Supplementary Material to “QRAD (Quantal Risk Assessment Database): a database for exploring patterns in quantal dose-response data in risk assessment and its application to develop priors for Bayesian dose-response analysis”

**Matthew W. Wheeler1\*, Walter W. Piegorsch2, and A. John Bailer3**

This supplementary document provides supporting material for *QRAD (Quantal Risk Assessment Database): a database for exploring patterns in quantal dose-response data in risk assessment and its application to develop priors for Bayesian dose-response analysis* by M.H. Wheeler, et al., (the “main document”). The various sections below address a variety of supplemental/supporting topics, and are not intended to flow naturally between each other. They are, however, presented in roughly the same order in which their counterpart topics appear in the main document. We assume the database has been imported as an R object into an R workspace, or as a **csv** file into SAS®.

S.1. Basic summaries: simple frequency tables and data sources

For a further illustration of QRAD’s use, consider construction of a two-way frequency table summarizing features of the data sources, say, numbers of dose groups (as in main document Fig. 1) vs. data source (i.e., IRIS, CalEPA, etc.). We start by constructing a data frame that has a variable with the number of dose groups in each experiment (ID) and a variable describing the data source. The R *dplyr*package ([Wickham and Francois, 2016](#_ENREF_8)) is again our primary tool:

Summary2 <- final.data %>%

group\_by(ID, data.source) %>%

summarise(nDoseGrps = n())

The resulting two-way frequency table is constructed via the **table** function, here,

group\_table <- with(Summary2, table(data.source, nDoseGrps))

print( group\_table )

which results in the following cross-classified table (output edited):

nDoseGrps

data.source 2 3 4 5 6 7 8 10 13 16

IRIS 27 116 155 36 30 3 0 3 1 0

CalEPA 61 83 71 12 3 1 1 0 1 2

PPRTV 2 26 41 1 5 0 0 0 0 0

HEAST 5 6 10 0 4 0 0 0 0 0

ATSDR 0 1 2 0 0 0 0 0 0 0

OPP 0 1 14 9 0 0 0 0 0 0

From this tabular output we see that only the IRIS and CalEPA data sources contributed experiments with more than 6 dose groups to the database. The fewest studies were contributions by ATSDR. The (sorted) row margins can be easily obtained as

sort(margin.table(group\_table,1))

producing (output edited)

data.source

ATSDR OPP HEAST PPRTV CalEPA IRIS

3 24 25 75 235 371

which displays the relative contribution of the different data sources to the database in increasing order. Namely, ATSDR only contributed 3 studies to this database while IRIS contributed 371. As an aside, the frequency output / numerator of the relative frequencies in main document §2.3 on number of IDs can be produced from this contingency table via the command

margin.table(group.table,2).

S.2. Subsetting data to only have more than two dose groups

Returning to the **Summary1** object from the main document’s §2.3, we find that 95 of the 733 of the quantal data sets (approximately 13%) in our database contain only two doses:

table(Summary1$nDoseGrps)

 2   3   4   5   6   7   8  10  13  16

95 233 293  58  42   4   1   3   2   2

In effect, these two-dose data sets are precluded from use in studying a dose-response pattern, although we feel they are still valuable as indicators of a toxicological effect in the experiments from which they were drawn. For investigators wishing to study dose responses in data sets with only three or more doses, those data sets with two doses can be ignored or removed. For instance, the R function **filter** (part of the *dplyr* package) can be used to select data sets with more than two doses.

Now, we select these data sets with a row for each unique dose group for a particular experiment using the R code below,

Summary3 <- final.data %>%

group\_by(ID) %>%

mutate(nDoseGrps = n()) %>%

select(ID, chemical, nDoseGrps, dose, r.dose, x, n) %>%

filter(nDoseGrps > 2)

One can easily confirm that chemicals with 3 or more dose groups are selected, e.g., via

**sort(unique(Summary3$nDoseGrps))**

we find the following numbers of dose groups in the subset database (output edited):

3 4 5 6 7 8 10 13 16

Further operations on this new data frame can be conducted in similar fashion to those seen above.

S.3. Applying a function to all experiments in the database

For a more-involved example of the type of toxicological calculations we might consider for these data, we fit the simple logistic regression model corresponding to main document Equation (6) to each data set in QRAD and then calculate a *median effective dose*, ED50, for the fit ([Piegorsch and Bailer, 2005, §4.1.1](#_ENREF_5)). That is, given proportions, *Y*i/*n*i, and a series of doses, *d*i (i = 1, ..., m), rescale the doses to *u*i = *d*i/max{*d*i}—the variable **r.dose** in the database—and assume a binomial model for the observed counts: *Y*i ~ indep. *Bin*.(*n*i,πi). Here the response probability πi is modeled as a function of the rescaled dose: πi = π(*u*i). The simple logistic dose-response model is π(*u*) = 1/(1 + exp{–0 – 1*u*}), with ED50 given by –0/1. The unknown -parameters and the ED50 are estimated from the data; maximum likelihood is a favored approach ([Piegorsch and Bailer, 2005, §A.4.3](#_ENREF_5)). In R, we can fit the logistic model to each experiment via the **glm** function with its **family= binomial** option, using the rescaled doses **r.dose**. The estimated ED50 based on this fit is then the ratio –0/1 from the R output.

We begin by creating a new data frame that includes two new variables: (i) **nobs**: a count of the number of observations in a dose group with the response and (ii) **nNotObs**: a count of the number of observations in a dose group without the response:

model\_final<- final.data %>%

  mutate(nobs=obs, nNotObs=n-obs)

Next, we split the data frame into lists by the original study’s ID and fit the logistic model separately to each study using R’s **glm** function:

split\_model\_final <- split(model\_final, model\_final$ID)

model\_logistic\_reg\_fits <- lapply( split\_model\_final,

                        function(x) glm(cbind(nobs, nNotObs)~r.dose,

                                data=x, family=binomial) )

Note that this will generate a number of R errors and warnings (e.g., algorithm did not converge, or fitted probabilities numerically 0 or 1 occurred). Some of these data sets are not fit well by a linear logistic model—see below.

We first extract the coefficients from each fit to estimate the ED50:

model\_coef <- lapply(model\_logistic\_reg\_fits, coef)

After the model coefficients are extracted from the fits, the ED50s are calculated and stored as a vector via

ED50.est <- lapply(model\_coef, function(x) -x[1]/x[2])

ED50\_vec <- unlist(ED50.est)

This produces 27 estimated ED50s less than zero and five ED50s greater than 5.0. Both reflect instability in the estimates since, as constructed here, (i) a dose cannot drop below zero and (ii) the doses have been scaled to have a maximum value of 1.0. For this analysis, we exclude any of these 32 possible values, which results in a new data frame containing 701 ED50 estimates. Viz.,

ED50\_tmp <- data.frame(ED50=ED50\_vec)

ED50\_df <- ED50\_tmp %>%

filter(ED50>=0, ED50<=5)

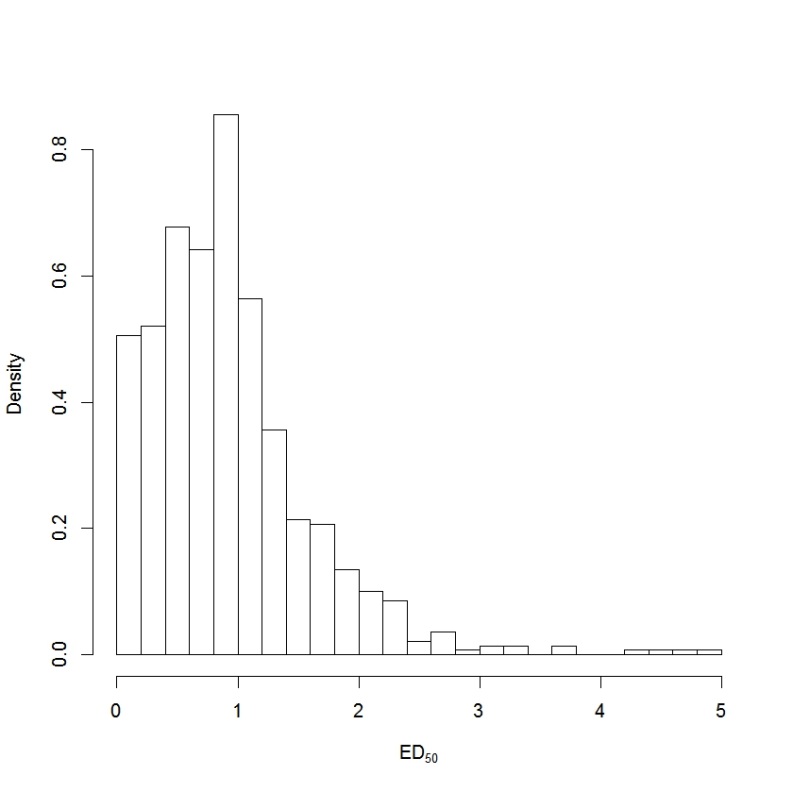
After excluding the missing ED50s in a new data frame **noMissED50.df** with complete information, we construct a histogram of the final, estimated ED50s in Fig. S.1 using R’s stock **hist** function. As might be expected, the sample histogram displays a substantial right skew. A similar graphic can be constructed via the **ggplot2** function **geom\_histogram** with **binwidth** specified. Sample R code is

# distribution of ED50

ggplot(ED50\_df,aes(x=ED50)) +

geom\_histogram(color='darkgrey',fill='white', binwidth=0.25) +

theme\_minimal()



**Fig. S.1**. Histogram of estimated median lethal doses (ED50) from QRAD.

It is also illustrative to examine the various empirical dose-response patterns from the various studies, stratified by the different data sources. This can provide insights into problems encountered when attempting formal model fits such as a logistic regression. The dose-response profiles for each experiment were plotted as a line graph with the display faceted by the data sources. Sample **ggplot** commands to produce this display are

ggplot(final.data, aes(x=r.dose,y=obs/n, group=ID)) +

 geom\_line(alpha=.2) +

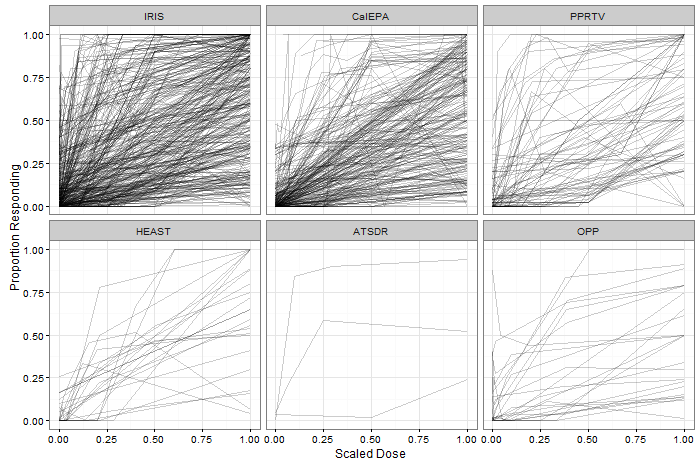
 theme\_bw() +

 xlab('Scaled Dose') +

 ylab('Proportion Responding') +

facet\_wrap(~data.source)

The results appear in Fig. S.2. While many of the dose-response patterns therein appear monotonic and non-decreasing, there remain a large number of experiments with a drop or reduction in the response at higher dose levels, colorfully known as an ‘umbrella pattern’ ([Mack and Wolfe, 1981](#_ENREF_3)). This is often seen when dose-related toxicity or other competing risks lead to mortality in the experimental subject before a non-lethal outcome such as cancer is observed.



