

Supplementary Data 4

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

```
1 # Error_Simulation.R
2 # A script simulating the error added by the endogenous concentration correction process,
3 # for two different distribution of standards with homo and heteroscedastic data.
4 # By Brigitte Desharnais, 2018-09-19.
5
6 # Set the working directory.
7 setwd("~/R/Endogene_2018-09-19")
8
9 # Set the total desired number of simulations (e.g. for 10 simulations, input 3*10 (10 x 3
10 # weights)).
11 Tot_It <- 3*1000
12
13 # Initialize the data frames which will receive the final results.
14 Results_GC_QC_60 <- data.frame(Iteration = integer(length=Tot_It), Weight = integer(length=
15 Tot_It),
16 Precision = numeric(length=Tot_It), Real_XE = numeric(length=
17 Tot_It), Calc_XE = numeric(length=Tot_It),
18 Un_Conc = numeric(length=Tot_It), Corr_Conc = numeric(length=
19 Tot_It),
20 E_Un_Conc = numeric(length=Tot_It), E_Endog = numeric(length=
21 Tot_It), E_Corr_Conc = numeric(length=Tot_It),
22 E_Un_Conc_Perc = numeric(length=Tot_It), E_Endog_Perc =
23 numeric(length=Tot_It), E_Corr_Conc_Perc = numeric(length=Tot_It),
24 Endog_Bias_Perc = numeric(length=Tot_It))
25 Results_GC_QC_150 <- data.frame(Iteration = integer(length=Tot_It), Weight = integer(length=
26 Tot_It),
27 Precision = numeric(length=Tot_It), Real_XE = numeric(length
28 =Tot_It), Calc_XE = numeric(length=Tot_It),
29 Un_Conc = numeric(length=Tot_It), Corr_Conc = numeric(length
30 =Tot_It),
31 E_Un_Conc = numeric(length=Tot_It), E_Endog = numeric(length
32 =Tot_It), E_Corr_Conc = numeric(length=Tot_It),
33 E_Un_Conc_Perc = numeric(length=Tot_It), E_Endog_Perc =
34 numeric(length=Tot_It), E_Corr_Conc_Perc = numeric(length=Tot_It),
35 Endog_Bias_Perc = numeric(length=Tot_It))
36 Results_GC_QC_375 <- data.frame(Iteration = integer(length=Tot_It), Weight = integer(length=
37 Tot_It),
38 Precision = numeric(length=Tot_It), Real_XE = numeric(length
39 =Tot_It), Calc_XE = numeric(length=Tot_It),
40 Un_Conc = numeric(length=Tot_It), Corr_Conc = numeric(length
41 =Tot_It),
42 E_Un_Conc = numeric(length=Tot_It), E_Endog = numeric(length
43 =Tot_It), E_Corr_Conc = numeric(length=Tot_It),
```

```

29         E_Un_Conc_Perc = numeric(length=Tot_It), E_Endog_Perc =
        numeric(length=Tot_It), E_Corr_Conc_Perc = numeric(length=Tot_It),
30         Endog_Bias_Perc = numeric(length=Tot_It))
31
32 # Model: linear, homoscedastic data, GC-MS calibration curve.
33 for(a in 1:(Tot_It/3)){
34
35     # Build the first three columns of the data frame to input in the error calculation script
        .
36     # NB Concentrations used are those of the GC-MS method.
37     Name <- c("STD0", "STD1", "STD2", "STD3", "STD4", "STD5", "STD6", "STD7",
38             "QC1", "QC1", "QC2", "QC2", "QC3", "QC3")
39     Type <- c("Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal",
40             "QC", "QC", "QC", "QC", "QC", "QC")
41     Spiked_Conc <- c(0, 10, 20, 50, 100, 200, 425, 500, 60, 60, 150, 150, 375, 375)
42
43     # Set an endogenous concentration (uniform distribution between 0 and 12).
44     # These limits were chosen based on experimentally observed BHB concentration in blood
        lots.
45     XE_R <- runif(1, min=0, max=12)
46
47     # Set B0 and B1 for the calibration model.
48     # A normal distribution of set mean and standard deviation was chosen to model B1,
49     # reflecting what was observed experimentally with GC-MS data.
50     B1_R <- rnorm(1, mean=0.004720322, sd=0.0004142351)
51     # B0 is back-calculated from the selected B1 and endogenous value.
