Supporting information

Robust fit of toxicokinetic-toxicodynamic models using prior knowledge contained in the design of survival toxicity tests.

Marie Laure Delignette-Muller[1,2*], Philippe Ruiz[1] and Philippe Veber[1]

[1] Université de Lyon, F-69000, Lyon; Université Lyon 1; CNRS, UMR5558, Laboratoire de Biométrie et Biologie Évolutive, F-69622, Villeurbanne, France

[2] Université de Lyon, F-69000, Lyon; VetAgro Sup Campus Vétérinaire de Lyon, F-69280 Marcy l'Etoile, France

Corresponding author :
* Tel.: +33-4-78-87-27-40, e-mail: marielaure.delignettemuller@vetagro-sup.fr

---

This document contains **18 pages, 4 tables and 6 figures**.

It is structured in five parts:

- S1) Tables with full experimental data sets.
- S2) The derivation of the loglikelihood.
- S3) Scripts for model fitting with the two approaches.
- S4) Results of the fit of the three experimental data sets: tables with parameter estimates and fitted survival curves.
- S5) scatterplot matrix of the joint posterior distribution of parameters for each experimental data set.

S1) Full experimental data sets

The three experimental data sets were reported in respective tables S1, S2 and S3 as used in the scripts described in part S3.

Table S1 The dichromate data set previously published in "Bedaux, J. J. M.; Kooijman, S. A. L. M. Statistical analysis of bioassays, based on hazard modeling. *Environ. Ecol. Stat.* 1994, *1*, 303–314".

conc	t	N	tprec	Nprec
0	2	50	0	50
0	5	50	2	50
0	7	50	5	50
0	9	49	7	50
0	12	49	9	49
0	14	49	12	49
0	16	49	14	49
0	19	49	16	49
0	21	49	19	49
0.1	2	50	0	50
0.1	5	50	2	50
0.1	7	50	5	50
0.1	9	50	7	50
0.1	12	50	9	50
0.1	14	50	12	50
0.1	16	50	14	50
0.1	19	50	16	50
0.1	21	50	19	50
0.18	2	50	0	50
0.18	5	50	2	50
0.18	7	50	5	50
0.18	9	50	7	50
0.18	12	50	9	50
0.18	14	50	12	50
0.18	16	50	14	50
0.18	19	50	16	50
0.18	21	50	19	50
0.32	2	50	0	50
0.32	5	50	2	50
0.32	7	50	5	50
0.32	9	50	7	50
0.32	12	50	9	50
0.32	14	48	12	50
0.32	16	47	14	48
0.32	19	47	16	47
0.32	21	45	19	47
0.56	2	50	0	50
0.56	5	48	2	50
0.56	7	48	5	48
0.56	9	48	7	48

0.56	12	40	9	48
0.56	14	32	12	40
0.56	16	30	14	32
0.56	19	23	16	30
0.56	21	16	19	23
1	2	48	0	50
1	5	36	2	48
1	7	35	5	36
1	9	31	7	35
1	12	15	9	31
1	14	9	12	15
1	16	3	14	9
1	19	0	16	3
1	21	0	19	0

Table S2 The propiconazole data previously published in "Nyman, A.-M.; Schirmer, K.; Ashauer, R. Toxicokinetic-toxicodynamic modelling of survival of *Gammarus pulex* in multiple pulse exposures to propiconazole : model assumptions , calibration data requirements and predictive power. *Ecotoxicology* 2012, *21*, 1828–1840".

conc	t	N	tprec	Nprec
0	1	9	0	10
0	2	9	1	9
0	3	9	2	9
0	4	9	3	9
8	1	20	0	20
8	2	20	1	20
8	3	20	2	20
8	4	19	3	20
12	1	19	0	20
12	2	19	1	19
12	3	19	2	19
12	4	17	3	19
14	1	19	0	20
14	2	19	1	19
14	3	18	2	19
14	4	16	3	18
18	1	21	0	21
18	2	20	1	21
18	3	16	2	20
18	4	16	3	16
24	1	17	0	20
24	2	6	1	17
24	3	2	2	6
24	4	1	3	2
29	1	11	0	20
29	2	4	1	11
29	3	0	2	4

29	4	0	3	0
36	1	11	0	20
36	2	1	1	11
36	3	0	2	1
36	4	0	3	0

Table S3 The zinc data set previously published in "Billoir, E.; Delignette-Muller, M. L.; Péry, A. R. R.; Charles, S. A Bayesian approach to analyzing ecotoxicological data. *Environ. Sci. Technol.* 2008, *42*, 8978–8984".

conc	t	N	tprec	Nprec
0	8	58	0	60
0	9	58	8	58
0	10	58	9	58
0	11	57	10	58
0	12	57	11	57
0	13	57	12	57
0	14	57	13	57
0	15	57	14	57
0	16	57	15	57
0	17	57	16	57
0	18	57	17	57
0	19	57	18	57
0	20	56	19	57
0	21	56	20	56
0.074	8	57	0	60
0.074	9	57	8	57
0.074	10	56	9	57
0.074	11	56	10	56
0.074	12	56	11	56
0.074	13	56	12	56
0.074	14	56	13	56
0.074	15	56	14	56
0.074	16	56	15	56
0.074	17	56	16	56
0.074	18	56	17	56
0.074	19	56	18	56
0.074	20	56	19	56
0.074	21	56	20	56
0.22	8	57	0	60
0.22	9	57	8	57
0.22	10	57	9	57
0.22	11	57	10	57
0.22	12	57	11	57
0.22	13	57	12	57
0.22	14	57	13	57
0.22	15	57	14	57

0.22	16	57	15	57
0.22	17	56	16	57
0.22	18	54	17	56
0.22	19	53	18	54
0.22	20	52	19	53
0.22	21	50	20	52
0.66	8	47	0	60
0.66	9	42	8	47
0.66	10	40	9	42
0.66	11	39	10	40
0.66	12	35	11	39
0.66	13	33	12	35
0.66	14	29	13	33
0.66	15	26	14	29
0.66	16	19	15	26
0.66	17	14	16	19
0.66	18	13	17	14
0.66	19	10	18	13
0.66	20	10	19	10
0.66	21	9	20	10

