Three perspectives on modelling for ecological risk assessment

A toy example with a simple one-compartment toxicokinetic model

Sandrine Charles and Roman Ashauer

2023-02-10

Abstract

We provide here a toy example based on the use of a simple one-compartment toxicokinetic model to describe the bioaccumulation of chemical substances within the whole body of living organisms. From a simple ODE model, we will illustrate : (P1) how to simulate both accumulation and depuration phases under constant exposure and to compare model outputs to observed data; (P2) how to fit such a model on data without using any prior information on the model (Frequentist point of view); (3) how to benefit of prior information in combination with knowledge in data to update the calibration results (Bayesian point of view).

Introduction

Perform calculations under the three perspectives as described within the main document

*** to be developed ***

Case study

Based on (Ashauer et al. 2010). Data set on *Gammarus pulex* exposed to Malathion.

A toxicokinetic (TK) model simply describing bioaccumulation of chemical substances within the whole body of living organisms is based on a set of two differential equations standing for the deterministic part (Charles, Ratier, and Lopes 2021):

$$\begin{cases} \dfrac{dC}{dt}(t) = k_u \times C_w - k_e \times C(t) \quad \forall\, 0 \leq t \leq t_c & \text{(1a)} \\ \dfrac{dC}{dt}(t) = -k_e \times C(t) \quad \forall\, t > t_c & \text{(1b)} \end{cases}$$

where t_c stands for the duration of the accumulation phase (namely the end of the exposure period, before organisms are transferred into a clean medium). Quantity C_w stands for the exposure concentration in water, while variable $C(t)$ corresponds to the internal concentration within the whole body of organisms over time t. Parameters k_u and k_e are the uptake and the elimination rates, respectively.

Given that state variables are concentrations, an appropriate stochastic part to describe the variability of the data around the mean tendency is the Gaussian distribution, so that:

$$C_{obs}(t_i) \simeq \mathcal{N}(C(t_i), \sigma) \tag{2}$$

where $C_{obs}(t_i)$ are the measured internal concentrations at time point t_i, $\forall\, i \in 1, n$, with n the total number of time points. Parameter σ stands for the standard deviation of the normal distribution.

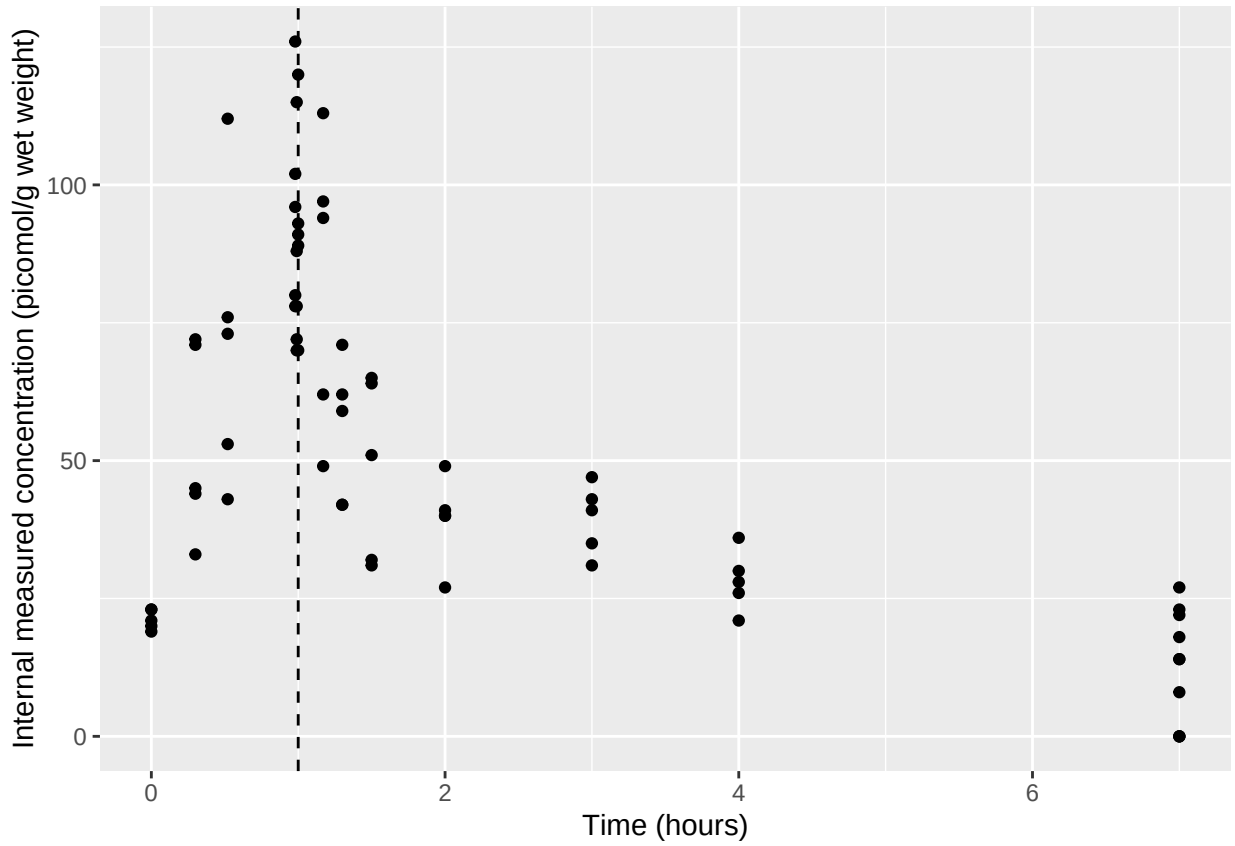


Figure 1: Raw data vizualisation (black dots). the vertical dashed line stands for the end of the accumulation phase ($t_c = 1$ day. Exposure concentration equals 1.485 picomol/ml.