**Fig. S.2**. QRAD dose-response patterns, stratified by data source

We can modify Fig. S.2 to highlight the patterns associated with a non-significant test of trend. In the last example of this section, we first fit a separate logistic regression to each experiment (ID) in the database. We extract the p-value associated with a test on the significance of the slope from the logistic regression and then highlight in color those dose-response patterns based on this quantity. To start, we use **lapply** to fit all of the logistic regressions and extract the coefficients and p-value (the 2nd row and 4th column of the matrix of results obtained from the coefficient extractor function):

model\_final<- final.data %>%

mutate(nobs=obs, nNotObs=n-obs)

split\_model\_final <- split(model\_final, model\_final$ID)

model\_logistic\_reg\_fits <-

lapply( split\_model\_final,

function(x) glm(cbind(nobs, nNotObs)~r.dose,

data=x, family=binomial) )

model\_summ <- lapply(model\_logistic\_reg\_fits, summary)

model\_coef <- lapply(model\_summ, coefficients)

model\_Trend\_Pvalue <- lapply(model\_coef, function(x) x[2,4])

These quantities are placed into a data frame that is then joined with the original database in preparation of plotting. We restrict attention in this plot to experiments with 3, 4, 5, or 6 dose groups.

Data\_3to6grps <- final.data %>%

group\_by(ID, data.source) %>%

mutate(nDoseGrps = n()) %>%

filter(nDoseGrps > 2 & nDoseGrps<7)

# Add p-value from logistic regression fit

Data\_3to6grps\_plus <- left\_join(Data\_3to6grps,trend\_df)

Data\_3to6grps\_plus %>%

select(ID, dose, r.dose, obs, n, nDoseGrps)

For instance, we plot dose-response patterns with p-values from the logistic slope test larger than 0.10 for the CalEPA experiments using the following code:

Data\_3to6grps\_plus %>%

mutate(NSig = (Pvalue > 0.10 )) %>%

filter(data.source=="CalEPA") %>%

ggplot(aes(x=r.dose,y=obs/n, group=ID)) +

geom\_line(aes(color=NSig, width=2), alpha=.5) +

theme\_bw() +

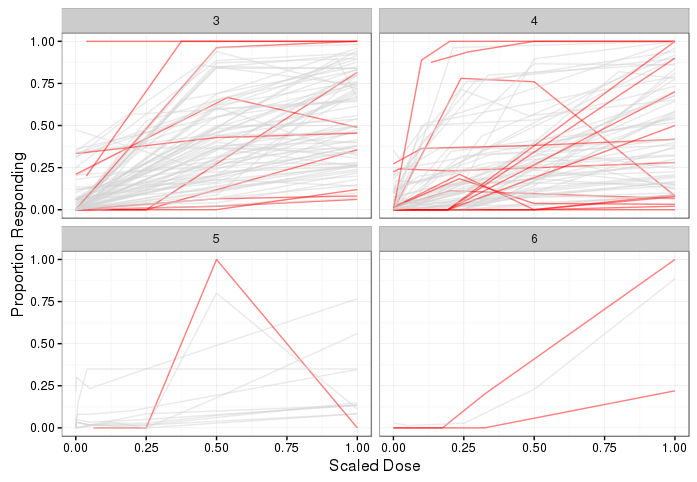
xlab('Scaled Dose') +

ylab('Proportion Responding') +

facet\_wrap(~nDoseGrps) +

scale\_color\_manual(values=c("light grey", "red"), guide=FALSE)

which produces Fig. S.3.



**Fig. S.3**. QRAD dose-response patterns, stratified by number of dose groups for CalEPA studies. Patterns with a p-value > 0.10 from a logistic regression test of the slope parameter are colored red in this plot.

While some highlighted patterns in this plot are flat and consistent with what we would expect from a case where the response does not change over dose (e.g. 0/48, 1/49, 3/49), others are surprising (e.g., 0/19, 0/46, 19/45). These latter cases illustrate enigmatic issues with the logistic regression test when zero or flat response patterns are observed at all but the highest dose(s).

S.4. Using EPA BMDS software to fit multistage models

As our database is based upon quantal-data studies employed in the regulatory process, interest may exist in comparing potential differences in Benchmark Dose (BMD) estimates ([Crump, 1984](#_ENREF_1)) and in their corresponding 100(1–α)% lower bounds (BMDLs), which are often starting points in quantitative risk assessments ([Izadi *et al.*, 2012](#_ENREF_2); [U.K. Committee on Carcinogenicity (COC), 2014](#_ENREF_6)). We do this using the multistage model as implemented in the U.S. EPA Benchmark Dose Software (BMDS) suite ([U.S. EPA, 2016](#_ENREF_7)). Expanding the multistage response function from main document Equation (2), we study multistage models for the response probability π(*u*) containing a first-degree, second-degree, or third-degree polynomial. That is,

π(*u*) =  + (1 – )(1 – exp{–1*u* – *u*2 – 3*u*3}),

where γ is the background response rate and the coefficients 1, , and 3 represent terms for the first-, second-, and third-degree components of the polynomial predictor, respectively. For this analysis, the first-degree multistage model sets  = = 0, while the second-degree model sets only *=* 0. For all model fits the coefficients are constrained to be greater than or equal to zero, which for *u* ≥ 0 imposes the necessary restriction that 0 ≤ π(*u*) ≤ 1. It also forces π(*u*) to be monotone increasing.

As the parameters are constrained, it is possible for higher order models to degenerate into other models and one may question the necessity of using higher order terms for regulatory purposes. To investigate this, an R script is given below to compare all three models. The following analysis and corresponding R code assumes that the files **multistage.exe**, **bmd.dll**, **libgcc\_s\_dw2-1.dll**, **libgfortran-3.dll** and **libquadmath-0.dll** have been copied from the BMDS install directory and placed in the directory from which this R script is executed. The user-defined function **BMDCompute**, which is employed below, calls the multistage model and is given in the next supplement Section. The BMDs are computed by setting the extra risk under the multistage model, RE(u) = 1 – exp{–1*u* – 2*u*2 – 3*u*3}, equal to 10%.

# fit multistage models with cubic, quadratic or linear components

BMD.MULTISTAGE3 <-final.data %>% group\_by(ID)  %>%

                 select(chemical,r.dose,n,obs) %>%

                 do(BMDCompute(.,3))

#The unique function is used to remove an extra BMD value that may be

#returned by the procedure

BMD.MULTISTAGE3<-unique(BMD.MULTISTAGE3)

BMD.MULTISTAGE2 <-final.data %>% group\_by(ID)  %>%

                        select(chemical,r.dose,n,obs) %>%

                        do(BMDCompute(.,2))

BMD.MULTISTAGE2<-unique(BMD.MULTISTAGE2)

BMD.MULTISTAGE1 <-final.data %>% group\_by(ID)  %>%

                          select(chemical,r.dose,n,obs) %>%

                     do(BMDCompute(.,1))

BMD.MULTISTAGE3<-unique(BMD.MULTISTAGE3)

# collect the BMD estimates from the 3 multistage models and

#   calculate their correlations

V = cov(cbind(BMD.MULTISTAGE1[,3],

             BMD.MULTISTAGE2[,3],

             BMD.MULTISTAGE3[,3]),use="pairwise.complete.obs")

cov2cor(V)

This yields the estimated BMD correlation matrix

BMD BMD BMD

BMD 1.0000000 0.8534608 0.8132934

BMD 0.8534608 1.0000000 0.9514441

BMD 0.8132934 0.9514441 1.0000000

Next, the R code

# collect the BMDLs from the 3 multistage models and

#   calculate their correlations

V = cov(cbind(BMD.MULTISTAGE1[,4],

             BMD.MULTISTAGE2[,4],

             BMD.MULTISTAGE3[,4]),use="pairwise.complete.obs")

cov2cor(V)

yields the BMDL correlation matrix

BMDL BMDL BMDL

BMDL 1.0000000 0.9116890 0.8690537

BMDL 0.9116890 1.0000000 0.9607213

BMDL 0.8690537 0.9607213 1.0000000

We see that the BMD estimates from the database exhibit rather high pairwise correlation coefficients: e.g., the correlation between the first-order model’s BMD estimates and those from the second-order model is found to be 0.85. Similarly, the correlation between the BMDLs from those two models is even higher, at 0.91. Further exploration would show that for these data sets, 40% of the first-order and second-order BMDLs are identical, where the quadratic coefficient is being estimated as zero for the second-order multistage model. Roughly similar values are evidenced for the other pairwise correlations. For example, with the third-order model there is even less divergence from the lower, second-order model: both the BMD estimates and the BMDLs show increases in their correlations, to 0.95 and 0.96, respectively.

This analysis shows that there is a complex interrelationship between the three model forms, and that a multistage model higher than second degree may not be necessary. This corresponds to precious indications that at most a second-degree multistage model is adequate for dichotomous dose-response relationships ([Nitcheva *et al.*, 2007](#_ENREF_4)). It requires the file `multistage.exe,’ which is available with BMDS 2.7.

S.5. R BMDCompute FUNCTION

# the following function performs the following operations...

# Call the EPA BMDS software to fit multistage models.

# Write the command file that is submitted to BMDS

# Input:

# data := data frame with the following structure

# deg := degree of the multistage model that was fit

# Output:

# dataframe containing two columns: BMD, BMDL

library (dplyr)

BMDCompute<- function(data,deg){

print(data)

# Check if there are more than two data points

# also check if there is a non 0% or 100% response.

if ((nrow(data) > 2) \* (sum((data[,5]/data[,4] < 1) \* ( data[,5]/data[,4]

> 0 )) > 0) ){

X = c( 'Multistage',

'BMDS\_Model\_Run',

'NoDataFile','.\\run.out',

sprintf('%d %d',nrow(data),deg),

'500 1.00E-08 1.00E-08 0 1 1 0 0',

'0.1000 0 0.95',

paste(rep('-9999 ',deg+1),collapse=""),

'0',

paste(rep('-9999 ',deg+1),collapse=""),

'Dose Effect NEGATIVE\_RESPONSE')

for (i in 1:nrow(data)){

X = c(X,sprintf('%f %d %d',data[i,3],as.numeric(data[i,5]),as.numeric(data[i,4]-data[i,5])))

}

write(X,"out.(d)",sep="\n")

system("multistage out.(d)")

infile <- file("out.out", "rt")

file.lines=readLines(infile)

if (length(grep("BMD computation failed",file.lines)) > 0){

close(infile)

return(data.frame(chemname = data[1,2],BMD = NA, BMDL=NA))

}else{

BMD = as.numeric(trimws(sub("BMD =\*","",file.lines[grep("BMD =\*",file.lines)]), "both"))

BMDL = as.numeric(trimws(sub("BMDL =\*","",file.lines[grep("BMDL =\*",file.lines)]),"both"))

close(infile)

return(data.frame(chemname = data[1,2],BMD=BMD,BMDL=BMDL))

}

}

else {

return(data.frame(chemname = data[1,2],BMD = NA, BMDL=NA))

}

} #end of function

S.6. Empirical distributions and recommended priors for all models in table 1

To complement the graphical summaries in main document Figs. 2–4 of the empirical distributions seen in QRAD, we present here similar graphic displays for the remaining model/parameter combinations in main document Table 1. For easy referral, these are as follows:

