openGUTS www.openguts.info

Interpretation of output of the openGUTS software

Tjalling Jager*

May 10, 2020

This documents is part of the openGUTS project, and can be downloaded from http://openguts.info/. The openGUTS project is made possible by funding from Cefic-LRI in project ECO39.2.

Contents

*DEBtox Research, De Bilt, The Netherlands. Email: tjalling@debtox.nl, http://www.debtox.nl/

1 Introduction

This documents helps to interpret the output produced by the openGUTS software; both the Matlab version and the standalone software (the outputs are basically the same). The openGUTS website also contains user manuals (for getting started and running the software) and technical background documentation. However, it is important for users of the software to understand the basic concepts behind GUTS. Therefore, reading Chapters 1 and 2 of the GUTS e-book [4] is highly recommended. Section 2 in this document provides a short summary, only considering the models implemented in openGUTS and their parameters.

In the remainder of this document, I first go through a well-behaved example, explaining the graphical and text output of the software (Section 3). Section 4 treats some typical nasty cases that may occur. The information in that section is intended to identify those cases and to suggest a prudent course of action.

Disclaimer. This document is meant to aid users to understand what the model output means, and what is signified by specific types of outputs. It deals with several aspects that are not treated, at least not in detail, by the EFSA scientific opinion [2]. However, this document is written from a scientific perspective, and does not have any explicit or implied regulatory status.

Version information. Updating from the version of 21 May 2019, I added another special case, which I have christened the Maltese-cross anomaly. Updating from the version of 10 December 2019, I added a chapter on the underlying model and its parameters.

2 The GUTS model and its parameters

A thorough discussion of the GUTS model is outside of the scope of this document; the reader is referred to the (freely-downloadable) e-book [4]. However, some knowloedge of the model structure and its parameters is needed to interpret the output of openGUTS. Therefore, I will present in this chapter a concise description of the model versions implemented in the software.

The openGUTS software implements two variants, special cases derived from the GUTS framework. These are reduced models (RED), which means that toxicokinetics (uptake and elimination of the chemical) and damage dynamics are lumped into a single first-order compartment of 'scaled damage'. This is needed to allow modelling survival data in the absence of information on body residues (which is the normal situation for risk assessment, and for ecotox applications in general). The implemented variants of the RED models are the pure stochastic death (SD) and individual tolerance model (IT). The GUTS framework unifies these two death mechanisms, but the 'proper' model can generally not be fitted to the type of survival data sets available for risk assessment. Note that the BYOM platform under Matlab (`https://www.debtox.info/byom.html`) contains code to use other variants, at the expense of limited user friendliness.

2.1 Scaled damage dynamics

The structure of the GUTS-RED models is schematically shown in Figure 1, keeping the death mechanism vague for the moment. There is only one compartment in between the exposure scenario (external concentrations over time) and the death mechanism. This compartment is referred to as 'scaled damage'. This compartment is the one-compartment approximation of an essentially two-(or more-)compartment system.

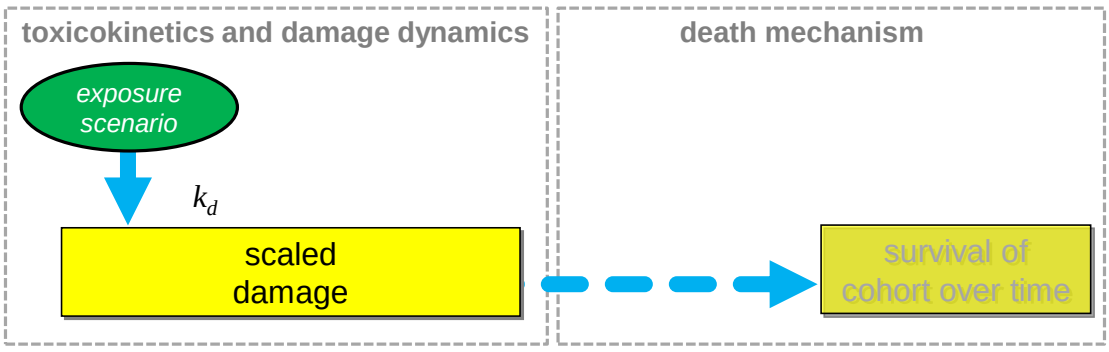

Figure 1: Schematic flow chart for GUTS-RED models; death mechanism not specified.

The equation for damage is a simple ordinary differential equation (ODE), linking the change in scaled damage (D_w) to the concentration in the exposure medium (C_w, which might be time varying):

$$\frac{dD_w}{dt} = k_d(C_w - D_w) \quad \text{with } D_w(0) = 0 \tag{1}$$

In openGUTS, this equation is solved analytically, also for time-varying exposure. The single parameter k_d is referred to as the 'dominant rate constant', and can generally be estimated from the patterns of survival over time.

Note that damage is scaled, which implies that it has the units of the external concentration. Under constant exposure, scaled damage will reach a steady-state value equal to the external concentration: $D_w(\infty) = C_w$ (illustrated in Fig. 2). The single parameter k_d governs the curvature, and thereby the time needed to reach steady state under constant exposure.

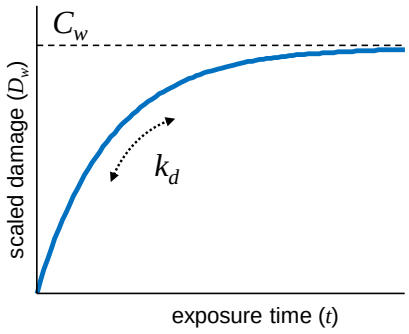

Figure 2: Pattern of scaled damage over time, under constant exposure.

2.2 GUTS-RED-SD

The structure of the GUTS-RED-SD model is schematically shown in Figure 3. The left part of the scheme was already discussed in the previous section, and here the right part will be filled in. Stochastic death implies that the individuals in a toxicity test are all identical, but have a probability to die, depending on the damage level in their bodies.

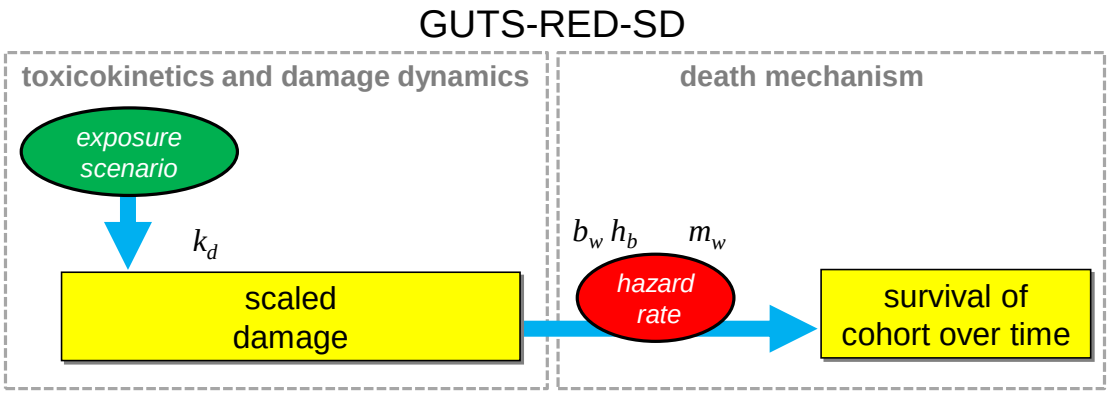

Figure 3: Schematic flow chart for GUTS-RED-SD.

Dealing with a continuously changing probability over time is the realm of hazard modelling. The hazard rate due to chemical stress (h_c) is the 'instantaneous probability to die', which is calculated using a linear-with-threshold relationship in D_w:

$$h_c = b_w \max(0, D_w - m_w) \tag{2}$$

As long as scaled damage is below the threshold m_w, the probability to die due to the chemical is zero. When scaled damage exceeds the threshold, the hazard rate increases linearly with a proportionality constant b_w (the killing rate or effect strength). This is illustrated in Figure 4.

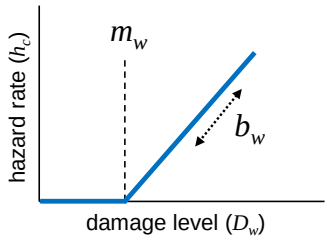

Figure 4: Linear-with-threshold relationship between hazard and scaled damage.

Turning the hazard rate into a survival probability (S_c) requires an integration over time:

$$S_c = \exp\left(-\int_0^t h_c(\tau)d\tau\right) \tag{3}$$

In openGUTS, this integration is performed analytically for constant exposure, and numerically (trapezium-rule integration) for time-varying exposure.

2.3 GUTS-RED-IT

The structure of the GUTS-RED-IT model is schematically shown in Figure 5. The left part of the scheme was already discussed above, and here the right part will be filled in. Individual tolerance implies that individuals differ in the value of their threshold for mortality, but deaths is instantaneous when scaled damage exceeds that threshold.

Calculating IT thus requires comparison of scaled damage D_w to the threshold distribution for the animals. In openGUTS, we assume a log-logistic distribution with median threshold m_w and a spread factor F_s. The spread factor is the factor by which you need to multiply and divide the median to cover 95% of the threshold distribution. E.g., $m_w = 10$ with $F_s = 2$ specifies a distribution with 95% of the threshold values between 5 and 20. The F_s is inversely proportional to the β of the log-logistic distribution but easier to interpret.

When comparing scaled damage to a threshold distribution, we need to take care when exposure is not constant: when damage decreases, the dead animals should not become alive again. Therefore, we first derive the maximum damage level up to that time point (D_{wm}). From that maximum, we can directly calculate survival probability S_c due to chemical exposure from the log-logistic distribution:

$$S_c = \frac{1}{1 + (D_{wm}/m_w)^\beta} \quad \text{with } \beta = \frac{\ln 39}{\ln F_s} \tag{4}$$

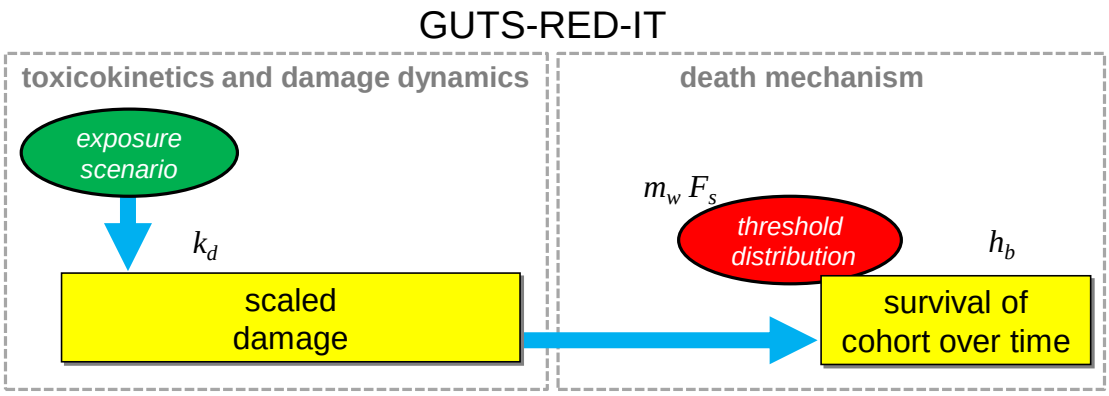

Figure 5: Schematic flow chart for GUTS-RED-IT.

Even though the same symbol m_w is used here for SD and IT, the interpretation is subtly different. Under IT, the threshold is distributed in the test population and m_w is the median of that distribution. Under SD, all individuals have the same threshold value m_w (which could still be seen as the median of a Dirac delta distribution: a distribution with all the probability density in a single point).

The threshold distribution for IT is illustrated in Figure 6. The threshold is a damage threshold. However, since we use scaled damage, it has the units of an external concentration (just like damage).

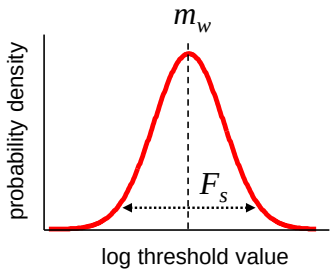

Figure 6: Threshold distribution, plotted on log scale.

2.4 Dealing with background mortality

The chemical stress may not be the only cause of death in the test system. Even in the control treatment, there may be mortality. Since the focus of openGUTS lies on the analysis of relatively short-term laboratory toxicity data, background mortality mostly relates to random events such as handling, and not to old-age effects. Therefore, we can treat background mortality as a constant hazard rate (h_b). Thus, when S_c is established, it needs to be multiplied with the background survival probability:

$$S = S_c \times \exp(-h_b t) \tag{5}$$

Note that $\exp(-h_b t)$ is the integration of a constant hazard rate over time as in Eq. 3.

3 A simple data set

Next, I will run through a simple data set to illustrate how everything goes in a rather best-case situation. This is a dataset for propiconazole in the amphipod *Gammarus pulex*. It was published by Nyman *et al* [5], and was also used as case study for GUTS in several places [4, 2]. The data set is distributed as example with the software.

Here, I will analyse this data set (keeping background hazard fixed to the value determined from the control treatment), validate with pulsed exposure data, and make predictions for a series of FOCUS profiles. Output shown is from the Matlab version, which is very similar to the output of the standalone version.[1] Only the stochastic death (SD) analysis is shown here, although one would generally perform both an SD and an IT (individual tolerance) analysis for the purpose of environmental risk assessment.

3.1 Input data

The Matlab version requires entering data as a formatted text file, whereas the standalone version additionally has an input grid to directly enter data or copy-paste from Excel. Both software versions can use the same text files, which are tab-delimited. The tabs don't line out nicely here, but the text file looks like this:

```
Part of the GUTS ring test. Real data set for
propiconazole in Gammarus pulex from Nyman et al
(2012). Ecotoxicology 21, 1828-1840.
Survival time [d] Control T1 T2 T3 T4 T5 T6 T7
0 20 20 20 20 21 20 20 20
1 19 20 19 19 21 17 11 11
2 19 20 19 19 20 6 4 1
3 19 20 19 18 16 2 0 0
4 19 19 17 16 16 1 0 0
Concentration unit: uM
Concentration time [d] Control T1 T2 T3 T4 T5 T6 T7
0 0 8.05 11.906 13.8 17.872 24.186 28.93 35.924
```

The first lines are a description of the data set (as many lines as you like), which is handy to keep track of what is what. Below that, a matrix with survivors over time (in days) in a series of treatments. Below that, the concentration unit (for displaying and archiving only). And below that, a matrix that provides the exposure scenario for this test; in this case (constant exposure) it is a single row of values (one for each treatment).