52     B0_R <- XE_R*B1_R
53
54     # Predicted measurements under the model are calculated.
55     Measure_R <- (B1_R*Spiked_Conc) + B0_R
56
57     # Standard deviation is chosen from a uniform distribution between 5% and 20% of the first
        non-null
58     # calibration level (based on SWGTOX recommendations).
59     # Since this models homoscedastic data, the variance (and thus SD) will be the same at all
        calibration levels.
60     SD <- runif(1, min=0.05*Measure_R[2], max=0.20*Measure_R[2])
61
62     # Actual measurements are modeled with homoscedastic noise.
63     Measure <- rnorm(14, mean=Measure_R, SD)
64
65     # Final data frame is assembled to pass to the error estimation script.
66     Data <- data.frame(Name, Type, Spiked_Conc, Measure)
67
68     # Weighting model set for the error estimation script.
69     WS <- "A"
70
71     # Error estimation script is run.
72     source('~/R/Endogene_2018-09-19/Error_Endogenous_Correction_Mod.R')
73
74     # Data frames are filled with results from this iteration.
75     Results_GC_QC_60$Iteration[a] <- a
76     Results_GC_QC_60$Weight[a] <- 1
77     Results_GC_QC_60$Precision[a] <- SD/Measure_R[2]

```

```

78 Results_GC_QC_60$Real_XE[a] <- XE_R
79 Results_GC_QC_60$Calc_XE[a] <- XE
80 Results_GC_QC_60$Un_Conc[a] <- 60
81 Results_GC_QC_60$Corr_Conc[a] <- 60 + XE
82 Results_GC_QC_60$E_Un_Conc[a] <- Results$EM[1]
83 Results_GC_QC_60$E_Endog[a] <- EE
84 Results_GC_QC_60$E_Corr_Conc[a] <- Results$EC[1]
85 Results_GC_QC_60$E_Un_Conc_Perc[a] <- (Results$EM[1]/60)*100
86 Results_GC_QC_60$E_Endog_Perc[a] <- (EE/60)*100
87 Results_GC_QC_60$E_Corr_Conc_Perc[a] <- (Results$EC[1]/Results$Corrected_Conc[1])*100
88 Results_GC_QC_60$Endog_Bias_Perc[a] <- (XE/60)*100
89
90 Results_GC_QC_150$Iteration[a] <- a
91 Results_GC_QC_150$Weight[a] <- 1
92 Results_GC_QC_150$Precision[a] <- SD/Measure_R[2]
93 Results_GC_QC_150$Real_XE[a] <- XE_R
94 Results_GC_QC_150$Calc_XE[a] <- XE
95 Results_GC_QC_150$Un_Conc[a] <- 150
96 Results_GC_QC_150$Corr_Conc[a] <- 150 + XE
97 Results_GC_QC_150$E_Un_Conc[a] <- Results$EM[2]
98 Results_GC_QC_150$E_Endog[a] <- EE
99 Results_GC_QC_150$E_Corr_Conc[a] <- Results$EC[2]
100 Results_GC_QC_150$E_Un_Conc_Perc[a] <- (Results$EM[2]/60)*100
101 Results_GC_QC_150$E_Endog_Perc[a] <- (EE/60)*100
102 Results_GC_QC_150$E_Corr_Conc_Perc[a] <- (Results$EC[2]/Results$Corrected_Conc[2])*100
103 Results_GC_QC_150$Endog_Bias_Perc[a] <- (XE/60)*100
104
105 Results_GC_QC_375$Iteration[a] <- a
106 Results_GC_QC_375$Weight[a] <- 1
107 Results_GC_QC_375$Precision[a] <- SD/Measure_R[2]
108 Results_GC_QC_375$Real_XE[a] <- XE_R
109 Results_GC_QC_375$Calc_XE[a] <- XE
110 Results_GC_QC_375$Un_Conc[a] <- 375
111 Results_GC_QC_375$Corr_Conc[a] <- 375 + XE
112 Results_GC_QC_375$E_Un_Conc[a] <- Results$EM[3]
113 Results_GC_QC_375$E_Endog[a] <- EE
114 Results_GC_QC_375$E_Corr_Conc[a] <- Results$EC[3]
115 Results_GC_QC_375$E_Un_Conc_Perc[a] <- (Results$EM[3]/60)*100
116 Results_GC_QC_375$E_Endog_Perc[a] <- (EE/60)*100
117 Results_GC_QC_375$E_Corr_Conc_Perc[a] <- (Results$EC[3]/Results$Corrected_Conc[3])*100
118 Results_GC_QC_375$Endog_Bias_Perc[a] <- (XE/60)*100
119
120 }
121
122 # Model: linear , heteroscedastic 1/x data, GC-MS calibration curve.
