**-Supplementary Information-**

**A simple Bayesian method for assessing the standard error of equivalent dose estimates**

Jun Peng1\*, ZhiBao Dong1

Cold and Arid Regions Environmental and Engineering Research Institute, Chinese Academy of Sciences, 730000 Lanzhou, China

\*Corresponding author: pengjun10@mails.ucas.ac.cn

***WinBUGS*** source-code for simulate ED values from a linear or a saturating exponential growth curve using a simple Bayesian method in which an ED value is treated as a node that depends on a variable whose posterior distribution can be constructed using the Bayesian theory and sampled.

**Appendix I: Simulate ED values from a linear growth curve.**

Parameter a (the slope) is stimulated using a prior of U (0, 10).

Parameter b (the intercept) and the ED value are regarded as nodes.

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

### Step 1: Check the model.

model {

 for (i in 1:n) {

 mu.mean[i] <- a\*x[i]+b ### fitted standardized OSL

 mu.precision[i] <- 1/pow(sy[i], 2)

 y[i] ~ dnorm(mu.mean[i], mu.precision[i]) ### Observed standardized OSL

 } # end for

 ###

 ### Prior of the slope (above zero).

 a ~ dunif(0, 10)

 ###

 ### Calculate the intercept (node b) using x, y, sy and a.

 for (i in 1:n) {

 w[i] <- 1/pow(sy[i], 2)

 x.w[i] <- x[i]\*w[i]

 y.w[i] <- y[i]\*w[i]

 } # end for

 b <- ( sum(y.w[]) -a\*sum(x.w[]) )/sum(w[])

 ###

 ### Natural standardized OSL follows Gaussian distribution.

 Y0 ~ dnorm(y0, y0.precision)

 y0.precision<- 1/pow(sy0, 2)

 ###

 ### Calculate node ED.

 ED<- (Y0-b)/a

} # end model

###

### Step 2: Load the data.

list(x=c(3.9, 7.8, 11.7, 15.6, 0.0, 3.9),

 y=c(0.273, 0.538, 0.808, 1.075, 0.005, 0.256),

 sy=c(0.024, 0.033, 0.043, 0.051, 0.013, 0.019),

 y0=0.732, sy0=0.046, n=6)

### Step 3: Compile the code.

### Step 4: Load the initial.

list(a=0.1)

### Step 5: Update the simulation.

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

**Appendix II: Simulate ED values from a saturating exponential growth curve.**

Parameter b (the reciprocal of the saturating level) is stimulated using a prior of U (0, 1). Parameter a, c and the ED value are regarded as nodes.

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

### Step 1: Check the model.

model {

 for (i in 1:n) {

 mu.mean[i] <- a\*(1-exp(-b\*x[i]))+c ### fitted standardized OSL

 mu.precision[i] <- 1/pow(sy[i], 2)

 y[i] ~ dnorm(mu.mean[i], mu.precision[i]) ### Observed standardized OSL

 } # end for

 ###

 ### Prior of b (the reciprocal of the saturating dose).

 b ~ dunif(0, 1)

 ###

 ### Calculate node a and node c using x, y, sy and b.

 for (i in 1:n) {

 w[i] <- 1/pow(sy[i], 2)

 y.w[i] <- y[i]\*w[i]

 X.w[i] <- (1-exp(-b\*x[i]))\*w[i]

 Xy.w[i] <- (1-exp(-b\*x[i]))\*y[i]\*w[i]

 XX.w[i] <- (1-exp(-b\*x[i]))\*(1-exp(-b\*x[i]))\*w[i]

 } # end for

 a <- (sum(w[])\*sum(Xy.w[])-sum(X.w[])\*sum(y.w[])) / (sum(w[])\*sum(XX.w[]) - sum(X.w[])\*sum(X.w[]))

 c <- (sum(y.w[]) - a\*sum(X.w[])) / sum(w[])

 ###

 ### Natural standardized OSL follows Gaussian distribution.

 Y0 ~ dnorm(y0, y0.precision)

 y0.precision<- 1/pow(sy0, 2)

 ###

 ### Calculate node ED.

 ED <- -log( (a-Y0+c)/a )/b

} # end model

###

### Step 2: Load the data.

list(x=c(10.4, 20.8, 31.2, 41.6, 0.0, 10.4),

 y=c(0.76, 1.443, 2.043, 2.496, 0.003, 0.869),

 sy=c(0.0331, 0.053, 0.069, 0.082, 0.004, 0.03),

 y0=1.416, sy0=0.06, n=6)

### Step3: Compile the code.

### Step 4: Load the initial.

list(b=0.05)

### Step 5: Update the simulation.

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