

Supplementary Data 2

R Script for Automatic Error Calculation of the Endogenous Concentration Correction

The functional version of the “Error_Endogenous_Correction.R” script can be found here: <https://www.dropbox.com/sh/8agalof79qfwe81/AADYzwHQqD4z9Ix8qSsVYd1Sa?dl=0>.

Users can watch a video tutorial detailing its use in R Studio here: https://youtu.be/_gpwQSKDWec.

To summarize, the user should copy to the clipboard a table containing the type of sample (“Cal” or “QC”), the spiked concentration and measurement (area, area ratio or response ratio). Columns can be in any order, but the exact headers “Type”, “Spiked_Conc” and “Measure” should be present. For example, in Excel, the zone in grey below should be copied (Ctrl+C):

	A	B	C	D
1	Sequence A			
2	Name	Type	Spiked_Conc	Measure
3	QC1	QC	60	0.3222
4	QC1	QC	60	0.3293
5	QC2	QC	150	0.7687
6	QC2	QC	150	0.7585
7	QC3	QC	375	1.7982
8	QC3	QC	375	1.7338
9	STD0	Cal	0	0.051
10	STD1	Cal	10	0.0919
11	STD2	Cal	20	0.1353
12	STD3	Cal	50	0.2791
13	STD4	Cal	100	0.5001
14	STD5	Cal	200	0.991
15	STD6	Cal	425	1.9131
16	STD7	Cal	500	2.2055

The script is then sourced in R Studio. At the end of the compilation procedure, the calculated parameters will be displayed in the console as a tibble:

```

Console ~/R/Endogene/ ↵
> source("H:/ENDOGÈNE/PAPER/2018-08-22 - v1/Error_Endogenous_Correction.R")
[1] "Should this analysis use no weight (type 'A'), 1/x weighting ('B') or 1/x^2 weighting ('C')?"
Created file named 'Answer.Rd'.
Edit the file and move it to the appropriate directory.
Answer.RdC
# A tibble: 3 x 6
  value Spiked_Conc   EM   Bias Corrected_Conc   EC
  <chr>   <dbl> <dbl> <dbl>   <dbl> <dbl>
1 QC1      60  1.47  16.6    69.9  1.52
2 QC2     150  3.73  6.63   160.  3.76
3 QC3     375  9.36  2.65   385.  9.37
Warning messages:
1: unknown or uninitialised column: 'EM'.
2: unknown or uninitialised column: 'Bias'.
3: unknown or uninitialised column: 'Corrected_conc'.
> |

```

With EM being the error on the uncorrected estimated concentration (in concentration units), $Bias$ being the percent bias on each QC level if the endogenous concentration was not corrected, $Corrected_Conc$ being the corrected concentration, i.e. $x_c = x_s + x_e$ and EC being the error on the corrected estimated concentration (in concentration units).

```

1 # Error_Endogenous_Correction.R
2 # A script calculating the error introduced by the endogenous concentration correction.
3 # By Brigitte Desharnais, 2018-08-01.
4
5 #####
6 ##### Parameters to be set by the user #####
7 #####
8
9 # Set the working directory.
10 setwd("~/R/Endogene_2018-09-19")
11
12 # Set the weighting to be used: none (A), 1/x (B), 1/x^2 (C)
13 WS <- "C"
14
15 #####
16 #####
17 #####
18
19 # Load the necessary packages
20 library(dplyr)
21
22 # Load data in the clipboard.
23 Data <- read.table(file="clipboard", sep="\t", header=TRUE)
24
25 # Generate a table containing standards and QCs.
26 # This table will be used to estimate the endogenous concentration.
27 Data_Cal <- tbl_df(Data) %>% select(Spiked_Conc, Measure)
28
29 # Generate a table containing the QCs only.
30 # Error (both corrected and uncorrected) will be calculated at those levels.
31 Data_QC <- tbl_df(Data) %>% filter(Type == "QC") %>% select(Name, Spiked_Conc, Measure)
32
33 # Sterilize data set to prevent NaN in weighted functions,
34 # i.e. replace concentrations of 0 by 1e-8.
35 if(WS!="A"){
36   for (k in 1:length(Data_Cal$Spiked_Conc)){
37     if (Data_Cal$Spiked_Conc[k] == 0){
38       Data_Cal$Spiked_Conc[k] <- 1E-8