123 for(a in ((Tot_It/3)+1):(Tot_It*2/3)){
124
125 # Build the first three columns of the data frame to input in the error calculation script
126
127 # NB Concentrations used are those of the GC-MS method.
128 Name <- c("STD0", "STD1", "STD2", "STD3", "STD4", "STD5", "STD6", "STD7",
129           "QC1", "QC1", "QC2", "QC2", "QC3", "QC3")
130 Type <- c("Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal",
131           "QC", "QC", "QC", "QC", "QC", "QC")

```

```

131 Spiked_Conc <- c(0, 10, 20, 50, 100, 200, 425, 500, 60, 60, 150, 150, 375, 375)
132
133 # Set an endogenous concentration (uniform distribution between 0 and 12).
134 # These limits were chosen based on experimentally observed BHB concentration in blood
135 # lots.
136 XE_R <- runif(1, min=0, max=12)
137
138 # Set B0 and B1 for the calibration model.
139 # A normal distribution of set mean and standard deviation was chosen to model B1,
140 # reflecting what was observed experimentally with GC-MS data.
141 B1_R <- rnorm(1, mean=0.004720322, sd=0.0004142351)
142 # B0 is back-calculated from the selected B1 and endogenous value.
143 B0_R <- XE_R*B1_R
144
145 # Predicted measurements under the model are calculated.
146 Measure_R <- (B1_R*Spiked_Conc) + B0_R
147
148 # Standard deviation for the first non-null calibration level is chosen from a uniform
149 # distribution
150 # between 5% and 20% (based on SWGTOX recommendations).
151 SD_L <- runif(1, min=0.05*Measure_R[2], max=0.20*Measure_R[2])
152
153 # For heteroscedastic, 1/x data, variance/x = constant.
154 CST <- (SD_L^2)/(Spiked_Conc[2]+XE_R)
155
156 # Creation of the standard deviation vector.
157 SD <- sqrt(CST*(Spiked_Conc+XE_R))
158
159 # Actual measurements are modeled with heteroscedastic noise.
160 Measure <- rnorm(14, mean=Measure_R, sd=SD)
161
162 # Final data frame is assembled to pass to the error estimation script.
163 Data <- data.frame(Name, Type, Spiked_Conc, Measure)
164
165 # Weighting model set for the error estimation script.
166 WS <- "B"
167
168 # Error estimation script is run.
169 source('~/R/Endogene_2018-09-19/Error_Endogenous_Correction_Mod.R')
170
171 # Data frames are filled with results from this iteration.
