Optimal Sampling Workflows with NMsim and NONMEM

Ahmed A. Abulfathi[1], Brian Reilly[2], Sergio Iadevaia[1], Stephen Duffull[1], Philip Delff[2]

[1] Certara USA, Inc; [2] Vertex Pharmaceuticals Incorporated

Introduction

Optimal sampling analyses are used to address important tasks such as designing optimal sampling schemes. The [NMsim] R package [1] offers automation of this workflow, providing a simple interface to optimal sampling analysis without need for editing the Nonmem model after estimation, all seamlessly in R.

Objectives

The objectives of this work were to:

- Leverage the NONMEM `$DESIGN` feature [2] using `NMsim` [1] to avoid the need for model reimplementation.
- Describe how a NONMEM model can be easily run with `$DESIGN` within `NMsim`, allowing for an automated workflow that can be executed with a single `NMsim` function call in R, only using the estimation control stream.

Methods

`NMsim` automates NONMEM simulation workflows into single R function calls. Provided with a path to a NONMEM control stream and a `data.frame` for design evaluation or optimization, `NMsim` will do the following:

- Save the evaluation/optimization input data in a csv file for NONMEM
- Create an evaluation/optimization input control stream based on the estimation control stream (including `$INPUT` and `$DATA` matching the saved evaluation/optimization data set; `$SIMULATE` instead of `$ESTIMATION` and `$COVARIANCE`)
- Update and fix initial values based on estimate (from `file.ext`)
- Further modify the control stream for `$DESIGN` using the `NMsim()` `modify` argument
- Run NONMEM on the generated evaluation/optimization control stream
- Collect output data tables, combine them, and merge with the evaluation/optimization input data
- Return the collected data in R

`NMsim` makes extensive use of functionality to handle NONMEM data and control streams provided by the R package [`NMdata`] [3].

Results

Design evaluation

The design evaluation dataset in this example consists of a single design with a bolus dose record and three observations (sample times). It's essential to note that the ID in this dataset represents a design, rather than an individual subject.

```
library(NMsim) ## Used version 0.2.5
# path to Estimated control stream
file.mod = "../models/warfarin01.mod"
dataset.1 <- NMsim::NMcreateDoses(TIME=0,AMT=70,CMT=NA) |>
    NMsim::NMaddSamples(TIME=c(1,4,8), DV=1) |>
  mutate(TIMEINIT = TIME)
kable(dataset.1, caption ="Dataset for design evaluation")
```

Table 1: Dataset for design evaluation

ID	TIME	EVID	AMT	MDV	DV	TIMEINIT
1	0	1	70	1	NA	0
1	1	0	NA	0	1	1
1	4	0	NA	0	1	4
1	8	0	NA	0	1	8

```
### Design evaluation
simres.eval <-
NMsim(
file.mod = file.mod, # path to NONMEM model
data = dataset.1, # dataset for design evaluation
name.sim = "eval-ex1", # output name suffix
# Replace $SIMULATION with $DESIGN
modify = list(simulation ="$DESIGN GROUPSIZE=32 FIMDIAG=1"),
table.vars = cc(TIMEINIT, TIME), # output table variables
reuse.results=reuse.results)
```

Notice that the *TIME* is not different from the *Initial TIME* in the input dataset. This is because we did not optimize the design.

Table 2: Design evaluation output

ID	Initial Time	TIME	EVID	AMT	DV	MDV
1	0	0	1	70	NA	1
1	1	1	0	NA	1	0
1	4	4	0	NA	1	0
1	8	8	0	NA	1	0

Design optimization

This section provides an example of design optimization with 5 samples. To perform design optimization, the dataset must include data items that describe the design space. In this case, we add variables such as `TSTRAT`, `TMIN`, and `TMAX` to the design. Our goal is to optimize the **TIME** variable.

```
samp.times <- c(1, 2, 4, 6, 8)
dataset.2b <-
NMsim::NMcreateDoses(TIME=data.frame(TIME=0,TSTRAT=0,TMIN=0,TMAX=200),
                     AMT=70,CMT=NA) |>
NMsim::NMaddSamples(TIME = data.frame(TIME=samp.times,
  TSTRAT=seq(1:length(samp.times)),TMIN=rep(0.01,length(samp.times)),
  TMAX=rep(200,length(samp.times))),DV=1) |> mutate(TIMEINIT = TIME)
kable(dataset.2b, caption="Dataset for design optimization")
```