```

```

39   }
40   }
41 }
42
43 # Load the function OptParam
44 # (Taken from CodeV5.R of the calibration project , written by FCL)
45 OptParam <- function(X,Y,poids , indice){
46   # Finds the calibration parameters for
47   # Weight (poids) 1 (=0) or 1/x (=1) or 1/(x^2) (=2).
48   # Model order (indice) linear (1) or quadratic (2).
49   W <- diag(1/abs(X)^poids)
50   xf <- matrix(rep(X,each = indice +1),ncol=indice+1,byrow=T)
51   ex <- matrix(rep(0:(indice),length(X)),ncol=indice +1 ,byrow=T)
52   Xp = xf^ex
53   matInv <- solve(t(Xp)%*%W%*%Xp)
54   param.optimaux <- rev(matInv%*%t(Xp)%*%W%*%Y)
55   return(param.optimaux)
56 }
57
58 # Calculate the error generated by the endogenous concentration correction according to the
59   weighting scheme selected.
60 # Weighting: none (A)
61 if (WS == "A"){
62
63   # Perform the linear regression and isolate the regression coefficients.
64   Reg_Params <- OptParam(Data_Cal$Spiked_Conc, Data_Cal$Measure, 0, 1)
65   B0 <- Reg_Params[2]
66   B1 <- Reg_Params[1]
67
68   # Calculate the individual (xi-xbar)^2 values for each level.
69   Data_Cal$X_Gap_Sq <- (Data_Cal$Spiked_Conc - mean(Data_Cal$Spiked_Conc))^2
70
71   # Calculate the squared residual (yi - yhat)^2 for each level.
72   Data_Cal$RES <- (Data_Cal$Measure - ((B1*Data_Cal$Spiked_Conc) + B0))^2
73
74   # Calculate the squared standard error (se^2) of the regression.
75   SE2 <- sum(Data_Cal$RES)/(n-2)
76
77   # Calculate the average response (ybar).
78   Y_Moy <- mean(Data_Cal$Measure)
79
80   # Calculate the n (calibration levels) value.
81   n <- as.numeric(length(Data_Cal$Spiked_Conc))
82
83   # Calculate XE, the estimated endogenous concentration.
84   XE <- abs(-B0/B1)
85
86   # Calculate EE, the error on the estimated endogenous analyte concentration.
87   EE <- (sqrt(SE2)/B1)*sqrt((1/n)+((Y_Moy^2)/((B1^2)*(sum(Data_Cal$X_Gap_Sq)))))
88
89   # Identify the different QC levels that need to be analyzed.
90   QC_LVL <- as.character(unique(Data_QC$Name))
91
92   # Create a DF to harbour the results.

