QC Report: round 0

Git repository: gilead_ghqc_demo

jenna

September 27, 2024

Contents

Milestone Summary

Title	Description	Status	Issues
round 0	NA	open	model/nonmem/1009.mod‡§ scripts/validate_mrgsolve_model.qmd‡§ scripts/DA.R‡§

‡ open issue

§ issue with unchecked items

round 0

Issue Summary

File Path	Author	QC Type	QCer	Issue Closer	Close Date
model/nonmem/1009-.mod	jenna-a2ai <jenna@a2-ai.com>	NONMEM Model	jenna-a2ai	NA	NA
scripts/validate_-mrgsolve_model.qm-d	jenna-a2ai <jenna@a2-ai.com>	mrgsolve Model Validation	jenna-a2ai	NA	NA
scripts/DA.R	jenna-a2ai <jenna@a2-ai.com>	Data Assembly	jenna-a2ai	NA	NA

model/nonmem/1009.mod

QC Data

- **File Author:** jenna-a2ai jenna@a2-ai.com
- **QC initializer:** jenna-a2ai at 2024-09-27 17:23:18
- **Issue number:** 3
- **Milestone:** round 0

Assigned QCers

- jenna-a2ai

Issue Status

open

Issue Body

NONMEM Model

Note: Please modify the checklist items to insert relevant QC context.

NONMEM Model Control Stream

- ☐ :red_circle: **Correct data files are used in model.**
  - correct data, file name and hash: [INSERT]
- ☐ $INPUT matches Data columns.
  - If columns are ignored, explain which ones (i.e. character columns at the end of a dataset): [INSERT]
- ☐ IGNORE statements subset the data correctly.
  - What are the IGNORE statements that SHOULD be there and why: [INSERT]
- ☐ Correct TIME and DV variables.
  - DV or transformed DV used: [INSERT]
  - time column: [INSERT]
- ☐ Number of THETAs in the model match in $THETA.
- ☐ Covariates on appropriate THETA.
  - covariates and the parameters they SHOULD be on and functional form: [INSERT]
- ☐ Correct Scale Factor and conversions used.
  - what the equation should be and why (e.g. S2 = V2/1000 because dose in mg and conc in ng/mL): [INSERT]
- ☐ ETAs on correct parameters and match labels in $OMEGA block.
- ☐ Residual Error model coded correctly.
  - confirm error model (i.e., proportional error): [INSERT]
- ☐ Description of PK model matches ADVAN/$DES.
  - PK model description (i.e., 2-CMT model with first-order absorption and LAG time): [INSERT]
- ☐ If Simulation, SEED included in $SIM.
- ☐ $EST method consistent with expectation.
  - expectation (I.e., FOCE-I): [INSERT]
- ☐ $COV step included for model runs, include MATRIX.

NONMEM Model Output

- ☐ Check Run Number matches the control stream.

- ☐ Successful $EST step.
  - MINIMIZATION SUCCESSFUL.
  - Absence of warnings like MINIMIZATION TERMINATED or ROUNDING ERRORS.
  - Check for any warnings or errors, such as SIGDIGIT, NO. OF FUNCTION EVALUATIONS EXCEEDED, or any numerical issues that might affect the model.
- ☐ Successful $COV step.
  - Ensure the matrix is positive definite. A non-positive definite matrix may indicate issues with model estimation.
- ☐ Check parameters are not highly correlated.
  - In the CORRELATION MATRIX OF ESTIMATE section, correlation coefficients <0.95.
- ☐ Condition Number < 1000.
  - Found in the EIGENVALUES OF THE COVARIANCE MATRIX section, largest eigenvalue divided by smallest eigenvalue.
- ☐ List parameters with high shrinkage (>30%) to flag for analyst/medical writer.

Metadata

- author: jenna-a2ai jenna@a2-ai.com
- qc type: NONMEM Model
- script hash: 88909315f4b0b2265faa6afe11c2fdc5
- git sha: aced09b5e323d40eeab922726cccb1d06ed00cd2
- file history: https://github.com/A2-ai/gilead_ghqc_demo/commits/main/model/nonmem/1009.mod

Comments

Events

- milestone set to round 0 by jenna-a2ai at 2024-09-27 17:23:18
- assigned to jenna-a2ai by jenna-a2ai at 2024-09-27 17:23:18

Detailed Timeline

- milestone set to round 0 by jenna-a2ai at 2024-09-27 17:23:18
- assigned to jenna-a2ai by jenna-a2ai at 2024-09-27 17:23:18

scripts/validate_mrgsolve_model.qmd

QC Data

- **File Author:** jenna-a2ai jenna@a2-ai.com
- **QC initializer:** jenna-a2ai at 2024-09-27 17:23:17
- **Issue number:** 2
- **Milestone:** round 0

Assigned QCers

- jenna-a2ai

Issue Status

open

Issue Body

mrgsolve Model Validation

Note: Please modify the checklist items to insert relevant QC context.

