

Supplementary Data 2

In silico decision point qualitative method simulation

1 Standard model

```
1 # QUAL_Simulation_Standard.R
2 # In silico decision point qualitative method simulation
3 # By Brigitte Desharnais, last modificaiton 2019-01-23
4
5 #####
6 ##### Parameters to be set by the user #####
7 #####
8
9 # Establishing the concentration at cut-off
10 CO_Conc <- 20
11
12 # Concentrations to be simulated
13 Conc <- c(0, 2, 4, 6, 8, 10, 11, 12, 14, 16, 17, 18, 20, 22, 23, 24, 26, 28, 29, 30,
14           32, 34, 36, 38, 40)
15
16 # Establish the range of possible measurements at cut-off
17 # To be set, for example, based on preliminary experimental values collected.
18 Min_Meas <- 0.008
19 Max_Meas <- 1.050
20
21 # Number of total simulations to be performed
22 # Suggested: >50. Purely guides the number of simulations, can increase this number
23 # a lot.
24 Nb_Sim <- 100
25
26 # Number of measurements (i.e. virtual "spiked samples") at each concentration
```

```

25 # Value will depend on what you are trying to model: an experimental setup? (Use
    experimental value.)
26 # The underlying true value? (Use at least > 50.)
27 Nb_Meas <- 100
28
29 # %RSD at cut-off (will be used to calculate the homoscedastic error)
30 RSD <- 0.15
31
32 #####
33 #####
34 #####
35
36 # Load necessary packages
37 library(dplyr)
38 library(ggplot2)
39 library(extrafont)
40
41 # Initializing an empty data frame to receive the results
42 Data <- data.frame(Conc = double(), Rate = double(), Iter = integer())
43
44 # Perform the specified number of simulations
45 for(i in 1:Nb_Sim){
46   # Setting a known, true measurement at cut-off
47   TR_CO <- runif(1, min = Min_Meas, max = Max_Meas)
48
49   # Calculating the B1 value in y = B1*x
50   B1 <- TR_CO/CO_Conc
51
52   # Calculating the standard deviation value (homoscedastic data is simulated here)
53   SD <- RSD*TR_CO
54
55   # Initializing a vector for the positivity rates to be calculated
56   Rate <- rep(0, times = length(Conc))
57
58   # For each concentration, calculate the positivity rate
59   for(j in 1:length(Conc)){

```

```

60 # Generate a measurements vector under a normal model
61 Meas <- rnorm(Nb_Meas, mean = B1*Conc[j], sd = SD)
62
63 # Calculate the positivity rate and store in the Rate vector
64 # Number of measurements which are above the known, true cut-off value
65 Rate[j] <- sum(Meas > TR_CO)/Nb_Meas*100
66 }
67
68 # Create an iteration vector
69 Iter <- rep(i, times = length(Conc))
70
71 # Bind together Conc, Rate and Iter column and append to Data data.frame.
72 Temp <- cbind(Conc, Rate, Iter)
73 Data <- rbind(Data, Temp)
74 }
75
76 # Convert data.frame to tbl.
77 Data <- tbl_df(Data)
78
79 # Generate a positivity graph (Figure 1(b))
80 ggplot(Data, aes(x = Conc, y = Rate)) +
81   geom_smooth(size = 2, col = "#1824cc") +
82   coord_cartesian(xlim = c(10, 30), ylim = c(0, 100)) +
83   scale_x_continuous(name = "Concentration (ng/mL)") +
84   scale_y_continuous(name = "Positivity Rate (%)") +
85   theme(axis.title = element_text(size = 26, family = "Century Gothic"),
86         axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0)),
87         axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0)),
88         axis.text = element_text(size = 22, family = "Century Gothic")) +
89   geom_vline(xintercept = 20, col = "black", linetype = "longdash", size = 1)