The first plot to make is an inspection plot (in the standalone version, press the button [Display data], in the Matlab version it is given automatically). This plot is only meant as a check on your data file: did you enter the correct data set in the correct manner? This is especially handy for checking the definition of time-varying exposure scenarios. The error bars on the survival data are the confidence intervals (CI) on the binomial proportion (the Wilson score interval). This provides an indication of how well the data are capable of

[1] Note that the output in this document is from an older version, and may (in non-essential ways) differ slightly from the latest version.

specifying survival probabilities (note that these interval will become tighter as the number of individuals underlying each point increases).

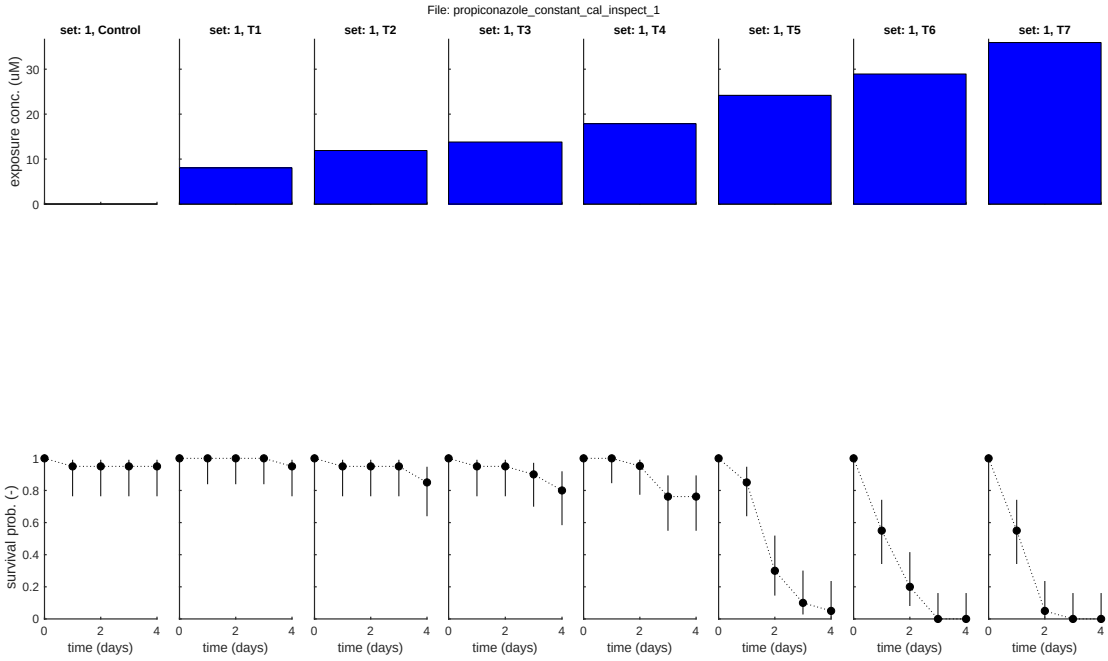

Figure 7: Inspection plot for the data set.

3.2 Calibration

Optimisation. Here, we are running the calibration with the background hazard rate h_b fixed. That is to say: it is fitted on the control treatment, and kept fixed when fitting the other model parameters.[2] We start by looking at the screen output:

```
Following data sets are loaded for calibration:
    set: 1, file: propiconazole_constant.txt

Settings for parameter search ranges:
=====================================================================
kd    bounds:   0.001641 -      143.8 1/d      fit: 1 (log)
mw    bounds:   0.002202 -      35.56 uM       fit: 1 (norm)
hb    bounds:    0.01307 -    0.01307 1/d      fit: 0 (norm)
bw    bounds:  0.0007332 -       9310 1/(uM d) fit: 1 (log)
Fs    bounds:          1 -          1 [-]      fit: 0 (norm)
=====================================================================
Special case: stochastic death (SD)
Background hazard rate fixed to a value fitted on controls
```

[2]Whether to fit h_b on the controls only or on all data is a choice that needs to be made by the user (or by a regulatory authority). There are pros and cons to both strategies. Section 2.3.4 of the e-book provides a discussion.

Search ranges have been automatically generated, based on the properties of the data set (time vector for survival data, and choice of exposure concentrations). The fit marks show that k_d, m_w and b_w are fitted. The other parameters are fixed: h_b to the value fitted on the control treatment, and F_s to 1 (which implies no differences between individuals, which is the consequence of having the pure-SD model here). Several parameters are fitted on log-scale, which is more efficient (they span a large search range and small values are highly relevant). In general, there will be no need to change these default settings. However, in some extreme cases, it may help to modify them.[3]

Next, the algorithm goes through several rounds of optimisation:

```
Starting round 1 with initial grid of 2016 parameter sets
  Status: best fit so far is (minloglik) 131.8616
Starting round 2, refining a selection of 200 parameter sets, with 60 tries each
  Status: 19 sets within total CI and 8 within inner. Best fit: 125.8153
Starting round 3, refining a selection of 200 parameter sets, with 40 tries each
  Status: 335 sets within total CI and 106 within inner. Best fit: 125.8153
Starting round 4, refining a selection of 867 parameter sets, with 57 tries each
  Status: 8497 sets within total CI and 3016 within inner. Best fit: 125.8153
  Finished sampling, running a simplex optimisation ...
  Status: 8498 sets within total CI and 3016 within inner. Best fit: 125.8152
Starting round 5, creating the profile likelihoods for each parameter
  Finished profiling, running a simplex optimisation on the best fit set found ...
  Status: 8499 sets within total CI and 3017 within inner. Best fit: 125.8152
```

Round one is an initial regular grid, trying parameter sets all over the search range to see where the promising sets are. The subsequent rounds mutate the most promising sets to obtain a good coverage of the parameter space around the best estimate. In the last round, profile likelihoods are generated (explained later). In this case, the analysis finishes after profiling. However, in more complex cases, additional sampling rounds (or even a new round of profiling) may be triggered.

The final plot of parameter space is shown in Figure 8. This plot is of major importance to judge the fit and whether the parameter's values can be identified from the data set. Therefore, this requires some explanation. Below the diagonal, there are three plots with some kind of 'clouds'. In fact, this is a single three-dimensional parameter cloud, projected in three 2-D plots. The yellow dot is the best estimate: the parameter set that leads to the highest value of the likelihood function. The inner cloud of green points are all within a certain goodness-of-fit distance from the best value, and the outer cloud of blue points are all within the joint 95% CI of the model parameters.

On the diagonal, we also see a projection of the parameter cloud (these are the same points as in the 2-D plots), but in a slightly different manner: the sample is plotted for a single parameter (x-axis), but with it's associated goodness-of-fit on the y-axis. The best value is again shown with a yellow dot, and plotted at a y-axis value of zero (in the scaling of these plots, higher values than zero mark a poorer fit). The green points in the profile plots are the same as the green plots in the 2-D plots, and the same goes for the blue

[3]This requires a larger degree of expertise from the user as it is certainly possible to modify these settings in such a way that the calibration algorithm will perform poorly or even fail.

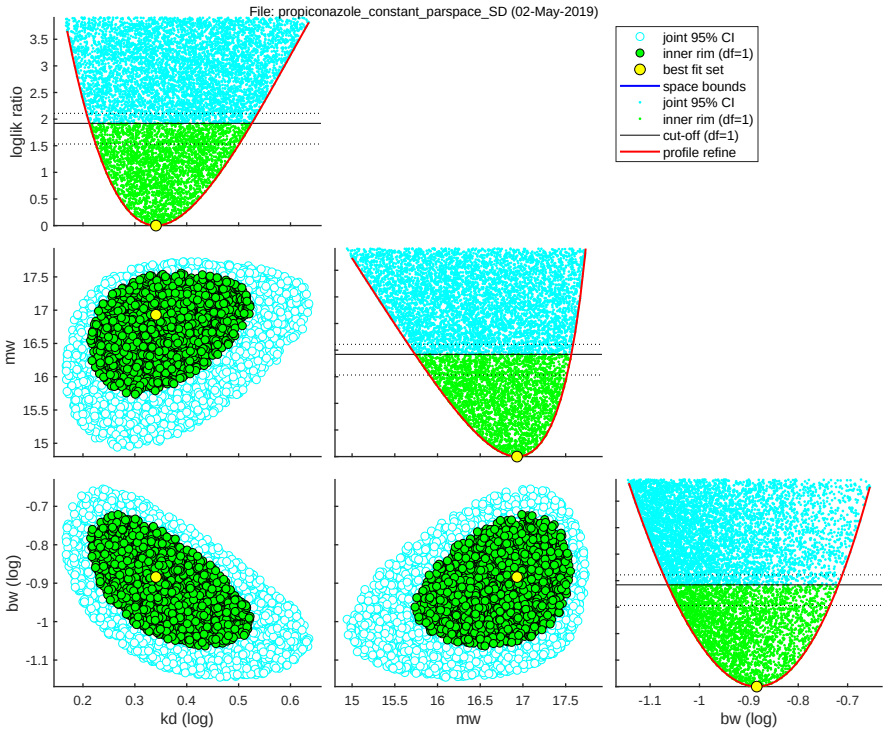

Figure 8: Final plot of parameter space. It shows the sample in all 2-D projections, and the profile likelihoods on the diagonal. The horizontal solid line in the profile plots is the critical value for the 95% CI. The dotted line above and below indicate the band from which the parameter sets will be used to propagate the uncertainty to model predictions.

points. The green points are all within a certain goodness-of-fit cut-off (a log-likelihood ratio of 1.92), which marks the single-parameter CI. The range between the minimum and maximum value of the green points in a profile plot for a parameter thus constitutes the CI of that model parameter.

The points were all derived by sampling. The red line in the profile plot is, however, a refinement, made by a clever algorithm to find the lowest possible log-likelihood ratio for each value of the parameter. If all goes well, the red line should be *just* below the sample points everywhere. There should be no sample points below the red line, and no gaps between the sample and the red line. Where the red line crosses the solid horizontal line is thus a refined estimate for our single-parameter CI. In well-behaved cases, there are two such crossings, and we have a nicely defined continuous CI. However, we'll see some cases later where this is less nice.

In principle, we can use the parameter sets that are on the horizontal line to propagate the uncertainty to model predictions (e.g., for our LC50 and LP50 estimates). However, the sample contains a discrete number of points from a continuous parameter space, so no point will be *exactly* on this line. Therefore, it is good to take all values within a certain band around this line. The dotted horizontal lines show the band of values that is used

for uncertainty propagation. This is a rather arbitrary bandwidth, but it is designed to yield representative CIs, also in cases where the parameter space is oddly shaped or even discontinuous.[4]

In this case, the parameter space plot looks very nice: well defined inner and outer clouds in the 2-D plots, not so much correlation between parameters, well-defined profiles that are almost parabola (which they will be for very large data sets), and a sample that nicely covers the relevant part of parameter space (no empty spots). We therefore can conclude that the optimisation worked well, and that all parameters are properly identifiable from the dataset. However, this does *not* mean that the fit was any good, or that this result is useful for risk assessment. Also, the other way around, an oddly-shaped parameter space does *not* imply a bad or useless fit (see Section 4). For that judgement, we would also need to look at the fit to the survival data in the next section.

The final result of the optimisation is output to screen:

```
================================================================================
Results of the parameter estimation
================================================================================
   GUTS prototype: version 0.5 of 29 April 2019
   Base name    : propiconazole_constant
   Analysis date: 02-May-2019 (12:37)
   Following data sets are loaded for calibration:
     set: 1, file: propiconazole_constant.txt
   Sample: 8499 sets in joint CI and 3017 in inner CI.
   Propagation set: 1308 sets will be used for error propagation.
   Minus log-likelihood has reached 125.8152 (AIC=257.63).
Best estimates and 95% CIs on single parameters
============================================================================
kd    best:      2.191 (     1.630  -     3.353 ) 1/d       fit: 1 (log)
mw    best:      16.93 (     15.74  -     17.57 ) uM        fit: 1 (norm)
hb    best:    0.01307 (       NaN  -       NaN ) 1/d       fit: 0 (norm)
bw    best:     0.1306 (   0.08626  -    0.1912 ) 1/(uM d)  fit: 1 (log)
Fs    best:      1.000 (       NaN  -       NaN ) [-]       fit: 0 (norm)
============================================================================
Special case: stochastic death (SD)
Background hazard rate fixed to a value fitted on controls
```

This provides some information on the fit, such as the size of the sample, and the final minus log-likelihood value. The parameter matrix provides the best estimate for each parameter and its 95% likelihood-based CI, and recaps the settings (fitting yes/no and fitting on log or normal scale). Note that `NaN` stands for 'not a number'. Below that, it gives some extra information:

```
Extra information from the fit:
============================================================================
Depuration/repair time (DRT95)    : 1.4 (0.89 - 1.8) days
Parameter ranges used for the optimisation:
kd    range:   0.001641 -      143.8
```

[4]The principles of using a frequentist sample from parameter space for CIs and error propagation is explained in more detail in the GUTS e-book [4], Appendix D.

```
mw    range:   0.002202 -      35.56
hb    range:    0.01307 -    0.01307
bw    range:  0.0007332 -       9310
Fs    range:          1 -          1
================================================================================
Time required for optimisation: 21.3 secs
```

It gives a depuration/repair time: when we would expose an animal to a pulse of the chemical, this is the estimated time needed for damage to return to 5% of the value that it was at the end of the pulse. The DRT95 can be used as a pragmatic approximation for the time needed between two pulses to make them independent (as proposed in the EFSA scientific opinion).[5] This value can thus be used to design validation tests with differently-spaced pulses; in this case, two pulses with 2 days or more in between would be treated as being independent, based on this fit. Below that, the search ranges are given once again.

