Supplementary Data 4

In Silico Evaluation of the Endogenous Concentration Correction Process and Modeling of the Error

```r
# Error_Simulation.R
# A script simulating the error added by the endogenous concentration correction process,
# for two different distribution of standards with homo and heteroscedastic data.
# By Brigitte Desharnais, 2018-09-19.

# Set the working directory.
setwd("~/R/Endogene_2018-09-19")

# Set the total desired number of simulations (e.g. for 10 simulations, input 3*10 (10 x 3 weights)).
Tot_It <- 3*1000

# Initialize the data frames which will receive the final results.
Results_GC_QC_60 <- data.frame(Iteration = integer(length=Tot_It), Weight = integer(length=Tot_It),
                               Precision = numeric(length=Tot_It), Real_XE = numeric(length=Tot_It), Calc_XE = numeric(length=Tot_It),
                               Un_Conc = numeric(length=Tot_It), Corr_Conc = numeric(length=Tot_It),
                               E_Un_Conc = numeric(length=Tot_It), E_Endog = numeric(length=Tot_It), E_Corr_Conc = numeric(length=Tot_It),
                               E_Un_Conc_Perc = numeric(length=Tot_It), E_Endog_Perc = numeric(length=Tot_It), E_Corr_Conc_Perc = numeric(length=Tot_It),
                               Endog_Bias_Perc = numeric(length=Tot_It))
Results_GC_QC_150 <- data.frame(Iteration = integer(length=Tot_It), Weight = integer(length=Tot_It),
                                Precision = numeric(length=Tot_It), Real_XE = numeric(length=Tot_It), Calc_XE = numeric(length=Tot_It),
                                Un_Conc = numeric(length=Tot_It), Corr_Conc = numeric(length=Tot_It),
                                E_Un_Conc = numeric(length=Tot_It), E_Endog = numeric(length=Tot_It), E_Corr_Conc = numeric(length=Tot_It),
                                E_Un_Conc_Perc = numeric(length=Tot_It), E_Endog_Perc = numeric(length=Tot_It), E_Corr_Conc_Perc = numeric(length=Tot_It),
                                Endog_Bias_Perc = numeric(length=Tot_It))
Results_GC_QC_375 <- data.frame(Iteration = integer(length=Tot_It), Weight = integer(length=Tot_It),
                                Precision = numeric(length=Tot_It), Real_XE = numeric(length=Tot_It), Calc_XE = numeric(length=Tot_It),
                                Un_Conc = numeric(length=Tot_It), Corr_Conc = numeric(length=Tot_It),
                                E_Un_Conc = numeric(length=Tot_It), E_Endog = numeric(length=Tot_It), E_Corr_Conc = numeric(length=Tot_It),
```

```r
                                    E_Un_Conc_Perc = numeric(length=Tot_It), E_Endog_Perc = numeric(length=Tot_It), E_Corr_Conc_Perc = numeric(length=Tot_It),
                                    Endog_Bias_Perc = numeric(length=Tot_It))

# Model: linear, homoscedastic data, GC-MS calibration curve.
for(a in 1:(Tot_It/3)){

  # Build the first three columns of the data frame to input in the error calculation script.
  # NB Concentrations used are those of the GC-MS method.
  Name <- c("STD0", "STD1", "STD2", "STD3", "STD4", "STD5", "STD6", "STD7",
            "QC1", "QC1", "QC2", "QC2", "QC3", "QC3")
  Type <- c("Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal",
            "QC", "QC", "QC", "QC", "QC", "QC")
  Spiked_Conc <- c(0, 10, 20, 50, 100, 200, 425, 500, 60, 60, 150, 150, 375, 375)

  # Set an endogenous concentration (uniform distribution between 0 and 12).
  # These limits were chosen based on experimentally observed BHB concentration in blood lots.
  XE_R <- runif(1, min=0, max=12)

