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).

```r
# Error_Endogenous_Correction.R
# A script calculating the error introduced by the endogenous concentration correction.
# By Brigitte Desharnais, 2018-08-01.

##############################################
#### Parameters to be set by the user ####
##############################################

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

# Set the weighting to be used: none (A), 1/x (B), 1/x^2 (C)
WS <- "C"

##############################################
##############################################
##############################################

# Load the necessary packages
library(dplyr)

# Load data in the clipboard.
Data <- read.table(file="clipboard", sep="\t", header=TRUE)

# Generate a table containing standards and QCs.
# This table will be used to estimate the endogenous concentration.
Data_Cal <- tbl_df(Data) %>% select(Spiked_Conc, Measure)

# Generate a table containing the QCs only.
# Error (both corrected and uncorrected) will be calculated at those levels.
Data_QC <- tbl_df(Data) %>% filter(Type == "QC") %>% select(Name, Spiked_Conc, Measure)

# Sterilize data set to prevent NaN in weighted functions,
# i.e. replace concentrations of 0 by 1e-8.
if(WS!="A"){
  for (k in 1:length(Data_Cal$Spiked_Conc)){
    if (Data_Cal$Spiked_Conc[k] == 0){
      Data_Cal$Spiked_Conc[k] <- 1E-8
```

```
    }
  }
}

# Load the function OptParam
# (Taken from CodeV5.R of the calibration project, written by FCL)
OptParam <- function(X,Y,poids,indice){
  # Finds the calibration parameters for
  # Weight (poids) 1 (=0) or 1/x (=1) or 1/(x^2) (=2).
  # Model order (indice) linear (1) or quadratic (2).
  W <- diag(1/abs(X)^poids)
  xf <- matrix(rep(X,each = indice +1),ncol=indice+1,byrow=T)
  ex <- matrix(rep(0:(indice),length(X)),ncol=indice +1 ,byrow=T)
  Xp = xf^ex
  matInv <- solve(t(Xp)%*%W%*%Xp)
  param.optimaux <- rev(matInv%*%t(Xp)%*%W%*%Y)
  return(param.optimaux)
}

# Calculate the error generated by the endogenous concentration correction according to the
    weighting scheme selected.
# Weighting: none (A)
if (WS == "A"){

  # Perform the linear regression and isolate the regression coefficients.
  Reg_Params <- OptParam(Data_Cal$Spiked_Conc, Data_Cal$Measure, 0, 1)
  B0 <- Reg_Params[2]
  B1 <- Reg_Params[1]

  # Calculate the individual (xi-xbar)^2 values for each level.
  Data_Cal$X_Gap_Sq <- (Data_Cal$Spiked_Conc - mean(Data_Cal$Spiked_Conc))^2

  # Calculate the squared residual (yi - yhat)^2 for each level.
  Data_Cal$RES <- (Data_Cal$Measure - ((B1*Data_Cal$Spiked_Conc) + B0))^2

  # Calculate the squared standard error (se^2) of the regression.
  SE2 <- sum(Data_Cal$RES)/(n-2)

  # Calculate the average response (ybar).
  Y_Moy <- mean(Data_Cal$Measure)

  # Calculate the n (calibration levels) value.
  n <- as.numeric(length(Data_Cal$Spiked_Conc))

  # Calculate XE, the estimated endogenous concentration.
  XE <- abs(-B0/B1)

  # Calculate EE, the error on the estimated endogenous analyte concentration.
  EE <- (sqrt(SE2)/B1)*sqrt((1/n)+((Y_Moy^2)/((B1^2)*(sum(Data_Cal$X_Gap_Sq)))))

  # Identify the different QC levels that need to be analyzed.
  QC_LVL <- as.character(unique(Data_QC$Name))

  # Create a DF to harbour the results.
```

```
  Results <- tbl_df(QC_LVL)
  Results$Spiked_Conc <- as.numeric(unique(Data_QC$Spiked_Conc))

  for (i in 1:length(QC_LVL)){

    # Segregate the data for the QC level analyzed.
    Data_QC_Temp <- filter(Data_QC, Name == QC_LVL[i])

    # Calculate the number of measurement replicates for this level.
    m <- length(Data_QC_Temp)

    # Calculate EM, the error on the estimated concentration in the sample (uncorrected).
    EM <-(sqrt(SE2)/B1)*sqrt((1/m)+(1/n)+(((mean(Data_QC_Temp$Measure) - Y_Moy)^2)/((B1^2)*(
    sum(Data_Cal$X_Gap_Sq)))))

    #Store this value in the Results DF.