S2) Derivation of the likelihood expression

In that part we detailed how the likelihood defined in eq 14 of the paper was derived from the model defined by eq 6. Naming N_{ij} the number or organisms alive at time t_i and exposure concentration c_j and N_i^P the number of organisms alive at time t_{i-1} and that have not disappeared between t_{i-1} and t_i, eq 6 can be reformulated as:

$$N_{ij} \sim B\left(N_{ij}^P, \frac{S\left(t_i, \mathrm{c}_j\right)}{S\left(t_{i-1}, \mathrm{c}_j\right)} \right) \tag{S1}$$

The conditional probability density function of N_{ij} can thus be written as:

$$f\left(N_{ij} \middle| N_{ij}^p, \theta\right) = \frac{N_{ij}^P!}{N_{ij}!\left(N_{ij}^P - N_{ij}\right)!} \times \left(\frac{S\left(t_i, c_j\right)}{S\left(t_{i-1}, c_j\right)} \right)^{N_{ij}} \times \left(1 - \frac{S\left(t_i, c_j\right)}{S\left(t_{i-1}, c_j\right)} \right)^{N_{ij}^P - N_{ij}}$$

$$= \frac{N_{ij}^P!}{N_{ij}!\left(N_{ij}^P - N_{ij}\right)!} \times \left(\frac{S\left(t_i, c_j\right)}{S\left(t_{i-1}, c_j\right)} \right)^{N_{ij}} \times \left(\frac{S\left(t_{i-1}, c_j\right) - S\left(t_i, c_j\right)}{S\left(t_{i-1}, c_j\right)} \right)^{N_{ij}^P - N_{ij}}$$

$$= \frac{N_{ij}^P!}{N_{ij}!\left(N_{ij}^P - N_{ij}\right)!} \times \frac{S\left(t_i, c_j\right)^{N_{ij}}}{S\left(t_{i-1}, c_j\right)^{N_{ij}^P}} \times \left(S\left(t_{i-1}, c_j\right) - S\left(t_i, c_j\right) \right)^{N_{ij}^P - N_{ij}} \tag{S2}$$

The likelihood of a set of parameters θ according to a global data set Y (consisting of all N_{ij} observed values) is:

$$L\left(\theta|Y\right)=\prod_{ij} f\left(N_{ij}\left|N_{ij}^{P},\theta\right.\right) \tag{S3}$$

The loglikelihood can thus be written as:

$$\begin{aligned}&\ln\left(L\left(\theta|Y\right)\right)=const+\\&\sum_{ij}\left(N_{ij}^{P}-N_{ij}\right)\ln\left(S\left(t_{i-1},c_{j}\right)-S\left(t_{i},c_{j}\right)\right)+N_{ij}\ln\left(S\left(t_{i},c_{j}\right)\right)-N_{ij}^{P}\ln\left(S\left(t_{i-1},c_{j}\right)\right)\end{aligned} \tag{S4}$$

which corresponds to eq 14.