```

```

92 Results <- tbl_df(QC.LVL)
93 Results$Spiked_Conc <- as.numeric(unique(Data_QC$Spiked_Conc))
94
95 for (i in 1:length(QC.LVL)){
96
97   # Segregate the data for the QC level analyzed.
98   Data_QC_Temp <- filter(Data_QC, Name == QC.LVL[i])
99
100  # Calculate the number of measurement replicates for this level.
101  m <- length(Data_QC_Temp)
102
103  # Calculate EM, the error on the estimated concentration in the sample (uncorrected).
104  EM <- (sqrt(SE2)/B1)*sqrt((1/m)+(1/n)+(((mean(Data_QC_Temp$Measure) - Y_Moy)^2)/((B1^2)*(
105    sum(Data_Cal$X_Gap_Sq))))))
106
107  #Store this value in the Results DF.
108  Results$EM[i] <- EM
109
110  # Calculate the bias incurred if no correction is made.
111  Results$Bias[i] <- (XE/Results$Spiked_Conc[i])*100
112
113  # Calculate the corrected expected concentration.
114  Results$Corrected_Conc[i] <- Results$Spiked_Conc[i] + XE
115 }
116
117 # Calculate EC, the error on the corrected concentration.
118 Results$EC <- Results$EM + EE
119 }
120
121 # Weighting: 1/x (B)
122 if (WS == "B"){
123
124   # Generate the wt vector to perform linear regression.
125   Data_Cal$WF <- 1/Data_Cal$Spiked_Conc
126
127   # Perform the linear regression and isolate the regression coefficients.
128   Reg_Params <- OptParam(Data_Cal$Spiked_Conc, Data_Cal$Measure, 1, 1)
129   B0 <- Reg_Params[2]
130   B1 <- Reg_Params[1]
131
132   # Calculate the concentrations weighted mean (XW).
133   XW <- sum(Data_Cal$WF*Data_Cal$Spiked_Conc)/sum(Data_Cal$WF)
134
135   # Calculate the measurements weighted mean (YW).
136   YW <- sum(Data_Cal$WF*Data_Cal$Measure)/sum(Data_Cal$WF)
137
138   # Calculate the residuals for each calibration level.
139   Data_Cal$RES <- (Data_Cal$Measure - ((B1*Data_Cal$Spiked_Conc) + B0))^2
140
141   # Calculate the n (calibration levels) value.
142   n <- as.numeric(length(Data_Cal$Spiked_Conc))
143
144   # Calculate the squared standard error (SE2) of the regression.

```

```

145 SE2 <- (sum(Data_Cal$WF*Data_Cal$RES))/(n-2)
146
147 # Calculate EE, the error on the estimated endogenous analyte concentration.
148 EE <- (sqrt(SE2)/B1)*sqrt((1/sum(Data_Cal$WF))+((YW^2)/((B1^2)*(sum(Data_Cal$WF*(Data_Cal$
149   Spiked_Conc^2))-sum(Data_Cal$WF*(XW^2))))))
150
151 # Calculate XE, the estimated endogenous concentration.
152 XE <- abs(-B0/B1)
153
154 # Identify the different QC levels that need to be analyzed.
155 QC_LVL <- as.character(unique(Data_QC$Name))
156
157 # Create a DF to harbour the results.
158 Results <- tbl_df(QC_LVL)
159 Results$Spiked_Conc <- as.numeric(unique(Data_QC$Spiked_Conc))
160
161 for (i in 1:length(QC_LVL)){
162   # Segregate the data for the QC level analyzed.
163   Data_QC_Temp <- filter(Data_QC, Name == QC_LVL[i])
164
165   # Calculate the weighting factor of the QCs.
166   Data_QC_Temp$WF <- 1/Data_QC_Temp$Spiked_Conc
167
168   # Calculate the number of measurement replicates for this level.
169   m <- length(Data_QC_Temp$Measure)
170
171   # Calculate EM, the error on the estimated concentration in the sample (uncorrected).
172   EM <- (sqrt(SE2)/B1)*sqrt((1/(Data_QC_Temp$WF[1]*m))+((1/sum(Data_Cal$WF))+
173     (((mean(Data_QC_Temp$Measure)-YW)^2)*(sum(Data_Cal$WF)))/
174     ((B1^2)*((sum(Data_Cal$WF)*(sum(Data_Cal$WF*(Data_Cal$
175     Spiked_Conc^2))))-
176     ((sum(Data_Cal$WF*Data_Cal$Spiked_Conc)^2))))))
177
178   #Store this value in the Results DF.
179   Results$EM[i] <- EM
180
181   # Calculate the bias incurred if no correction is made.
182   Results$Bias[i] <- (XE/Results$Spiked_Conc[i])*100
183
184   # Calculate the corrected expected concentration.
185   Results$Corrected_Conc[i] <- Results$Spiked_Conc[i] + XE
186 }
187
188 # Calculate EC, the error on the corrected concentration.
189 Results$EC <- Results$EM + EE
190 }
191
192 # Weighting: 1/x^2 (C)
193 if (WS == "C"){
194
195