Perspective 1 -

Under perspective 1, the idea is to simulate under the model equations (1) and to compare with observed data. In this perspective, we can use the k_u and k_e parameter estimates as provided in Table 1 of (Ashauer et al. 2010). However, we need to choose one of the two estimates: the one from the Marquardt fit, or the one from the MCMC fit. Note that these estimates are provide as aggregated values (mean $\pm$ standard deviation, and that the estimation of parameter σ (equation (2)) is missing.

Then, two simulation options can be considered: option 1, using only mean values for the simulations; option 2: accounting for the uncertainty. This latter raises the question of the probability distribution to consider. Because estimates are provided as means and standard deviations, this invites us to consider a normal distribution for both k_u and k_e parameters.

Below is a summary table with estimates from (Ashauer et al. 2010) corresponding to the raw data we are dealing with in this document:

```
tab1 <- read.table("table1.txt", header = TRUE)
kable(tab1)
```

Method	ku_mean	ku_sd	ke_mean	ke_sd	BCF_mean	BCF_low	BCF_up	t95
Marquardt	92.58	5.1	0.81	0.083	114.3	NA	NA	3.70
MCMC	93.40	10.2	0.82	0.173	115.3	144.4	144.4	3.64

Simulated model

Because the exposure concentration is here considered as constant, equations (1) can be analytically integrated as follows:

$$\begin{cases} C(t) = \dfrac{k_u}{k_e} C_w \left(1 - e^{-k_e t}\right) & \forall\, 0 \leq t \leq t_c \quad (3a) \\ C(t) = \dfrac{k_u}{k_e} C_w \left(e^{k_e (t_c - t)} - e^{-k_e t}\right) & \forall\, t > t_c \quad (3b) \end{cases}$$

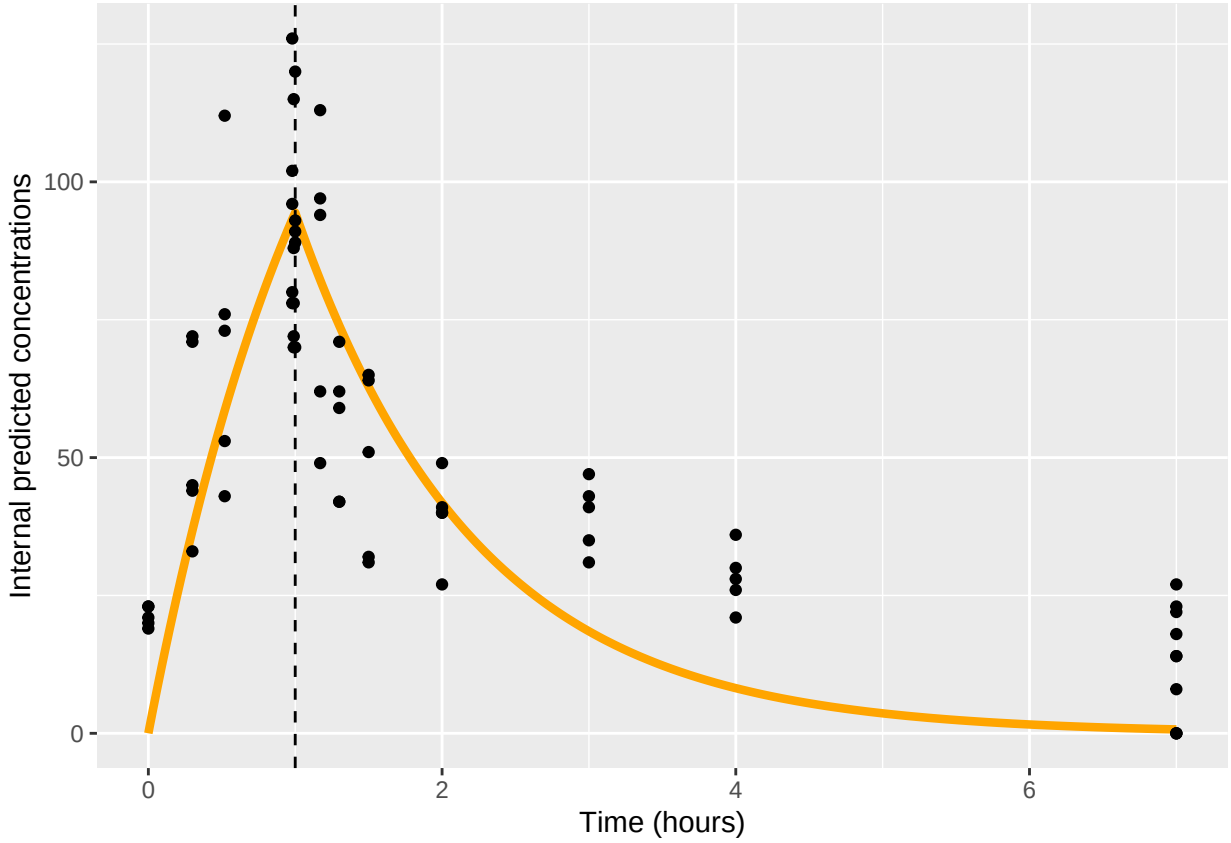
```
# Create a function to simulation bioaccumulation
# along both accumulation and depuration phases
bioacc <- function(parameters, expw, tc, tmax){
  tacc <- seq(0, tc, length.out = 100)
  Cacc <- parameters[1]*expw*(1 - exp(-parameters[2]*tacc))/parameters[2]
  tdep <- seq(tc, tmax, length.out = 200)
  Cdep <- parameters[1]*expw*(exp(parameters[2]*(tc - tdep)) - exp(-parameters[2]*tdep))/parameters[2]
  result <- data.frame(time = c(tacc, tdep),
                       conc = c(Cacc, Cdep))
  return(result)
}
```