Note. The optimisation algorithm applies random mutation of candidate parameter sets. Therefore, each run with the software will produce slightly different results. I would expect the same minus log-likelihood in all cases (with 4 digits behind the decimal point), maximum differences in the parameter estimates within 0.1%, and maximum difference in single parameter CIs within 1%. For predictions, the difference may be slightly higher, as prediction relies directly on the sample, but generally those will be within 2%.

Plotting. The results of the calibration are plotted with the standard plotting format for the openGUTS software: three rows of plots with exposure, survival, damage versus time, for each treatment separately.

Judging the fit will be mainly based on the lower row of plots: the estimated and observed survival probability over time. Judging these plots requires some expertise. The main trick is to see whether the patterns in the data are well represented by the model patterns. The CIs are of help in this respect, but it is good to keep in mind what they represent. The CIs on the model curves (the green areas) represent the uncertainty in the model curve. In other words, if we would repeat the experiment, and make best-fit model curves again, we would expect to find these new *curves* within the green bands.[6] However, these bands tell us nothing about where to find new *observations* on survival probability. Where we can expect new observations depends on how many individuals we will test. If we test 1000 individuals in each treatment, the observations should be within the green bands, but not if we test 4 individuals per treatment. Therefore, we should not demand that all observations fall within the green bands.[7] The observations can only give an estimate for the survival probability at each point, and these estimates are also uncertain. The Wilson

[5] Technically, two pulses will never be independent. Damage will decrease exponentially after a pulse but never truly reach zero.

[6] More accurately: 95% of the CIs created in this way will capture the 'true' curve.

[7] It would be possible to create additional bands that should capture where the data points can be expected (prediction intervals). However, this requires some further theoretical work before such an option can be implemented into a future update.

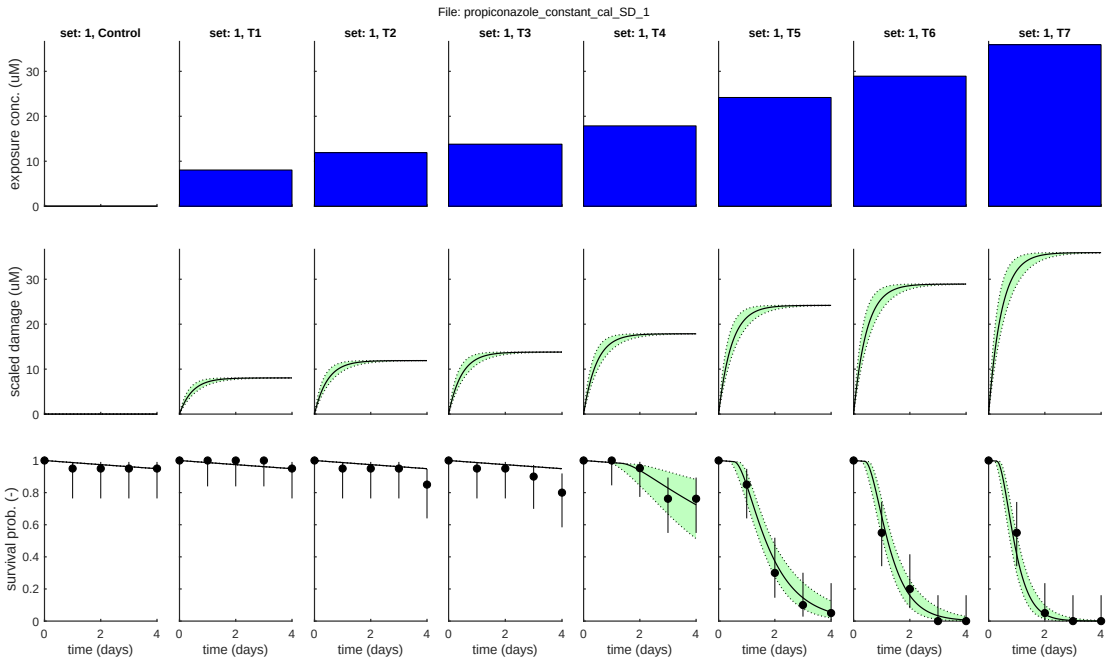

Figure 9: Model fit. Top row: exposure scenario, middle row: modelled damage with 95% CI, lower row: modelled survival probability (with 95% CI) and survival observations (with Wilson score interval).

score (error bars on the data points) gives an indication of this uncertainty. Together, these two CIs give an impression of whether the deviations between model and data are acceptable or not.

However, the CIs should not be our sole focus for judging the fit. We should consider the CIs together with the first statement I made: "see whether the patterns in the data are well represented by the model patterns." Looking at the fit in Figure 9, both the model and the data span a nice range from no effects to complete mortality. Furthermore, the pattern in the high treatments is well captured by the model and gives no reason for concern. In the lower treatments (T1-T3) there is, however, a peculiar pattern: the model fit judges all mortality in these treatments to be background mortality, but the observations show a slight dose-dependent increase in mortality. The model cannot fit these low-dose effects together with the high-dose effects and focusses on a good fit of the latter. This will in general be the case: observation intervals with high mortality carry more weight in the likelihood function than observation intervals with low mortality. It is possible that these low-dose effects are simply caused by randomness, and that they will not show up when repeating the test. However, it is good to keep the possibility of low-dose effects in mind.

On screen, the model output will include some goodness-of-fit measures, as proposed in the EFSA scientific opinion [2]:

```
Goodness of fit measures for calibration
  Special case: stochastic death (SD)
```

```
=========================================================================
Model efficiency (NSE, r-square)          : 0.9811
Normalised root-means-square error (NRMSE): 7.948 %
Survival probability prediction error (SPPE) for each treatment
     Data set     treatment    value
        1         Control     +0.0938 %
        1              T1     +0.0938 %
        1              T2     -9.91 %
        1              T3     -14.9 %
        1              T4     +3.79 %
        1              T5     -0.559 %
        1              T6     -0.690 %
        1              T7     -0.0297 %
=========================================================================
Warning: these measures need to be interpreted more qualitatively
  as they, strictly speaking, do not apply to quantal data
```

The last message warns us that these measures are limited; it is difficult to come up with a useful goodness-of-fit measure for survival data over time. From a scientific viewpoint, it is therefore not recommended to apply a strict pass-fail criterion on such measures.[8]

Model fits are also plotted as predicted-versus-observed plots (Fig. 10). This provides an additional means to judge the goodness-of-fit. The dotted lines show a rather arbitrary 20% deviation.[9]

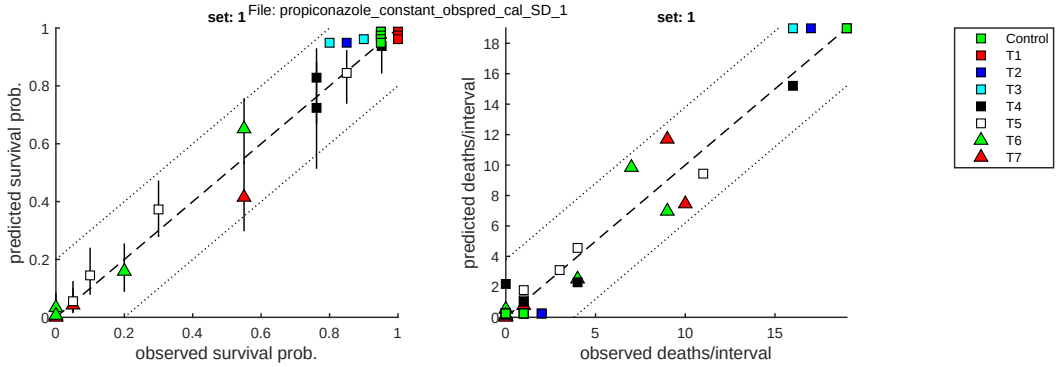

Figure 10: Predicted-versus-observed plots for survival probability (left) and number of deaths in each time interval (right). Error bars on predicted survival probability are the CI resulting from the parameter uncertainty (green bands in Fig. 9). Dotted lines mark a deviation of 20% from the 1:1 line, which can help to guide the eye.

As a final word of warning: the goodness-of-fit measures given above, and most of the plots, are comparing predicted and observed survival probabilities (or numbers). However,

[8] To calculate the NSE and NRMSE, the data and model value at $t = 0$ are excluded. Especially for the NSE, this may not be such a good idea. Therefore, this needs to be reconsidered.

[9] The EFSA opinion, Fig. 23, shows lines for 25 and 50% deviation, but this is based on the distance at a 45-degree angle from the solid line (as percentage of the initial number of animals). Since this is a rather awkward (and arbitrary) definition, we decided on a simple alternative that provides roughly the same distance.

the likelihood function that is optimised is *not* based on such comparison. The likelihood function compares observed and predicted deaths in each interval between observations. The right plot of Figure 10 is thus closest to how the model is optimised. In practice, we are usually more interested in whether the model is able to accurately predict the survival probability over time. For a good fit, it will generally be the case that the best fit on the deaths also provides a good representation of the survival probability. However, this complication needs to be kept in mind when judging data sets that fit less well.

3.3 Estimation of LCx

The parameters (and the parameter cloud) can be used to make predictions with CIs. One useful prediction is for the LC$x, t$: the concentration that, when applied as a constant exposure, produces x% mortality (relative to the control) after t days of exposure. The model provides output to screen for LC50, LC20 and LC10 at various standard time points, and a plot for LC50 and LC10 (Fig. 11).

```
Results table for LCx,t (uM), with 95% CI
  Special case: stochastic death (SD)
======================================================================
time (d)       LC50               LC20               LC10
======================================================================
   1   33.3 ( 29.9- 37.7)  25.5 ( 23.1- 28.1)  23.1 ( 20.8- 25.3)
   2   22.6 ( 21.5- 24.0)  19.5 ( 18.5- 20.3)  18.6 ( 17.5- 19.2)
   3   20.1 ( 19.2- 21.0)  18.2 ( 17.2- 18.8)  17.7 ( 16.6- 18.2)
   4   19.0 ( 18.1- 19.8)  17.7 ( 16.7- 18.3)  17.4 ( 16.2- 18.0)
   7   17.9 ( 16.9- 18.6)  17.3 ( 16.1- 17.9)  17.1 ( 15.9- 17.7)
  14   17.4 ( 16.2- 18.0)  17.1 ( 15.9- 17.7)  17.0 ( 15.8- 17.7)
  21   17.2 ( 16.0- 17.9)  17.0 ( 15.8- 17.7)  17.0 ( 15.7- 17.6)
  28   17.1 ( 15.9- 17.8)  17.0 ( 15.8- 17.6)  17.0 ( 15.7- 17.6)
  42   17.1 ( 15.8- 17.7)  17.0 ( 15.7- 17.6)  17.0 ( 15.7- 17.6)
  50   17.0 ( 15.8- 17.7)  17.0 ( 15.7- 17.6)  16.9 ( 15.7- 17.6)
 100   17.0 ( 15.7- 17.6)  16.9 ( 15.7- 17.6)  16.9 ( 15.7- 17.6)
======================================================================
```

All LCx values will decrease over time until they reach a stable or incipient value. For SD, the LCx values for all x will go towards the same value: the threshold m_w (compare the last value at $t = 100$ in the LCx table to the best estimate for m_w in Section 3.2). This is a property of the SD model. For the IT model, the LCx values will ultimately run parallel for different values of x. The rate at which the LCx values decrease over time depends on the model parameters: on k_d, and for SD on b_w as well. The uncertainties in the parameter estimates (including their correlations) is propagated to a CI for the LC$x, t$ values.

The interesting thing is thus that openGUTS can provide robust estimates for LC$x, t$ values, using all observations at all time points, even for time points beyond the test observations, even when exposure has not been constant in a test, and even when the dataset would not be suitable for classic dose-response analysis (see [3]).

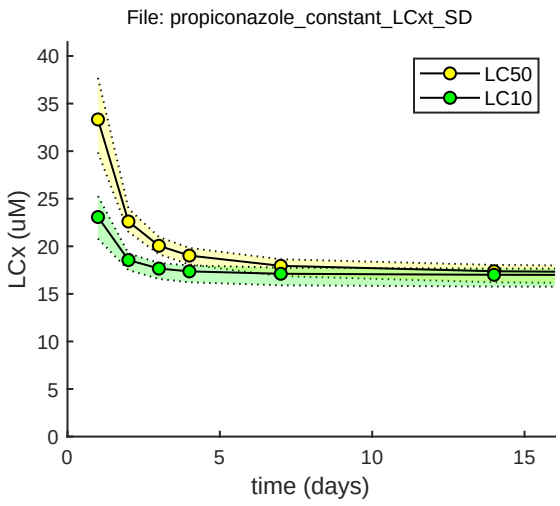

Figure 11: Plot of LC50 and LC10 over time, with 95% CIs.

3.4 Validation

The data set. Next, we can validate the calibrated model using data for pulsed exposure, from the same publication. The data set comprises a control (which will be used to fix the background hazard rate), two pulse treatments, and a longer-term constant exposure. The pulses have 2 or 6 days in between. In view of the estimated DRT95, both scenarios would be treated as being 'toxicologically independent' under the SD hypothesis: negligible carry-over toxicity is expected as the damage caused by the first pulse will be almost completely repaired by the time the second pulse hits the animals.[10] The text file of the data set looks like this:

```
Part of the GUTS ring test. Real data for propiconazole in
Gammarus pulex from Nyman et al  (2012). Ecotoxicology 21, 1828-
1840. Note that the pulses are specified as a linear series of time
points and concentration, with a short (but non-zero) transition time
from one exposure situation to the next.
Survival time [d] Control close pulses wide pulses constant
0   60 70 70 70
1   59 50 57 70
2   58 49 53 69
3   58 49 52 69
4   57 45 50 68
5   57 45 46 66
6   56 45 45 65
7   56 42 44 64
8   56 38 40 60
9   55 37 38 55
10  54 36 37 54
Concentration unit: uM
```

[10] Note that for IT, the DRT95 will be (very) different. This implies that the DRT95 criterion should not be applied too strictly in the design of validation tests.

```
Concentration time [d] Control close pulses wide pulses constant
0    0 30.56 28.98 4.93
0.96 0 27.93 27.66 4.69
1 0 0 0 4.69
1.96 0 0.26 0.27 4.58
2.96 0 0.21 0.26 4.58
3 0 27.69 0.26 4.58
3.96 0 26.49 0.26 4.54
4 0 0 0.26 4.54
4.96 0 0.18 0.25 4.58
4.97 0 0.18 0.25 4.71
5.96 0 0.18 0.03 4.71
6.96 0 0.14 0 4.6
7 0 0.14 26.98 4.6
7.96 0 0.18 26.28 4.59
8 0 0.18 0 4.59
9 0 0 0.12 4.46
9.96 0 0 0.12 4.51
```

Again, it is a good idea to make an inspection plot (Fig. 12) to check that the exposure scenario was represented appropriately.