  # Set B0 and B1 for the calibration model.
  # A normal distribution of set mean and standard deviation was chosen to model B1,
  # reflecting what was observed experimentally with GC-MS data.
  B1_R <- rnorm(1, mean=0.004720322, sd=0.0004142351)
  # B0 is back-calculated from the selected B1 and endogenous value.
  B0_R <- XE_R*B1_R

  # Predicted measurements under the model are calculated.
  Measure_R <- (B1_R*Spiked_Conc) + B0_R

  # Standard deviation is chosen from a uniform distribution between 5% and 20% of the first non-null
  # calibration level (based on SWGTOX recommendations).
  # Since this models homoscedastic data, the variance (and thus SD) will be the same at all calibration levels.
  SD <- runif(1, min=0.05*Measure_R[2], max=0.20*Measure_R[2])

  # Actual measurements are modeled with homoscedastic noise.
  Measure <- rnorm(14, mean=Measure_R, SD)

  # Final data frame is assembled to pass to the error estimation script.
  Data <- data.frame(Name, Type, Spiked_Conc, Measure)

  # Weighting model set for the error estimation script.
  WS <- "A"

  # Error estimation script is run.
  source('~/R/Endogene_2018-09-19/Error_Endogenous_Correction_Mod.R')

  # Data frames are filled with results from this iteration.
  Results_GC_QC_60$Iteration[a] <- a
  Results_GC_QC_60$Weight[a] <- 1
  Results_GC_QC_60$Precision[a] <- SD/Measure_R[2]
```

```
  Results_GC_QC_60$Real_XE[a] <- XE_R
  Results_GC_QC_60$Calc_XE[a] <- XE
  Results_GC_QC_60$Un_Conc[a] <- 60
  Results_GC_QC_60$Corr_Conc[a] <- 60 + XE
  Results_GC_QC_60$E_Un_Conc[a] <- Results$EM[1]
  Results_GC_QC_60$E_Endog[a] <- EE
  Results_GC_QC_60$E_Corr_Conc[a] <- Results$EC[1]
  Results_GC_QC_60$E_Un_Conc_Perc[a] <- (Results$EM[1]/60)*100
  Results_GC_QC_60$E_Endog_Perc[a] <- (EE/60)*100
  Results_GC_QC_60$E_Corr_Conc_Perc[a] <- (Results$EC[1]/Results$Corrected_Conc[1])*100
  Results_GC_QC_60$Endog_Bias_Perc[a] <- (XE/60)*100

  Results_GC_QC_150$Iteration[a] <- a
  Results_GC_QC_150$Weight[a] <- 1
  Results_GC_QC_150$Precision[a] <- SD/Measure_R[2]
  Results_GC_QC_150$Real_XE[a] <- XE_R
  Results_GC_QC_150$Calc_XE[a] <- XE
  Results_GC_QC_150$Un_Conc[a] <- 150
  Results_GC_QC_150$Corr_Conc[a] <- 150 + XE
  Results_GC_QC_150$E_Un_Conc[a] <- Results$EM[2]
  Results_GC_QC_150$E_Endog[a] <- EE
  Results_GC_QC_150$E_Corr_Conc[a] <- Results$EC[2]
  Results_GC_QC_150$E_Un_Conc_Perc[a] <- (Results$EM[2]/60)*100
  Results_GC_QC_150$E_Endog_Perc[a] <- (EE/60)*100
  Results_GC_QC_150$E_Corr_Conc_Perc[a] <- (Results$EC[2]/Results$Corrected_Conc[2])*100
  Results_GC_QC_150$Endog_Bias_Perc[a] <- (XE/60)*100

  Results_GC_QC_375$Iteration[a] <- a
  Results_GC_QC_375$Weight[a] <- 1
  Results_GC_QC_375$Precision[a] <- SD/Measure_R[2]
  Results_GC_QC_375$Real_XE[a] <- XE_R
  Results_GC_QC_375$Calc_XE[a] <- XE
  Results_GC_QC_375$Un_Conc[a] <- 375
  Results_GC_QC_375$Corr_Conc[a] <- 375 + XE
  Results_GC_QC_375$E_Un_Conc[a] <- Results$EM[3]
  Results_GC_QC_375$E_Endog[a] <- EE
  Results_GC_QC_375$E_Corr_Conc[a] <- Results$EC[3]
  Results_GC_QC_375$E_Un_Conc_Perc[a] <- (Results$EM[3]/60)*100
  Results_GC_QC_375$E_Endog_Perc[a] <- (EE/60)*100
  Results_GC_QC_375$E_Corr_Conc_Perc[a] <- (Results$EC[3]/Results$Corrected_Conc[3])*100
  Results_GC_QC_375$Endog_Bias_Perc[a] <- (XE/60)*100

}