Persp.1 - option 1

```
# Assign input parameter values
# Use the mean of Marquardt and MCMC fits
ku.mean <- mean(tab1$ku_mean)
ke.mean <- mean(tab1$ke_mean)
parameters <- c(ku.mean, ke.mean)
expw <- unique(df$expw)
tc <- 1 # duration of the accumulation phase
tmax <- max(df$time) # final time of the experiment
```

```r
# Launch simulations
simu.mean <- bioacc(parameters, expw, tc, tmax)
```

```r
# Plot simulations
ymax <- max(unique(df$conc)) # maximal observed concentrations
ggplot(simu.mean, aes(time, conc)) +
  geom_line(col = "orange", size = 1.5) +
  xlab("Time (hours)") +
  ylab("Internal predicted concentrations") +
  geom_vline(xintercept = 1, linetype="dashed") +
  coord_cartesian(xlim = c(0, tmax), ylim = c(0, ymax)) +
  # Add observed data
  geom_point(data = df, aes(time, conc))
```



Comparison between data and simulations

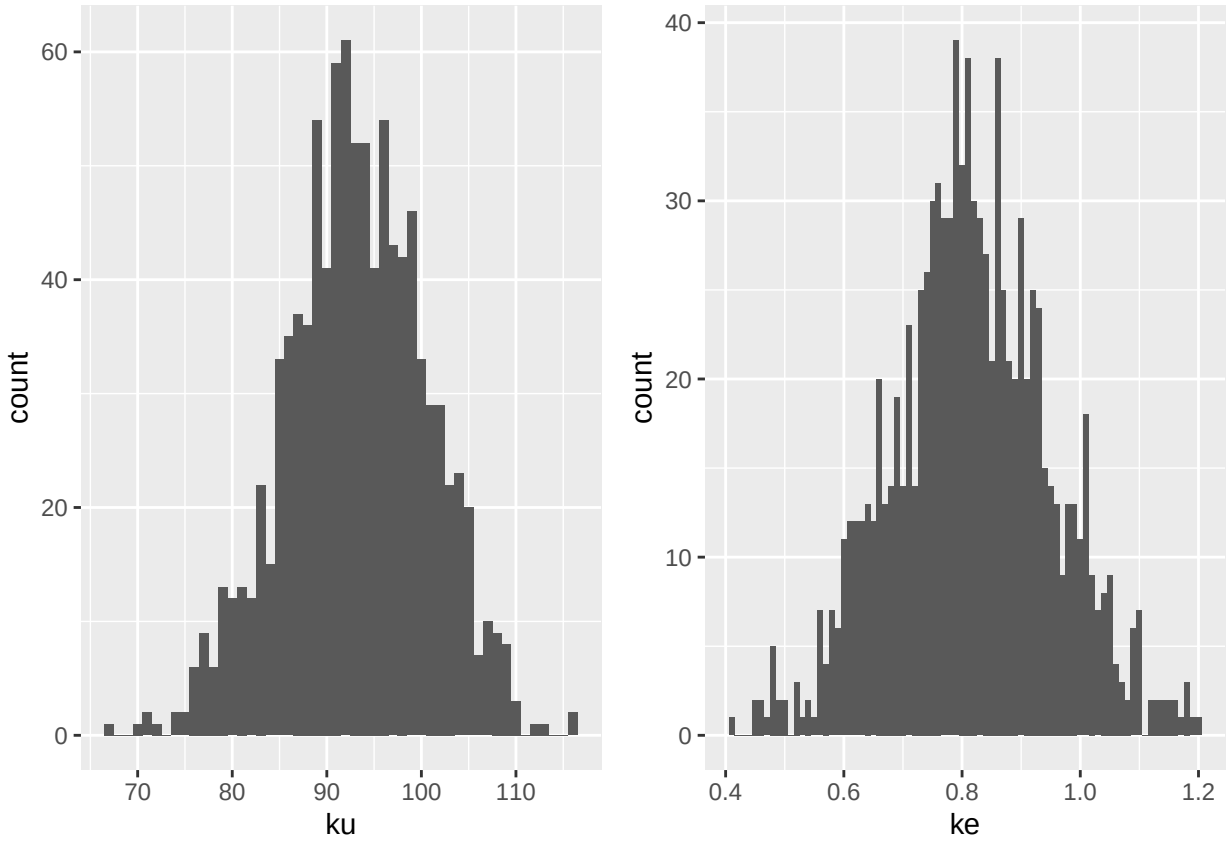
Persp.1 - option 2

Given that k_u and k_e parameters were provided with their standard deviations, it is tempting to add uncertainties around the previous mean predicted curve. However, no information is available on parameter σ to appropriately characterize the normal distribution of the stochastic part of the model (equation (2)). Without additional information, all parameters will be assumed to be normally distributed. Note also that, in doing as such, we do account for any correlation between model parameters.