Table 3: Dataset for design optimization

ID	TIME	EVID	AMT	MDV	TMAX	TMIN	TSTRAT	DV	TIMEINIT
1	0	1	70	1	200	0.00	0	NA	0
1	1	0	NA	0	200	0.01	1	1	1
1	2	0	NA	0	200	0.01	2	1	2
1	4	0	NA	0	200	0.01	3	1	4
1	6	0	NA	0	200	0.01	4	1	6
1	8	0	NA	0	200	0.01	5	1	8

To indicate that design optimization is being performed, we specify `MAXEVAL>0`.

```
### Design optimization
simres.opt.ex2.b <-
NMsim(file.mod = file.mod, # path to NONMEM model
data = dataset.2b, # dataset for design evaluation
name.sim = "opt-ex2b", # output name suffix
# adding NONMEM $DESIGN
modify = list(simulation="$DESIGN GROUPSIZE=32 FIMDIAG=1 MAXEVAL=9999
PRINT=20 DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN DESELMAX=TMAX"),
table.vars = cc(TIMEINIT, TIME, TSTRAT, TMIN, TMAX),
reuse.results=reuse.results)
```

Notice that the resulting `TIME` is now different from the `TIME` in the input dataset (i.e, Initial Time). This is because we optimized the design.

Table 4: Design optimization output

ID	Initial Time	TIME	EVID	AMT	DV	MDV	TMAX	TMIN	TSTRAT
1	0	0.0000000	1	70	NA	1	200	0.00	0
1	1	0.1316852	0	NA	1	0	200	0.01	1
1	4	7.0065718	0	NA	1	0	200	0.01	3
1	2	7.0463475	0	NA	1	0	200	0.01	2
1	6	7.1110970	0	NA	1	0	200	0.01	4
1	8	159.2489315	0	NA	1	0	200	0.01	5

Table 5 compares the -log(det(FIM)) and %RSE values between evaluated and optimized designs.

Table 5: Comparison between evaluated and optimized designs

Item	Evaluated (warfarin example 1, 3 sample times)	Optimized (warfarin example 2b, 5 sample times)
%RSE CL	36.90000	4.78000
%RSE V	4.95000	2.71000
%RSE KA	15.70000	13.90000
%RSE PPVCL	731.00000	26.00000
%RSE PPVV	42.40000	29.50000
%RSE PPVKA	27.00000	25.70000
%RSE propErr	28.10000	17.70000
-log(det(FIM))	-39.51821	-51.59772

Evaluate differences in sample times due to uncertainty in population parameters

We observed that some optimized times in the example above tend to cluster together. This clustering suggests that some of the repetition in the design can be leveraged to spread out points and make the design more robust against variations in PK parameters.

To further explore this idea, we can investigate how differences in sample times may arise due to uncertainty in population parameters. To do this, we will use prior information to randomly generate theta sets, and then obtain optimal time points for each of these theta sets using `$SIM TRUE = PRIOR` [2].

This approach enables us to perform a robust design using the `NMsim()`, specifically `method.sim=NMsim_NWPRI` [2].

The input dataset for the robust design includes the following sampling times 0.5, 7, 7, 7 and 72 hours.

```
### Robust Design
simres.robust <-NMsim(file.mod = file.mod.2, # NONMEM model
  data = dataset.priort, # dataset for robust design
  name.sim = "robust-ex3", # output name suffix
  method.sim = NMsim_NWPRI, # to use Normal Wishart Prior
  seed.nm = 4442223, #NONMEM seed number for reproducibility
  subproblems = 1000, # Number of subproblems
  onlysim = F, #to remove ONLYSIM
  # adding NONMEM $DESIGN
  modify = list(simulation = NMsim::add("$DESIGN NELDER FIMDIAG=1
  GROUPSIZE=32 OFVTYPE=0 DESEL=TIME DESELSTRAT=TSTRAT DESELMIN=TMIN
  DESELMAX=TMAX MAXEVAL=4000 SIGL=10 nohabort PRINT=100 NOPRIOR=1")),
  typical = FALSE,
  table.vars = cc(TIME, TIMEINIT, TSTRAT, TMIN, TMAX, IPRED),
  reuse.results=reuse.results)
```