# Model: linear, heteroscedastic 1/x data, GC-MS calibration curve.
for(a in ((Tot_It/3)+1):(Tot_It*2/3)){

  # Build the first three columns of the data frame to input in the error calculation script
     .
  # NB Concentrations used are those of the GC-MS method.
  Name <- c("STD0", "STD1", "STD2", "STD3", "STD4", "STD5", "STD6", "STD7",
            "QC1", "QC1", "QC2", "QC2", "QC3", "QC3")
  Type <- c("Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal",
            "QC", "QC", "QC", "QC", "QC", "QC")
```

```r
Spiked_Conc <- c(0, 10, 20, 50, 100, 200, 425, 500, 60, 60, 150, 150, 375, 375)

# Set an endogenous concentration (uniform distribution between 0 and 12).
# These limits were chosen based on experimentally observed BHB concentration in blood lots.
XE_R <- runif(1, min=0, max=12)

# Set B0 and B1 for the calibration model.
# A normal distribution of set mean and standard deviation was chosen to model B1,
# reflecting what was observed experimentally with GC-MS data.
B1_R <- rnorm(1, mean=0.004720322, sd=0.0004142351)
# B0 is back-calculated from the selected B1 and endogenous value.
B0_R <- XE_R*B1_R

# Predicted measurements under the model are calculated.
Measure_R <- (B1_R*Spiked_Conc) + B0_R

# Standard deviation for the first non-null calibration level is chosen from a uniform distribution
# between 5% and 20% (based on SWGTOX recommendations).
SD_L <- runif(1, min=0.05*Measure_R[2], max=0.20*Measure_R[2])

# For heteroscedastic, 1/x data, variance/x = constant.
CST <- (SD_L^2)/(Spiked_Conc[2]+XE_R)

# Creation of the standard deviation vector.
SD <- sqrt(CST*(Spiked_Conc+XE_R))

# Actual measurements are modeled with heteroscedastic noise.
Measure <- rnorm(14, mean=Measure_R, sd=SD)

# Final data frame is assembled to pass to the error estimation script.
Data <- data.frame(Name, Type, Spiked_Conc, Measure)

# Weighting model set for the error estimation script.
WS <- "B"

# Error estimation script is run.
source('~/R/Endogene_2018-09-19/Error_Endogenous_Correction_Mod.R')