As σ is expressed in the same unit as the observed concentrations, in first intention, we could assume that σ follows a normal distribution of mean and standard deviation equal to the mean and the standard deviation of the observations. Nevertheless, an alternative that would be less computing demanding, could be to fix $\sigma = 0.1$ (or any other arbitrary value).

Pers 1 - option 2 - ficed σ

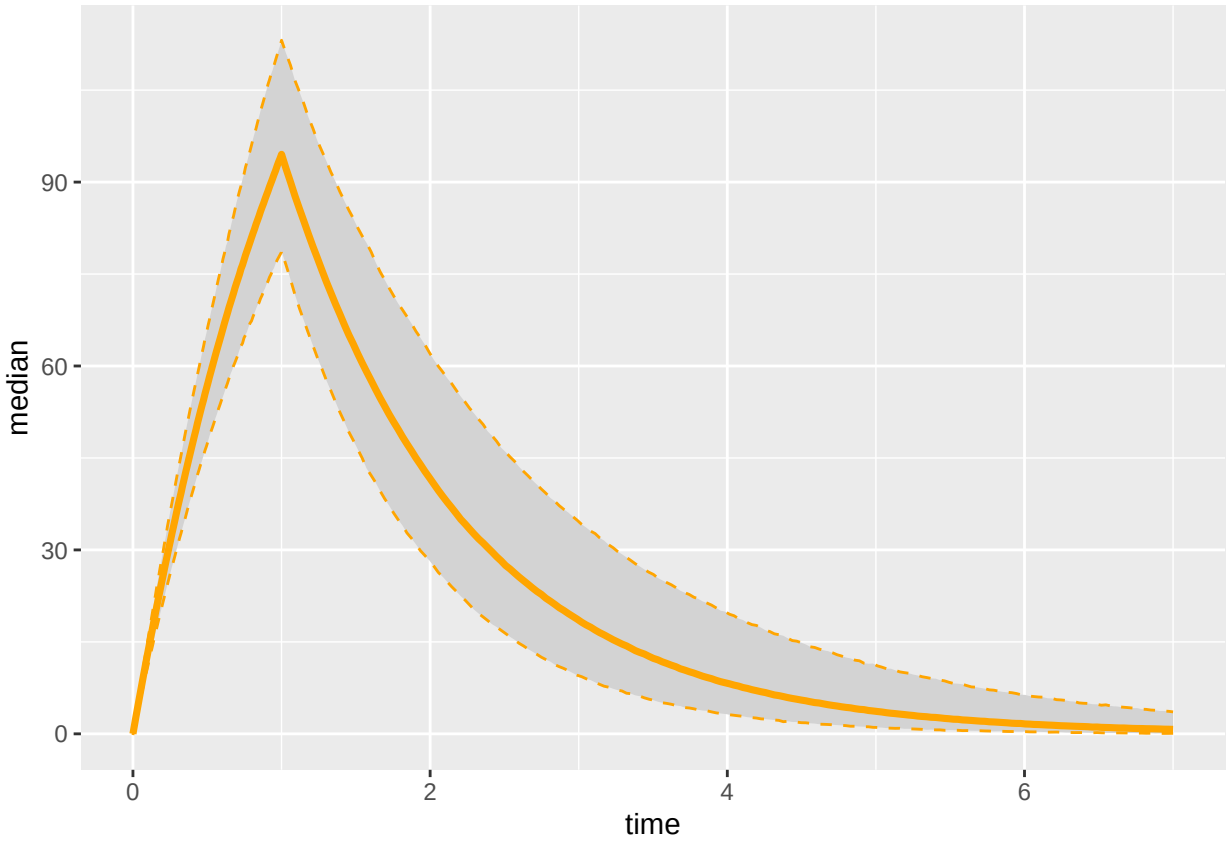
```
# Built normal distributions for model parameters
niter <- 1000 # sampling size in parameter distributions
ku.sd <- mean(tab1$ku_sd) # ku standard deviation
ku <- rnorm(niter, ku.mean, ku.sd) # ku distribution
ke.sd <- mean(tab1$ke_sd) # ke standard deviation
ke <- rnorm(niter, ke.mean, ke.sd) # ke distribution
sigma <- 0.1 # 50% of the observed sd
# Plot the simulated normal distributions
# for the model parameters ku and ke
g1 <- ggplot(as.data.frame(ku), aes(ku)) +
  geom_histogram(binwidth = 1)
g2 <- ggplot(as.data.frame(ke), aes(ke)) +
  geom_histogram(binwidth = 0.01)
grid.arrange(g1, g2, ncol = 2)
```




```
# Simulation of internal concentrations
# considering the above probability distributions
# and a fixed sigma value
pred <- c()
for(k in 1:niter){
  tmp.ku <- sample(ku, 1)
  tmp.ke <- sample(ke, 1)
  tmp.theo <- bioacc(parameters = c(tmp.ku, tmp.ke),
                     expw = expw, tc = 1, tmax = tmax)
  alea <- rnorm(n = 300, 0, sigma)
```

```
  tmp.pred <- tmp.theo$conc + alea
  pred <- rbind(pred, c(tmp.ku, tmp.ke, tmp.pred))
}
colnames(pred) <- c("ku", "ke", tmp.theo$time)
Qpred <- t(apply(pred[,3:302], MARGIN = 2,
                 FUN = function(x) quantile(x, probs = c(0.025, 0.5, 0.975))))
rownames(Qpred) <- NULL
Qpred <- as.data.frame(Qpred)
Qpred$time <- tmp.theo$time
colnames(Qpred) <- c("Lower", "median", "Upper", "time")
```

```
ggplot(data = Qpred, aes(x = time, y = median)) +
  geom_ribbon(aes(ymin = Lower, ymax = Upper),
              col = "orange", linetype = "dashed",
              fill = "lightgrey") +
  geom_line(col = "orange", size = 1.25)
```



Perspective 2

Perspective 3

In the section, we fit the one compartment model (equations (1)) under a Bayesian framework with the R-package `rbioacc` (Ratier and Charles 2022). The same calculation can be easily reproduced on-line with the MOSAIC web platform and its `bioacc` module: https://mosaic.univ-lyon1.fr/bioacc.