172 Results_GC_QC_60$Iteration[a] <- a
173 Results_GC_QC_60$Weight[a] <- 2
174 Results_GC_QC_60$Precision[a] <- SD/Measure_R[2]
175 Results_GC_QC_60$Real_XE[a] <- XE_R
176 Results_GC_QC_60$Calc_XE[a] <- XE
177 Results_GC_QC_60$Un_Conc[a] <- 60
178 Results_GC_QC_60$Corr_Conc[a] <- 60 + XE
179 Results_GC_QC_60$E_Un_Conc[a] <- Results$EM[1]
180 Results_GC_QC_60$E_Endog[a] <- EE
181 Results_GC_QC_60$E_Corr_Conc[a] <- Results$EC[1]
182 Results_GC_QC_60$E_Un_Conc_Perc[a] <- (Results$EM[1]/60)*100
183 Results_GC_QC_60$E_Endog_Perc[a] <- (EE/60)*100
184 Results_GC_QC_60$E_Corr_Conc_Perc[a] <- (Results$EC[1]/Results$Corrected_Conc[1])*100

```

```

183 Results_GC-QC_60$Endog_Bias_Perc[a] <- (XE/60)*100
184
185 Results_GC-QC_150$Iteration[a] <- a
186 Results_GC-QC_150$Weight[a] <- 2
187 Results_GC-QC_150$Precision[a] <- SD/Measure_R[2]
188 Results_GC-QC_150$Real_XE[a] <- XE_R
189 Results_GC-QC_150$Calc_XE[a] <- XE
190 Results_GC-QC_150$Un_Conc[a] <- 150
191 Results_GC-QC_150$Corr_Conc[a] <- 150 + XE
192 Results_GC-QC_150$E_Un_Conc[a] <- Results$EM[2]
193 Results_GC-QC_150$E_Endog[a] <- EE
194 Results_GC-QC_150$E_Corr_Conc[a] <- Results$EC[2]
195 Results_GC-QC_150$E_Un_Conc_Perc[a] <- (Results$EM[2]/60)*100
196 Results_GC-QC_150$E_Endog_Perc[a] <- (EE/60)*100
197 Results_GC-QC_150$E_Corr_Conc_Perc[a] <- (Results$EC[2]/Results$Corrected_Conc[2])*100
198 Results_GC-QC_150$Endog_Bias_Perc[a] <- (XE/60)*100
199
200 Results_GC-QC_375$Iteration[a] <- a
201 Results_GC-QC_375$Weight[a] <- 2
202 Results_GC-QC_375$Precision[a] <- SD/Measure_R[2]
203 Results_GC-QC_375$Real_XE[a] <- XE_R
204 Results_GC-QC_375$Calc_XE[a] <- XE
205 Results_GC-QC_375$Un_Conc[a] <- 375
206 Results_GC-QC_375$Corr_Conc[a] <- 375 + XE
207 Results_GC-QC_375$E_Un_Conc[a] <- Results$EM[3]
208 Results_GC-QC_375$E_Endog[a] <- EE
209 Results_GC-QC_375$E_Corr_Conc[a] <- Results$EC[3]
210 Results_GC-QC_375$E_Un_Conc_Perc[a] <- (Results$EM[3]/60)*100
211 Results_GC-QC_375$E_Endog_Perc[a] <- (EE/60)*100
212 Results_GC-QC_375$E_Corr_Conc_Perc[a] <- (Results$EC[3]/Results$Corrected_Conc[3])*100
213 Results_GC-QC_375$Endog_Bias_Perc[a] <- (XE/60)*100
214
215 }
216
217 # Model: linear , heteroscedastic 1/x^2 data, GC-MS calibration curve.
218 for(a in ((Tot_It*2/3)+1):Tot_It){
219
220 # Build the first three columns of the data frame to input in the error calculation script
221 .
222 # NB Concentrations used are those of the GC-MS method.
223 Name <- c("STD0", "STD1", "STD2", "STD3", "STD4", "STD5", "STD6", "STD7",
224           "QC1", "QC1", "QC2", "QC2", "QC3", "QC3")
225 Type <- c("Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal", "Cal",
226           "QC", "QC", "QC", "QC", "QC", "QC")
227 Spiked_Conc <- c(0, 10, 20, 50, 100, 200, 425, 500, 60, 60, 150, 150, 375, 375)
228
229 # Set an endogenous concentration (uniform distribution between 0 and 12).
230 # These limits were chosen based on experimentally observed BHB concentration in blood
231 # lots.
232 XE_R <- runif(1, min=0, max=12)
233
234 # Set B0 and B1 for the calibration model.
235 # A normal distribution of set mean and standard deviation was chosen to model B1,
236 # reflecting what was observed experimentally with GC-MS data.

```

```

235 B1_R <- rnorm(1, mean=0.004720322, sd=0.0004142351)
236 # B0 is back-calculated from the selected B1 and endogenous value.
237 B0_R <- XE_R*B1_R
238
239 # Predicted measurements under the model are calculated.
240 Measure_R <- (B1_R*Spiked_Conc) + B0_R
241
242 # Standard deviation for the first non-null calibration level is chosen from a uniform
    distribution
243 # between 5% and 20% (based on SWGTOX recommendations).
244 SD_L <- runif(1, min=0.05*Measure_R[2], max=0.20*Measure_R[2])
245
246 # For heteroscedastic, 1/x data, variance/x = constant.
247 CST <- (SD_L^2)/((Spiked_Conc[2]+XE_R)^2)
248
249 # Creation of the standard deviation vector.