    Results$EM[i] <- EM

    # Calculate the bias incurred if no correction is made.
    Results$Bias[i] <- (XE/Results$Spiked_Conc[i])*100

    # Calculate the corrected expected concentration.
    Results$Corrected_Conc[i] <- Results$Spiked_Conc[i] + XE

  }

  # Calculate EC, the error on the corrected concentration.
  Results$EC <- Results$EM + EE
}

# Weighting: 1/x (B)
if (WS == "B"){

  # Generate the wt vector to perform linear regression.
  Data_Cal$WF <- 1/Data_Cal$Spiked_Conc

  # Perform the linear regression and isolate the regression coefficients.
  Reg_Params <- OptParam(Data_Cal$Spiked_Conc, Data_Cal$Measure, 1, 1)
  B0 <- Reg_Params[2]
  B1 <- Reg_Params[1]

  # Calculate the concentrations weighted mean (XW).
  XW <- sum(Data_Cal$WF*Data_Cal$Spiked_Conc)/sum(Data_Cal$WF)

  # Calculate the measurements weighted mean (YW).
  YW <- sum(Data_Cal$WF*Data_Cal$Measure)/sum(Data_Cal$WF)

  # Calculate the residuals for each calibration level.
  Data_Cal$RES <- (Data_Cal$Measure - ((B1*Data_Cal$Spiked_Conc) + B0))^2

  # Calculate the n (calibration levels) value.
  n <- as.numeric(length(Data_Cal$Spiked_Conc))

  # Calculate the squared standard error (SE2) of the regression.
```

```
  SE2 <- (sum(Data_Cal$WF*Data_Cal$RES))/(n-2)

  # Calculate EE, the error on the estimated endogenous analyte concentration.
  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))))))

  # Calculate XE, the estimated endogenous concentration.
  XE <- abs(-B0/B1)

  # Identify the different QC levels that need to be analyzed.
  QC_LVL <- as.character(unique(Data_QC$Name))

  # Create a DF to harbour the results.
  Results <- tbl_df(QC_LVL)
  Results$Spiked_Conc <- as.numeric(unique(Data_QC$Spiked_Conc))

  for (i in 1:length(QC_LVL)){

    # Segregate the data for the QC level analyzed.
    Data_QC_Temp <- filter(Data_QC, Name == QC_LVL[i])

    # Calculate the weighting factor of the QCs.
    Data_QC_Temp$WF <- 1/Data_QC_Temp$Spiked_Conc

    # Calculate the number of measurement replicates for this level.
    m <- length(Data_QC_Temp$Measure)

    # Calculate EM, the error on the estimated concentration in the sample (uncorrected).
    EM <- (sqrt(SE2)/B1)*sqrt((1/(Data_QC_Temp$WF[1]*m))+(1/sum(Data_Cal$WF))+
                                ((((mean(Data_QC_Temp$Measure)-YW)^2)*(sum(Data_Cal$WF)))/
                                   ((B1^2)*((sum(Data_Cal$WF)*(sum(Data_Cal$WF*(Data_Cal$
    Spiked_Conc^2))))-
                                                              ((sum(Data_Cal$WF*Data_Cal$Spiked_Conc))^2))))
    )

    #Store this value in the Results DF.
    Results$EM[i] <- EM

    # Calculate the bias incurred if no correction is made.
    Results$Bias[i] <- (XE/Results$Spiked_Conc[i])*100

    # Calculate the corrected expected concentration.
    Results$Corrected_Conc[i] <- Results$Spiked_Conc[i] + XE

  }

  # Calculate EC, the error on the corrected concentration.
  Results$EC <- Results$EM + EE
}

# Weighting: 1/x^2 (C)
if (WS == "C"){


```

```r

# Perform the linear regression and isolate the regression coefficients.
Reg_Params <- OptParam(Data_Cal$Spiked_Conc, Data_Cal$Measure, 2, 1)
B0 <- Reg_Params[2]
B1 <- Reg_Params[1]



# Sterilize data set to remove "missing" standards.
#Data_Cal <- tbl_df(Data_Cal) %>% filter(Spiked_Conc > 1E-8)

# Generate the wt vector to perform linear regression.
Data_Cal$WF <- 1/(Data_Cal$Spiked_Conc^2)


# Calculate the concentrations weighted mean (XW).
XW <- sum(Data_Cal$WF*Data_Cal$Spiked_Conc)/sum(Data_Cal$WF)

# Calculate the measurements weighted mean (YW).
YW <- sum(Data_Cal$WF*Data_Cal$Measure)/sum(Data_Cal$WF)

# Calculate the residuals for each calibration level.
Data_Cal$RES <- (Data_Cal$Measure - ((B1*Data_Cal$Spiked_Conc) + B0))^2

# Calculate the n (calibration levels) value.
n <- as.numeric(length(Data_Cal$Spiked_Conc))

# Calculate the squared standard error (SE2) of the regression.
SE2 <- (sum(Data_Cal$WF*Data_Cal$RES))/(n-2)

# Calculate EE, the error on the estimated endogenous analyte concentration.
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))))))

# Calculate XE, the estimated endogenous concentration.
XE <- abs(-B0/B1)

# Identify the different QC levels that need to be analyzed.
QC_LVL <- as.character(unique(Data_QC$Name))

# Create a DF to harbour the results.
Results <- tbl_df(QC_LVL)
Results$Spiked_Conc <- as.numeric(unique(Data_QC$Spiked_Conc))

for (i in 1:length(QC_LVL)){

  # Segregate the data for the QC level analyzed.
  Data_QC_Temp <- filter(Data_QC, Name == QC_LVL[i])

  # Calculate the weighting factor of the QCs.
  Data_QC_Temp$WF <- 1/(Data_QC_Temp$Spiked_Conc^2)

  # Calculate the number of measurement replicates for this level.
  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$
    Spiked_Conc^2))))-
254                                                                ((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')
```