1. Prepare needed R objects

- ☐ Build your mrgsolve model, in this case my_model. Copy parameter estimates from the NONMEM model, preferably through adding an $NMXML code block.
- ☐ Construct a validation dataset, in this case validation_data. This should contain the same dosing and observation records that went into the NONMEM model (i.e., exclude records that were flagged to be ignored by the NONMEM model), plus variables for ETA1, ETA2, etc.
- ☐ Read in the NONMEM output file with observation-level predictions. Some mild post-processing may be needed to ensure the data will merge properly - for example, adding USUBJID or renaming the time variable to TIME as required by mrgsolve. In this case, the post-processed data frame is called nm_output.

2. Generate exposure predictions from mrgsolve

- ☐ Generate exposure predictions from mrgsolve.

```
set.seed(20240531)
mrg_test_raw <- mrgsim(my_model,
                       data = validation_data,
                       etasrc = "data.all",
                       obsonly = TRUE)
```

The argument etasrc = "data.all" tells mrgsolve to look for all of the ETAs as columns in validation_data; it will throw an error if it does not find them. The obsonly = TRUE tells mrgsolve to exclude dosing records from the output.

3. Combine exposure predictions from mrgsolve and NONMEM

- ☐ Combine exposure predictions from mrgsolve and NONMEM.

```
# Post-process mrgsim output
mrg_test1 <- mrg_test_raw |>
  as.data.frame() |>
  pivot_longer(cols = c(IPRED, CL, VC, Q1, VP1, Q2, VP2)) |>
  mutate(SOURCE = "mrgsolve") |>
  select(USUBJID, TIME, SOURCE, PARAM = name, value)

# NONMEM predictions of the same values
nm_preds <- nm_output |>
  filter(EVID == 0) |>
  pivot_longer(cols = c(IPRED, CL, VC, Q1, VP1, Q2, VP2)) |>
  mutate(SOURCE = "NONMEM") |>
  select(USUBJID, TIME, SOURCE, PARAM = name, value)

# Merge mrgsolve and NONMEM predictions
mrg_test <- bind_rows(mrg_test1, nm_preds) |>
  pivot_wider(id_cols = c(USUBJID, TIME, PARAM), names_from = SOURCE)
```

In lines 4 and 11, substitute in whatever parameter values are relevant in this scenario. Make sure they are included in the $CAPTURE code block of your mrgsolve model specification! When this code is run, mrg_test will have columns USUBJID, TIME, PARAM, mrgsolve, and NONMEM. The column PARAM will have the value IPRED, CL, VC, Q1, VP1, Q2, or VP2, the column mrgsolve will have the value of that parameter as predicted by mrgsolve, and the column NONMEM will have the value that parameter as predicted by NONMEM.

:bell: "TIME" from NONMEM output may not precisely match the times given in the source dataset, since NONMEM output typically has less precision than we use while saving our derived datasets. If you are finding that mrg_test1 and nm_preds have rows that cannot be matched on USUBJID and TIME, you can try retrieving the original TIME values from the source data and adding them to nm_output, or perhaps using line number (LINE or NUM) as the ID variable for rows instead.

4. Graphically examine output

☐ Graphically examine output.

```
# Plot the comparison
ggplot(data = mrg_test, aes(x = NONMEM, y = mrgsolve)) +
  geom_abline(aes(color = "red", slope = 1, intercept = 0),
              key_glyph = "abline") +
  geom_point(size = 0.7) +
  facet_wrap(vars(PARAM), scales = "free") +
  theme(legend.position = "bottom",
        legend.direction = "horizontal") +
  scale_color_identity(guide = "legend", name = NULL,
                       breaks = c("red"),
                       labels = c("y = x reference line"))
```

:x: If any figures have any points visibly deviating from the line y = x, validation is not complete until these differences are eliminated (or verified to be harmless and documented in the validation file).

If all of the figures look wrong, you may not be matching the right subjects, or you might not be pulling model parameter estimates from the NONMEM output correctly.

If most of the PK parameters are falling on that line, but one or two of them are not (e.g., VC, Q1, VP1, Q2, and VP2 all look great, but CL and IPRED are all over the place - IPRED presumably being affected by CL), then your problem may be a covariate or ETA value that affect a PK parameter and are not constructed correctly in validation_data.