# Data frames are filled with results from this iteration.
Results_GC_QC_60$Iteration[a] <- a
Results_GC_QC_60$Weight[a] <- 2
Results_GC_QC_60$Precision[a] <- SD/Measure_R[2]
Results_GC_QC_60$Real_XE[a] <- XE_R
Results_GC_QC_60$Calc_XE[a] <- XE
Results_GC_QC_60$Un_Conc[a] <- 60
Results_GC_QC_60$Corr_Conc[a] <- 60 + XE
Results_GC_QC_60$E_Un_Conc[a] <- Results$EM[1]
Results_GC_QC_60$E_Endog[a] <- EE
Results_GC_QC_60$E_Corr_Conc[a] <- Results$EC[1]
Results_GC_QC_60$E_Un_Conc_Perc[a] <- (Results$EM[1]/60)*100
Results_GC_QC_60$E_Endog_Perc[a] <- (EE/60)*100
Results_GC_QC_60$E_Corr_Conc_Perc[a] <- (Results$EC[1]/Results$Corrected_Conc[1])*100
```

```
  Results_GC_QC_60$Endog_Bias_Perc[a] <- (XE/60)*100

  Results_GC_QC_150$Iteration[a] <- a
  Results_GC_QC_150$Weight[a] <- 2
  Results_GC_QC_150$Precision[a] <- SD/Measure_R[2]
  Results_GC_QC_150$Real_XE[a] <- XE_R
  Results_GC_QC_150$Calc_XE[a] <- XE
  Results_GC_QC_150$Un_Conc[a] <- 150
  Results_GC_QC_150$Corr_Conc[a] <- 150 + XE
  Results_GC_QC_150$E_Un_Conc[a] <- Results$EM[2]
  Results_GC_QC_150$E_Endog[a] <- EE
  Results_GC_QC_150$E_Corr_Conc[a] <- Results$EC[2]
  Results_GC_QC_150$E_Un_Conc_Perc[a] <- (Results$EM[2]/60)*100
  Results_GC_QC_150$E_Endog_Perc[a] <- (EE/60)*100
  Results_GC_QC_150$E_Corr_Conc_Perc[a] <- (Results$EC[2]/Results$Corrected_Conc[2])*100
  Results_GC_QC_150$Endog_Bias_Perc[a] <- (XE/60)*100

  Results_GC_QC_375$Iteration[a] <- a
  Results_GC_QC_375$Weight[a] <- 2
  Results_GC_QC_375$Precision[a] <- SD/Measure_R[2]
  Results_GC_QC_375$Real_XE[a] <- XE_R
  Results_GC_QC_375$Calc_XE[a] <- XE
  Results_GC_QC_375$Un_Conc[a] <- 375
  Results_GC_QC_375$Corr_Conc[a] <- 375 + XE
  Results_GC_QC_375$E_Un_Conc[a] <- Results$EM[3]
  Results_GC_QC_375$E_Endog[a] <- EE
  Results_GC_QC_375$E_Corr_Conc[a] <- Results$EC[3]
  Results_GC_QC_375$E_Un_Conc_Perc[a] <- (Results$EM[3]/60)*100
  Results_GC_QC_375$E_Endog_Perc[a] <- (EE/60)*100
  Results_GC_QC_375$E_Corr_Conc_Perc[a] <- (Results$EC[3]/Results$Corrected_Conc[3])*100
  Results_GC_QC_375$Endog_Bias_Perc[a] <- (XE/60)*100

}

# Model: linear, heteroscedastic 1/x^2 data, GC-MS calibration curve.
for(a in ((Tot_It*2/3)+1):Tot_It){

  # Build the first three columns of the data frame to input in the error calculation script.
  # NB Concentrations used are those of the GC-MS method.
  Name <- c("STD0", "STD1", "STD2", "STD3", "STD4", "STD5", "STD6", "STD7",
            "QC1", "QC1", "QC2", "QC2", "QC3", "QC3")
  Type <- c("Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal",
            "QC", "QC", "QC", "QC", "QC", "QC")
  Spiked_Conc <- c(0, 10, 20, 50, 100, 200, 425, 500, 60, 60, 150, 150, 375, 375)

  # Set an endogenous concentration (uniform distribution between 0 and 12).
  # These limits were chosen based on experimentally observed BHB concentration in blood lots.
  XE_R <- runif(1, min=0, max=12)

  # Set B0 and B1 for the calibration model.
  # A normal distribution of set mean and standard deviation was chosen to model B1,
  # reflecting what was observed experimentally with GC-MS data.
```

```r
B1_R <- rnorm(1, mean=0.004720322, sd=0.0004142351)
# B0 is back-calculated from the selected B1 and endogenous value.
B0_R <- XE_R*B1_R

# Predicted measurements under the model are calculated.
Measure_R <- (B1_R*Spiked_Conc) + B0_R

# Standard deviation for the first non-null calibration level is chosen from a uniform distribution
# between 5% and 20% (based on SWGTOX recommendations).
SD_L <- runif(1, min=0.05*Measure_R[2], max=0.20*Measure_R[2])