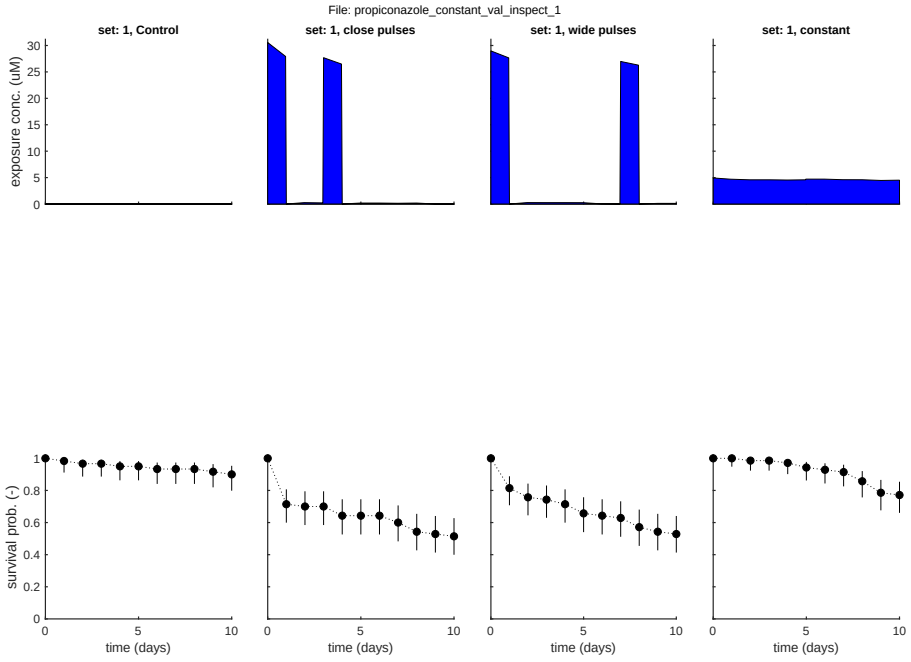

Figure 12: Inspection plot for the validation data set.

The comparison. Next, we can calculate the model output for damage and survival for these new exposure scenarios (which were not used for calibration). The background hazard rate is fixed, to the fitted value for the control of the validation stage. There might

be some difference in the background mortality between the two experiments (which were done at different times of the year), and it is generally a good idea to use the value relevant for the current control situation.

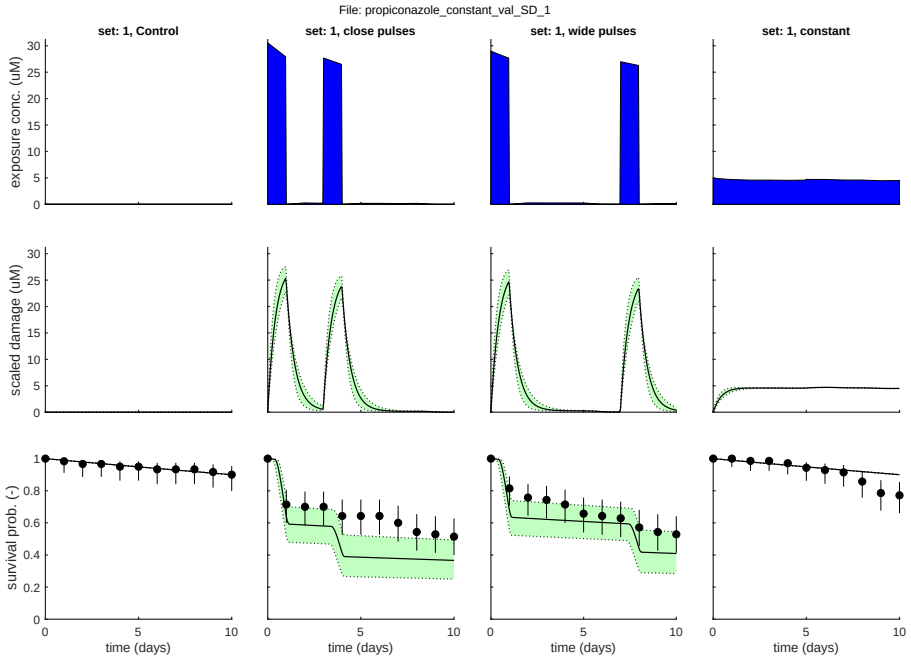

Figure 13: Standard plot for the validation data set. Note that the only thing that is fitted here is the background hazard rate on the control treatment.

The comparison between model and independent data is made with the same type of figures as for the calibration: a standard plot of the state variables versus time (Fig. 13) and a predicted-observed plot (Fig. 14). Additionally, the same goodness-of-fit measures as for calibration are printed on screen:

```
Following data sets are loaded for validation:
    set: 1, file: propiconazole_pulsed_linear.txt
Goodness of fit measures for validation
  Special case: stochastic death (SD)
========================================================================
Model efficiency (NSE, r-square)          : 0.5214
Normalised root-means-square error (NRMSE): 14.821 %
Survival probability prediction error (SPPE) for each treatment
     Data set     treatment    value
        1          Control     +0.00623 %
        1       close pulses     +14.8 %
        1       wide pulses      +11.9 %
        1         constant      -12.9 %
========================================================================
Warning: these measures need to be interpreted more qualitatively
  as they, strictly speaking, do not apply to quantal data
```

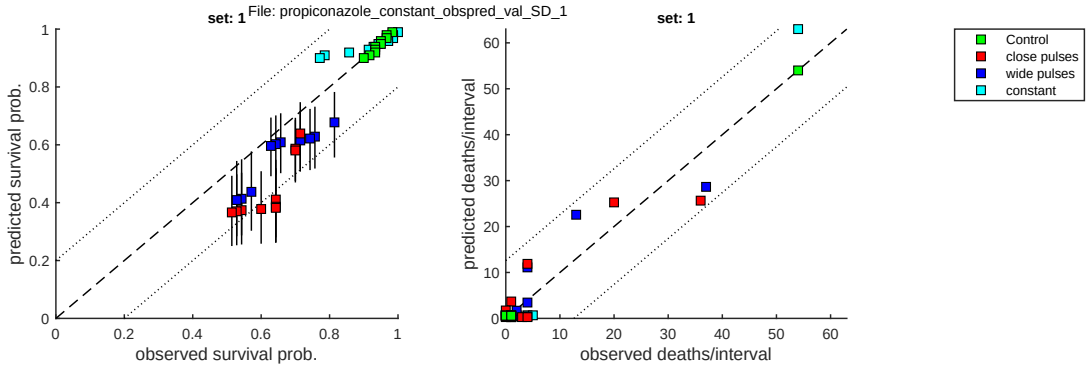

Figure 14: Predicted-versus-observed plots for survival probability (left) and number of deaths in each time interval (right). Error bars on predicted survival probability are the CI resulting from the parameter uncertainty. Dotted lines mark a deviation of 20% from the 1:1 line, which can help to guide the eye.

Clearly, the correspondence of the model to the data is not as good as for the calibration. The CI on the observed survival probabilities almost overlap with the green bands on the model predictions. However, what is more interesting is that the model predicts a pattern with a clear change in survival at each pulse, which is not obvious from the data (the second pulse does not have a clear impact; Fig. 13).[11] Another interesting point is seen for the constant exposure scenario: the model predicts no mortality other than background, but the data show a clear increase in mortality by the end of the test (which is not seen in the control treatment). Taken together with the dose-dependent mortality in the low doses for the calibration (Fig. 9), this points at the possibility for an additional mechanism of action (acting slowly and with a lower threshold).

3.5 Prediction

The data sets. The prediction capabilities for the model are geared towards the estimation of LPx values: the factor by which an entire exposure profile needs to be multiplied to yield x% mortality by the end of the profile. This factor can thus be viewed as a 'margin-of-safety' (the higher the value, the less mortality risk). The exposure profiles are entered as text files (the standalone software has an input grid for the exposure profiles as well), in a simple tab-delimited two-column format without headers. The first column contains time points (in days) and the second the exposure concentration (in the same unit as used for the calibration!). As with the calibration and validation data, an inspection plot is made as in Figure 15.

Batch-wise screening. In practical application of GUTS in risk assessment of pesticides, we expect that there is a common need to screen a large number of exposure profiles

[11] Although the fact that the calibrated model overestimates the mortality in these cases may be important for risk-assessment applications.

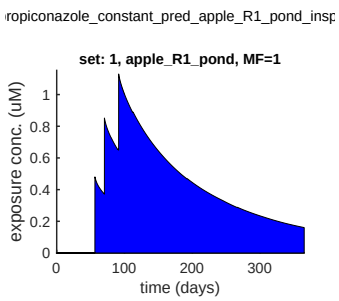

Figure 15: Inspection plot for one exposure profile (scenario 'apple R1 pond').

to see where the problems will be. Therefore, the openGUTS software contains the possibility to perform such a screening automatically (in a batch process). The user selects the input profiles (text files), and the model runs through them one by one and calculates LP50 and LP10. The final result is displayed on screen as a table (unsorted in the standalone software, and sorted on LP50 in the Matlab version):

```
Starting batch processing ...
Running through scenarios (10): 1 2 3 4 5 6 7 8 9 10
LPx values, scenarios ordered by descending risk (LP50)
  Special case: stochastic death (SD)
file analysed             LP10       LP50
=================================================================
     cereal_D1_ditch       2.29       2.65
      cereal_D5_pond       11.0       11.7
      cereal_D4_pond       11.1       11.7
     cereal_D3_ditch       8.59       12.3
       apple_R1_pond       15.9       16.7
    cereal_D1_stream       13.6       20.2
    cereal_R4_stream       30.6       38.4
     apple_R2_stream       31.8       43.4
    cereal_D5_stream       114.       193.
    cereal_D4_stream       120.       204.
=================================================================
```

These are the same values as given in the EFSA opinion [2], Table 4. However, the authors of the opinion made the error that the FOCUS profiles were given in μg/L whereas the calibration was done on exposure concentrations in μmol/L. I did not make a correction here, to demonstrate that the openGUTS software gives the same values as the authors of the opinion derived. However, it shows how easy it is to make a mistake here.

The LPx values in batch processing are calculated without CIs. This is done to increase speed, and under the assumption that CIs will only be needed when looking at a smaller number of profiles one-by-one. The Matlab version does offer the option to calculate CIs in batch processing, and saves plots in the output directory, in the standard format. Two examples are provided in Figure 16. These examples are not that interesting, as the main pulse exposure event does all of the mortality.

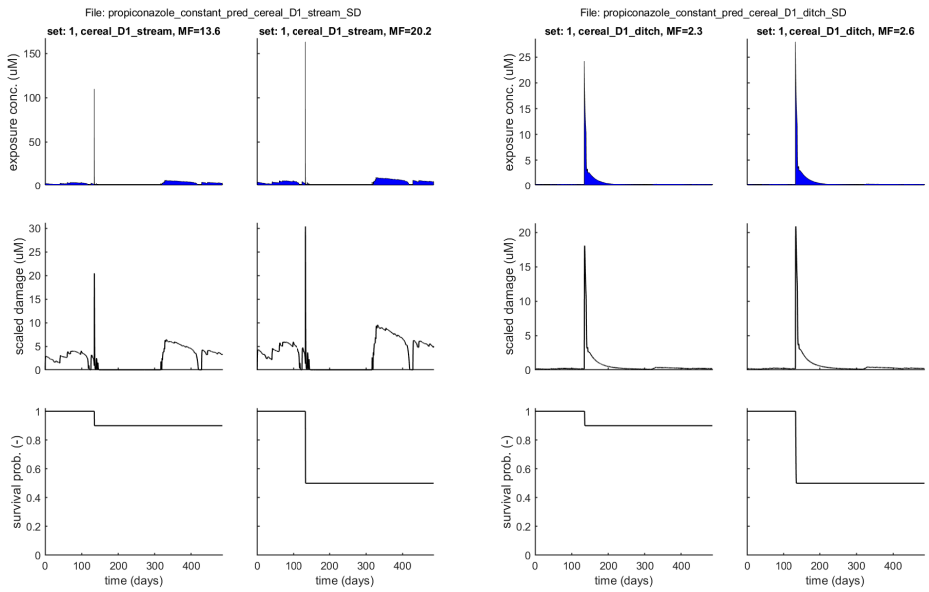

Figure 16: Output plots for two exposure profile (scenario 'cereal D1 stream' and 'cereal D1 pond'). The header of the columns show the multiplication factor applied (and hence specify LP10 and LP50).

In-depth assessment. We can also make detailed calculations for one profile: calculating LPx values with CIs, and making plots as well.

```
Following exposure profiles are loaded for predictions:
    set: 1, file: apple_R1_pond.txt
Results table for LPx (-) with 95% CI: apple_R1_pond
  Special case: stochastic death (SD)
=====================================================================
LPx     best       CI
=====================================================================
LP10:  15.9 ( 14.8- 16.5)
LP50:  16.7 ( 15.7- 17.3)
=====================================================================
Time required for the LPx calculations: 15 mins, 55.8 secs
```

The standard plot is given in Figure 17, but also a kind of dose-response plot is made in Figure 18. The x-axis has the multiplication factor for the profile and the y-axis the survival probability at the end of the profile. Clearly, one can read the LP10 and LP50 from this plot (with their CIs), but also judge how far one is away from a certain trigger value for the LPx.