Model fitting

```
# Prepare the data to be use in the `rbioacc` package
mdf <- modelData(df, time_accumulation = 1, )
# fit the TK model built by default from the data
fit <- fitTK(mdf, refresh = 0)
```

Model equations

```
# Below is the code line allowing to get
# the used model equations
equations(fit, df)
```

Fitting results

```
# Get parameter estimates
# medians and 95% credible intervals
quantile_table(fit)
```

```
               2.5%       50%     97.5% parameter
ku        53.6643165 65.841757 82.958215        ku
kee        0.3714643  0.562678  0.861489       kee
sigmaConc 15.5384738 18.210791 21.766015 sigmaConc
```

```
# Fitting plot
plot(fit)
```

Bioaccumulation metric

```
# Calculation of the posterior probability distribution
# of the kinetic bioconcentration factor
bm <- bioacc_metric(fit)
# Display median and 95% credible interval of the BCF_k
quantile(bm$BCFk, probs = c(0.025, 0.5, 0.975))
```

```
    2.5%      50%    97.5%
 93.2962 116.7346 149.9672
```

The 95% elimination time can also be easily calulated.

```
signif(t95(fit), digits = 3)
```

```
[1] 5.32
```

Goodness-of-fit criteria

```
# Posterior Predictive Check (PPC)
# The expectation is to get ~95% of data
# within their prediction interval
ppc(fit)
```