(1) Quantal-linear → see Fig. S.4

Quantal-linear 1→ see Fig. S.5

(2) Multi-stage → see Fig. S.6

Multi-stage 1→ see Fig. S.7

Multi-stage 2→ see Fig. S.8

(3) Weibull → see Fig. S.9

Weibull → see main document Fig. 4

Weibull → see Fig. S.10

(4) Gamma → see Fig. S.11

Gamma → see Fig. S.12

Gamma → see Fig. S.13

(5) Logistic 0→ see Fig. S.14

Logistic 1→ see Fig. S.15

(6) Log-logistic → see Fig. S.16

Log-logistic 0→ see main document Fig. 3

Log-logistic 1→ see Fig. S.17

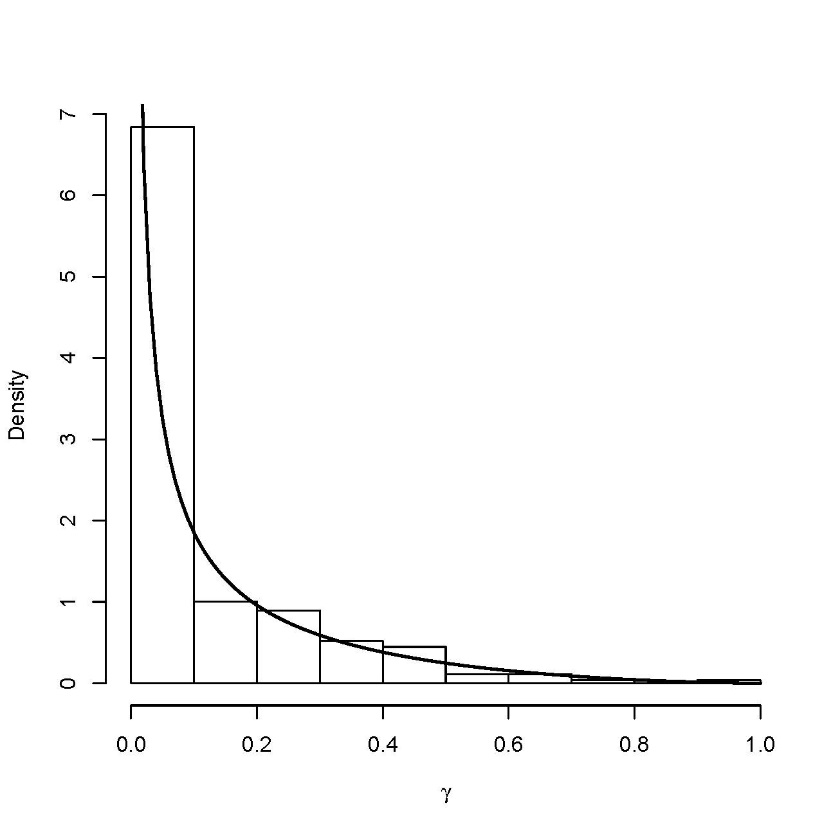
(7) Probit 0→ see Fig. S.18

Probit 1→ see main document Fig. 2

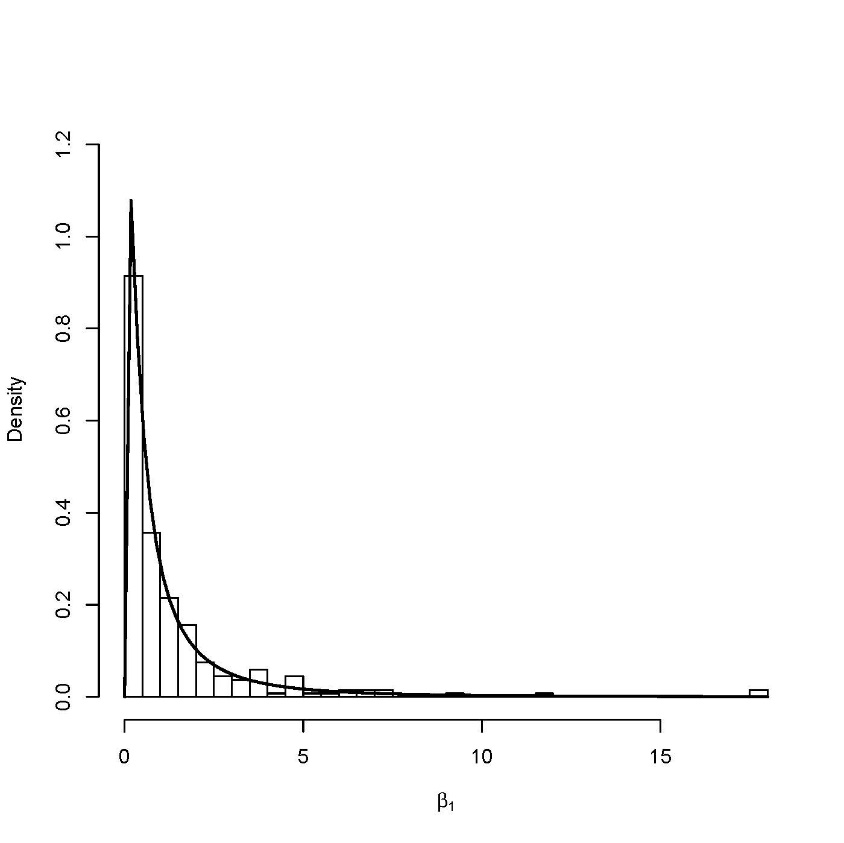
(8) Log-probit → see Fig. S.19

Log-probit 0→ see Fig. S.20

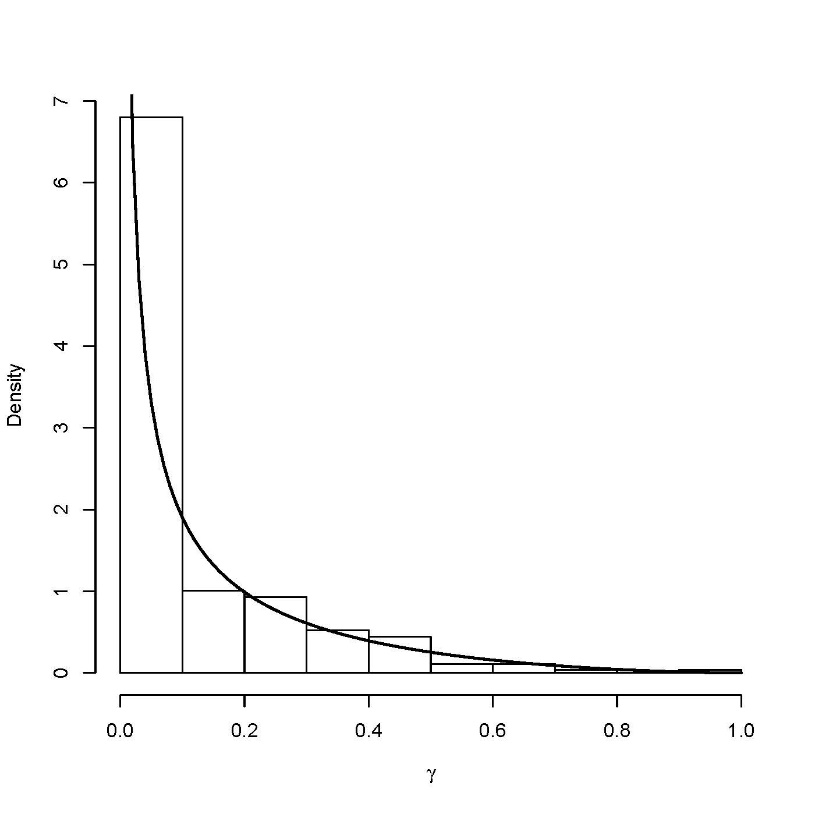
Log-probit 1→ see Fig. S.21



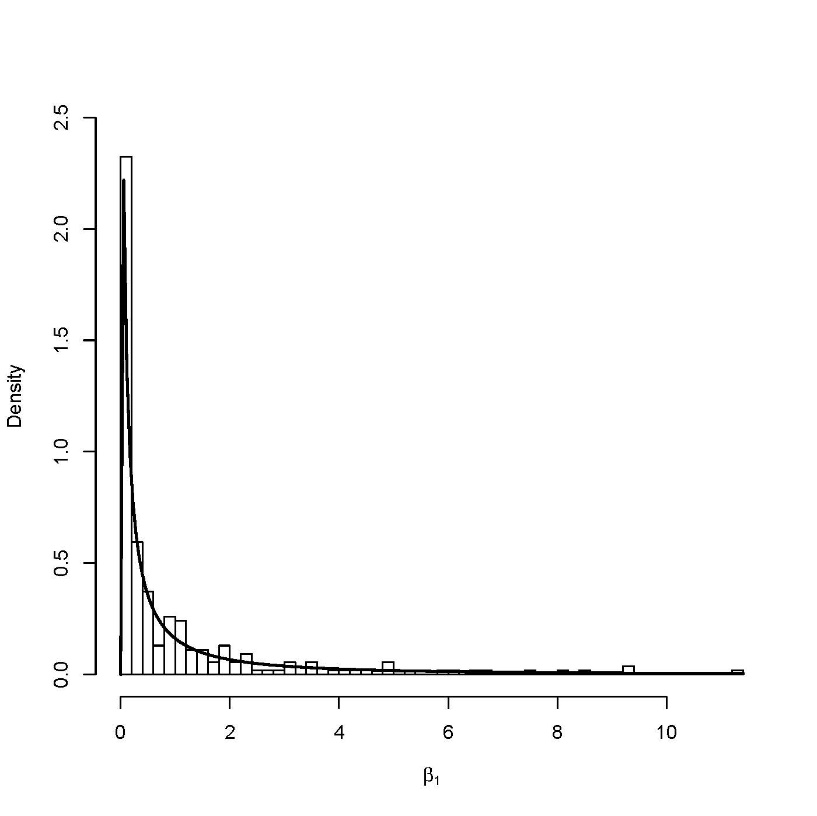
**Fig. S.4:** Empirical distribution (histogram) relating to background response parameter  for the Quantal-linear dose-response model (1) and corresponding recommended prior (dark curve),  ~ Beta(0.313, 2.545), from Table 1.



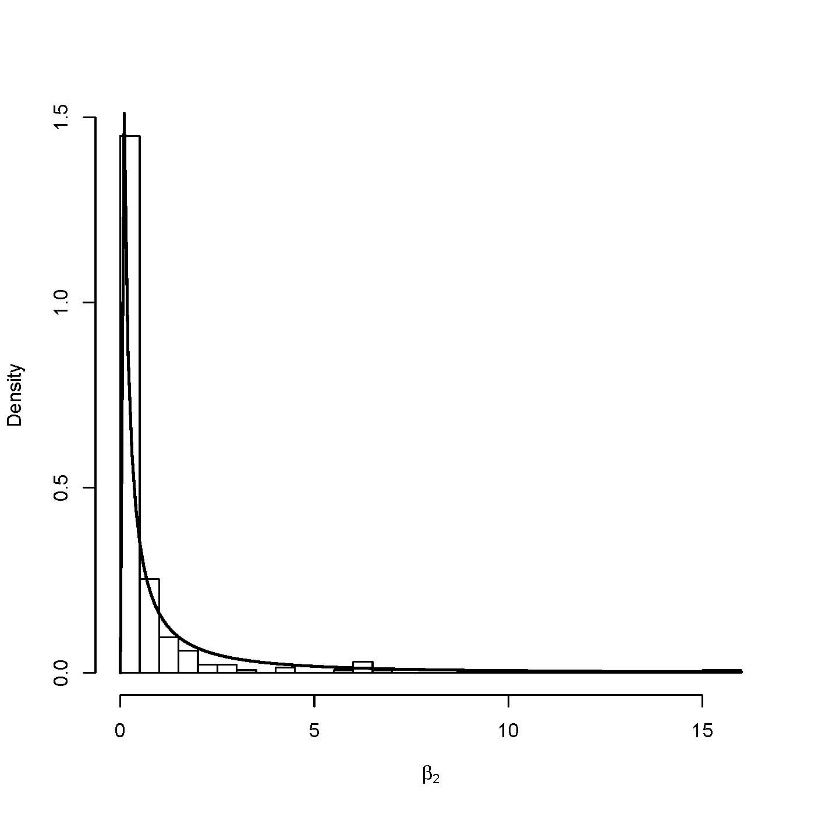
**Fig. S.5:** Empirical distribution (histogram) relating to 1 parameter for the Quantal-linear dose-response model (1) and corresponding recommended prior (dark curve), 1 ~ LN(–0.470, 1.688), from Table 1.