Extra: test design. In principle, the prediction mode can also be used to design further testing. For example, after calibrating the model for constant exposure, the prediction mode can be used to select useful pulse-exposure treatments for a validation test. This is easier to do in the Matlab version, and is demonstrated in the main script (`BLOCK 7`).

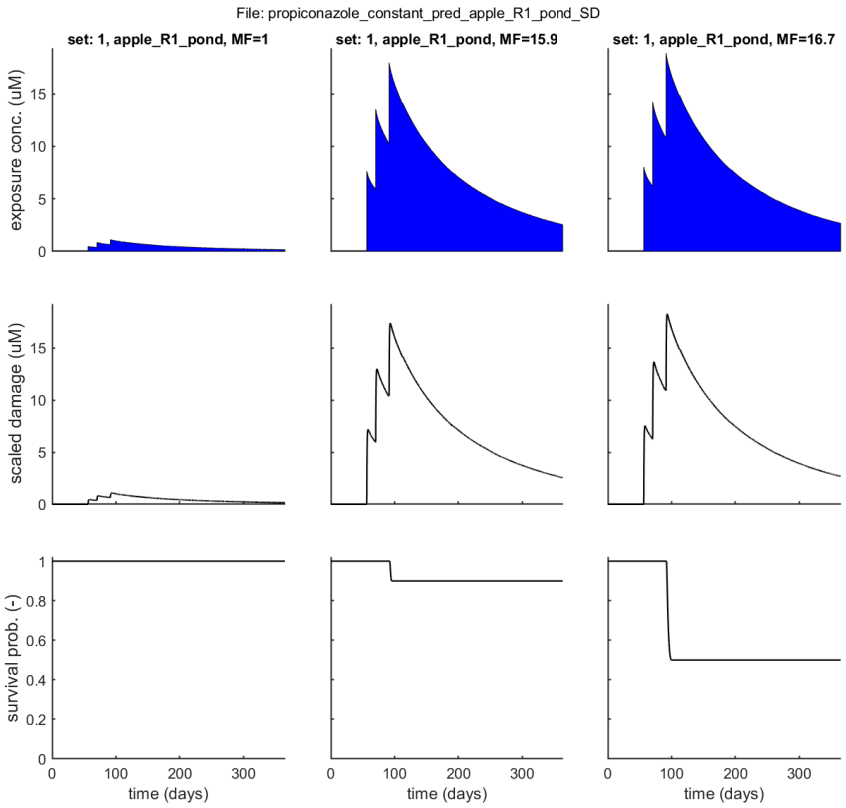

Figure 17: Detailed result plot for one exposure profile (scenario 'apple R1 pond'), including the survival due to the unmodified profile (MF=1) in the left panel.

For example, we can take two pulse scenarios, each with two pulses, with an exposure concentration of 10 μmol/L. In Figure 19, the predictions are shown. I attempted to create carry-over toxicity by taking the close pulses within the DRT95. However, there will be little carry-over toxicity as a 1-day pulse is already close to sufficient for achieving steady-state.

3.6 Conclusions on this data set

The calibration data set for this case study is rather straightforward, in that it allows all parameters to be identified with nicely defined CIs. The validation raises some questions; the performance of the calibrated model is not bad, but not as good as one would like to see. However, it is good to realise that these experiments have been performed with field-collected animals, which were collected and tested at a different time of the year in both experiments. It is therefore possible that the GUTS model parameters are somewhat different in both experiments.

The calibrated model largely overestimated toxicity in the validation test, so it seems to represent a worst-case approach. However, there are indications that prolonged exposure to low concentrations triggers another mechanism of action. This was indicated by

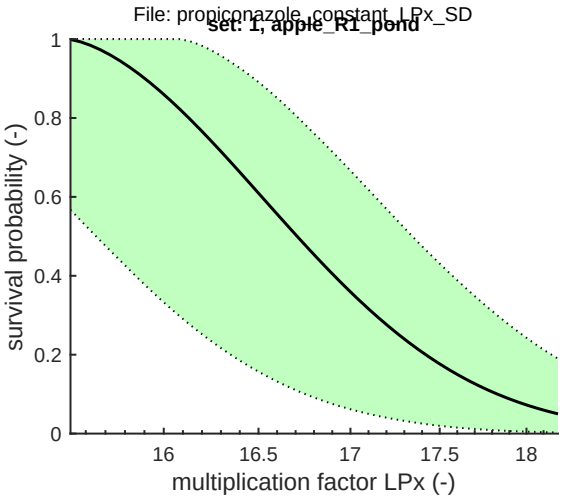

Figure 18: Survival probability versus the multiplication for one exposure profile (scenario 'apple R1 pond'), including 95% CI.

the dose-related mortality in the low doses in Figure 9, and confirmed in the validation experiment with prolonged constant exposure in Figure 13. Is this deviation relevant for a risk assessment? For exposure scenarios with one short dominant exposure pulse, as in Figure 16, such slow effect at low doses are probably not relevant. However, for more prolonged low-level exposure. as in Figure 17, this additional mechanism of action may lead to more mortality than predicted from the calibrated model. Simulations with an extended GUTS model with two mechanisms of action (basically a mixtures model) can shed more light on this, but is beyond the current version of the software. Such simulations could be performed with the BYOM framework in Matlab (`http://www.debtox.info/byom.html`). Also, dedicated experimental work (with low-dose and long exposure) may help to decide whether this slowly developing low-dose effect is a true effect of the chemical.

It is good to note that I here only showed the SD analysis. In general, one would apply both SD and IT to the same data set, and take the worst-case results. The IT model *does* capture the low-dose effects in calibration quite well, even though the overall fit is slightly worse (and it also fails to capture the low-dose effect in the validation experiment). Using both SD and IT provides more reassurance that the toxicity of the chemical is captured, although it will not be sufficient in all cases.

Model analyses with mechanistic models will inevitably provide much more information on the toxic effects than descriptive methods such as dose-response curves and peak or time-weighted average concentrations. That is simply in their nature. In some cases this will lead to more confidence in the assessment, and higher allowable exposure concentrations. In other cases, model analysis may indicate more subtle effects of the chemical that are missed in the first tier, and could lead to lower allowable exposure levels. However, it is good to consider that the classical treatment of exposure profiles (using LC50 and a peak or average concentration) would have completely ignored such subleties, and also would not identify them.

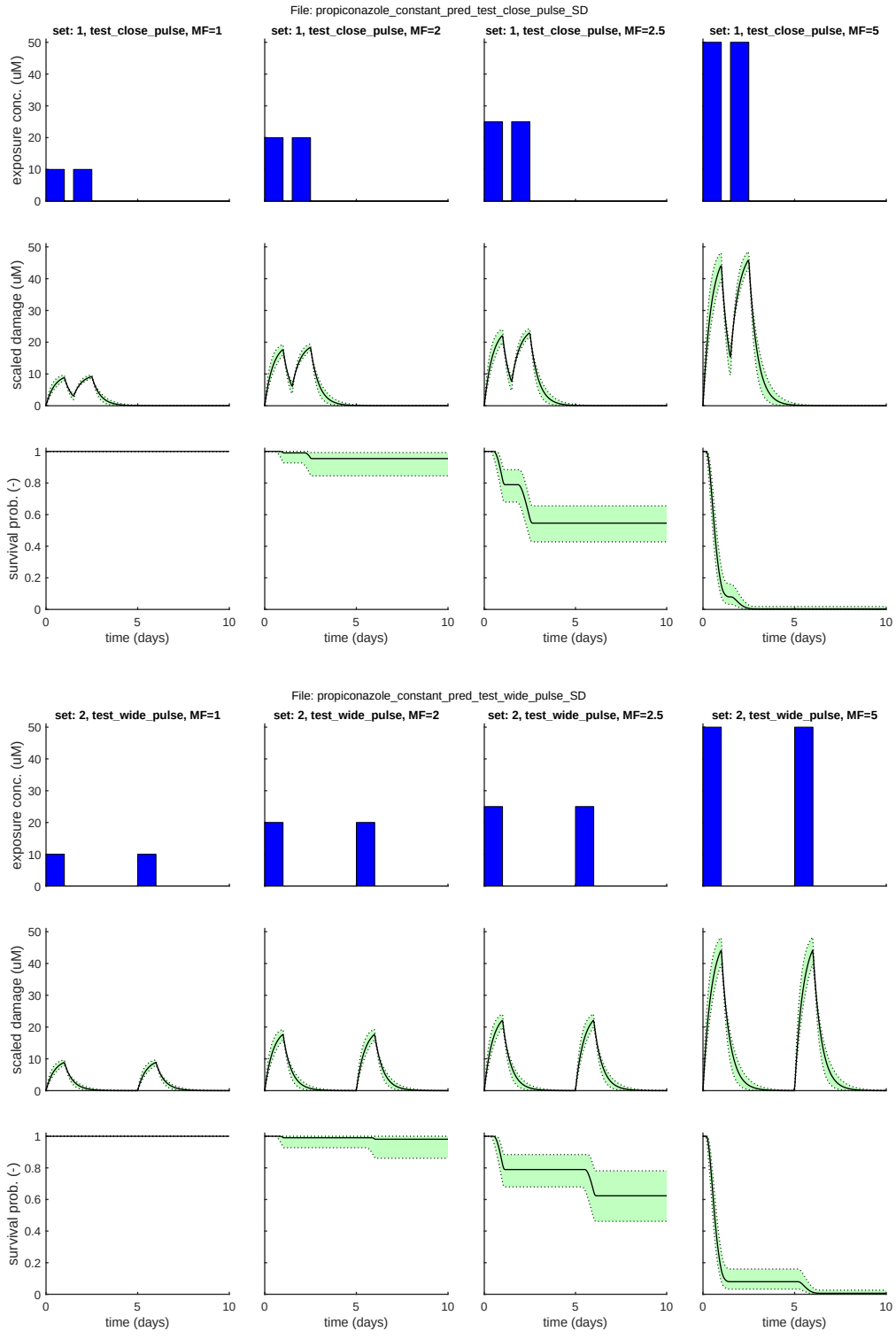

Figure 19: Predicted effects for two pulse scenarios with different intensities. Such calculations can help design toxicity tests.

4 Typical special cases

In this chapter, I will present several data sets that do not lead to the nicely defined cloud in parameter space that we saw in the previous chapter. To recap: in the best case, the profile likelihoods will look like parabola's, and the 2-D projections like (pretty circular) ellipsoids. In many cases, we will see deviations from this best-case situation. Such deviations are tough on the optimisation algorithm (the parameter-space explorer) but should not worry us overly; a lot of effort has gone into development of a robust algorithm that is able to deal with these difficult cases. However, it is good to understand what causes these deviations, what they mean, and when they will influence our model results. Up front, there are a number of points to make:

1. Weirdly-shaped parameter clouds do not preclude useful parameter estimation or predictions (the optimisation algorithm is designed to still retrieve a representative sample). However, they may lead to increased uncertainty in the model predictions, so it is good to consider the CIs and not just the best estimate. Further, some cases require scrutiny as they may affect predictions under some circumstances.

2. In many (but not all) cases, weirdly-shaped parameter clouds result from data sets that are probably not acceptable for risk assessment anyway (e.g., because there is only one treatment with partial effects). Therefore, weird parameter clouds can be a sign of a lack of information in the data set. However, from a modelling perspective, there is no need to discard them (as long as the CIs are considered in the assessment).

3. Many of these cases result from, or are exacerbated by, the fact that the test was done at constant exposure. The EFSA opinion [2] prescribes validation of the calibrated model with pulse-exposure tests. Together, tests at constant and pulsed exposure provide a wealth of information for parameter estimation. If the two sets of tests are consistent, it makes scientific sense to combine them for making predictions. This will deliver model predictions with the greatest degree of accuracy.

4. Beautifully-shaped parameter-space plots do not mean that the fit is any good. And, *vice versa*, awful parameter plots do not mean that the fit will be bad. Both the parameter-space plot and the model fit to the data need to be considered in judging the quality of the calibration.

It should be noted that considering CIs and combining data sets is, strictly speaking, not in line with the proposed work flow of the current EFSA opinion [2].

4.1 General: hitting bounds

Each parameter has a minimum-maximum range. In general, we like to see parameter clouds that are well-defined and well away from these boundaries. However, it is possible that the outer parameter cloud (blue points) runs into one of these boundaries. It may also be that the inner cloud (green points) run into a boundary, which implies that there are consequences for the confidence interval of the parameter (it is indicated by an asterisk in the output for the CIs to screen). When the parameter cloud (and/or the CI) run into a bound, this is not generally a problem for the model analysis or for the model predictions. However, in each case, some further scrutiny of the model output will be needed. First, it has to be established whether this behaviour has occurred with the standard settings or with user-modified settings (in the latter case, there has to be a very good reason to deviate from the defaults).[12]

Dominant rate constant k_d. When the parameter cloud runs into the lower boundary of parameter space, this is called 'slow kinetics'. This case is dealt with in detail in Section 4.4. When the cloud runs into the upper boundary, this is called 'fast kinetics', which is dealt with in Section 4.3.

Threshold parameter m_w. Running into the lower boundary could signify 'slow kinetics'. Check whether k_d is also going to low values and whether there is strong correlation between m_w and k_d (if there is, see Section 4.4, although generally, k_d will hit its bound first). If this is not the case, the threshold is just very low (and possibly not significantly different from zero). The openGUTS software does not allow the threshold to be set to zero, but it can be tried whether lowering the lower bound of m_w leads to a better fit. In general, it should not, and the parameter cloud can be used for model predictions.