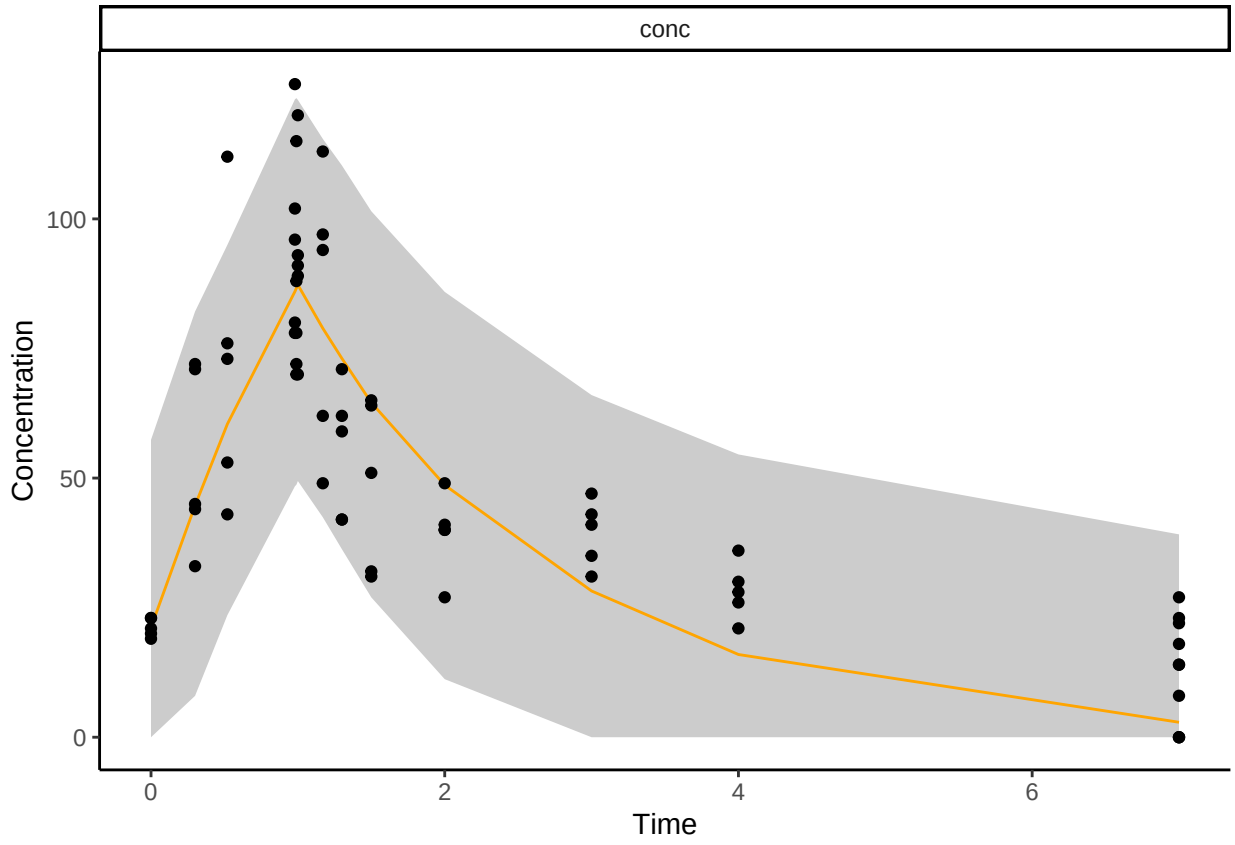
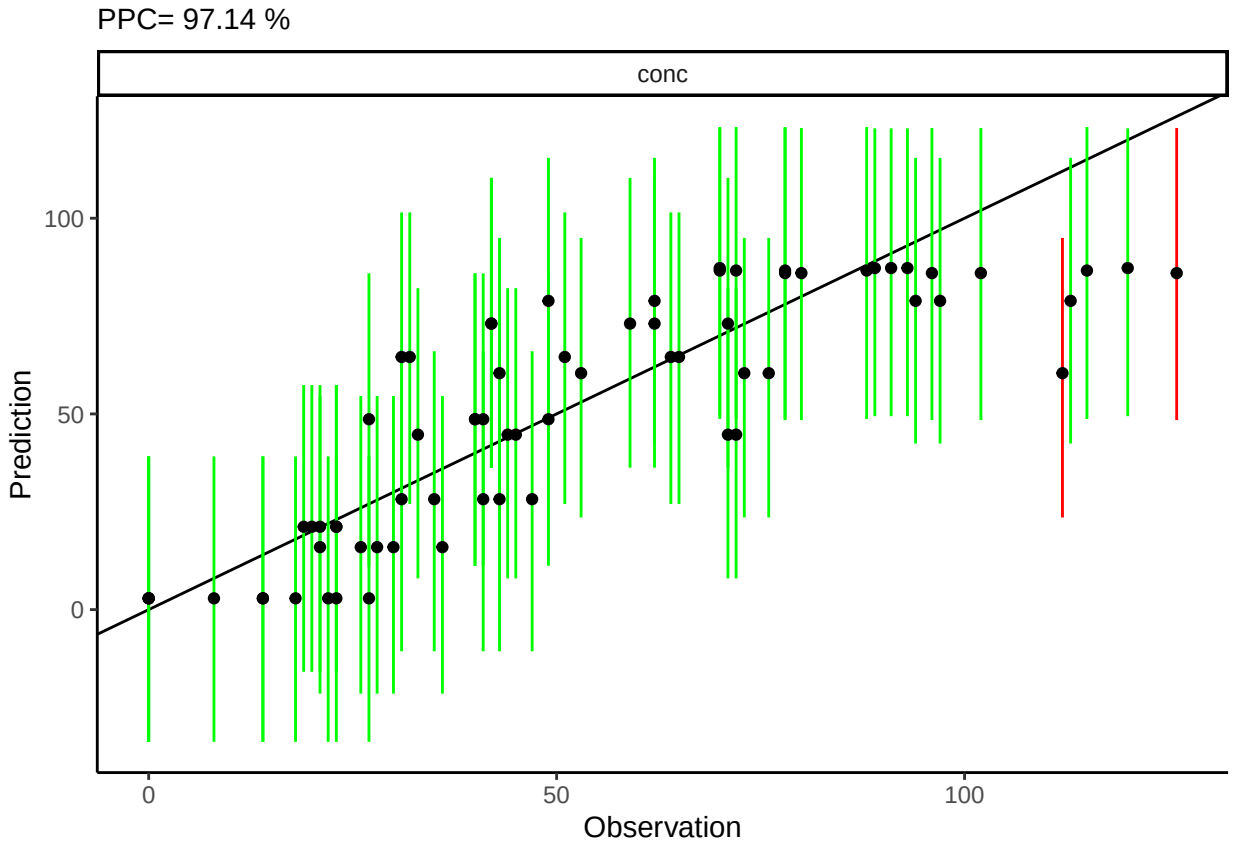
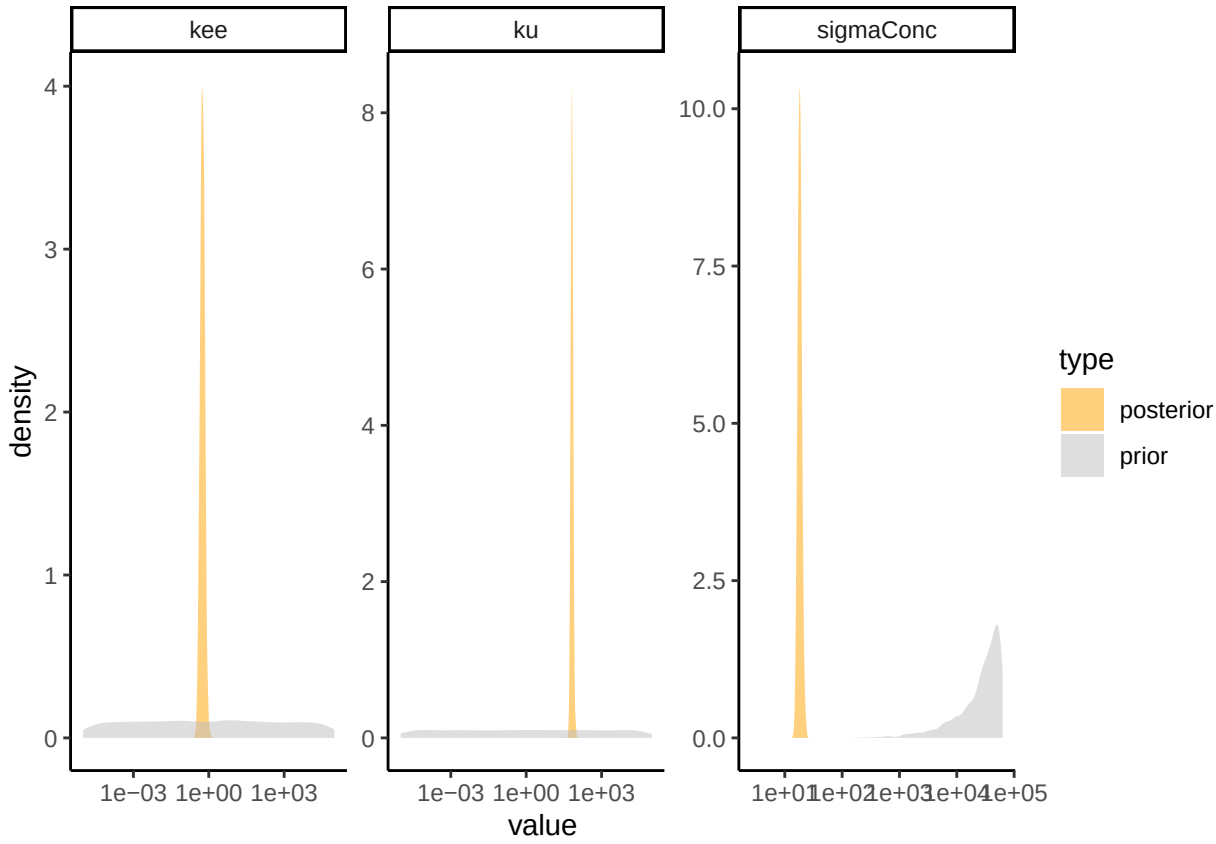