If all the PK parameters look good but IPRED is inconsistent, then the concentration predictions may not be scaled to the volume in the observation compartment (and/or the correct units). Unlike in NONMEM, there are no reserved terms for scale parameters that are automatically handled; if IPRED is observed in compartment 1, then you must manually specify that IPRED = CENT/S1 (or whatever the appropriate variable names would be).

5. Look at observations with the most extreme differences

☐ Look at observations with the most extreme differences.

```
mrg_test_ipred <- mrg_test |>
  filter(PARAM == "IPRED") |>
  mutate(ABSDIFF = abs(mrgsolve - NONMEM),
         PCTDIFF = 100*(ABSDIFF/NONMEM))

mrg_test_ipred |>
  arrange(desc(ABSDIFF)) |>
  head() |>
  flextable()

mrg_test_ipred |>
  ggplot(aes(y = ABSDIFF)) +
  geom_boxplot()


# Discard records that could have large relative differences but minuscule abs
    diffs
mrg_test_ipred_reldiffs <- mrg_test_ipred |>
  filter(mrgsolve > 1E-5 | NONMEM > 1E-5)

mrg_test_ipred_reldiffs |>
  arrange(desc(PCTDIFF)) |>
  head() |>
  flextable()

mrg_test_ipred_reldiffs |>
  ggplot(aes(y = PCTDIFF)) +
  geom_boxplot()
```

When looking at these values, it should be relatively obvious if the differences can be entirely explained by rounding errors or not. NONMEM output is usually converted to scientific notation before rounding, so it won't necessarily match how R rounds numbers or the model precision specified for mrgsolve.

:x: Relative differences should be $< 5\%$. If mrgsolve is automatically retrieving parameter estimates from NONMEM with high precision and the exact same dataset of doses/observations is used for validation, then relative differences should be $< 1\%$.

If the first few observations are dramatically wrong, but others are not, then there might be an issue with how variables are assigned initial values. There have been several cases where analysts programmed something along these lines:

```
X = exp(Y)
double Y = 0.5;
```

mrgsolve will allow you to do this without throwing any errors or warnings, but in the first record, X will be defined as though Y = 0 instead of the actual value of Y. If X is a function of Y, then X should be defined in the model specification file after Y.

6. Create arbitrary additional observations and plot them

☐ Create arbitrary additional observations and plot them.

:bell: mrgsolve can have strange and obscure bugs that are not apparent when only comparing predictions made at the actual PK observation times, but will mangle the exposure predictions you may need for forest plots or ER analysis.

Examples of problems that were not apparent until something like this was done: - Analyst misunderstood how to program multiple observation compartments (antibody and drug for ADC), and calculated the correct value in observation compartments only for records that had a CMT value corresponding to the NONMEM observation compartment (mrgsolve outputs all compartments at all observations, in wide format) - The wrong nocb value was applied, so that F1 for each dose was actually the F1 for the previous dose

Try something along these lines:

```
dose_data <- filter(validation_data, EVID == 1)
new_test <- mrgsim(my_model,
                   data = dose_data,
                   tgrid = tgrid(start = 0, end = 500, delta = 0.05),
                   etasrc = "data.all",
                   obsonly = TRUE)

new_test |>
  as.data.frame() |>
  filter(ID %in% c(1:5)) |>
  ggplot(aes(x = TIME, y = IPRED, color = as.factor(ID), group = as.factor(ID)
      )) +
  geom_line()
```

The output should appear smooth and continuous (except for sharp increases where IV boluses are administered). The relative height of the concentration peaks should make sense given the dose given before each peak.

7. (Optional) Check residual errors.

☐ (Optional) Check residual errors

```
nm_res <- nm_output |>
  select(IRES)
mrg_res <- mrg_test_raw |>
  mutate(IRES = log(DV) - log(IPRED))

ggplot(nm_res, aes(x = IRES)) +
  geom_histogram(aes(y = after_stat(density)), binwidth = 0.1, color = "white"
      ) +
  geom_density(data = mrg_res, color = "blue", linetype = "dashed")
```

Tweak code snippet as needed to define IRES correctly and give an appropriate bin width.

:x: This diagnostic should end up looking very similar to the pmtables::res_hist() family of plots. The histogram and the density curve should both be visible and have similar spread.