Running into the upper boundary is not expected. The parameter cloud may be extremely close to that bound under 'single-dose runaway' for SD (when there are only effects at the highest dose), which is discussed in Section 4.5. Furthermore, it could happen under IT, when the highest treatment in the test produces only little effect. In the latter case, the upper boundary can be set higher, although it should be questioned whether the data has sufficient information for the purpose that it is used for.

Background hazard rate h_b. Running into the lower boundary signifies that there is no control mortality. With the default settings, setting the lower boundary to even lower values will not change the fit. Therefore, the results can be used as is.

Running into the upper boundary implies that the data set suggests very high control mortality. This could obviously be caused by high observed mortality in the control (which places question marks on the validity of the test), or because a poor fit in the treatments is compensated for by increasing the background mortality to unrealistic levels. The latter case should be avoided, and it is better to fix the background hazard rate to the controls (although this will likely lead to an unconvincing model fit).

[12] Please note that the standalone version of openGUTS does *not* plot the bounds in the parameter-space plots. The Matlab version shows them as thick blue lines.

Killing rate b_w. Running into the lower boundary is not expected. That would signify that exposure above the threshold will lead to minimal effects, relative to the control. If the default setting of the boundaries indeed are limiting, it may be opportune to lower the minimum bound to see if a better fit results.

Running into the upper boundary could signify 'slow kinetics', which is treated in Section 4.4 (although generally, k_d will hit its bound first), or 'single-dose runaway', which is treated in Section 4.5. In all cases, it is unlikely that a better fit will result by increasing the upper boundary.

Spread of threshold distribution F_s. Running into the lower boundary means that there is basically no difference in sensitivity between individuals: when the damage level exceeds the threshold, all animals will die immediately. This is (at least theoretically) possible, so running into this boundary does not cause any problems.

Running into the upper boundary means that there are large differences in sensitivity among individuals. This could happen, and it is a good idea to increase the maximum bound to make sure that the inner rim (green points) does not hit the bound anymore. For standardised laboratory test, such large difference are not expected and should trigger further scrutiny into the data set and the goodness-of-fit of the model.

4.2 Islands in parameter space

Data set. The data set here is for dieldrin in guppies. This set was used in the original publication of the hazard model for survival by Bedaux and Kooijman [1], and was also used as case study in the GUTS e-book [4]. It is a rather extensive data set with 7 treatments (plus a control), 7 days of observations, and 20 individuals per treatment. This data set is available from the example files for the openGUTS software.

Analysis type. The SD model is applied, fitting background hazard along with the other parameters (hence, 4 parameters are fitted).

Parameter space. The plot of parameter space (Fig. 20) reveals a small 'island' in parameter space (isolated green dots separate from the main cloud of green points). This is reflected in the profile for m_w as a local minimum (an extra 'stalactite' in the plot, left of the best fit), and a small drop in the profile for h_b close to the value of zero. Also, the estimate for h_b runs into its lower boundary (basically no background mortality), which should not concern us.

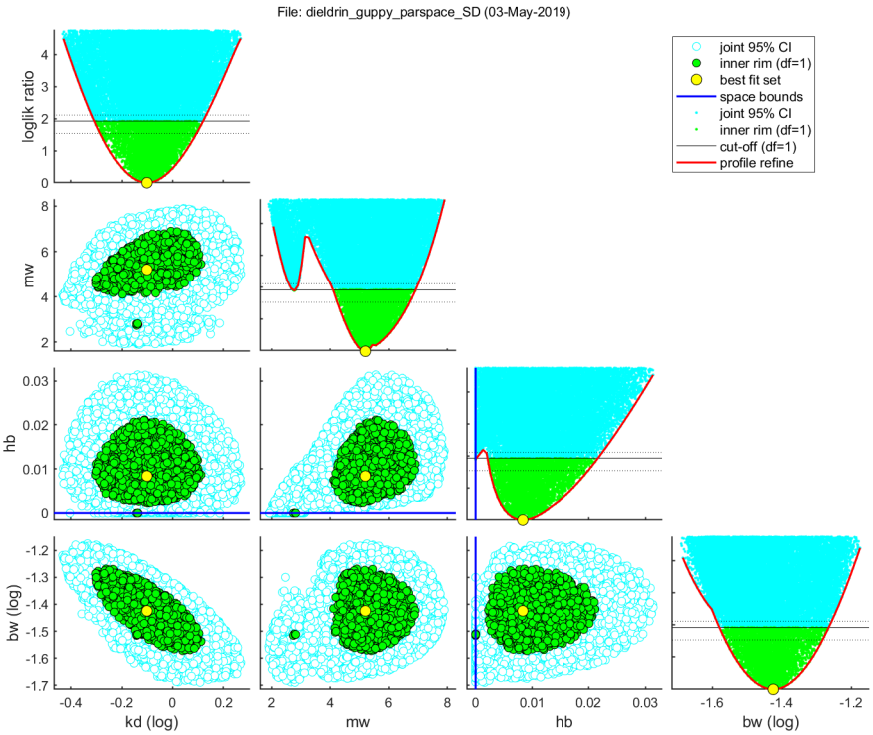

Figure 20: Final plot of parameter space for dieldrin in guppies, using the SD model and fitting h_b.

Cause of this behaviour. This behaviour results from the fact that the data set allows two explanations for the mortality at lower doses. The best fit treats that mortality as

background (high values for h_b and m_w). This fit is shown in Figure 21. However, an alternative explanation is that this mortality is actually caused by the chemical (h_b close to zero, and low value for m_w). This latter explanation provides a poorer fit than the first one, but is not significantly worse.

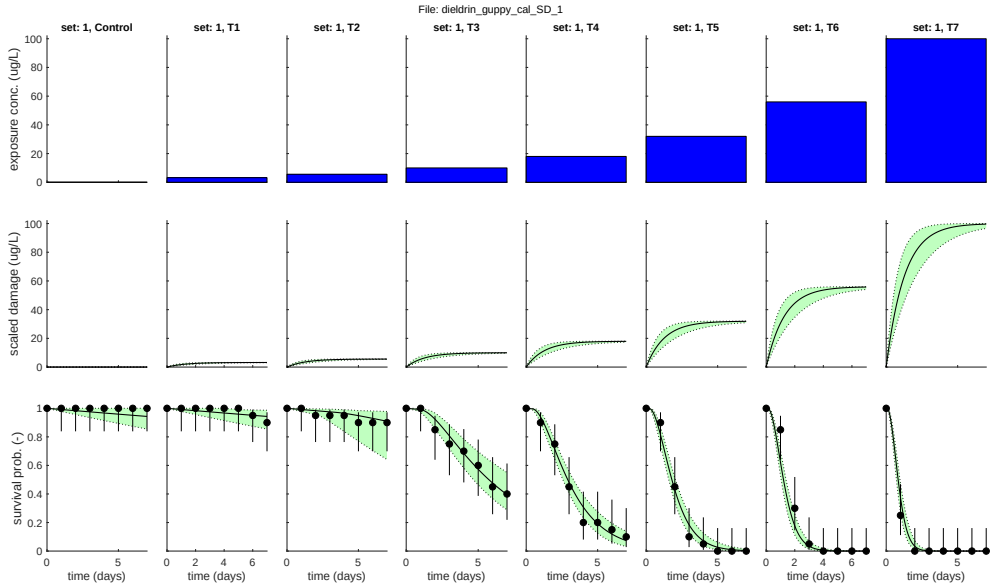

Figure 21: Model fit for dieldrin in guppies, using the SD model and fitting h_b.

Consequences. The CIs for m_w and h_b are broken sets, rather than a single continuous interval. The model will report the total interval, i.e., the widest continuous interval that catches all confidence sets. Under the heading "Extra information from the fit", the screen output will also report:

```
Confidence interval for parameter mw is a broken set.
Confidence interval for parameter hb is a broken set.
```

The fact that there can be islands of confidence, rather than continuous ellipsoids, has been one of the reasons for using a rather elaborate algorithm for optimisation and scanning of parameter space. The occurrence of such islands is not a problem for the parameter estimates or the subsequent analyses; the optimisation is completely acceptable. The sample from parameter space that is used for uncertainty propagation will contain these islands (the parameter sets within the dotted horizontal lines are used for propagation), and hence the model predictions will represent the complete uncertainty in the parameter values. The uncertainty in the model predictions will likely also contain islands. However, these are not shown. For example, the green uncertainty bands in Figure 21 are single bands, whereas they in reality might actually be two bands per model curve. The CIs on model predictions thus cover the maximum width of the actual CI without showing any islands in there.

For some exposure scenarios (especially those with low prolonged exposure), the small island at low m_w might produce more mortality than the large cloud at high m_w. The lower part of the CI on LCx and LPx values might thus represent this island. If the lower part of the CI causes concern, it is possible to provide more insight by setting up an additional experiment with low-dose prolonged constant exposure. This again stresses the importance of considering the CIs as well.

4.3 Fast kinetics

Data set. The data set here is for methomyl in fathead minnows, and is part of the fathead minnow database (see [6]). It is a 4-day test, but additional measurements are available within the first day of the test (after 1 and 2 hours). There are 5 treatments (plus a control), and 20 animals per treatment. This data set is available from the example files for the openGUTS software.

Analysis type. The SD model is applied, fitting background hazard along with the other parameters (hence, 4 parameters are fitted).

Parameter space. The plot of parameter space (Fig. 22) shows more-or-less parabolic profiles for m_w, h_b and b_w, but not for k_d. The dominant rate constant runs into its upper bound (blue vertical line, plotted in the Matlab version only). It is also clear from the 2-D plots of the parameter cloud (first column, below the profile for k_d) that the sample (including the best fit) is squashed into the upper boundary. This is also flagged in the output to screen:

```
kd    best:      1105. (     270.5  -     1105.*) 1/d       fit: 1 (log)
...
* edge of 95% parameter CI has run into a boundary
```

For h_b, the sample is running into its upper boundary. However, this only concerns the blue points, and not the green ones (only the latter ones are used for uncertainty propagation).

Cause of this behaviour. This behaviour results from the fact that the scaled damage dynamics are becoming very fast in the optimisation. This implies that the data suggest that steady state between the external concentration and the damage level is reached very rapidly, and might as well be instantaneous. This causes a problem in optimisation: if $k_d = 1000$ d^{-1} fits well, $k_d = 1 \cdot 10^6$ will also fit well, and $k_d = \infty$ as well. In fact, there is no natural upper limit to the value of k_d, and, without setting some boundary, the optimisation routine would run away to infinity.

Consequences. We cannot obtain an estimate for the upper CI of k_d, which might as well be infinite. However, as long as we have found the best estimate (that we know the lowest value for the minus-log-likelihood), the lower boundary will be representative. The profile for k_d in Figure 23 suggests that the profile can decrease further if we increase k_d beyond the current boundary. It is possible to manually extend the search range for k_d, which might yield a slightly better optimum (though hardly any changes to the visual correspondence to the data), which will shrink the CIs of all parameters a bit (if the optimum improves, some parameter sets that are currently within the CI will then be not good enough anymore). The current analysis will thus be slightly worst case in terms of the width of the CIs.

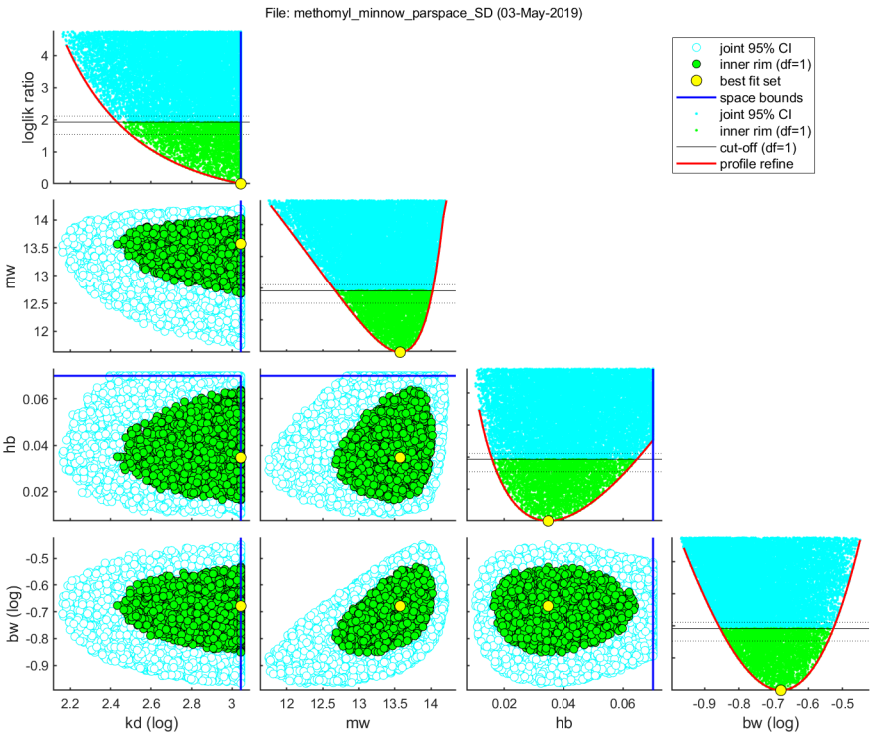

Figure 22: Final plot of parameter space for methomyl in fathead minnows, using the SD model and fitting h_b.

For the predictions, the fact that k_d runs to infinity is not generally a problem. The LCx values will be representative. For the LPx, the estimates will also be fine, although some care will be needed if we analyse an exposure profile with exposure peaks on an extremely short time scale. The default upper boundary for k_d is set such that 95% of steady state will be achieved in half an hour. Since the FOCUS profiles are on an hourly resolution, this should allow the model to respond adequately to short peaks. In this case study, there already is an observation on survival after one hour, which automatically prompts an increase in the search range for k_d: the maximum boundary is set at a point where 99% of steady state is reached at one-tenth of the first observation time. This is done because such early observation times allow the identification of very high values for k_d. In this case, the upper boundary of k_d is so high that we do not need to worry when entering an exposure profile with changes on a very short time scale.

In short: the optimisation is representative and can be used for model predictions. However, the maximum value of k_d has to be judged in relation to the time resolution of the exposure profile. In a general setting (when using FOCUS profiles), the default settings of the search ranges are designed to provide representative results, even when the fit runs into fast kinetics.