```

2 Model corrected for sampled cut-off and heteroscedasticity

```
1 # QUAL_Simulation_Corrected.R
2 # In silico decision point qualitative method simulation
3 # Modified simulation: sampled cut-off, heteroscedasticity
4 # By Brigitte Desharnais, last modificaiton 2019-01-23
5
6 #####
7 ##### Parameters to be set by the user #####
8 #####
9 # Establishing the concentration at cut-off
10 CO_Conc <- 20
11
12 # Concentrations to be simulated
13 Conc <- c(0, 2, 4, 6, 8, 10, 11, 12, 14, 16, 17, 18, 20, 22, 23, 24, 26, 28, 29, 30,
14           32, 34, 36, 38, 40)
15
16 # Establish the range of possible measurements at cut-off
17 # To be set, for example, based on preliminary experimental values collected.
18 Min_Meas <- 0.008
19 Max_Meas <- 1.050
20
21 # Number of total simulations to be performed
22 # Suggested: >50. Purely guides the number of simulations, can increase this number
23 # a lot.
24 Nb_Sim <- 100
25
26 # Number of measurements (i.e. virtual "spiked samples") at each concentration
27 # Value will depend on what you are trying to model: an experimental setup? (Use
28 # experimental value.)
29 # The underlying true value? (Use at least > 50.)
30 Nb_Meas <- 30
31
32 # Number of cut-off measurements to be performed with each batch (sampled cut-off)
33 # Can set it up to reflect actual experimental work, or to explore the impact of
34 # changing its value.
35 Nb_CO <- 2
```

```

32
33 # %RSD at all levels (heteroscedastic data)
34 # Set based on experimental data or hypothesis (e.g. SWGTOX dictates %RSD below 20%)
35 RSD <- 0.15
36
37 #####
38 #####
39 #####
40
41 # Load necessary packages
42 library(dplyr)
43 library(ggplot2)
44 library(extrafont)
45
46 # Initializing an empty data frame to receive the results
47 Data <- data.frame(Conc = double(), Rate = double(), Iter = integer())
48
49 # Perform the specified number of simulations
50 for(i in 1:Nb_Sim){
51   # Setting a known, true measurement at cut-off
52   TR_TR_CO <- runif(1, min = Min_Meas, max = Max_Meas)
53   TR_CO <- mean(rnorm(Nb_CO, mean = TR_TR_CO, sd = RSD*TR_TR_CO))
54
55   # Calculating the B1 value in  $y = B1*x$ 
56   B1 <- TR_CO/CO_Conc
57
58   # Initializing a vector for the positivity rates to be calculated
59   Rate <- rep(0, times = length(Conc))
60
61   # For each concentration, calculate the positivity rate
62   for(j in 1:length(Conc)){
63     # Generate a measurements vector under a normal model
64     Meas <- rnorm(Nb_Meas, mean = B1*Conc[j], sd = RSD*B1*Conc[j])
65
66     # Calculate the positivity rate and store in the Rate vector

```

```

67   # Number of measurements which are above the known, true cut-off value
68   Rate[j] <- sum(Meas > TR_CO)/Nb_Meas*100
69 }
70
71 # Create an iteration vector
72 Iter <- rep(i, times = length(Conc))
73
74 # Bind together Conc, Rate and Iter column and append to Data data.frame.
75 Temp <- cbind(Conc, Rate, Iter)
76 Data <- rbind(Data, Temp)
77 }
78
79 # Convert data.frame to tbl.
80 Data <- tbl_df(Data)
81
82 # Generate a positivity graph
83 ggplot(Data, aes(x = Conc, y = Rate)) +
84   geom_smooth(size = 2, col = "#1824cc") +
85   coord_cartesian(xlim = c(10, 30), ylim = c(0, 100)) +
86   scale_x_continuous(name = "Concentration (ng/mL)") +
87   scale_y_continuous(name = "Positivity Rate (%)") +
88   theme(axis.title = element_text(size = 26, family = "Century Gothic"),
89         axis.title.y = element_text(margin = margin(t = 0, r = 20, b = 0, l = 0)),
90         axis.title.x = element_text(margin = margin(t = 20, r = 0, b = 0, l = 0)),
91         axis.text = element_text(size = 22, family = "Century Gothic")) +
92   geom_vline(xintercept = 20, col = "black", linetype = "longdash", size = 1)
93
94 Summary <- Data %>% group_by(Conc) %>%
95   summarise(min = min(Rate), max = max(Rate), mean = mean(Rate),
96             Q5 = quantile(Rate, prob = 0.05), Q95 = quantile(Rate, prob = 0.95))

```

3 Corrected model and expected performance evaluation

```
1 # QUAL_Simulation_Performance.R
2 # In silico decision point qualitative method simulation
3 # Modified simulation & calculation of all performance statistics
4 # By Brigitte Desharnais, last modification 2019-01-23
5
6 #####
7 ##### Parameters to be set by the user #####
8 #####
9 # Establishing the concentration at cut-off
10 CO_Conc <- 20
11
12 # Concentrations to be simulated
13 Conc <- c(0, 2, 4, 6, 8, 10, 11, 12, 14, 16, 17, 18, 20, 22, 23, 24, 26, 28, 29, 30,
14           32, 34, 36, 38, 40)
15
16 # Establish the range of possible measurements at cut-off
17 Min_Meas <- 0.008
18 Max_Meas <- 1.050
19
20 # Number of total simulations to be performed
21 # Suggested: >50. Purely guides the number of simulations, can increase this number
22 # a lot.
23 Nb_Sim <- 100
24
25 # Number of measurements (i.e. virtual "spiked samples") at each concentration
26 # Value will depend on what you are trying to model: an experimental setup? (Use
27 # experimental value.)
28 # The underlying true value? (Use at least > 50.)
29 Nb_Meas <- 30
30
31 # Number of cut-off measurements to be performed with each batch (sampled cut-off)
32 # Can set it up to reflect actual experimental work, or to explore the impact of
33 # changing its value.
34 Nb_CO <- 2
35
```

```

32 # %RSD at all levels (heteroscedastic data)
33 # Set based on experimental data or hypothesis (e.g. SWGTOX dictates %RSD below 20%)
34 RSD <- 0.15
35
36 #####
37 #####
38 #####
39
40 # Load necessary packages
41 library(dplyr)
42 library(ggplot2)
43 library(extrafont)
44
45 # Initializing an empty data frame to receive the results
46 Data <- data.frame(Conc = double(), Nb_Pos = double(), Iter = integer())
47
48 # Perform the specified number of simulations
49 for(i in 1:Nb_Sim){
50   # Setting a known, true measurement at cut-off
51   TR_TR_CO <- runif(1, min = Min_Meas, max = Max_Meas)
52   TR_CO <- mean(rnorm(Nb_CO, mean = TR_TR_CO, sd = RSD*TR_TR_CO))
53
54   # Calculating the B1 value in  $y = B1*x$ 
55   B1 <- TR_CO/CO_Conc
56
57   # Initializing a vector for the positivity rates to be calculated
58   Nb_Pos <- rep(0, times = length(Conc))
59
60   # For each concentration, calculate the positivity rate
61   for(j in 1:length(Conc)){
62     # Generate a measurements vector under a normal model
63     Meas <- rnorm(Nb_Meas, mean = B1*Conc[j], sd = RSD*B1*Conc[j])
64
65     # Calculate the positivity rate and store in the Rate vector
66     # Number of measurements which are above the known, true cut-off value