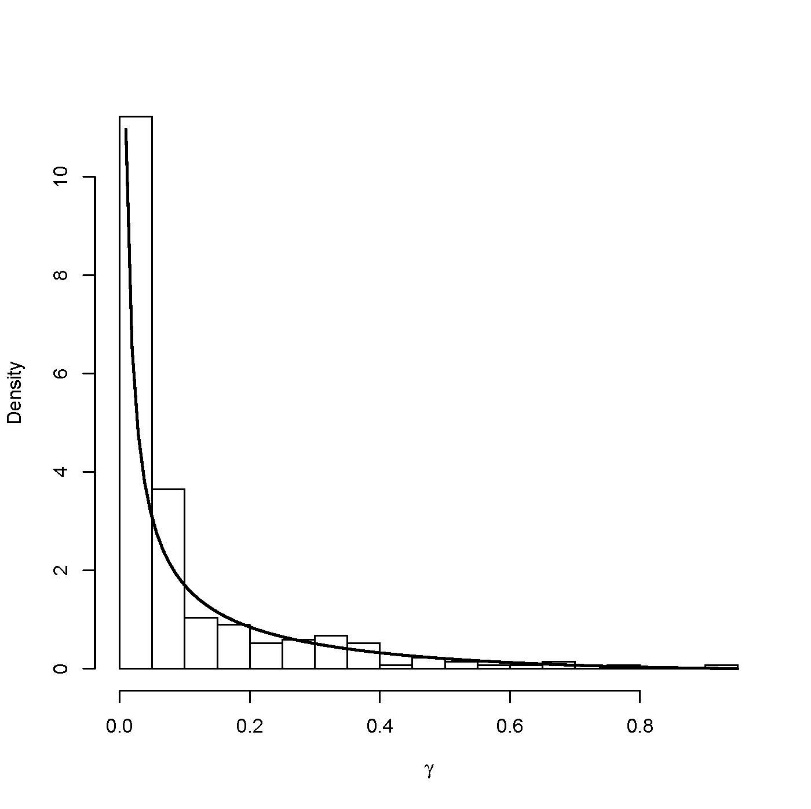
**Fig. S.6:** Empirical distribution (histogram) relating to background response parameter  for the Multi-stage dose-response model (2) and corresponding recommended prior (dark curve),  ~ Beta(0.326, 2.582), from Table 1.



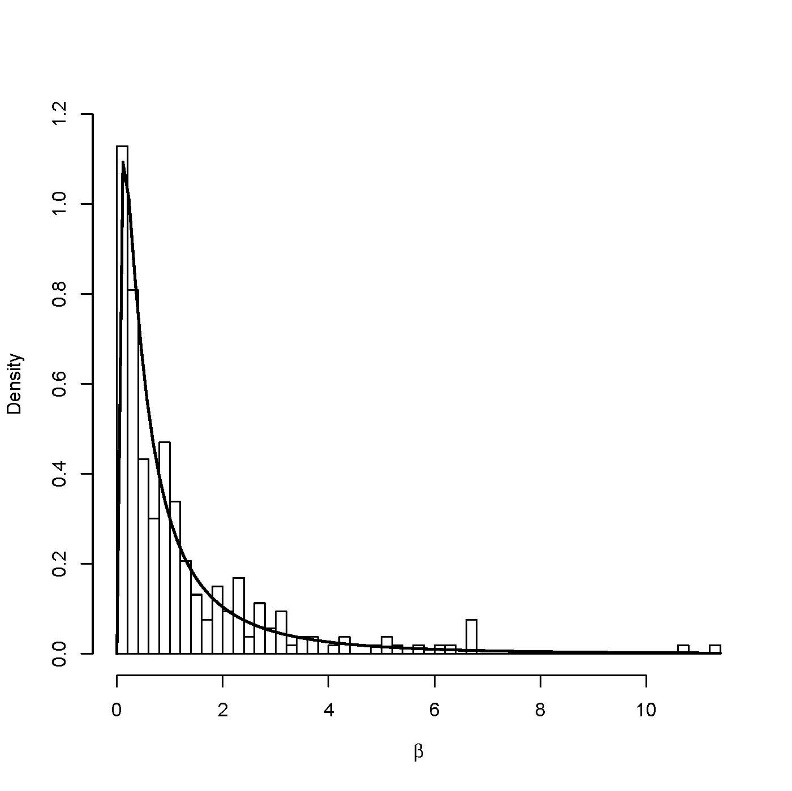
**Fig. S.7:** Empirical distribution (histogram) relating to 1 parameter for the Multi-stage dose-response model (2) and corresponding recommended prior (dark curve), 1 ~ LN(–1.006, 4.949), from Table 1.



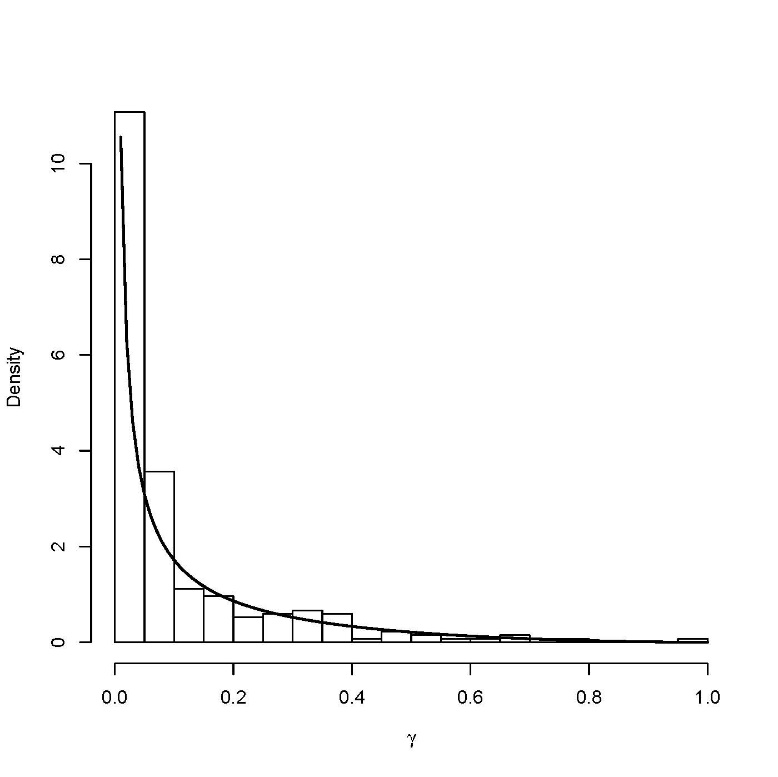
**Fig. S.8:** Empirical distribution (histogram) relating to 2 parameter for the Multi-stage dose-response model (2) and corresponding recommended prior (dark curve), 2 ~ LN(–1.016, 5.013), from Table 1.



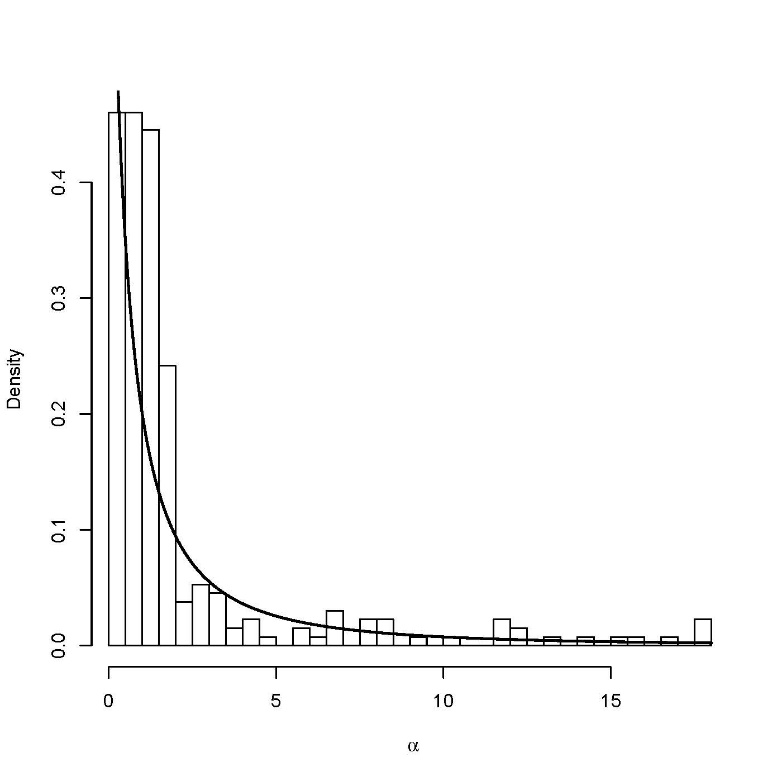
**Fig. S.9:** Empirical distribution (histogram) relating to background response parameter  for the Weibull dose-response model (3) and corresponding recommended prior (dark curve),  ~ Beta(0.271, 2.583), from Table 1.



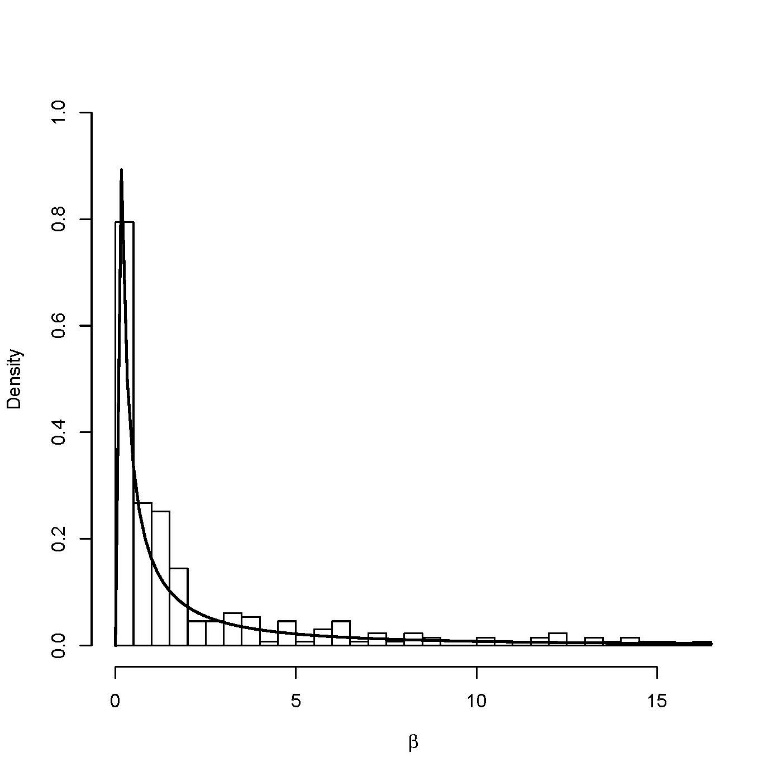
**Fig. S.10:** Empirical distribution (histogram) relating to  parameter for the Weibull dose-response model (3) and corresponding recommended prior (dark curve),  ~ LN(–0.464, 1.535), from Table 1.



