/packages/simpar https://github.com/metrumresearchgroup/simpar

```
sampleParsSimpar <- function(file.mod,nsim){
  library(NMsim) ## version 0.1.4 (includes sampleParsSimpar)
  library(NMdata) ## version 0.1.8
  library(simpar) ## version 0.1.1

  ## read param distributions from ext file
  pars <- NMreadExt(file=file.mod,as.fun="data.table")

  ## calculate degrees of freedom
  omega.sigma.dfs <- NMsim:::NWPRI_df(pars)

  ## variance-covariance for THETAs
  covar <- NMreadCov(file=file.mod)
  theta.covar <- covar[grep("^THETA",rownames(covar))
                       ,grep("^THETA",colnames(covar))]

  ## variance-covariance for OMEGAs
  omegas <- pars[par.type=="OMEGA" & !is.na(iblock),]
  omegas.list <- split(omegas,by="iblock")
  omega.mat.list <- lapply(omegas.list,NMdata::dt2mat)

  ## variance-covariance for SIGMAs
  sigmas <- pars[par.type=="SIGMA" & !is.na(iblock),]
  sigmas.list <- split(sigmas,by="iblock")
  sigma.mat.list <- lapply(sigmas.list,NMdata::dt2mat)

  ## use simpar to sample params
  pars <- simpar::simpar(
              nsim = nsim,
              theta = pars[par.type=="THETA",value],
              covar = theta.covar,
              omega = omega.mat.list,
              odf = omega.sigma.dfs[par.type=="OMEGA",DF2],
              sigma = sigma.mat.list,
              sdf = omega.sigma.dfs[par.type=="SIGMA",DF2]
              ) %>% as.data.table()

  ## read in parameters simulated with simpar and return
  pars <- readParsWide(data=pars)
  return(pars)
}
```