```



```

67   Nb_Pos[j] <- sum(Meas > TR.CO)
68   }
69
70   # Create an iteration vector
71   Iter <- rep(i, times = length(Conc))
72
73   # Bind together Conc, Rate and Iter column and append to Data data.frame.
74   Temp <- cbind(Conc, Nb_Pos, Iter)
75   Data <- rbind(Data, Temp)
76 }
77
78 # Convert data.frame to tbl.
79 Data <- tbl_df(Data)
80
81 # Remove all observations at the cut-off concentration
82 #(which can't be classified into "true" or "false" positive/negative)
83 Data <- Data %>% filter(Conc != CO_Conc)
84
85 # Initialize empty columns for true/false positives/negatives
86 Data$TP <- 0
87 Data$FP <- 0
88 Data$TN <- 0
89 Data$FN <- 0
90
91 # Calculate the number of TP, TN, FP, FN for each concentration
92 for(k in 1:length(Data$Nb_Pos)){
93   if(Data$Conc[k] < CO_Conc){
94     # For samples actually negative, only a TN or FP result is possible.
95     Data$FP[k] <- Data$Nb_Pos[k]
96     Data$TN[k] <- Nb_Meas - Data$Nb_Pos[k]
97   } else {
98     # For samples actually positive, only a TP or FN result is possible.
99     Data$TP[k] <- Data$Nb_Pos[k]
100    Data$FN[k] <- Nb_Meas - Data$Nb_Pos[k]
101  }
102 }

```

```

103
104 # Calculation of the performance statistics
105 Data$FNR <- Data$FN/(Data$FN + Data$TP)*100
106 Data$FPR <- Data$FP/(Data$FP + Data$TN)*100
107 Data$REL <- (Data$TP + Data$TN) / (Data$TP + Data$TN + Data$FP + Data$FN)
108 Data$SEL <- Data$TN / (Data$TN + Data$FP)*100
109 Data$SEN <- Data$TP / (Data$TP + Data$FN)*100
110
111 # Summary table of the performance statistic per concentration level.
112 # Table lists the minimum and maximum value observed, the mean rate for all
    simulations
113 # performed, and the 5% and 95% percentile values observed.
114 # These summary tables were used to set the threshold values stated in the paper.
115 Summary_FNR <- Data %>% group_by(Conc) %>% arrange(FNR) %>%
116   summarise(min = min(FNR), Q5 = nth(FNR, 5), mean = mean(FNR),
117             Q95 = nth(FNR, 95), max = max(FNR))
118
119 Summary_FPR <- Data %>% group_by(Conc) %>% arrange(FPR) %>%
120   summarise(min = min(FPR), Q5 = nth(FPR, 5), mean = mean(FPR),
121             Q95 = nth(FPR, 95), max = max(FPR))
122
123 Summary_REL <- Data %>% group_by(Conc) %>% arrange(REL) %>%
124   summarise(min = min(REL), Q5 = nth(REL, 5), mean = mean(REL),
125             Q95 = nth(REL, 95), max = max(REL))
126
127 Summary_SEL <- Data %>% group_by(Conc) %>% arrange(SEL) %>%
128   summarise(min = min(SEL), Q5 = nth(SEL, 5), mean = mean(SEL),
129             Q95 = nth(SEL, 95), max = max(SEL))
130
131 Summary_SEN <- Data %>% group_by(Conc) %>% arrange(SEN) %>%
132   summarise(min = min(SEN), Q5 = nth(SEN, 5), mean = mean(SEN),
133             Q95 = nth(SEN, 95), max = max(SEN))

```