250 SD <- sqrt(CST*((Spiked_Conc+XE_R)^2))
251
252 # Actual measurements are modeled with heteroscedastic noise.
253 Measure <- rnorm(14, mean=Measure_R, sd=SD)
254
255 # Final data frame is assembled to pass to the error estimation script.
256 Data <- data.frame(Name, Type, Spiked_Conc, Measure)
257
258 # Weighting model set for the error estimation script.
259 WS <- "C"
260
261 # Error estimation script is run.
262 source("~/R/Endogene_2018-09-19/Error_Endogenous_Correction_Mod.R")
263
264 # Data frames are filled with results from this iteration.
265 Results_GC_QC_60$Iteration[a] <- a
266 Results_GC_QC_60$Weight[a] <- 3
267 Results_GC_QC_60$Precision[a] <- SD/Measure_R[2]
268 Results_GC_QC_60$Real_XE[a] <- XE_R
269 Results_GC_QC_60$Calc_XE[a] <- XE
270 Results_GC_QC_60$Un_Conc[a] <- 60
271 Results_GC_QC_60$Corr_Conc[a] <- 60 + XE
272 Results_GC_QC_60$E_Un_Conc[a] <- Results$EM[1]
273 Results_GC_QC_60$E_Endog[a] <- EE
274 Results_GC_QC_60$E_Corr_Conc[a] <- Results$EC[1]
275 Results_GC_QC_60$E_Un_Conc_Perc[a] <- (Results$EM[1]/60)*100
276 Results_GC_QC_60$E_Endog_Perc[a] <- (EE/60)*100
277 Results_GC_QC_60$E_Corr_Conc_Perc[a] <- (Results$EC[1]/Results$Corrected_Conc[1])*100
278 Results_GC_QC_60$Endog_Bias_Perc[a] <- (XE/60)*100
279
280 Results_GC_QC_150$Iteration[a] <- a
281 Results_GC_QC_150$Weight[a] <- 3
282 Results_GC_QC_150$Precision[a] <- SD/Measure_R[2]
283 Results_GC_QC_150$Real_XE[a] <- XE_R
284 Results_GC_QC_150$Calc_XE[a] <- XE
285 Results_GC_QC_150$Un_Conc[a] <- 150
286 Results_GC_QC_150$Corr_Conc[a] <- 150 + XE
287 Results_GC_QC_150$E_Un_Conc[a] <- Results$EM[2]

```

```

288 Results_GC-QC_150$E_Endog[a] <- EE
289 Results_GC-QC_150$E_Corr_Conc[a] <- Results$EC[2]
290 Results_GC-QC_150$E_Un_Conc_Perc[a] <- (Results$EM[2]/60)*100
291 Results_GC-QC_150$E_Endog_Perc[a] <- (EE/60)*100
292 Results_GC-QC_150$E_Corr_Conc_Perc[a] <- (Results$EC[2]/Results$Corrected_Conc[2])*100
293 Results_GC-QC_150$Endog_Bias_Perc[a] <- (XE/60)*100
294
295 Results_GC-QC_375$Iteration[a] <- a
296 Results_GC-QC_375$Weight[a] <- 3
297 Results_GC-QC_375$Precision[a] <- SD/Measure_R[2]
298 Results_GC-QC_375$Real_XE[a] <- XE_R
299 Results_GC-QC_375$Calc_XE[a] <- XE
300 Results_GC-QC_375$Un_Conc[a] <- 375
301 Results_GC-QC_375$Corr_Conc[a] <- 375 + XE
302 Results_GC-QC_375$E_Un_Conc[a] <- Results$EM[3]
303 Results_GC-QC_375$E_Endog[a] <- EE
304 Results_GC-QC_375$E_Corr_Conc[a] <- Results$EC[3]
305 Results_GC-QC_375$E_Un_Conc_Perc[a] <- (Results$EM[3]/60)*100
306 Results_GC-QC_375$E_Endog_Perc[a] <- (EE/60)*100
307 Results_GC-QC_375$E_Corr_Conc_Perc[a] <- (Results$EC[3]/Results$Corrected_Conc[3])*100
308 Results_GC-QC_375$Endog_Bias_Perc[a] <- (XE/60)*100
309
310 }

```