In this case study, more concern should be given to the fit (Fig. 23), which is rather poor. The model fit attempts to capture the high-dose effects by increasing background

mortality to rather unrealistic values, and still ends up with a rather poor representation of the effects patterns. There seems to be a rapid onset of mortality followed by a slower progression (and even a stop of further mortality at some point in the highest two doses). The overall fit is not very convincing.

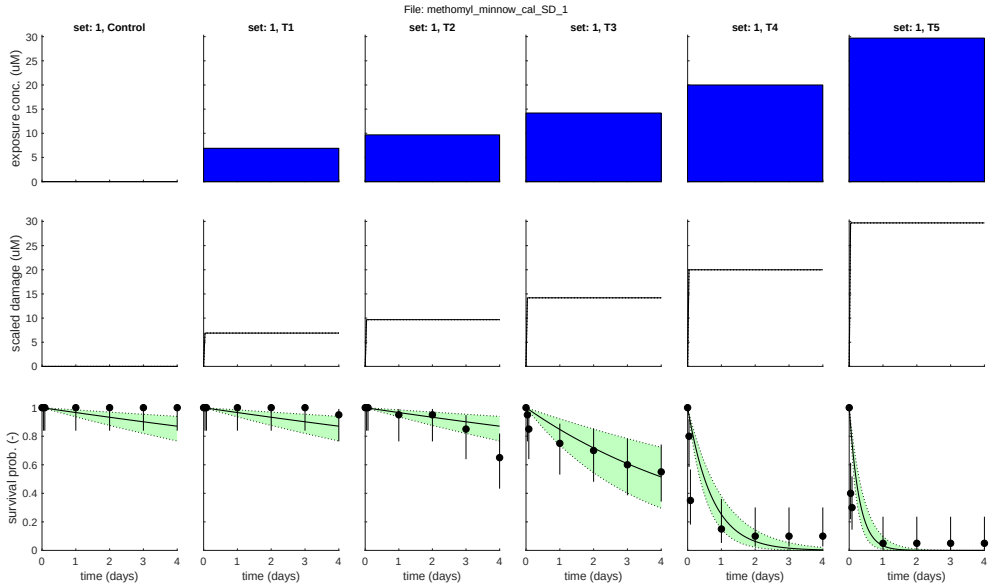

Figure 23: Model fit for methomyl in fathead minnows, using the SD model and fitting h_b.

4.4 Slow kinetics

Data set. The data set here is for fluorophenyl in fathead minnows, and is part of the fathead minnow database (see [6]). It is a 4-day test, with 5 treatments (plus a control), and 10 animals per treatment. This data set is available from the example files for the openGUTS software.

Analysis type. The SD model is applied, fitting background hazard along with the other parameters (hence, 4 parameters are fitted).

Parameter space. The plot of parameter space (Fig. 24) shows some interesting patterns. Most strikingly, strong correlations between k_d, m_w and b_w, with the green points for k_d running into the lower boundary (hence slow kinetics). For m_w, we see a rather sudden switch in the profile at some low value, and for b_w something similar at a high value. The background hazard rate h_b runs into its lower boundary (basically no background mortality), which should not concern us.

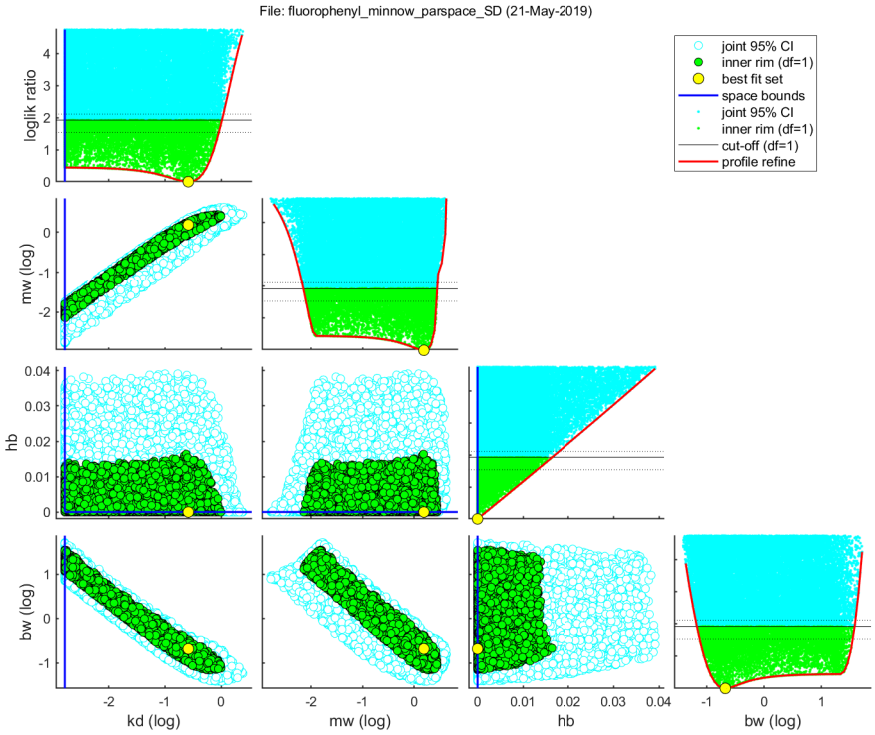

Figure 24: Final plot of parameter space for fluorphenyl in fathead minnows, using the SD model and fitting h_b.

Slow kinetics is hard on the optimisation routine. The threshold m_w is, by default, estimated on normal scale, which is efficient in most cases. However, as a low k_d will drag m_w to very low values, a log-scale is needed. The optimisation routine will recognise that

the parameter space is moving towards slow kinetics, will stop the analysis, and restart with m_w on log-scale.[13] The output to screen looks like this:

```
Following data sets are loaded for calibration:
    set: 1, file: slow kinetics fluorophenyl.txt

Settings for parameter search ranges:
=====================================================================
kd    bounds:   0.001641 -      143.8 1/d      fit: 1 (log)
mw    bounds:  0.0004909 -      8.306 uM       fit: 1 (norm)
hb    bounds:      1e-06 -       0.07 1/d      fit: 1 (norm)
bw    bounds:   0.003139 -  3.986e+04 1/(uM d) fit: 1 (log)
Fs    bounds:          1 -          1 [-]      fit: 0 (norm)
=====================================================================
Special case: stochastic death (SD)

Starting round 1 with initial grid of 16128 parameter sets
  Status: best fit so far is (minloglik) 38.0953
Starting round 2, refining a selection of 1871 parameter sets, with 37 tries each
  Status: 486 sets within total CI and 54 within inner. Best fit: 37.7386

Slow kinetics indicated: parameter-space explorer is restarting with these settings:
=====================================================================
kd    bounds:   0.001641 -      18.11 1/d      fit: 1 (log)
mw    bounds:  0.0004909 -      8.306 uM       fit: 1 (log)
hb    bounds:      1e-06 -       0.07 1/d      fit: 1 (norm)
bw    bounds:    0.01369 -  3.986e+04 1/(uM d) fit: 1 (log)
Fs    bounds:          1 -          1 [-]      fit: 0 (norm)
=====================================================================
Special case: stochastic death (SD)

Starting round 1 with initial grid of 16128 parameter sets
...
```

Cause of this behaviour. This behaviour results from the fact that k_d becomes very low in the optimisation. Because of the scaling of damage in GUTS, this causes a series of problems. A low value for k_d implies very slow increase in damage over time, in fact, the damage level will increase almost linearly in time (see also the damage plots in Fig. 25). This implies that the scaled damage levels are very low (compared to the external concentrations) for most of the test duration. As a consequence, the threshold m_w must be very low, and the killing rate b_w very high, to get effects on survival. The lower k_d becomes, the lower m_w and the higher b_w. Therefore, these strong correlations are inherent in the model when low values of k_d are consistent with the survival patterns in the data set.

With all these parameters running away to zero or infinity, one of them is going to hit its boundary first. In most cases, that will be k_d (as it is in this case). When k_d hits its lower boundary, lower values of m_w will produce a bad fit as the fit cannot be compensated by lower values of k_d anymore. Hence the switch in the profile for m_w around a value of -2

[13] In the output report of the standalone software, the search ranges are reported as they were set *after* the restart.

(on log scale). The same situation occurs for b_w at a high value (around 1.5). For k_d, the model output will show that this parameter runs into its lower boundary:

```
========================================================================
kd    best:     0.2606 (  0.001641* -     1.035 ) 1/d       fit: 1 (log)
mw    best:      1.536 (  0.007052  -     2.789 ) uM        fit: 1 (log)
hb    best:  1.000e-06 ( 1.000e-06* -   0.01641 ) 1/d       fit: 1 (norm)
bw    best:     0.2088 (   0.06640  -     38.28 ) 1/(uM d)  fit: 1 (log)
Fs    best:      1.000 (       NaN  -       NaN ) [-]       fit: 0 (norm)
========================================================================
* edge of 95% parameter CI has run into a boundary
  (this may also have affected CIs of other parameters)
```

Consequences. What does this mean for the CIs on the model parameters? For k_d, it means that the CI should be interpreted as half-open (the lower boundary is zero, or minus infinity on log-scale). The last sentence of the output to screen is crucial here: "this may also have affected CIs of other parameters." In this case, because of the strong correlations, the lower CI of m_w and the upper CI of b_w are affected. Even though the model will report nicely defined values here, they should be interpreted as half-open intervals as well. Therefore, the CIs should be summarised as $0 < k_d < 1.035$, $0 < m_w < 2.805$ and $0.06643 < b_w < \infty$.

The fit to the data set does not look particularly troublesome (Fig. 25), which means that we can use this data set for model predictions. The fact that three of the model parameters are running away, and are unbounded at one end of their CI, should not concern us: due to the strong correlations, the model behaviour will be very similar for all parameter sets with low k_d values. The uncertainty, and the strong correlations, are part of the parameter cloud (the green points), and will be propagated to the model predictions. Note that the CIs on the predicted scaled damage are very wide. This is a consequence of the strong correlations between the parameters, and is of no concern (the CIs on predicted survival are more relevant, and they are quite reasonable).

In combination with exposure profiles that have long periods of (low) exposure over the year, or many peaks, slow kinetics will lead to low values for the LPx. The reason is that there is very little elimination/damage repair and all exposure events will basically add up. In such situations, a GUTS analysis can easily lead to lower allowable exposure levels than a 4-day LC50 combined with a peak concentration. In this particular example, the best estimate for k_d is still well defined even though the lower bound of the CI is not. In this case, $k_d = 0$ would yield only a marginally worse fit. Therefore, it makes sense to carefully consider the CIs on the model predictions, as they may include very low values for the LCx and LPx. In this case, for the monitoring profile as used in the GUTS ring test, the CIs on the LPx values span a range of more than a factor of 20:

```
Results table for LPx (-) with 95% CI: profile_monit
  Special case: stochastic death (SD)
=====================================================================
LPx     best       CI
=====================================================================
LP10: 2.15e+04 (1.28e+03 - 2.95e+04)
```

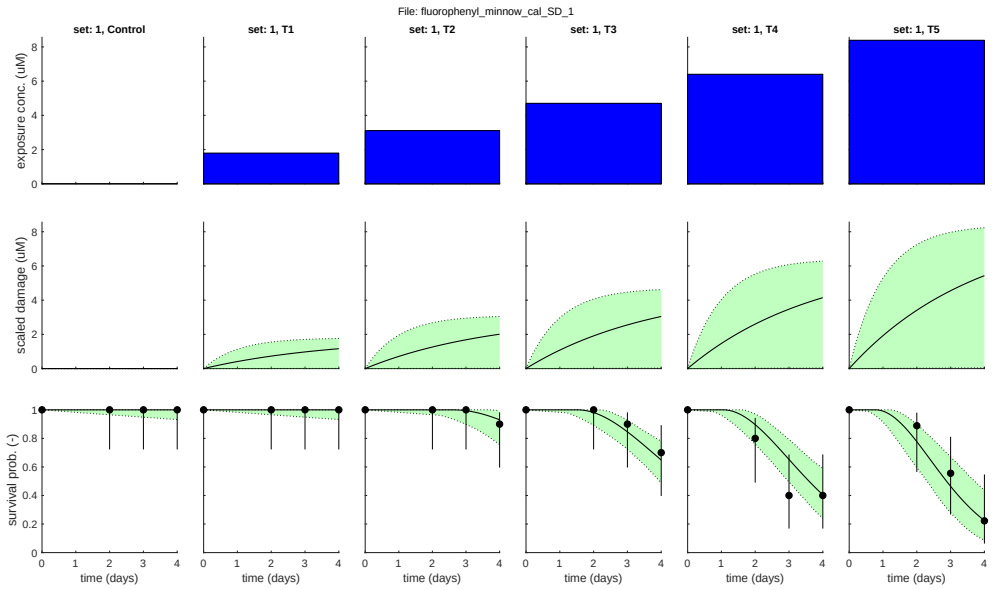

Figure 25: Model fit for fluorphenyl in fathead minnows, using the SD model and fitting h_b.

```
LP50: 2.77e+04 (1.64e+03 - 4.31e+04)
=====================================================================
```

Figure 26 shows the predictions for LPx with the corresponding survival plots. In this example, the CIs on the model results are plotted as well (which is not possible in the standalone version). Clearly, at the LP10, the uncertainty is so high that survival probability could easily decline to almost zero. This is also reflected in the large CIs for the LPx given above (and will also be clear from the plot of survival versus multiplication factor as in Fig. 18). The possibility of low LPx values, and low survival at the LP10, relates to the possibility of very slow kinetics. If slow kinetics can be ruled out (e.g., by information from pulsed-exposure tests), the CIs will be much decreased. Slow kinetics is associated with low values for damage (hence the large green area under the solid line in the damage plot of Fig. 26). However, it is also associated with low values for m_w and high values for b_w, the net result being that toxic effects can become quite severe when there is prolonged exposure.