S3) Scripts for model fitting with the two approaches

In that part we just copied the stand-alone script that is provided in the Supporting information as a text file directly usable as an R script. You just have to read the preliminary comments of the script to have a description of each part of the script and to know which tools to install before running it.

```
################################################################################################
# This script contains five parts
#
# 1/ a script for defining an example of data set (the dichromate data set) directly
# in the R script. If you want to use it with another data set
# just import it classically using function read.table() or read.csv() for example.
#
# 2/ a script for defining priors from the experimental design of a data set. Here
# the dataset is read directly from the survdichromate variable, but it can easily be
# changed to read data from a file.
#
# 3/ a script for maximum likelihood estimation (MLE)
# using previously defined priors
# (you should run part 1/ and 2/ before)
#
# 4/ a script for Bayesian inference using previously defined priors
# (you should run part 1/ and 2/ before)
#
# 5/ three independent simple scripts for Bayesian procedure using package morse
# (one for each data set presented in the paper)
# including the definition of priors from the experimental design
# and the loading of data sets which are available in the package.
#
# It is a stand-alone script that you can run as soon as you have installed
# - R (https://cran.r-project.org/)
# - JAGS (https://sourceforge.net/projects/mcmc-jags/files/)
# - the R packages rjags (needed for parts 4 and 5) and morse (for part 5 only)
################################################################################################

################################################################################################
# 1/Script for defining the dichromate data set (as an example).
# Usually such a data set is imported
# from a text file using function read.table() or from a csv file using functionread.csv()
# but we directly wrote it in the present R script in order
# to provide a standalone script.
################################################################################################
survdichromate <- "
conc t N tprec Nprec
0 2 50 0 50
0 5 50 2 50
0 7 50 5 50
0 9 49 7 50
0 12 49 9 49
0 14 49 12 49
```

```
0 16 49 14 49
0 19 49 16 49
0 21 49 19 49
0.1 2 50 0 50
0.1 5 50 2 50
0.1 7 50 5 50
0.1 9 50 7 50
0.1 12 50 9 50
0.1 14 50 12 50
0.1 16 50 14 50
0.1 19 50 16 50
0.1 21 50 19 50
0.18 2 50 0 50
0.18 5 50 2 50
0.18 7 50 5 50
0.18 9 50 7 50
0.18 12 50 9 50
0.18 14 50 12 50
0.18 16 50 14 50
0.18 19 50 16 50
0.18 21 50 19 50
0.32 2 50 0 50
0.32 5 50 2 50
0.32 7 50 5 50
0.32 9 50 7 50
0.32 12 50 9 50
0.32 14 48 12 50
0.32 16 47 14 48
0.32 19 47 16 47
0.32 21 45 19 47
0.56 2 50 0 50
0.56 5 48 2 50
0.56 7 48 5 48
0.56 9 48 7 48
0.56 12 40 9 48
0.56 14 32 12 40
0.56 16 30 14 32
0.56 19 23 16 30
0.56 21 16 19 23
1 2 48 0 50
1 5 36 2 48
1 7 35 5 36
1 9 31 7 35
1 12 15 9 31
1 14 9 12 15
1 16 3 14 9
1 19 0 16 3
1 21 0 19 0
"

##########################################################################
# 2/ Script for defining priors from the experimental design of a data set
# stored in the R object datasurv (run script 1/ before)
##########################################################################
datasurv <- read.table(file = textConnection(survdichromate), header = TRUE)

# Characteristics of the experimental design
tmin <- min(datasurv$t)
tmax <- max(datasurv$t)
cmin <- min(datasurv$conc[datasurv$conc > 0])
cmax <- max(datasurv$conc)
seqc <- sort(unique(datasurv$conc)) # ordered sequence of tested concentrations
nseqc <- length(seqc) # length of this sequence
deltaseqc <- seqc[3:nseqc] - seqc[2:(nseqc-1)] # differences between 2 successive non nul
concentrations
deltacmin <- min(deltaseqc) # minimum of those differences

# Characteristics of the normal prior on log10(NEC)
meanlog10nec <- (log10(cmax) + log10(cmin))/2 # mean
sdlog10nec <- (log10(cmax) - log10(cmin)) / 4 # standard deviation
taulog10nec <- 1/ (sdlog10nec)^2 # precision (for use in JAGS)

# Characteristics of the normal prior for log10(m0)
m0max <- - log(0.5) / tmin
m0min <- - log(0.999) / tmax
meanlog10m0 <- (log10(m0max) + log10(m0min))/2 # mean
sdlog10m0 <- (log10(m0max) - log10(m0min)) / 4 # standard deviation
```

```
taulog10m0 <- 1/ (sdlog10m0)^2 # precision for use in JAGS

# Characteristics of the normal prior for log10(kd)
kdmax <- - log(0.001) / tmin
kdmin <- - log(0.999) / tmax
meanlog10kd <- (log10(kdmax) + log10(kdmin))/2 # mean
sdlog10kd <- (log10(kdmax) - log10(kdmin)) / 4 # standard deviation
taulog10kd <- 1/ (sdlog10kd)^2 # precision for use in JAGS

# Characteristics of the normal prior for log10(ks)
ksmax <- - log(0.001) / (tmin * deltacmin)
ksmin <- - log(0.999) / (tmax * (cmax - cmin))
meanlog10ks <- (log10(ksmax) + log10(ksmin))/2 # mean
sdlog10ks <- (log10(ksmax) - log10(ksmin)) / 4 # standard deviation
taulog10ks <- 1/ (sdlog10ks)^2 # precision for use in JAGS


################################################################################
# 3/ script for maximum likelihood estimation (MLE)
# using previously defined priors (run scripts 1/ and 2/ before)
################################################################################
require(stats4)

# Survival function according to the TKTD model
Surv = function (conc, time , ks, kd, NEC, m0)
{
  S <- exp(-m0*time) # survie de base avec mortalite naturelle seule
  if(conc > NEC) {
    tNEC = -(1/kd)*log(1 - NEC/conc)
    if (time > tNEC) {
      # ajoute de la mortalite due au toxique
      S <- S * exp( ks/kd*conc*(exp(-kd*tNEC) -exp(-kd*time)) - ks*(conc-NEC)*(time - tNEC) )
    }
  }
  return(S)
}

# Function to compute minus the loglikelihood for a set
# of parameter values
minuslogL <- function(log10ks, log10kd, log10NEC, log10m0)
{
  ks <- 10^log10ks
  kd <- 10^log10kd
  NEC <- 10^log10NEC
  m0 <- 10^log10m0
  # function to calculate loglikelihood for data at one concentration
  logLoneC <- function(dsurvoneC)
  {
    n <- nrow(dsurvoneC)
    Cw <- dsurvoneC$conc[1]
    vSurv <- numeric(length = n)
    for (i in 1:n)
    {
      vSurv[i] <- Surv(Cw, time = dsurvoneC$t[i], ks, kd, NEC, m0)
    }
    vSurvprec <- c(1, vSurv[1:n-1])
    N <- dsurvoneC$N
    Nprec <- dsurvoneC$Nprec
    logL <- sum( (Nprec - N) * log(vSurvprec - vSurv) + N * log(vSurv) - Nprec*
log(vSurvprec))
    return(logL)
  }
  vlogL <- by(datasurv, datasurv$conc, logLoneC)  # application of the previous function
                                                  # for each concentration
  return(- sum(vlogL))
}

# randomization of 100 sets of initial values in the prior for the parameters
ninit <- 100
set.seed(1234)
vlog10ks <- rnorm(ninit, mean = meanlog10ks, sd = sdlog10ks)
vlog10kd <- rnorm(ninit, mean = meanlog10kd, sd =  sdlog10kd)
vlog10NEC <- rnorm(ninit, mean = meanlog10nec, sd = sdlog10nec)
vlog10m0 <- rnorm(ninit, mean = meanlog10m0, sd = sdlog10m0)

# dataframe to store results for each set of initial parameters
dres <- data.frame(log10ksest = rep(0,ninit), log10kdest = rep(0,ninit),
```

```
                  log10NECest = rep(0,ninit), log10m0est = rep(0,ninit),
                  logl = rep(0,ninit))