```

```

196
197 # Perform the linear regression and isolate the regression coefficients.
198 Reg_Params <- OptParam(Data_Cal$Spiked_Conc, Data_Cal$Measure, 2, 1)
199 B0 <- Reg_Params[2]
200 B1 <- Reg_Params[1]
201
202
203
204 # Sterilize data set to remove "missing" standards.
205 #Data_Cal <- tbl_df(Data_Cal) %>% filter(Spiked_Conc > 1E-8)
206
207 # Generate the wt vector to perform linear regression.
208 Data_Cal$WF <- 1/(Data_Cal$Spiked_Conc^2)
209
210
211 # Calculate the concentrations weighted mean (XW).
212 XW <- sum(Data_Cal$WF*Data_Cal$Spiked_Conc)/sum(Data_Cal$WF)
213
214 # Calculate the measurements weighted mean (YW).
215 YW <- sum(Data_Cal$WF*Data_Cal$Measure)/sum(Data_Cal$WF)
216
217 # Calculate the residuals for each calibration level.
218 Data_Cal$RES <- (Data_Cal$Measure - ((B1*Data_Cal$Spiked_Conc) + B0))^2
219
220 # Calculate the n (calibration levels) value.
221 n <- as.numeric(length(Data_Cal$Spiked_Conc))
222
223 # Calculate the squared standard error (SE2) of the regression.
224 SE2 <- (sum(Data_Cal$WF*Data_Cal$RES))/(n-2)
225
226 # Calculate EE, the error on the estimated endogenous analyte concentration.
227 EE <- (sqrt(SE2)/B1)*sqrt((1/sum(Data_Cal$WF))+((YW^2)/((B1^2)*(sum(Data_Cal$WF*(Data_Cal$
    Spiked_Conc^2))-sum(Data_Cal$WF*(XW^2)))))
228
229 # Calculate XE, the estimated endogenous concentration.
230 XE <- abs(-B0/B1)
231
232 # Identify the different QC levels that need to be analyzed.
233 QC_LVL <- as.character(unique(Data_QC$Name))
234
235 # Create a DF to harbour the results.
236 Results <- tbl_df(QC_LVL)
237 Results$Spiked_Conc <- as.numeric(unique(Data_QC$Spiked_Conc))
238
239 for (i in 1:length(QC_LVL)){
240
241   # Segregate the data for the QC level analyzed.
242   Data_QC_Temp <- filter(Data_QC, Name == QC_LVL[i])
243
244   # Calculate the weighting factor of the QCs.
245   Data_QC_Temp$WF <- 1/(Data_QC_Temp$Spiked_Conc^2)
246
247   # Calculate the number of measurement replicates for this level.
248   m <- length(Data_QC_Temp$Measure)

```

```

249
250 # Calculate EM, the error on the estimated concentration in the sample (uncorrected).
251 EM <- (sqrt(SE2)/B1)*sqrt((1/(Data_QC_Temp$WF[1]*m))+(1/sum(Data_Cal$WF))+
252 (((mean(Data_QC_Temp$Measure)-YW)^2)*(sum(Data_Cal$WF)))/
253 ((B1^2)*((sum(Data_Cal$WF)*(sum(Data_Cal$WF*(Data_Cal$
254 Spiked_Conc^2))))-
((sum(Data_Cal$WF*Data_Cal$Spiked_Conc))^2))))
255 )
256 #Store this value in the Results DF.
257 Results$EM[i] <- EM
258
259 # Calculate the bias incurred if no correction is made.
260 Results$Bias[i] <- (XE/Results$Spiked_Conc[i])*100
261
262 # Calculate the corrected expected concentration.
263 Results$Corrected_Conc[i] <- Results$Spiked_Conc[i] + XE
264
265 }
266
267 # Calculate EC, the error on the corrected concentration.
268 Results$EC <- Results$EM + EE
269
270 }
271
272 # Print results.
273 print(Results)
274
275 # Export final results table in the clipboard (e.g. to paste in Excel).
276 write.table(Results, 'clipboard', sep='\t')

```