Summary of the robust design is easily obtained as follows:

```
robust3.table <- simres.robust %>% filter(TSTRAT>0)
robust3.table$GROUP <- rep(1:5, length.out = nrow(robust3.table))
robust3.table %>% select(GROUP,NMREP,TIMEINIT,TIME,TSTRAT,TMIN,
                         TMAX)%>% group_by(GROUP) %>%
  summarize(N=n(),q2.5=quantile(TIME,probs=0.025),Mean=mean(TIME),
            q50=quantile(TIME,probs=0.5),
            q97.5=quantile(TIME,probs=0.975)) %>%
  kable(caption = "Robust design")
```

Table 6: Robust design

GROUP	N	q2.5	Mean	q50	q97.5
1	1000	0.0624486	0.1393475	0.1305439	0.2656241
2	1000	4.0600185	7.1184415	6.9567215	11.0854360
3	1000	4.0600185	7.1184415	6.9567215	11.0854360
4	1000	4.0600185	7.1184415	6.9567215	11.0854360
5	1000	81.8591034	154.8636497	156.4208187	199.9799767

Multiple design evaluation of varying sample sizes

In this example, we aim to show how easy it is to use `NMsim` to evaluate multiple designs at once saving the pharmacometrician valuable time.

Let's use the design in the first example but instead of using sample size of 32, we will vary it as follows: 5, 10, 20, 32, 60, 90, 120, 160, 200, 300, and 400. We will compare the designs using the %RSE and log(det(FIM)).

```
dt.sim <- dataset.1

### Create a vector with the number of subjects i.e.,
### GROUPSIZE for each design
ns <- c(5,10,20,32,60,90,120,160,200,300,400)

# Get the $DESIGN section for each of the 11 designs to be
# evaluated
strings.sim <- sprintf("$DESIGN GROUPSIZE=%d FIMDIAG=1",ns)
paths.eval.n <- lapply(1:length(strings.sim),function(nmod){
    NMsim(file.mod = file.mod, # path to NONMEM model
        data = transform(dt.sim,N=ns[nmod]),
        name.sim = paste0("eval-ex1-n_",ns[nmod]),
        modify = list(simulation = strings.sim[nmod]),
        table.vars = c("TIME", "TIMEINIT"),
        wait=FALSE,
        reuse.results=reuse.results)})
```

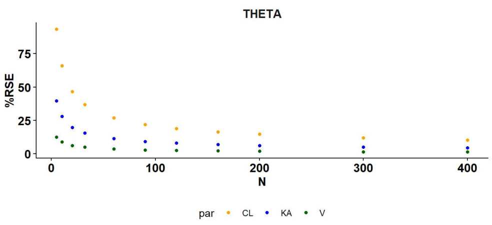

Figure 1: Multiple designs with varying number of subjects evaluated with a single NMsim() call.

Supplementary Information

Learn about how NMsim can help with:

- Optimal Sampling Workflows with NMsim and NONMEM
- Visual Predictive Checks
- Modifying NONMEM models on the fly
- Simulation of known subjects using EBEs and more…

References:

1. Delff P (2024). NMsim: Seamless 'Nonmem' Simulation Platform. https://philipdelff.github.io/NMsim/.
2. Bauer RJ, Hooker AC, Mentre F. Tutorial for $DESIGN in NONMEM: Clinical trial evaluation and optimization. CPT Pharmacometrics Syst Pharmacol. 2021 Dec;10(12):1452-1465. doi: 10.1002/psp4.12713.
3. Delff P. 2025. NMdata: Preparation, Checking and Post-Processing Data for PK/PD Modeling. https://nmautoverse.github.io/NMdata/.