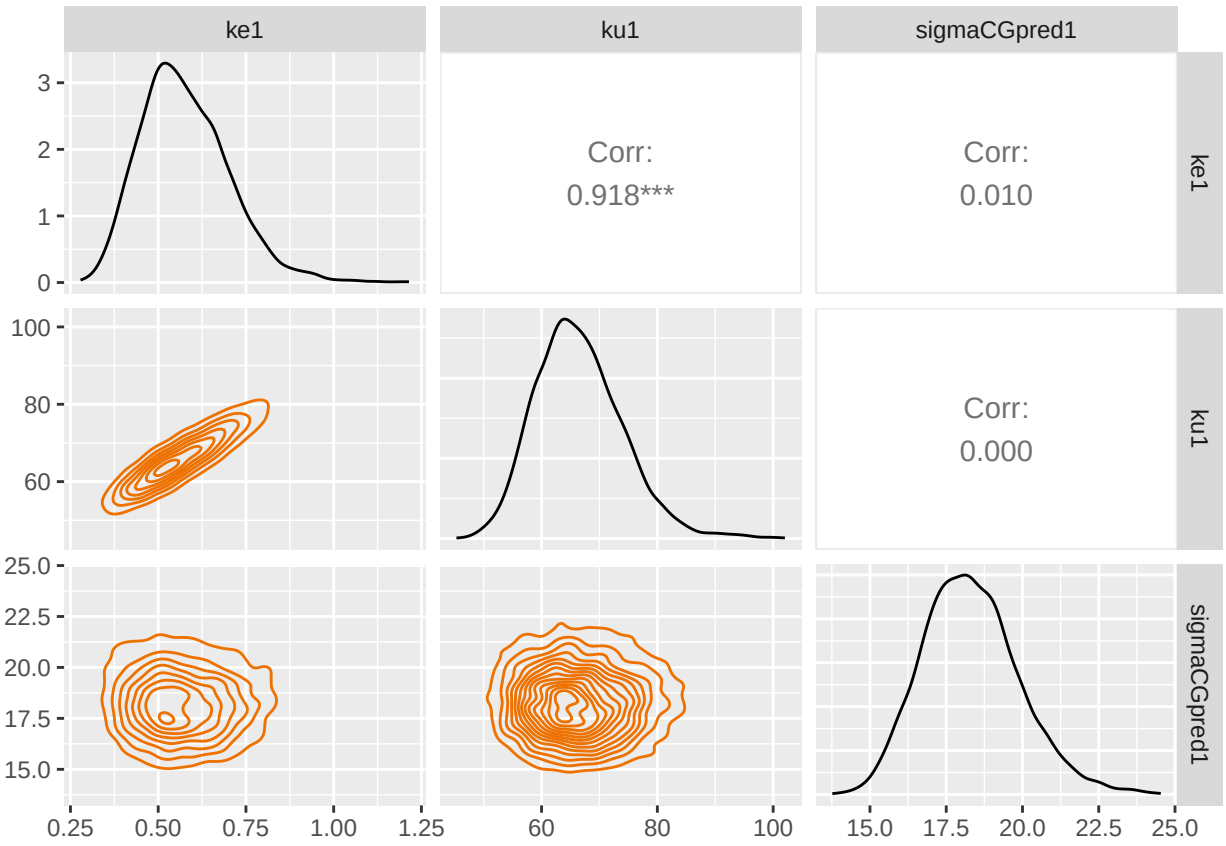
Figure 2: Fitting plot with black dots representing the observed data, the solid orange line the median predictive model and the grey area the 95% uncertainty band including the uncertainty on the model parameter estimates as well as the stochastic part of the model.




```
# Compare priors and posteriors
plot_PriorPost(fit)
```