# MLE try from each set of initial parameters
# !!!!!!!!!!!!!!!!!!! takes a few minutes to run !!!!!!!!!!!!!!!!!!!!!!!!!!!!
for (i in 1:ninit)
{
  resfit <- try(do.call(mle, list(minuslogl = minuslogL, method = "Nelder-Mead",
                    start = list(log10ks = vlog10ks[i], log10kd = vlog10kd[i],
                    log10NEC = vlog10NEC[i], log10m0 = vlog10m0[i]))), silent = TRUE)
  if(inherits(resfit, "try-error"))
  {
    dres$log10kdest[i] <- NA
    dres$log10ksest[i] <- NA
    dres$log10NECest[i] <- NA
    dres$log10m0est[i] <- NA
    dres$logl[i] <- NA
    cat("MLE try from set of initial parameters nb.", i,": no convergence","\n")
  } else
  {
    coeff <- coef(resfit)
    dres$log10kdest[i] <- coeff["log10kd"]
    dres$log10ksest[i] <- coeff["log10ks"]
    dres$log10NECest[i] <- coeff["log10NEC"]
    dres$log10m0est[i] <- coeff["log10m0"]
    dres$logl[i] <- logLik(resfit)
    cat("MLE try from set of initial parameters nb.", i,": convergence","\n")
  }
}

# Index of the results ordered by decreasing loglikelihood
orderedi <- order(dres$logl, decreasing = TRUE)

# Estimation with the retained set of initial parameters
fitMLE <- mle(minuslogL, method = "Nelder-Mead",
              start = list(log10ks = vlog10ks[orderedi[1]], log10kd = vlog10kd[orderedi[1]],
                           log10NEC = vlog10NEC[orderedi[1]], log10m0 =
vlog10m0[orderedi[1]]))
# Point estimates
(coeff <- coef(fitMLE))

# Wald confidence intervals
varcov <- fitMLE@vcov
SD <- sqrt(diag(varcov))
(WaldCI_lower <- coeff - 1.96*SD)
(WaldCI_upper <- coeff + 1.96*SD)


#############################################################################################
# 4/ Script for Bayesian inference using previously defined priors
# (run scripts 1/ and 2/ before)
#############################################################################################
# Definition in one list of all data needed in function jags.model (see below)
data4JAGS <- list( c = datasurv$conc, N = datasurv$N,
                t = datasurv$t, tprec = datasurv$tprec,
                Nprec = datasurv$Nprec,
                meanlog10nec = meanlog10nec, taulog10nec = taulog10nec,
                meanlog10ks = meanlog10ks, taulog10ks = taulog10ks,
                meanlog10kd = meanlog10kd, taulog10kd = taulog10kd,
                meanlog10m0 = meanlog10m0, taulog10m0 = taulog10m0,
                ndat = length(datasurv$conc),
                bigtime = max(datasurv$t) + 10 ) # arbitrary time above every observed time
                                                 # needed for implementation in JAGS (see
below)

# Declaration of the model in the JAGS language
model_TKTD <- "model {
# Following notations are used in this model :
# c for the concentration
# t for the measurement date
# tprec for the date of previous measurement
# N for the observed data (number alive at time t)
# Nprec for the number of alive organisms at previous measurement tprec
# ndat for the total number of measurements

  ######### Definition of priors
  lNEC ~ dnorm(meanlog10nec , taulog10nec)
```

```
  lks ~ dnorm(meanlog10ks , taulog10ks)
  lkd ~ dnorm(meanlog10kd , taulog10kd)
  lm0 ~ dnorm(meanlog10m0 , taulog10m0)

  ##### parameter transformation
  ks <- 10**lks
  NEC <- 10**lNEC
  kd <- 10**lkd
  m0 <- 10**lm0

  ########## Definition of the links of the model
  for (i in 1:ndat)
  {
    # definition of tNEC a little bit tricky to accomodate when c <= NEC or c = 0
    ###############################################################################
    tNEC[i] <- ifelse(c[i] > NEC, -1/kd * log( 1- R[i]), bigtime)
    # with bigtime greater than all the observed times

    R[i] <- ifelse(c[i] > NEC, NEC/ccor[i], 0.1)
    # 0.1 is an arbitrary small value ( < 1) necessary to enable the definition
    # log(1 - R) in any case, but this constant has no impact on the final calculation of N
(see below).

    ccor[i] <- ifelse(c[i] > 0, c[i], 10)
    # 10 is an arbitrary positive constant necessary to enable the definition
    # of R in any case, but this constant has no impact on the final calculation of N (see
below).

    # if c = 0,  ccor = 10, R = 0.1 et tNEC = bigtime never reached in the data
    # if 0 < c <= NEC, ccor = c, R = 0.1, tNEC = bigtime never reached in the data

    # definition of the conditional probability of an organism to survive
    # until t if it is alive at tprec : psurv = S(t) / S(tprec)
    ###############################################################################
    tref[i] <- max(tprec[i], tNEC[i])
    # necessary if tprec < tNEC < t : time from which there is an effect of the toxicant

    psurv[i] <- exp(- m0*(t[i] - tprec[i]) + ifelse(t[i] > tNEC[i], - ks*( (c[i]-NEC)*(t[i] -
tref[i]) + c[i]/kd*( exp(-kd*t[i]) - exp(-kd*tref[i])) ), 0))

    # definition of the number of alive organisms at time t
    ###############################################################################
    N[i] ~ dbin(psurv[i] , Nprec[i])

  }
}"

require(rjags)
set.seed(1234)
mTKTD <- jags.model(file = textConnection(model_TKTD), data = data4JAGS, n.chains=3) # build
of the JAGS model
update(mTKTD, 5000) # burn-in phase

# calculation of the optimal number of iterations and thin from a
# short pilot run of the chains
mctrial <- coda.samples(mTKTD, c("lkd", "lNEC","lks", "lm0"), n.iter = 5000, thin = 1)
RL <- raftery.diag(mctrial)
resmatrixtot <- rbind(RL[[1]]$resmatrix,RL[[2]]$resmatrix,RL[[3]]$resmatrix )
thin <- round(max(resmatrixtot[,"I"])+0.5)
niter <- max(resmatrixtot[,"Nmin"])*thin

# Final run and storage of MCMC iterations
# !!!!!!!!!!!!!!!!!!!! takes a few minutes to run !!!!!!!!!!!!!!!!!!!!!!!!!!!!
mcTKTD <- coda.samples(mTKTD, c("lkd", "lNEC","lks", "lm0"), n.iter = niter, thin = thin)
plot(mcTKTD) # plot of the traces and marginal posterior densities
gelman.diag(mcTKTD) # calculation of the Gelman and Rubin statistics of convergence
summary(mcTKTD) # calculation of statistics to characterize the posterior distribution

###############################################################################
# 5/ Independent simple scripts for Bayesian procedure using package morse
# from each of the three data sets studied in the paper
# including the definition of priors from the experimental design of the data set
# (no need for running any script before).
#
# The package works with data sets classically coded,
# with columns named "replicate", "conc", "time" and "Nsurv",
# and computes by itself the columns needed for the fit of the TKTD model
```

```
# (columns named "conc", "t", "N", "tprec" and "Nprec" in the previous scripts)
################################################################################
require(morse)

# For the dichromate data set
data(dichromate)
dat <- survData(dichromate)
plot(dat, pool.replicate = TRUE, style = "ggplot")
fit <- survFitTKTD(dat) # !!!!!!!!!! takes a few minutes !!!!!!!!!!!
summary(fit)
plot(fit, style = "ggplot", adddata = TRUE)

# For the dichromate data set
data(propiconazole)
dat <- survData(propiconazole)
plot(dat, pool.replicate = TRUE, style = "ggplot")
fit <- survFitTKTD(dat)  # !!!!!!!!!! takes a few minutes !!!!!!!!!!!
summary(fit)
plot(fit, style = "ggplot", adddata = TRUE)

# For the zinc data set
data(zinc)
dat <- survData(zinc)
plot(dat, pool.replicate = TRUE, style = "ggplot")
fit <- survFitTKTD(dat)  # !!!!!!!!!! takes a few minutes !!!!!!!!!!!
summary(fit)
plot(fit, style = "ggplot", adddata = TRUE)
```