# For heteroscedastic, 1/x data, variance/x = constant.
CST <- (SD_L^2)/((Spiked_Conc[2]+XE_R)^2)

# Creation of the standard deviation vector.
SD <- sqrt(CST*((Spiked_Conc+XE_R)^2))

# Actual measurements are modeled with heteroscedastic noise.
Measure <- rnorm(14, mean=Measure_R, sd=SD)

# Final data frame is assembled to pass to the error estimation script.
Data <- data.frame(Name, Type, Spiked_Conc, Measure)

# Weighting model set for the error estimation script.
WS <- "C"

# Error estimation script is run.
source('~/R/Endogene_2018-09-19/Error_Endogenous_Correction_Mod.R')

# Data frames are filled with results from this iteration.
Results_GC_QC_60$Iteration[a] <- a
Results_GC_QC_60$Weight[a] <- 3
Results_GC_QC_60$Precision[a] <- SD/Measure_R[2]
Results_GC_QC_60$Real_XE[a] <- XE_R
Results_GC_QC_60$Calc_XE[a] <- XE
Results_GC_QC_60$Un_Conc[a] <- 60
Results_GC_QC_60$Corr_Conc[a] <- 60 + XE
Results_GC_QC_60$E_Un_Conc[a] <- Results$EM[1]
Results_GC_QC_60$E_Endog[a] <- EE
Results_GC_QC_60$E_Corr_Conc[a] <- Results$EC[1]
Results_GC_QC_60$E_Un_Conc_Perc[a] <- (Results$EM[1]/60)*100
Results_GC_QC_60$E_Endog_Perc[a] <- (EE/60)*100
Results_GC_QC_60$E_Corr_Conc_Perc[a] <- (Results$EC[1]/Results$Corrected_Conc[1])*100
Results_GC_QC_60$Endog_Bias_Perc[a] <- (XE/60)*100

Results_GC_QC_150$Iteration[a] <- a
Results_GC_QC_150$Weight[a] <- 3
Results_GC_QC_150$Precision[a] <- SD/Measure_R[2]
Results_GC_QC_150$Real_XE[a] <- XE_R
Results_GC_QC_150$Calc_XE[a] <- XE
Results_GC_QC_150$Un_Conc[a] <- 150
Results_GC_QC_150$Corr_Conc[a] <- 150 + XE
Results_GC_QC_150$E_Un_Conc[a] <- Results$EM[2]
```

```
  Results_GC_QC_150$E_Endog[a] <- EE
  Results_GC_QC_150$E_Corr_Conc[a] <- Results$EC[2]
  Results_GC_QC_150$E_Un_Conc_Perc[a] <- (Results$EM[2]/60)*100
  Results_GC_QC_150$E_Endog_Perc[a] <- (EE/60)*100
  Results_GC_QC_150$E_Corr_Conc_Perc[a] <- (Results$EC[2]/Results$Corrected_Conc[2])*100
  Results_GC_QC_150$Endog_Bias_Perc[a] <- (XE/60)*100

  Results_GC_QC_375$Iteration[a] <- a
  Results_GC_QC_375$Weight[a] <- 3
  Results_GC_QC_375$Precision[a] <- SD/Measure_R[2]
  Results_GC_QC_375$Real_XE[a] <- XE_R
  Results_GC_QC_375$Calc_XE[a] <- XE
  Results_GC_QC_375$Un_Conc[a] <- 375
  Results_GC_QC_375$Corr_Conc[a] <- 375 + XE
  Results_GC_QC_375$E_Un_Conc[a] <- Results$EM[3]
  Results_GC_QC_375$E_Endog[a] <- EE
  Results_GC_QC_375$E_Corr_Conc[a] <- Results$EC[3]
  Results_GC_QC_375$E_Un_Conc_Perc[a] <- (Results$EM[3]/60)*100
  Results_GC_QC_375$E_Endog_Perc[a] <- (EE/60)*100
  Results_GC_QC_375$E_Corr_Conc_Perc[a] <- (Results$EC[3]/Results$Corrected_Conc[3])*100
  Results_GC_QC_375$Endog_Bias_Perc[a] <- (XE/60)*100

}
```