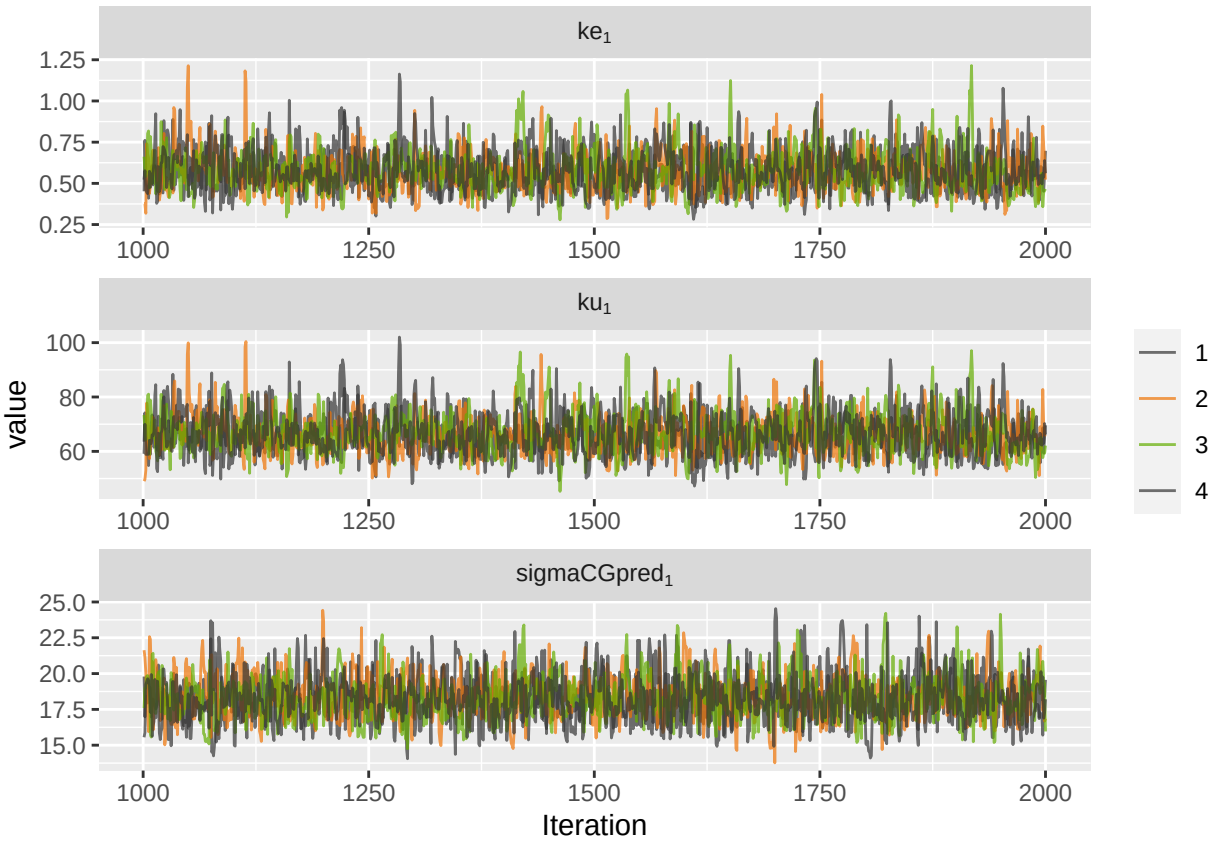

```
# Check for correlations between parameters
corrPlot(fit)
```




```
# Check  for non-signigicantly different traces
# of the four MCMC chains run in paralell
psrf(fit)
```

```
           PSRF parameter
ku         1.003        ku
kee        1.002       kee
sigmaConc 1.000 sigmaConc
```

```
# Look at the traces of the 4 MCMC chains
mcmcTraces(fit)
```



References

Ashauer, Roman, Ivo Caravatti, Anita Hintermeister, and Beate Escher. 2010. "Bioaccumulation kinetics of organic xenobiotic pollutants in the freshwater invertebrate Gammarus pulex modeled with prediction intervals." *Environmental Toxicology and Chemistry* 29 (7): 1625–36. https://doi.org/10.1002/etc.175.

Charles, Sandrine, Aude Ratier, and Christelle Lopes. 2021. "Generic Solving of One-compartment Toxicokinetic Models." *Journal of Exploratory Research in Pharmacology* 6 (4): 158–67. https://doi.org/10.14218/jerp.2021.00024.

Ratier, Aude, and Sandrine Charles. 2022. "Accumulation-depuration data collection in support of toxicokinetic modelling." *Nature, Scientific Data* 9 (1): 130. https://doi.org/10.1038/s41597-022-01248-y.

APPENDIX

Table of raw data

```
df <- read.table("data.txt", header = TRUE, sep = "")
kable(df[1:25,], format="latex")
```

time	conc	replicate	expw
0.00	23	1	1.485
0.00	19	2	1.485
0.00	20	3	1.485
0.00	21	4	1.485
0.00	23	5	1.485
0.30	44	1	1.485
0.30	72	2	1.485
0.30	33	3	1.485
0.30	71	4	1.485
0.30	45	5	1.485
0.52	43	1	1.485
0.52	76	2	1.485
0.52	53	3	1.485
0.52	112	4	1.485
0.52	73	5	1.485
0.98	102	1	1.485
0.98	78	2	1.485
0.98	96	3	1.485
0.98	126	4	1.485
0.98	80	5	1.485
0.99	72	1	1.485
0.99	115	2	1.485
0.99	70	3	1.485
0.99	88	4	1.485
0.99	78	5	1.485

One more thing

This will be Appendix B.