S4) Results of the fit of the three experimental data sets

In that part were reported the estimates of each model parameter for each experimental data set and each inference method, together with the corresponding prior credible intervals (Table S4) and the survival fitted curves (with a 95% credible band for the Bayesian inference) together with the observed survival rates (Figures S1, S2, S3).

Table S4. Prior ranges ($[\log_{10}\theta_{min}, \log_{10}\theta_{max}]$ considered as a prior 95% credible on $\log_{10}\theta$ for the definition of the normal priors on $\log_{10}\theta$) and estimates of model parameters on experimental data sets using both approaches with 95% confidence intervals (or credible intervals for Bayesian inference) in brackets. Estimated values previously reported in the source papers were also reported when available, with the correspondance between raw values and log-transformed values.

	Dichromate	**Propiconazole**	**Zinc**
Prior range			
$\log_{10} NEC$	[-0.992, -0.013]	[0.911, 1.550]	[-1.130, -0.200]
$\log_{10} k_d$	[-4.298, 0.498]	[-3.573, 0.825]	[-4.290, -0.133]
$\log_{10} k_s$	[-4.266, 1.574]	[-5.032, 0.515]	[-4.040, 0.742]
$\log_{10} m_0$	[-4.286, -0.487]	[-3.552, -0.189]	[-4.290, -1.094]
ML estimations			
$\log_{10} NEC$	-0.565 [-0.624, -0.506]	1.236 [1.214, 1.258]	-0.901 [-1.171, -0.746]
$\log_{10} k_d$	-0.669 [-0.813, -0.525]	0.338 [0.188, 0.488]	-1.164 [-1.546, -0.833]
$\log_{10} k_s$	-0.556 [-0.675, -0.437]	-0.874 [-1.051, -0.696]	-0.341 [-0.613, -0.005]
$\log_{10} m_0$	-3.511 [-4.364, -2.659]	-1.503 [-1.786, -1.220]	-2.468 [-2.744, -2.230]
Bayesian estimations			
$\log_{10} NEC$	-0.583 [-0.706, -0.530]	1.232 [1.194, 1.277]	-0.919 [-1.209, -0.629]
$\log_{10} k_d$	-0.695 [-0.989, -0.521]	0.344 [0.202, 0.544]	-1.220 [-1.679, -0.761]
$\log_{10} k_s$	-0.563 [-0.698, -0.380]	-0.902 [-1.098, -0.715]	-0.281 [-0.647, 0.086]
$\log_{10} m_0$	-3.363 [-4.262, -2.761]	-1.518 [-1.841, -1.259]	-2.454 [-2.709, -2.199]
ML estimations reported in the source papers when available			
$\log_{10} NEC$	-0.565 ($\log_{10}(0.272)$)	1.220 ($\log_{10}(16.6)$)	Not available
$\log_{10} k_d$	-0.669 ($\log_{10}(0.214)$)	0.322 ($\log_{10}(2.1)$)	Not available
$\log_{10} k_s$	-0.556 ($\log_{10}(0.278)$)	-0.873 ($\log_{10}(0.1339)$)	Not available
$\log_{10} m_0$	-3.511 ($\log_{10}(3.08\ 10^{-4})$)	Not available	Not available