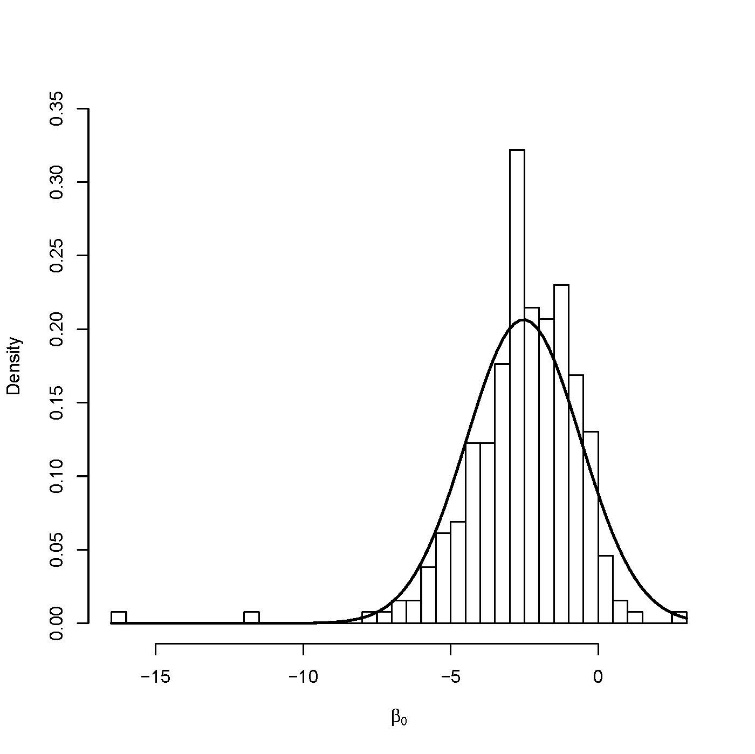
**Fig. S.11:** Empirical distribution (histogram) of background response parameter  for the gamma dose-response model (4) and corresponding recommended prior (dark curve),  ~ Beta(0.276, 2.572), from Table 1.



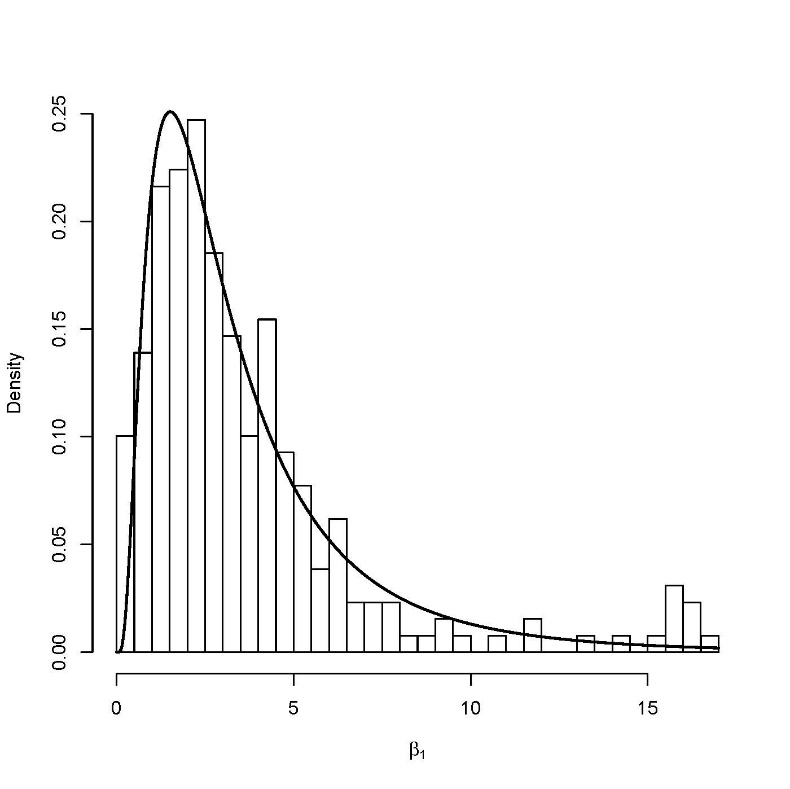
**Fig. S.12:** Empirical distribution (histogram) of shape parameter α for the gamma dose-response model (4) and corresponding recommended prior (dark curve), based on ′ = –0.2 ~ LN(–0.153, 2.637), from Table 1.



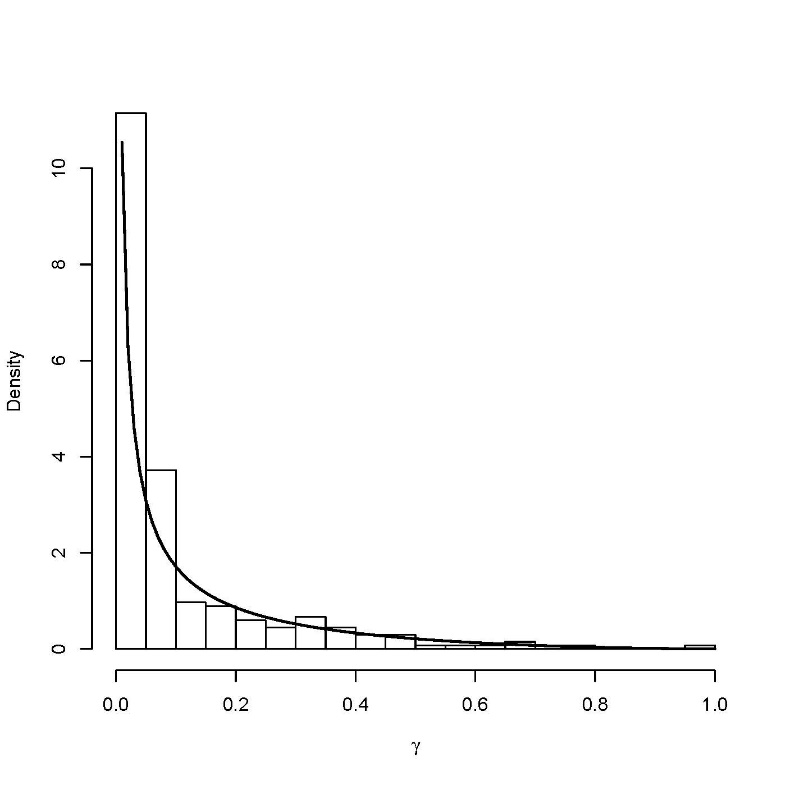
**Fig. S.13:** Empirical distribution (histogram) relating to  parameter for the gamma dose-response model (4) and corresponding recommended prior (dark curve),  ~ LN(–0.587, 5.659), from Table 1.



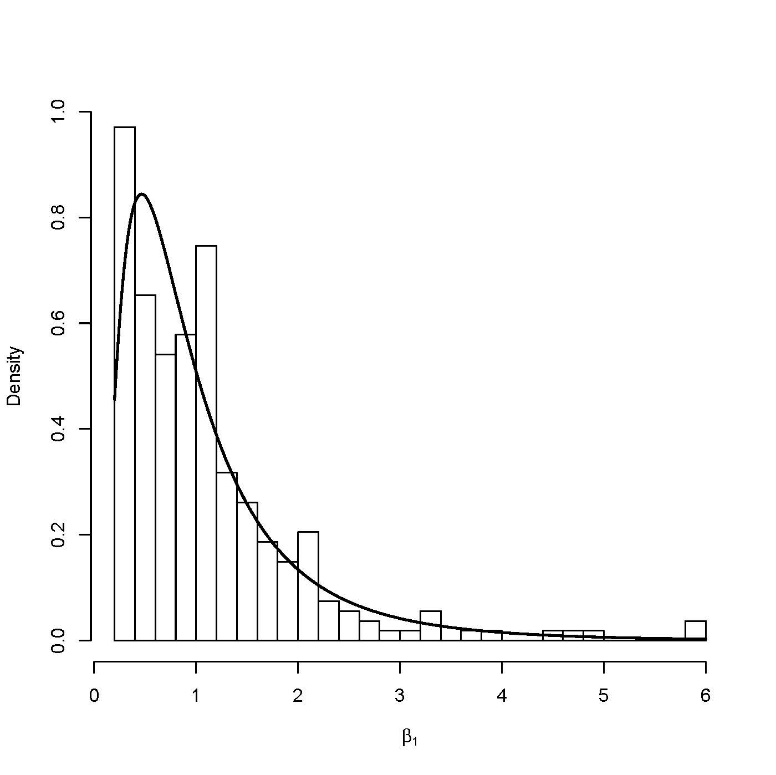
**Fig. S.14**. Empirical distribution (histogram) relating to parameter β0 for the logistic dose-response model (5) and corresponding recommended prior (dark curve), β0 ~ N(–2.526, 3.733), from Table 1.



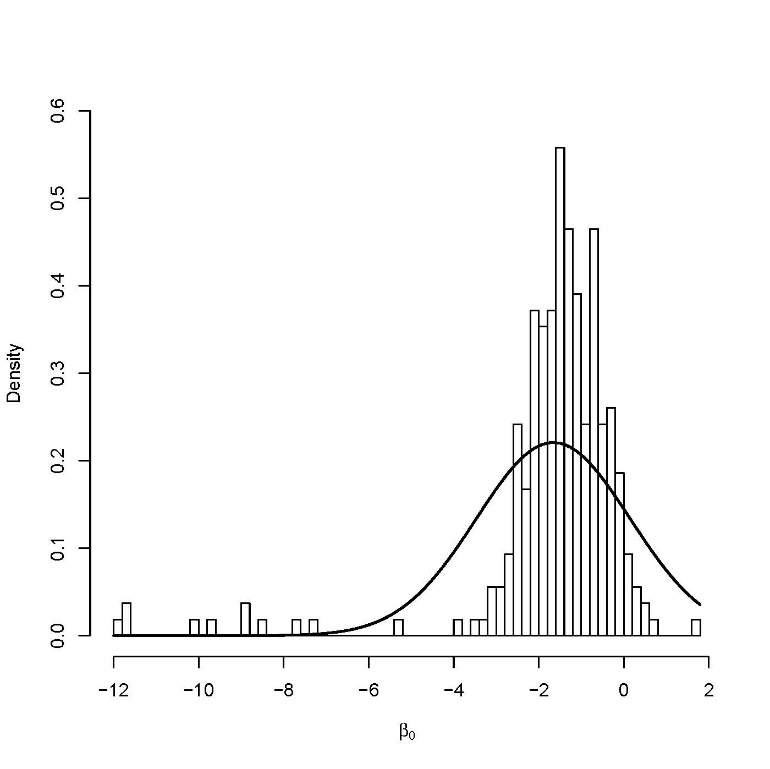
**Fig. S.15**. Empirical distribution (histogram) relating to parameter β1 for the logistic dose-response model (5) and corresponding recommended prior (dark curve), β1 ~ LN(1.018, 0.603), from Table 1.



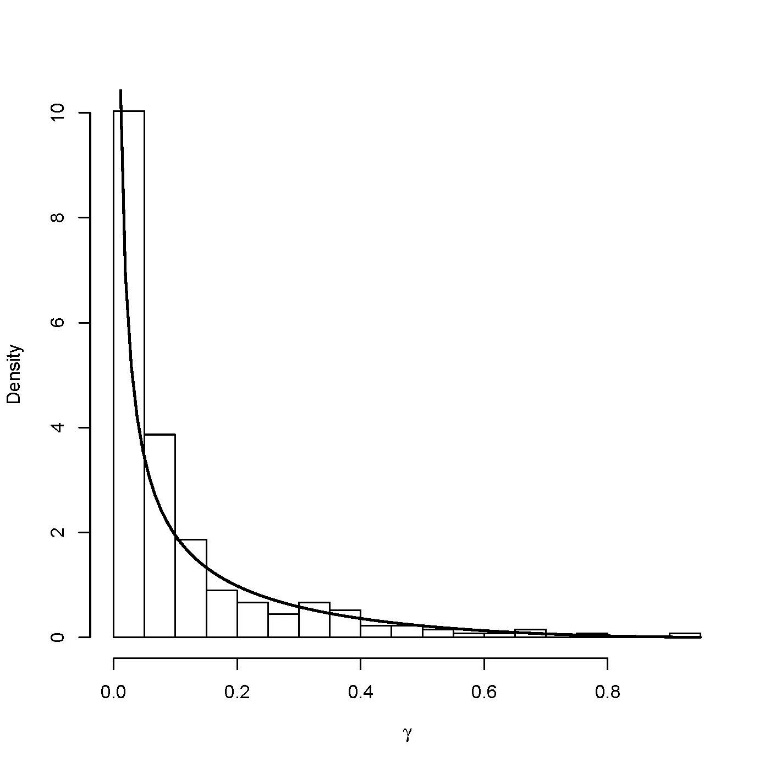
**Fig. S.16:** Empirical distribution (histogram) of background response parameter  for the log-logistic dose-response model (6) and corresponding recommended prior (dark curve),  ~ Beta(0.275, 2.571), from Table 1.