Metadata

- author: jenna-a2ai jenna@a2-ai.com
- qc type: mrgsolve Model Validation
- script hash: f2dc8ceb442bf77aee8824d91a11e0f3

- git sha: aced09b5e323d40eeab922726cccb1d06ed00cd2
- file history: https://github.com/A2-ai/gilead_ghqc_demo/commits/main/scripts/validate_mrgsolve_model.qmd

Comments

Events

- milestone set to round 0 by jenna-a2ai at 2024-09-27 17:23:17
- assigned to jenna-a2ai by jenna-a2ai at 2024-09-27 17:23:17

Detailed Timeline

- milestone set to round 0 by jenna-a2ai at 2024-09-27 17:23:17
- assigned to jenna-a2ai by jenna-a2ai at 2024-09-27 17:23:17

scripts/DA.R

QC Data

- **File Author:** jenna-a2ai jenna@a2-ai.com
- **QC initializer:** jenna-a2ai at 2024-09-27 17:23:16
- **Issue number:** 1
- **Milestone:** round 0

Assigned QCers

- jenna-a2ai

Issue Status

open

Issue Body

Data Assembly

Note: Please modify the checklist items to insert relevant QC context.

Reproducibility and Organization

- ☐ Script is a quarto document.
- ☐ Script renders free of error within the Rproject space.
- ☐ renv::status() present at beginning of script and passes.
- ☐ Relative paths are used.
- ☐ .Rprofile is present in same directory as final script.
- ☐ :red_circle: **Parquet files are used where available.**
  - files that should be parquet: [INSERT]
- ☐ :red_circle: **Data hash codes are printed and match source and output data.**
  - expected hash files: [INSERT]
- ☐ :red_circle: **Correct source data files are read in.**
  - correct source data: [INSERT]
- ☐ Final dataset outputs to expected location as a csv, converts all NA values to a period, and outputs a hash.
- ☐ All code chunks are labeled.
- ☐ session.info() included at the conclusion of script.

yspec

- ☐ All yspec files can be found in <project>/data/derived and are excluded from gitignore.
- ☐ All variables have descriptions. All descriptions are easy to read and understand. All variables have units where applicable.
- ☐ All categorical variables have all options outlined and a 1:1 numerical decoded value. All categorical variables match the decoding present in the yspec.
- ☐ :red_circle: **All continuous variables specify units and all units match those of the final dataset. If multiple analytes are included in a single column, units are clear for each analyte.**
- ☐ :red_circle: **The yspec file contains all necessary variables for analysis, including both project specific variables and NONMEM required variables.**

Dataset Assumptions - Automated

:red_circle: These can be done using pointblank and QCer can just confirm these pass!

- ☐ :red_circle: **LIST POINTBLANK ASSUMPTIONS.**
- ☐ The final dataset includes both a numeric (ID) and character (e.g. USUBJID) version of the unique subject identifier.
- ☐ The final dataset includes all columns listed in the corresponding yspec file.
- ☐ Any missing columns or observations are confirmed to be missing in source data and marked as -999.
- ☐ Missing data is documented appropriately in dataset.

Data Assumptions - Manual

:red_circle: These CAN'T be done using pointblank and QCer must look closely at specific sections of code.

- ☐ :red_circle: **All manual conversions are accurate and consistent.**
- ☐ :red_circle: **All manual imputations are well documented to outside sources or sufficiently justified.**
- ☐ :red_circle: **All manual imputations are accurate according to their specified source.**
- ☐ :red_circle: **All assumptions are well justified (e.g. by the protocol, project lead decision, client decision, etc.), documented, and reasonable.**
- ☐ :red_circle: **All records to be excluded (e.g. for all analyses, for sensitivity analysis only) are properly identified, justified, and documented (i.e., CFLAG or other flag column).**

Misc

- ☐ Code is easily readable and follows general coding guidelines (link to code hub).
- ☐ Helper functions are well documented, make sense, make reasonable assumptions, and follow good coding practices.
- ☐ :red_circle: **Helper function arguments match as stated in yspec (e.g. SEX variable in Cockcroft-Gault equation matches decoding in yspec).**

Metadata

- author: jenna-a2ai jenna@a2-ai.com
- qc type: Data Assembly
- script hash: 9de0992cc46272ab53f38ac5f79fe6f4
- git sha: aced09b5e323d40eeab922726cccb1d06ed00cd2
- file history: https://github.com/A2-ai/gilead_ghqc_demo/commits/main/scripts/DA.R

Comments

Events

- milestone set to round 0 by jenna-a2ai at 2024-09-27 17:23:16
- assigned to jenna-a2ai by jenna-a2ai at 2024-09-27 17:23:16

Detailed Timeline

- milestone set to round 0 by jenna-a2ai at 2024-09-27 17:23:16
- assigned to jenna-a2ai by jenna-a2ai at 2024-09-27 17:23:16