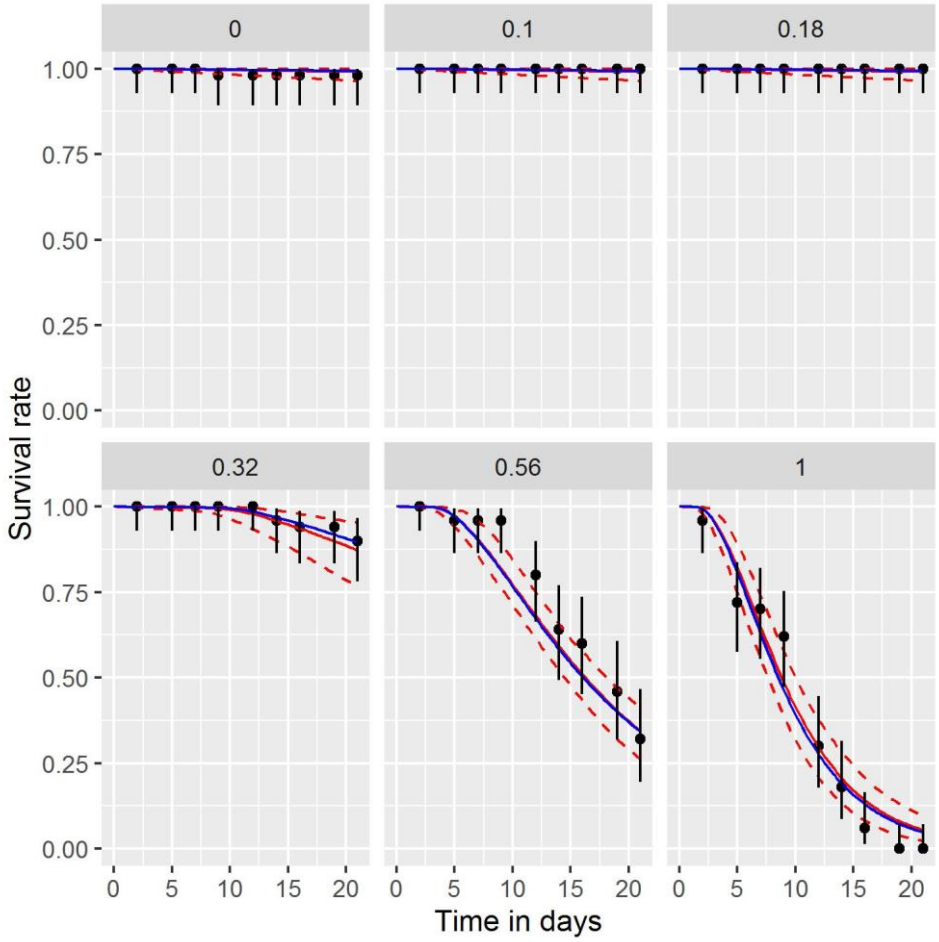

Figure S1 Fit of the stochastic death model to the dichromate data set with survival rate on y-axis, time on x-axis and one subplot per concentration of exposure. The survival rate observed at each time and exposure condition is surrounded by its 95% confidence interval computed by the Clopper and Pearson procedure using the function binom.test() of the R package stats.Theoretical curves obtained with ML estimates of parameters are plotted in blue. Using the Bayesian posterior joint distribution of parameters, a theoretical curve (red plain line) surrounded by its 95% credible interval (red dotted lines) are plotted.

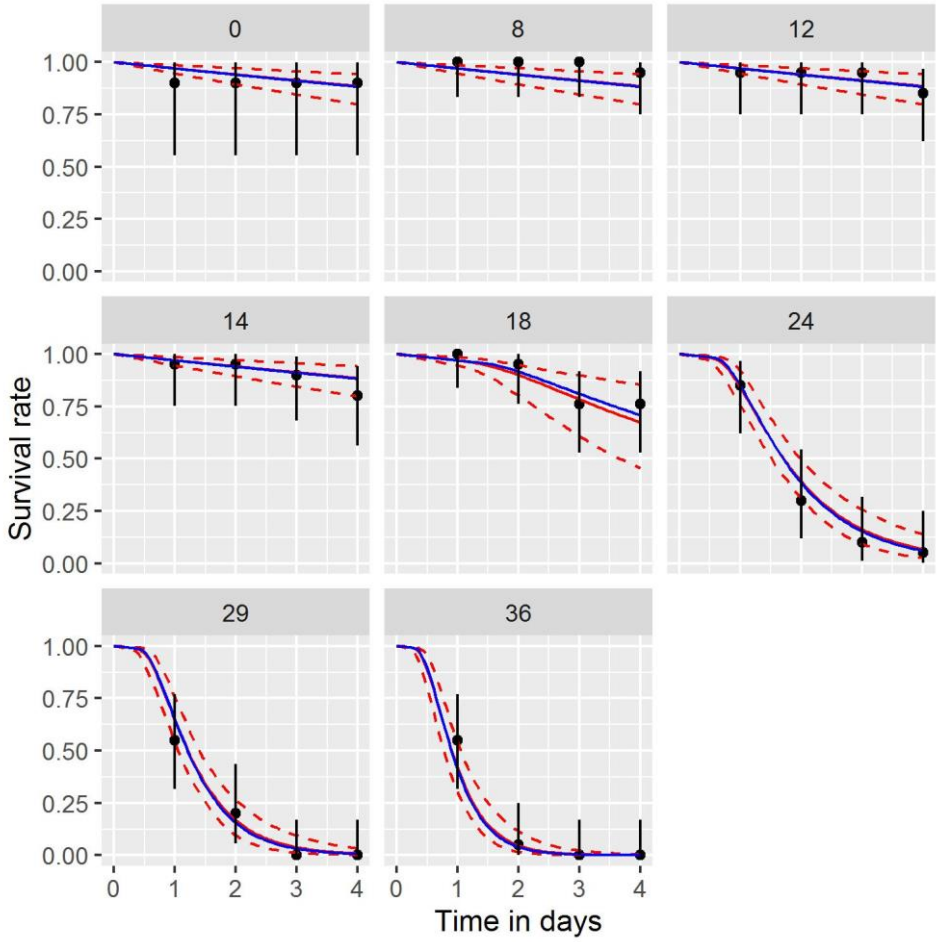

Figure S2 Fit of the stochastic death model to the propiconazole data set with survival rate on y-axis, time on x-axis and one subplot per concentration of exposure. The survival rate observed at each time and exposure condition is surrounded by its 95% confidence interval computed by the Clopper and Pearson procedure using the function binom.test() of the R package stats.Theoretical curves obtained with ML estimates of parameters are plotted in blue. Using the Bayesian posterior joint distribution of parameters, a theoretical curve (red plain line) surrounded by its 95% credible interval (red dotted lines) are plotted.