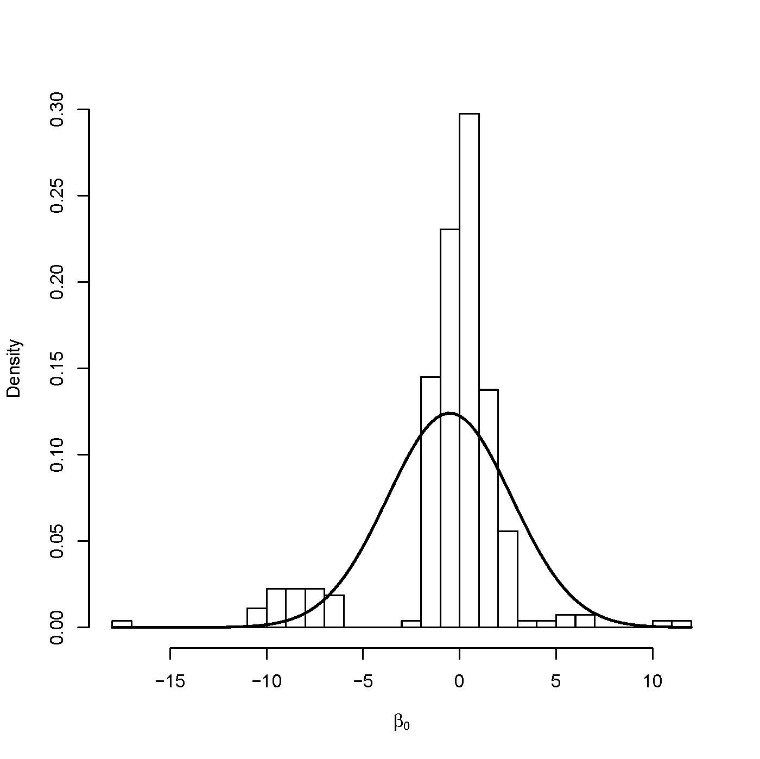
**Fig. S.17**. Empirical distribution (histogram) relating to parameter β1 for the log-logistic dose-response model (6) and corresponding recommended prior (dark curve), β1 ~ LN(0.274, 0.960), from Table 1.



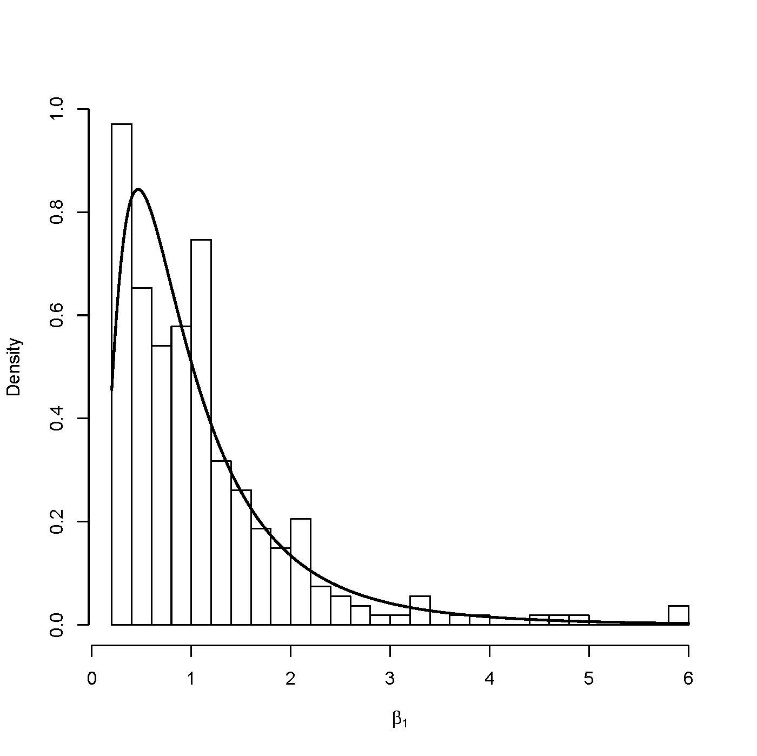
**Fig. S.18**. Empirical distribution (histogram) relating to parameter β0 for the probit dose-response model (7) and corresponding recommended prior (dark curve), β0 ~ N(–1.66, 3.264), from Table 1.



**Fig. S.19**. Empirical distribution (histogram) of background response parameter  for the log-probit dose-response model (8) and corresponding recommended prior (dark curve),  ~ Beta(0.333, 2.877), from Table 1.



**Fig. S.20**. Empirical distribution (histogram) relating to parameter β0 for the log-probit dose-response model (8) and corresponding recommended prior (dark curve), β0 ~ N(–0.514,10.337), from Table 1.



**Fig. S.21**. Empirical distribution (histogram) relating to parameter β1 for the log-probit dose-response model (8) and corresponding recommended prior (dark curve), β1 ~ LN(–0.186, 0.579), from Table 1.

S.7. R Code to for Developing Priors

The following code uses the database to develop the priors.

**library(MABMDDichotomous)**

**library(nloptr)**

**library(dplyr)**

**load(url("http://www.users.miamioh.edu/baileraj/research/database.RData"))**

**#Weibull Dose-Response**

**weibull <- function(BETAS,n,o,d){**

**gamma <- BETAS[1]**

**alpha <- BETAS[3]**

**beta <- BETAS[2]**

**p <- gamma + (1-gamma)\*(1-exp(-beta\*d^alpha))**

**p[d==0] = gamma**

**return(p)**

**}**

**#Binomial likelihood**

**weibbinLike <- function(par,n,o,d){**

**p <- weibull(par,n,o,d)**

**log.like = o\*log(p) + (n-o)\*log(1-p)**

**log.like[p==0] = log(1e-8)**

**log.like[p==1] = log(1e-8)**

**return(-sum(log.like))**

**}**

**#Weibull Dose-Response**

**gamma.D <- function(BETAS,n,o,d){**

**gamma <- BETAS[1]**

**alpha <- BETAS[3]**

**beta <- BETAS[2]**

**p <- gamma + (1-gamma)\*pgamma(beta\*d,alpha)**

**p[d==0] = gamma**

**return(p)**

**}**

**#Binomial likelihood**

**gambinLike <- function(par,n,o,d){**

**p <- gamma.D(par,n,o,d)**

**log.like = o\*log(p) + (n-o)\*log(1-p)**

**log.like[p==0] = log(1e-8)**

**log.like[p==1] = log(1e-8)**

**return(-sum(log.like))**

**}**

**BMDComputeWeibull<- function(data){**

**rval3<-nloptr(c(0.1,1,1),eval\_f=weibbinLike,**

**lb=c(0,0,0.2),**

**ub=c(1,18,18),**

**n=data$n,o=data$obs,d=data$r.dose,**

**opts = list("algorithm"="NLOPT\_LN\_BOBYQA",xtol\_rel = 1e-6,maxeval=3000))**

**return(data.frame(ID = data$ID[1],gamma=rval3$solution[1],beta=rval3$solution[2],alpha=rval3$solution[3],AIC = 2\*rval3$objective+2\*3))**

**}**

**BMDComputeGamma<- function(data){**

**rval3<-nloptr(c(0.1,1,1),eval\_f=gambinLike,**

**lb=c(0,0,0.2),**

**ub=c(1,18,18),**

**n=data$n,o=data$obs,d=data$r.dose,**

**opts = list("algorithm"="NLOPT\_LN\_BOBYQA",xtol\_rel = 1e-6,maxeval=3000))**

**return(data.frame(ID = data$ID[1],gamma=rval3$solution[1],beta=rval3$solution[2],alpha=rval3$solution[3],AIC = 2\*rval3$objective+2\*3))**

**}**

**####################################################################333**

**###**

**####################################################################**

**#log-logistic**

**log.logistic <- function(BETAS,n,o,d){**

**gamma <- BETAS[1]**

**beta0 <- BETAS[2]**

**beta1 <- BETAS[3]**

**p <- gamma + (1-gamma)\*(1/(1+exp(-beta0-beta1\*log(d))))**

**p[d==0] = gamma**

**return(p)**

**}**

**loglogitlike <- function(par,n,o,d){**

**p <- log.logistic(par,n,o,d)**

**log.like = o\*log(p) + (n-o)\*log(1-p)**

**log.like[p==0] = log(1e-8)**

**log.like[p==1] = log(1e-8)**

**return(-sum(log.like))**

**}**

**BMDComputeLlogistic <- function(data){**

**rval3<-nloptr(c(0.1,1,1),eval\_f=loglogitlike,**

**lb=c(0,-18,0.2),**

**ub=c(1,18,18),**

**n=data$n,o=data$obs,d=data$r.dose,**

**opts = list("algorithm"="NLOPT\_LN\_BOBYQA",xtol\_rel = 1e-6,maxeval=3000))**

**return(data.frame(ID = data$ID[1],gamma=rval3$solution[1],beta0=rval3$solution[2],beta1=rval3$solution[3],AIC = 2\*rval3$objective+2\*3))**

**}**

**##########################################################**

**##########################################################**

**#log-probit**

**log.probit <- function(BETAS,n,o,d){**

**gamma <- BETAS[1]**

**beta0 <- BETAS[2]**

**beta1 <- BETAS[3]**

**p <- gamma + (1-gamma)\*pnorm(beta0+beta1\*log(d))**

**p[d==0] = gamma**

**return(p)**

**}**

**logprobitlike <- function(par,n,o,d){**

**p <- log.probit(par,n,o,d)**

**log.like = o\*log(p) + (n-o)\*log(1-p)**

**log.like[p==0] = log(1e-8)**

**log.like[p==1] = log(1e-8)**

**return(-sum(log.like))**

**}**

**BMDComputeLprobit <- function(data){**

**rval3<-nloptr(c(0.1,0,1),eval\_f=logprobitlike,**

**lb=c(0,-18,0.2),**

**ub=c(1,18,18),**

**n=data$n,o=data$obs,d=data$r.dose,**

**opts = list("algorithm"="NLOPT\_LN\_BOBYQA",xtol\_rel = 1e-6,maxeval=3000))**

**return(data.frame(ID = data$ID[1],gamma=rval3$solution[1],beta0=rval3$solution[2],beta1=rval3$solution[3],AIC = 2\*rval3$objective+2\*3))**

**}**

**##########################################################################**

**#Quantal Linear and Multistage**

**##########################################################################**

**#Binomial likelihood Logistic**

**#log-probit**

**multistage <- function(BETAS,n,o,d){**

**gamma <- BETAS[1]**

**beta1 <- BETAS[2]**

**beta2 <- BETAS[3]**

**p <- gamma + (1-gamma)\*(1-exp(-beta1\*d-beta2\*d^2))**

**return(p)**

**}**

**qlinear <- function(BETAS,n,o,d){**

**gamma <- BETAS[1]**

**beta1 <- BETAS[2]**

**p <- gamma + (1-gamma)\*(1-exp(-beta1\*d))**

**return(p)**

**}**

**multistagelike <- function(par,n,o,d){**

**p <- multistage(par,n,o,d)**

**log.like = o\*log(p) + (n-o)\*log(1-p)**

**log.like[p==0] = log(1e-8)**

**log.like[p==1] = log(1e-8)**

**return(-sum(log.like))**

**}**

**qlinearlike <- function(par,n,o,d){**

**p <- qlinear(par,n,o,d)**

**log.like = o\*log(p) + (n-o)\*log(1-p)**

**log.like[p==0] = log(1e-8)**

**log.like[p==1] = log(1e-8)**

**return(-sum(log.like))**

**}**

**BMDComputeMstage <- function(data){**

**rval3<-nloptr(c(0.1,1,1),eval\_f=multistagelike,**

**lb=c(0,0,0),**

**ub=c(1,18,18),**

**n=data$n,o=data$obs,d=data$r.dose,**

**opts = list("algorithm"="NLOPT\_LN\_BOBYQA",xtol\_rel = 1e-6,maxeval=3000))**

**return(data.frame(ID = data$ID[1],gamma=rval3$solution[1],beta1=rval3$solution[2],beta1=rval3$solution[3],AIC = 2\*rval3$objective+2\*3))**

**}**

**BMDComputeQlinear<- function(data){**

**rval3<-nloptr(c(0.1,1),eval\_f=qlinearlike,**

**lb=c(0,0),**

**ub=c(1,18),**

**n=data$n,o=data$obs,d=data$r.dose,**

**opts = list("algorithm"="NLOPT\_LN\_BOBYQA",xtol\_rel = 1e-6,maxeval=3000))**

**return(data.frame(ID = data$ID[1],gamma=rval3$solution[1],beta1=rval3$solution[2],AIC = 2\*rval3$objective+2\*2))**

**}**

**##########################################################################**

**#Quantal Linear and Multistage**

**##########################################################################**

**#Binomial likelihood Logistic**

**#log-probit**

**probit <- function(BETAS,n,o,d){**

**beta0 <- BETAS[1]**

**beta1 <- BETAS[2]**

**p <- pnorm(beta0+beta1\*d)**

**return(p)**

**}**

**logistic <- function(BETAS,n,o,d){**

**beta0 <- BETAS[1]**

**beta1 <- BETAS[2]**

**p <- 1/(1+ exp(-beta0 -beta1\*d))**

**return(p)**

**}**

**probitlike <- function(par,n,o,d){**

**p <- probit(par,n,o,d)**

**log.like = o\*log(p) + (n-o)\*log(1-p)**

**log.like[p==0] = log(1e-8)**

**log.like[p==1] = log(1e-8)**

**return(-sum(log.like))**

**}**

**logisticlike <- function(par,n,o,d){**

**p <- logistic(par,n,o,d)**

**log.like = o\*log(p) + (n-o)\*log(1-p)**

**log.like[p==0] = log(1e-8)**

**log.like[p==1] = log(1e-8)**

**return(-sum(log.like))**

**}**

**BMDComputelogistic <- function(data){**

**rval3<-nloptr(c(0,0),eval\_f=logisticlike,**

**lb=c(-18,0),**

**ub=c(18,18),**

**n=data$n,o=data$obs,d=data$r.dose,**

**opts = list("algorithm"="NLOPT\_LN\_BOBYQA",xtol\_rel = 1e-6,maxeval=3000))**

**return(data.frame(ID = data$ID[1],beta0=rval3$solution[1],beta1=rval3$solution[2],AIC = 2\*rval3$objective+2\*2))**

**}**

**BMDComputeprobit<- function(data){**

**rval3<-nloptr(c(-1.2,0),eval\_f=probitlike,**

**lb=c(-18,0),**

**ub=c(18,18),**

**n=data$n,o=data$obs,d=data$r.dose,**

**opts = list("algorithm"="NLOPT\_LN\_BOBYQA",xtol\_rel = 1e-6,maxeval=3000))**

**return(data.frame(ID = data$ID[1],beta0=rval3$solution[1],beta1=rval3$solution[2],AIC = 2\*rval3$objective+2\*2))**

**}**

**results.probit <- final.data %>% group\_by(ID) %>%**

**select(ID,r.dose,n,obs) %>%**

**filter(length(r.dose) >= 4) %>%**

**filter(sum(n) > 100) %>%**

**do(BMDComputeprobit(.))**

**results.logistic <- final.data %>% group\_by(ID) %>%**

**select(ID,r.dose,n,obs) %>%**

**filter(length(r.dose) >= 4) %>%**

**filter(sum(n) > 100) %>%**

**do(BMDComputelogistic(.))**

**results.llogistic <- final.data %>% group\_by(ID) %>%**

**select(ID,r.dose,n,obs) %>%**

**filter(length(r.dose) >= 4) %>%**

**filter(sum(n) > 100) %>%**

**do(BMDComputeLlogistic(.))**

**results.multistage <- final.data %>% group\_by(ID) %>%**

**select(ID,r.dose,n,obs) %>%**

**filter(length(r.dose) >= 4) %>%**

**filter(sum(n) > 100) %>%**

**do(BMDComputeMstage(.))**

**results.Qlinear <- final.data %>% group\_by(ID) %>%**

**select(ID,r.dose,n,obs) %>%**

**filter(length(r.dose) >= 4) %>%**

**filter(sum(n) > 100) %>%**

**do(BMDComputeQlinear(.))**

**results.lprobit <- final.data %>% group\_by(ID) %>%**

**select(ID,r.dose,n,obs) %>%**

**filter(length(r.dose) >= 4) %>%**

**filter(sum(n) > 100) %>%**

**do(BMDComputeLprobit(.))**

**results.weibull <- final.data %>% group\_by(ID) %>%**

**select(ID,r.dose,n,obs) %>%**

**filter(length(r.dose) >= 4) %>%**

**filter(sum(n) > 100) %>%**

**do(BMDComputeWeibull(.))**

**results.gamma <- final.data %>% group\_by(ID) %>%**

**select(ID,r.dose,n,obs) %>%**

**filter(length(r.dose) >= 4) %>%**

**filter(sum(n) > 100) %>%**

**do(BMDComputeGamma(.))**

**AIC.MAT<-cbind(results.weibull$AIC,results.gamma$AIC,results.llogistic$AIC,results.logistic$AIC,**

**results.probit$AIC, results.lprobit$AIC,results.multistage$AIC,results.Qlinear$AIC)**

**mins <- apply(AIC.MAT,1,which.min)**

**model <- matrix(1:8,nrow=nrow(AIC.MAT),ncol=8,byrow=TRUE)**

**priorWeights3 <- colMeans(model==mins)**

**names(priorWeights3) <- c("Weibull","Gamma","Log-Logistic","Logistic","Probit","Log-Probit","Multistage","QLinear")**

**#########################################################################**

**##################################################3**

**#Create the emperical prior distributions**

**###################################################**

**results.llogistic**

**pdf(file="prior-histograms.pdf")**

**beta0 <- ggplot(results.llogistic, aes(x = beta0)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.4) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Log-Logistic") +**

**xlab(expression(paste(beta[0]))) +**

**stat\_function(fun = dnorm, colour = "black",**

**args = list(mean = mean(results.llogistic$beta0, na.rm = TRUE),**

**sd = sd(results.llogistic$beta0, na.rm = TRUE)),size=1.25)**

**beta0**

**mean(results.llogistic$beta0)**

**sd(results.llogistic$beta0)**

**beta0 <- ggplot(results.llogistic, aes(x = beta1)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.4) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**xlab(expression(paste(beta[1]))) +**

**ggtitle("Log-Logistic") +**

**scale\_x\_continuous(limits = c(0, 10)) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.llogistic$beta1), na.rm = TRUE),**

**sd = sd(log(results.llogistic$beta1), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.llogistic$beta1),na.rm = T)**

**sd(log(results.llogistic$beta1),na.rm = T)**

**######################################################**

**# Log Probit**

**#############################################33**

**beta0 <- ggplot(results.lprobit, aes(x = beta0)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.4) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Log-Probit") +**

**xlab(expression(paste(beta[0]))) +**

**stat\_function(fun = dnorm, colour = "black",**

**args = list(mean = mean(results.lprobit$beta0, na.rm = TRUE),**

**sd = sd(results.lprobit$beta0, na.rm = TRUE)),size=1.25)**

**beta0**

**mean(results.lprobit$beta0)**

**sd(results.lprobit$beta0)**

**beta0 <- ggplot(results.lprobit, aes(x = beta1)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.4) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**xlab(expression(paste(beta[1]))) +**

**ggtitle("Log-Probit") +**

**scale\_x\_continuous(limits = c(0, 10)) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.lprobit$beta1), na.rm = TRUE),**

**sd = sd(log(results.lprobit$beta1), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.lprobit$beta1),na.rm = T)**

**sd(log(results.lprobit$beta1),na.rm = T)**

**#################################################################################33**

**##################################################3**

**#Create the emperical prior distributions**

**###################################################**

**beta0 <- ggplot(results.logistic[(results.logistic$beta0 > -10)\*(results.logistic$beta0 < 10)==1,], aes(x = beta0)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.4) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Logistic") +**

**xlab(expression(paste(beta[0]))) +**

**stat\_function(fun = dnorm, colour = "black",**

**args = list(mean = mean(results.logistic$beta0[(results.logistic$beta0 > -10)\*(results.logistic$beta0 < 10)==1], na.rm = TRUE),**

**sd = sd(results.logistic$beta0[(results.logistic$beta0 > -10)\*(results.logistic$beta0 < 10)==1], na.rm = TRUE)),size=1.25)**

**beta0**

**mean(results.logistic$beta0[(results.logistic$beta0 > -10)\*(results.logistic$beta0 < 10)==1])**

**sd(results.logistic$beta0[(results.logistic$beta0 > -10)\*(results.logistic$beta0 < 10)==1])**

**beta0 <- ggplot(results.logistic, aes(x = beta1)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.4) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**xlab(expression(paste(beta[1]))) +**

**ggtitle("Logistic") +**

**scale\_x\_continuous(limits = c(0, 10)) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.logistic$beta1+0.1), na.rm = TRUE),**

**sd = sd(log(results.logistic$beta1+0.1), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.logistic$beta1+0.1),na.rm = T)**

**sd(log(results.logistic$beta1+0.1),na.rm = T)**

**######################################################**

**# Log Probit**

**#############################################33**

**beta0 <- ggplot(results.probit[(results.probit$beta0 > -5)\*(results.probit$beta0 < 5)==1,], aes(x = beta0)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.4) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Probit") +**

**xlab(expression(paste(beta[0]))) +**

**stat\_function(fun = dnorm, colour = "black",**

**args = list(mean = mean(results.probit$beta0[(results.probit$beta0 > -5)\*(results.probit$beta0 < 5)==1], na.rm = TRUE),**

**sd = sd(results.probit$beta0[(results.probit$beta0 > -5)\*(results.probit$beta0 < 5)==1], na.rm = TRUE)),size=1.25)**

**beta0**

**mean(results.probit$beta0[(results.probit$beta0 > -5)\*(results.probit$beta0 < 5)==1])**

**sd(results.probit$beta0[(results.probit$beta0 > -5)\*(results.probit$beta0 < 5)==1])**

**beta0 <- ggplot(results.probit[results.probit$beta1 >0,], aes(x = beta1)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.4) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**xlab(expression(paste(beta[1]))) +**

**ggtitle("Probit") +**

**scale\_x\_continuous(limits = c(0, 10)) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.probit$beta1[results.probit$beta1 >0]), na.rm = TRUE),**

**sd = sd(log(results.probit$beta1[results.probit$beta1 >0]), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.probit$beta1),na.rm = T)**

**sd(log(results.probit$beta1),na.rm = T)**

**#################################################################################**

**######################################################**

**# Log Quantal Linear**

**######################################################**

**beta0 <- ggplot(results.Qlinear[results.Qlinear$beta1 > 0,], aes(x = beta1)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.25) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Quantal-Linear") +**

**xlab(expression(paste(beta[1]))) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.Qlinear$beta1[results.Qlinear$beta1 > 0]), na.rm = TRUE),**

**sd = sd(log(results.Qlinear$beta1[results.Qlinear$beta1 > 0]), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.Qlinear$beta1[results.Qlinear$beta1 > 0]),na.rm = T)**