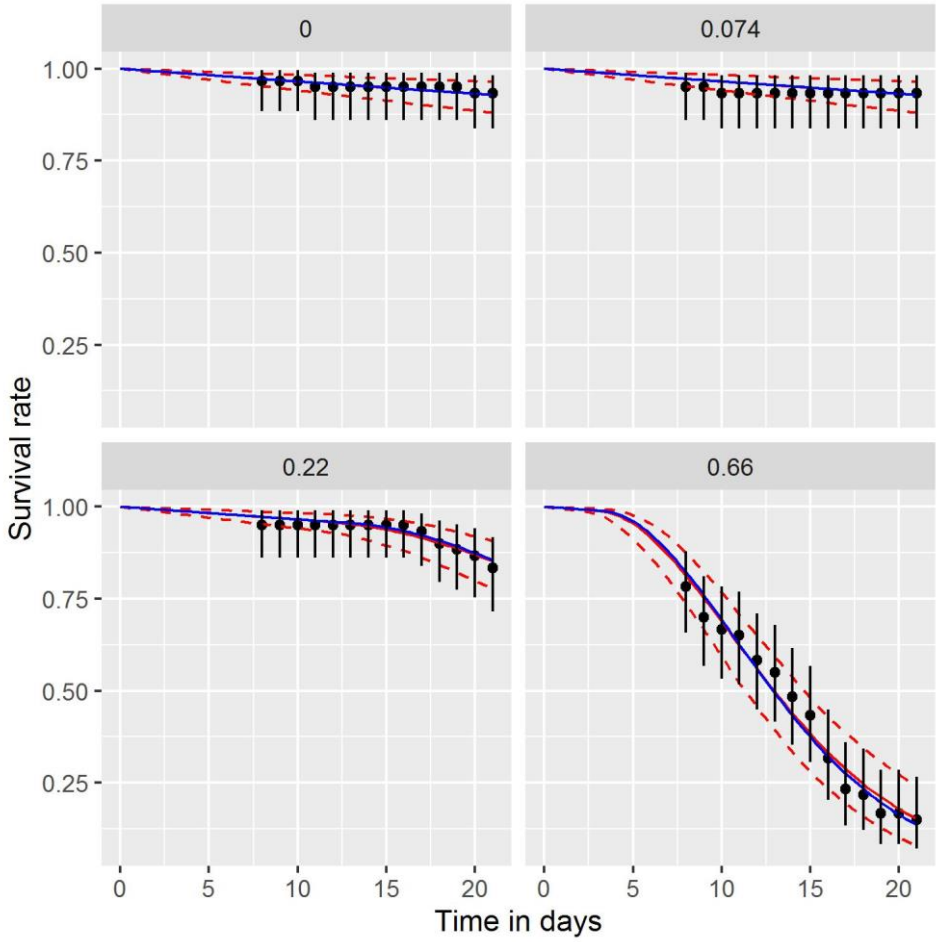

Figure S3 Fit of the stochastic death model to the zinc data set with survival rate on y-axis, time on x-axis and one subplot per concentration of exposure. The survival rate observed at each time and exposure condition is surrounded by its 95% confidence interval computed by the Clopper and Pearson procedure using the function binom.test() of the R package stats.Theoretical curves obtained with ML estimates of parameters are plotted in blue. Using the Bayesian posterior joint distribution of parameters, a theoretical curve (red plain line) surrounded by its 95% credible interval (red dotted lines) are plotted.

S5) Joint posterior distributions

Here we reported, for each experimental data set, a representation of the joint posterior distribution estimated by Bayesian inference (Figures S4, S5 and S6).

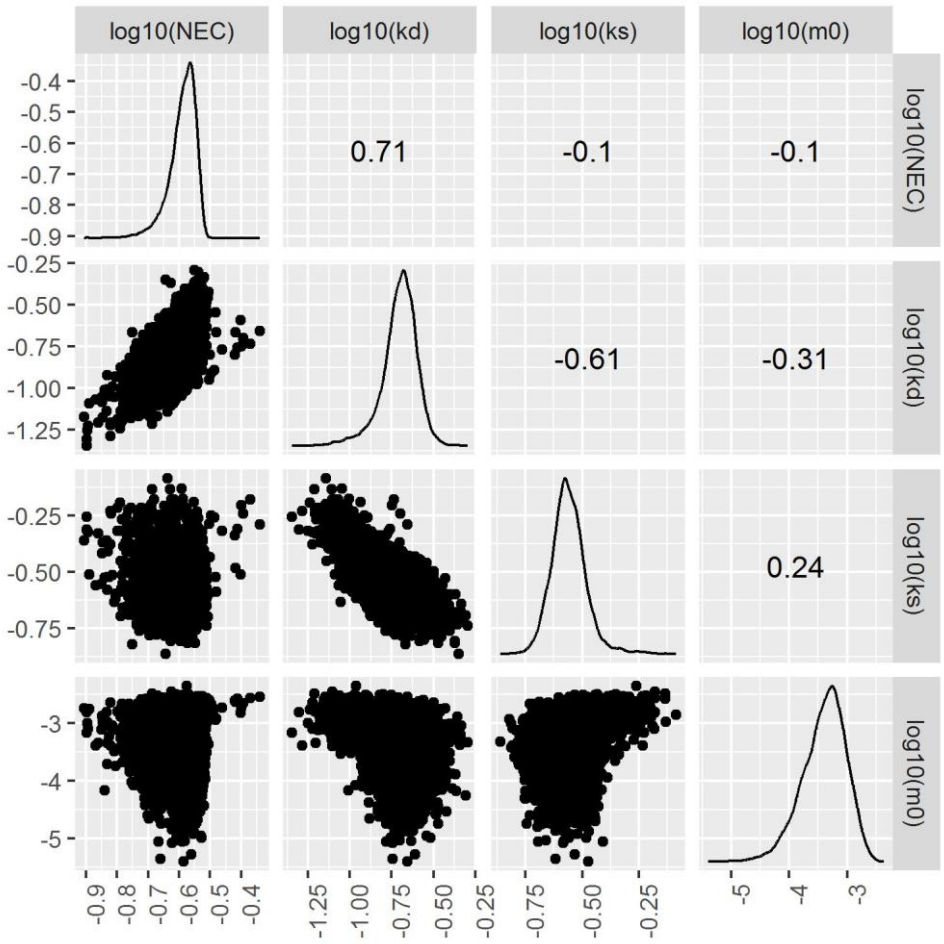

Figure S4 Scatterplot matrix of the joint posterior distribution of parameters estimated using the dichromate data set, with density plots on the diagonal and correlation printed in the upper triangle.

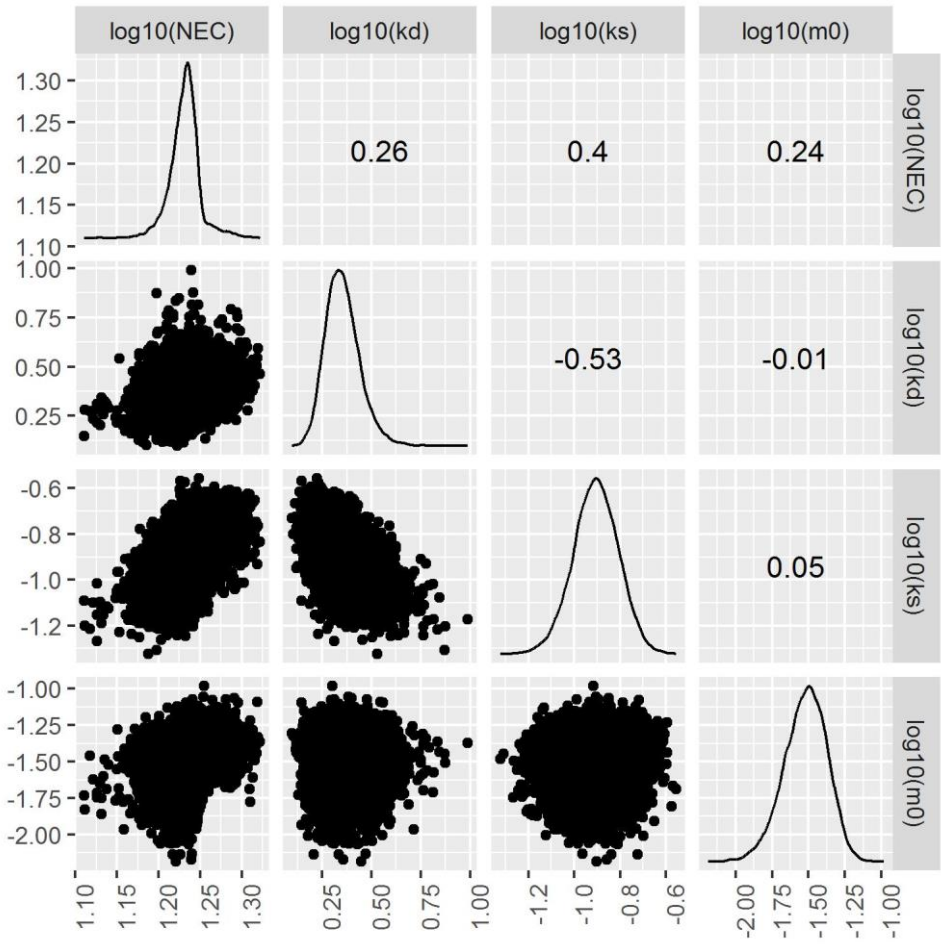

Figure S5 Scatterplot matrix of the joint posterior distribution of parameters estimated using the propiconazole data set, with density plots on the diagonal and correlation printed in the upper triangle.

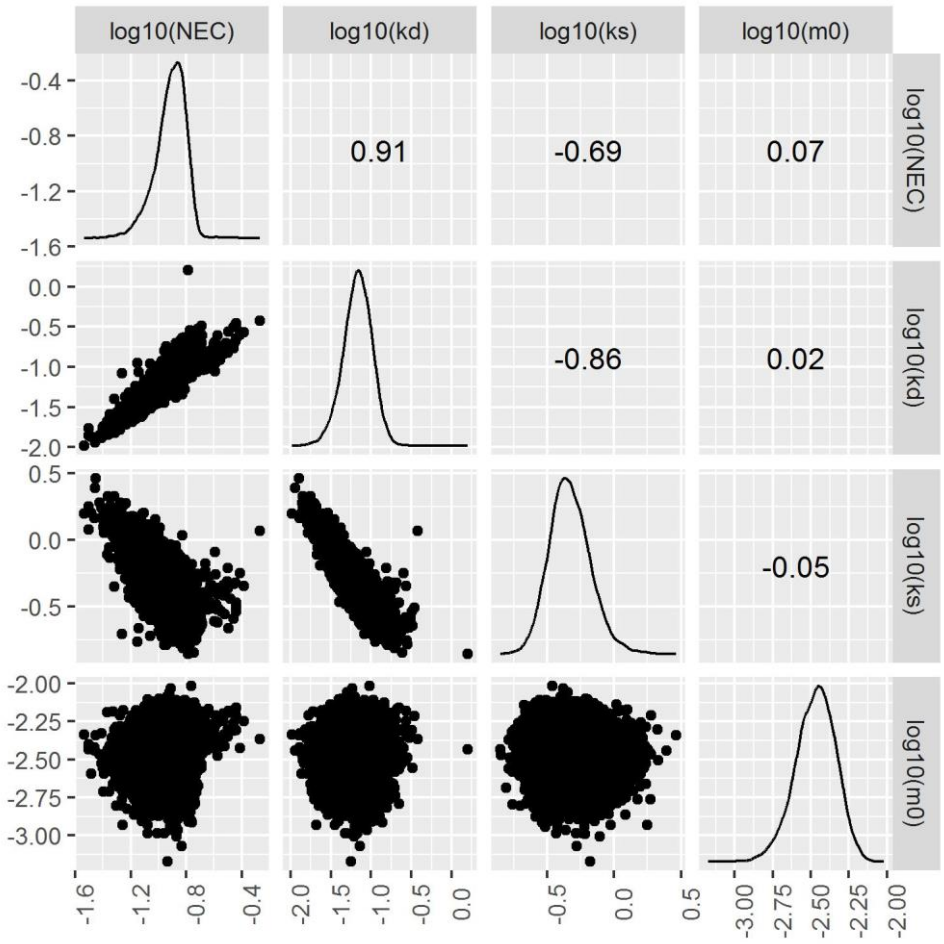

Figure S6 Scatterplot matrix of the joint posterior distribution of parameters estimated using the zinc data set, with density plots on the diagonal and correlation printed in the upper triangle.