**sd(log(results.Qlinear$beta1[results.Qlinear$beta1 > 0]),na.rm = T)**

**#################################################################################**

**######################################################**

**# Multistage**

**######################################################**

**beta0 <- ggplot(results.multistage[results.multistage$beta1 > 0,], aes(x = beta1)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.25) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Multistage 2-Degree") +**

**xlab(expression(paste(beta[1]))) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.multistage$beta1[results.multistage$beta1 > 0]), na.rm = TRUE),**

**sd = sd(log(results.multistage$beta1[results.multistage$beta1 > 0]), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.multistage$beta1[results.multistage$beta1 > 0]),na.rm = T)**

**sd(log(results.multistage$beta1[results.multistage$beta1 > 0]),na.rm = T)**

**beta0 <- ggplot(results.multistage[results.multistage$beta1.1 > 0,], aes(x = beta1.1)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.25) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Multistage 2-Degree") +**

**xlab(expression(paste(beta[2]))) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.multistage$beta1.1[results.multistage$beta1.1 > 0]), na.rm = TRUE),**

**sd = sd(log(results.multistage$beta1.1[results.multistage$beta1.1 > 0]), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.multistage$beta1.1[results.multistage$beta1.1 > 0]),na.rm = T)**

**sd(log(results.multistage$beta1.1[results.multistage$beta1.1 > 0]),na.rm = T)**

**######################################################**

**# Gamma**

**######################################################**

**beta0 <- ggplot(results.gamma[results.gamma$beta > 0,], aes(x = beta)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.20) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Gamma") +**

**xlab(expression(paste(beta))) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.gamma$beta[results.gamma$beta > 0]), na.rm = TRUE),**

**sd = sd(log(results.gamma$beta[results.gamma$beta > 0]), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.gamma$beta[results.gamma$beta > 0]),na.rm = T)**

**sd(log(results.gamma$beta[results.gamma$beta > 0]),na.rm = T)**

**beta0 <- ggplot(results.gamma[results.gamma$alpha > 0,], aes(x = alpha)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.30) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Gamma") +**

**xlab(expression(paste(alpha))) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.gamma$alpha[results.gamma$alpha > 0]), na.rm = TRUE),**

**sd = sd(log(results.gamma$alpha[results.gamma$alpha > 0]), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.gamma$alpha[results.gamma$alpha > 0]),na.rm = T)**

**sd(log(results.gamma$alpha[results.gamma$alpha > 0]),na.rm = T)**

**######################################################**

**# weibull**

**######################################################**

**beta0 <- ggplot(results.weibull[results.weibull$beta > 0,], aes(x = beta)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.20) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Weibull") +**

**xlab(expression(paste(beta))) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.weibull$beta[results.weibull$beta > 0]), na.rm = TRUE),**

**sd = sd(log(results.weibull$beta[results.weibull$beta > 0]), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.weibull$beta[results.weibull$beta > 0]),na.rm = T)**

**sd(log(results.weibull$beta[results.weibull$beta > 0]),na.rm = T)**

**beta0 <- ggplot(results.weibull[results.weibull$alpha > 0,], aes(x = alpha)) +**

**geom\_histogram(aes(y = ..density..),center = 0,colour="grey20",fill="grey90",binwidth = 0.30) +**

**theme\_bw() + theme(panel.border = element\_blank(), axis.line = element\_line(colour = "black")) +**

**ggtitle("Weibull") +**

**xlab(expression(paste(alpha))) +**

**stat\_function(fun = dlnorm, colour = "black",**

**args = list(mean = mean(log(results.weibull$alpha[results.weibull$alpha > 0]), na.rm = TRUE),**

**sd = sd(log(results.weibull$alpha[results.weibull$alpha > 0]), na.rm = TRUE)),size=1.25)**

**beta0**

**mean(log(results.weibull$alpha[results.weibull$alpha > 0]),na.rm = T)**

**sd(log(results.weibull$alpha[results.weibull$alpha > 0]),na.rm = T)**

**dev.off()**

S.7. R Code for Weibull Bayesian Analysis for 3-monochloropropane-1,2-diol

The following code is the R/RSTAN code for the Bayesian analysis.

**library(MABMDDichotomous)**

**library(dplyr)**

**library(rstan)**

**N <- c(50,50,50,50)**

**D <- c(0,1.97,8.27,29.5)/29.5**

**O <- c(1,11,21,36)**

**Adata <- list(len=length(N),y=O,n=N,d=D)**

**wei\_model.p <- "**

**data {**

**int<lower=0> len; //number of dose groups**

**int<lower=0> y[len]; //observed number of cases**

**int<lower=0> n[len]; //number of subjects**

**real<lower=0> d[len]; //dose levels**

**}**

**parameters {**

**real <lower=0,upper=1> gamma;**

**real <lower = 0> beta;**

**real <lower = 0> alpha;**

**}**

**model {**

**alpha ~ lognormal(-0.243,1.61); //note: STD devations are specified here**

**beta ~ lognormal(-0.587,1.57);// variances are in the manuscript**

**gamma ~ beta(0.271, 2.583);**

**//likelihood**

**for (i in 1:len){**

**y[i] ~ binomial(n[i],gamma+(1-gamma)\*(1-exp(-beta\*(d[i])^(0.2+alpha))));**

**}**

**}**

**"**

**mcmc\_iter.wei = 25000**

**mcmc\_warmup.wei = 1000**

**fit\_wei = stan(model\_code=wei\_model.p,data=Adata,iter=mcmc\_iter.wei,warmup=mcmc\_warmup.wei,chains=1,seed=8675309)**

**list\_of\_draws <- extract(fit\_wei)**

**BMD.P <- (-log(0.9)/list\_of\_draws$beta)^(1/(list\_of\_draws$alpha+0.2))\*29.5**

**wei\_model <- "**

**data {**

**int<lower=0> len; //number of dose groups**

**int<lower=0> y[len]; //observed number of cases**

**int<lower=0> n[len]; //number of subjects**

**real<lower=0> d[len]; //dose levels**

**}**

**parameters {**

**real <lower=0,upper=1> gamma;**

**real <lower = 0> beta;**

**real <lower = 0> alpha;**

**}**

**model {**

**alpha ~ uniform(0.2,18); //note: STD devations are specified here**

**beta ~ uniform(0,18);// variances are in the manuscript**

**gamma ~ uniform(0,18);**

**//likelihood**

**for (i in 1:len){**

**y[i] ~ binomial(n[i],gamma+(1-gamma)\*(1-exp(-beta\*(d[i])^(alpha))));**

**}**

**}**

**"**

**mcmc\_iter.wei = 25000**

**mcmc\_warmup.wei = 1000**

**fit\_wei2 = stan(model\_code=wei\_model,data=Adata,iter=mcmc\_iter.wei,warmup=mcmc\_warmup.wei,chains=1,seed=8675309)**

**list\_of\_draws2 <- extract(fit\_wei2)**

**BMD <- (-log(0.9)/list\_of\_draws2$beta)^(1/(list\_of\_draws2$alpha))\*29.5**

**####################################################**

**###################################################**

**#gamma**

**###################################################**

**N <- c(50,50,50,50)**

**D <- c(0,1.97,8.27,29.5)/29.5**

**O <- c(1,11,21,36)**

**Adata <- list(len=length(N),y=O,n=N,d=D)**

**gamma\_model.p <- "**

**data {**

**int<lower=0> len; //number of dose groups**

**int<lower=0> y[len]; //observed number of cases**

**int<lower=0> n[len]; //number of subjects**

**real<lower=0> d[len]; //dose levels**

**}**

**parameters {**

**real <lower=0,upper=1> gamma;**

**real <lower = 0> beta;**

**real <lower = 0> alpha;**

**}**

**model {**

**alpha ~ lognormal(-0.243,1.60); //note: STD devations are specified here**

**beta ~ lognormal(-0.587,1.62);// variances are in the manuscript**

**gamma ~ beta(0.276, 2.572);**

**//likelihood**

**for (i in 1:len){**

**if (d[i] > 0){**

**y[i] ~ binomial(n[i],gamma+(1-gamma)\*gamma\_cdf(beta\*d[i],alpha+0.2,1));**

**}else{**

**y[i] ~ binomial(n[i],gamma);**

**}**

**}**

**}**

**"**

**mcmc\_iter.wei = 25000**

**mcmc\_warmup.wei = 1000**

**fit\_gamma = stan(model\_code=gamma\_model.p,data=Adata,iter=mcmc\_iter.wei,warmup=mcmc\_warmup.wei,chains=1,seed=8675309)**

**list\_of\_draws <- extract(fit\_gamma)**

**GAMMA.BMD.P <- (qgamma(0.1,list\_of\_draws$alpha+0.2,1)/list\_of\_draws$beta)\*29.5**

**gam\_model <- "**

**data {**

**int<lower=0> len; //number of dose groups**

**int<lower=0> y[len]; //observed number of cases**

**int<lower=0> n[len]; //number of subjects**

**real<lower=0> d[len]; //dose levels**

**}**

**parameters {**

**real <lower=0,upper=1> gamma;**

**real <lower = 0> beta;**

**real <lower = 0> alpha;**

**}**

**model {**

**alpha ~ uniform(0.2,18); //note: STD devations are specified here**

**beta ~ uniform(0,18);// variances are in the manuscript**

**gamma ~ uniform(0,18);**

**//likelihood**

**for (i in 1:len){**

**if (d[i] > 0){**

**y[i] ~ binomial(n[i],gamma+(1-gamma)\*gamma\_cdf(beta\*d[i],alpha,1));**

**}else{**

**y[i] ~ binomial(n[i],gamma);**

**}**

**}**

**}**

**"**

**mcmc\_iter.wei = 25000**

**mcmc\_warmup.wei = 1000**

**fit\_gamma2 = stan(model\_code=gam\_model,data=Adata,iter=mcmc\_iter.wei,warmup=mcmc\_warmup.wei,chains=1,seed=8675309)**

**list\_of\_draws2 <- extract(fit\_gamma2)**

**GAMMA.BMD <- (qgamma(0.1,list\_of\_draws2$alpha,1)/list\_of\_draws2$beta)\*29.5**

## References

Crump, K. S. (1984). A new method for determining allowable daily intake. *Fundamental and Applied Toxicology* **4**, 854-871. PMCID/PMID: PMID6510615.

Izadi, H., Grundy, J. E., and Bose, R. (2012). Evaluation of the benchmark dose for point of departure determination for a variety of chemical classes in applied regulatory settings. *Risk Analysis* **32**, 830-835.

Mack, G. A., and Wolfe, D. A. (1981). *K*-sample rank tests for umbrella alternatives. *Journal of the American Statistical Association* **76**, 175-181.

Nitcheva, D. K., Piegorsch, W. W., and West, R. W. (2007). On use of the multistage dose-response model for assessing laboratory animal carcinogenicity. *Regulatory Toxicology and Pharmacology* **48**, 135-147. PMCID/PMID: PMC2040324.

Piegorsch, W. W., and Bailer, A. J. (2005). *Analyzing Environmental Data*. Chichester: John Wiley & Sons.

U.K. Committee on Carcinogenicity (COC) (2014). Defining a Point of Departure and Potency Estimates in Carcinogenic Dose Response. Technical Report number COC/G 05. Public Health England, Committee on Carcinogenicity of Chemicals in Food, Consumer Products and the Environment London.

U.S. EPA (2016). Benchmark Dose Software (BMDS) User Manual Version 2.6.0.1. Technical Report. National Center for Environmental Assessment, U.S. Environmental Protection Agency, Research Triangle Park, NC.

Wickham, H., and Francois, R. (2016). *dplyr*: A grammar of data manipulation. R package version 0.5.0. http://CRAN.R-project.org/package=dplyr. http://CRAN.R-project.org/package=dplyr.