The current lower value for the search range of k_d is defined, rather arbitrarily, as the rate constant that will lead to 95% of steady state after 5 years of constant exposure. This is pretty slow, probably biologically unrealistic (as it would be hard to imagine a toxicant for which repair or elimination is completely impossible), and should not affect extrapolation to 485-day FOCUS profiles. However, for extremely long-lived species, and extrapolation to much longer exposure profiles, the lower boundary of k_d may need to be reconsidered. When strong correlations, typical of slow kinetic, are seen, but either m_w or b_w is hitting its boundary before k_d, it makes sense to modify the boundaries of m_w and/or b_w to make sure that k_d will hit its lower boundary. That ensures that the interpretation

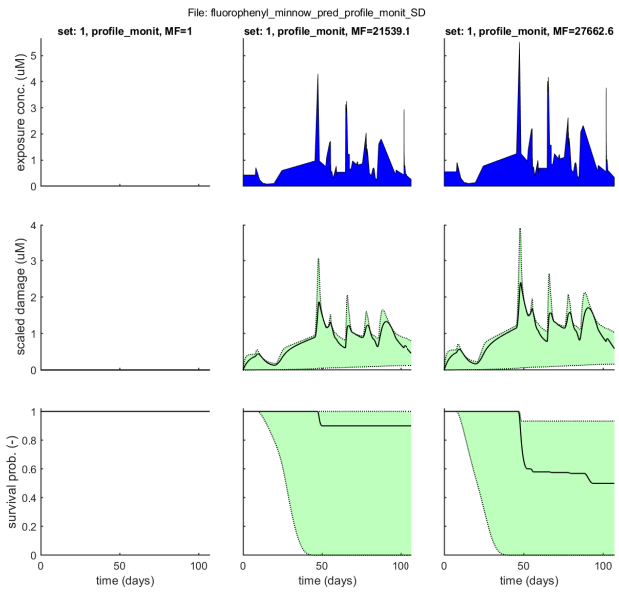

Figure 26: Model predictions for fluorphenyl in fathead minnows, using the SD model and fitting h_b. The wide CIs, stretching to low values for damage and survival, are the result of the possibility for slow kinetics.

of "95% of steady state after 5 years" still applies.

Slow kinetics is the most troublesome case to deal with in GUTS analysis, so it is important for users of (results from) openGUTS to recognise this case and to consider it with a bit of extra care.

Even though the example above deals with SD, it is good to note that slow kinetics also occurs for IT (in fact, it will be more common than for SD); an example is shown in the GUTS e-book [4] for the dieldrin-guppy case study. Difference is that IT does not have b_w as model parameter, and that F_s will not be correlated to k_d. The only striking correlation will thus be between k_d and m_w.

When slow kinetics is missed by the optimisation algorithm. The optimisation algorithm of openGUTS will check for slow kinetics during the main rounds of sampling. If slow kinetics is detected, it will restart the optimisation with m_w on log-scale. However, there are cases when this check fails; an example is shown is Figure 27. There is a clear correlation between b_w and k_d, indicative of slow kinetics. The correlation between m_w and k_d is obscured since the latter is on log scale while the former is on normal scale (same for the correlation between m_w and b_w). In this case, the optimisation still produces the correct best fit and provides a reasonable coverage of parameter space. Nevertheless, it would be best to repeat the optimisation with m_w set manually to log scale. Note that you can here spot problems with the sampling in the parameter-space plots as there is a distinct unevenness in the points in the profiles.

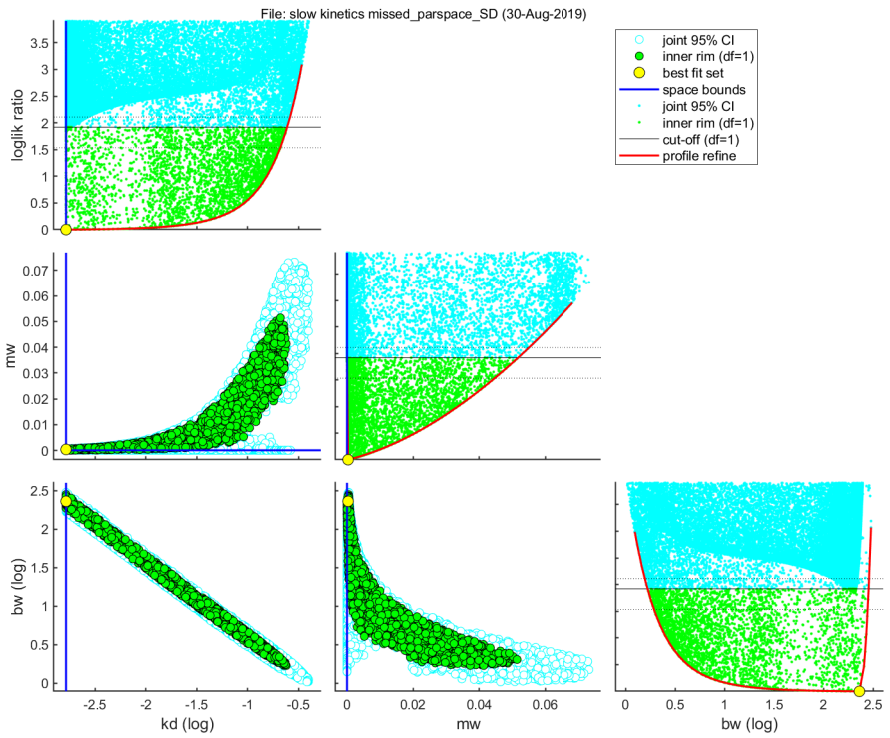

Figure 27: Plot of parameter space for a data set where slow kinetics was not identified by the optimisation algorithm.

4.5 Single-dose runaway

Data set. The data set here is for flucythrinate in fathead minnows, and is part of the fathead minnow database (see [6]). It is a 4-day test, with 5 treatments (plus a control), and 20 animals per treatment. This data set is available from the example files for the openGUTS software.

Analysis type. The SD model is applied, fitting background hazard along with the other parameters (hence, 4 parameters are fitted).

Parameter space. The plot of parameter space (Fig. 28) shows several nasty features. The 2-D projections of the parameter cloud are nothing like ellipsoids; they are ragged, and several have a sort of tube attached. Furthermore, the profile plots clearly show that sampling has been poor in the areas where k_d and b_w are high.

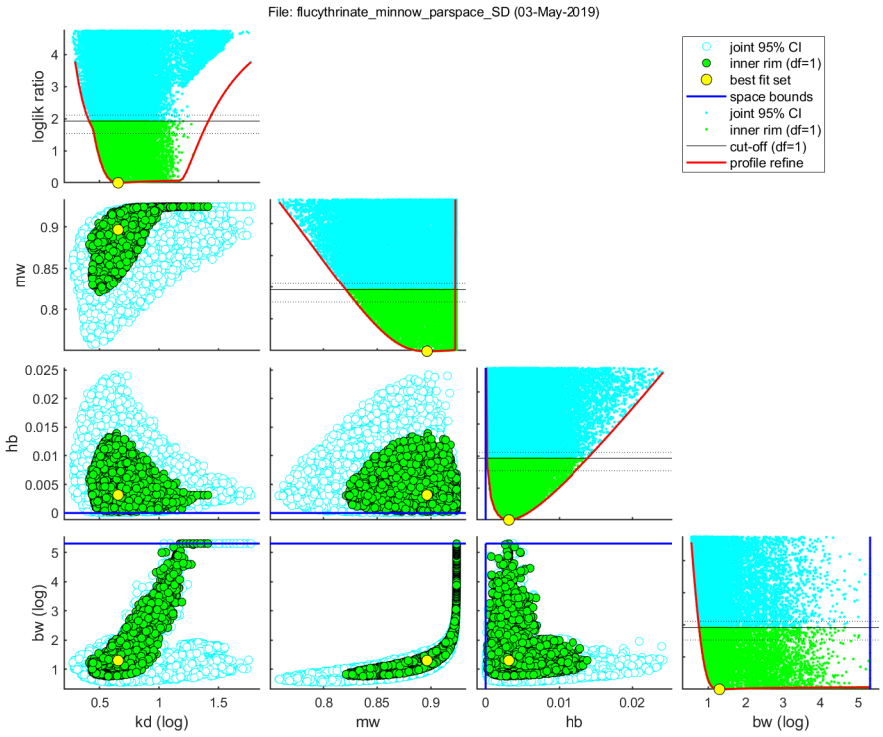

Figure 28: Final plot of parameter space for flucythrinate in fathead minnows, using the SD model and fitting h_b.

This behaviour is particularly tough on the optimisation routine. It has a lot of trouble reaching an acceptable sample, and triggers several rounds of additional sampling and profiling. Even after several rounds of extra sampling, it fails to fill the open areas in the profile plots in Figure 28, and also warns about that at the end of the optimisation process:

```
Starting round 6, creating the profile likelihoods for each parameter
```

```
  Extending profile for mw to higher parameter values
  Extending profile for hb to lower parameter values
  Extending profile for bw to higher parameter values
  Finished profiling, running a simplex optimisation on the best fit set found ...
  Status: 20546 sets within total CI and 5533 within inner. Best fit: 37.0639
  Profiling has detected gaps between profile and sample, which requires extra sampling rounds.
Starting round 7, (extra 1) refining a selection of 311 parameter sets, with 112 tries each
  Status: 23763 sets within total CI and 6914 within inner. Best fit: 37.0639
Starting round 8, (extra 2) refining a selection of 464 parameter sets, with 75 tries each
  Status: 25176 sets within total CI and 7429 within inner. Best fit: 37.0639
Starting round 9, (extra 3) refining a selection of 390 parameter sets, with 89 tries each
  Status: 26539 sets within total CI and 7956 within inner. Best fit: 37.0639
Starting round 10, (extra 4) refining a selection of 324 parameter sets, with 108 tries each
  Status: 27698 sets within total CI and 8340 within inner. Best fit: 37.0639
Starting round 11, (extra 5) refining a selection of 282 parameter sets, with 124 tries each
  Status: 29005 sets within total CI and 8773 within inner. Best fit: 37.0639
Exiting parameter space explorer, but still 220 sets flagged (too much distance between profile and sample).
  Check plot.
```

Cause of this behaviour. This behaviour results from the fact that there is basically only one treatment (with constant exposure) showing partial effects (Fig. 29). This implies that the threshold m_w can creep up, arbitrarily close, to the concentration in that treatment. The closer m_w gets to the concentration in the treatment, the higher b_w needs to be to produce sufficient mortality (the hazard rate is proportional to the difference between the scaled damage and the threshold, and the scaled damage level will be close to the external concentration). The result is a rather odd correlation between b_w on the one hand, and the distance between m_w and the concentration in the treatment on the other. Proper sampling would perhaps be easier to achieve by using a kind of reverse log-scale for m_w, counting back from the concentration in the treatment with partial effects. However, programming such a sample scheme would be more trouble than it's worth.

The strong correlation between b_w and 'm_w-minus-concentration-T4' produces the tube in the 2-D plot for b_w versus m_w: as m_w runs into the concentration of treatment T4, b_w goes to infinity. The tube in the plot of m_w versus k_d also relates to this, as running m_w close to treatment T4 requires fast kinetics.

Consequences. Due to this behaviour, the upper bound of the CI for b_w cannot be established. Despite the fact that parameter space is rather rough, and not sampled in a very detailed way, especially in the tails of k_d and b_w, the sample is still representative for making model predictions. The parameter sets in the section of parameter space where sampling is poor will all lead to the same model behaviour: no mortality as long as damage is below the threshold, and immediate death when it is above. Therefore, there is really no need for a detailed coverage of this part of parameter space.

The data set, in this case, contains only limited information about the parameter values. As can be seen in the model fit (Fig. 29), there is basically only one treatment with partial effects. This is not necessarily a reason to discard the data set, as the sample will still be representative of the uncertainty. However, it is again a good idea to consider the CIs on the model predictions, as they may be very wide.

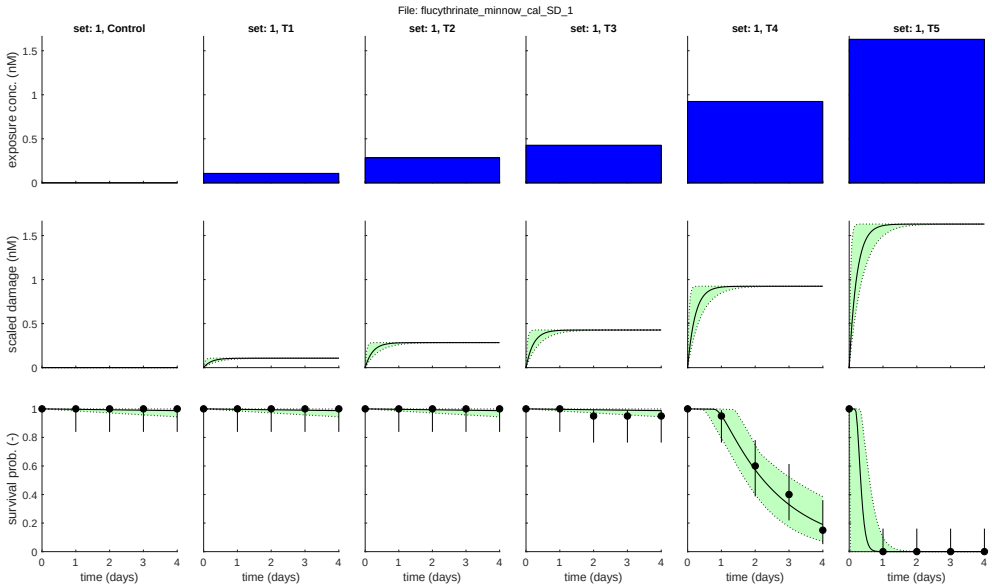

Figure 29: Model fit for flucythrinate in fathead minnows, using the SD model and fitting h_b.

4.6 Either SD or IT fits better

Data set. The data set here is for bromacil in fathead minnows, and is part of the fathead minnow database (see [6]). It is a 4-day test, with 5 treatments (plus a control), and 40 animals per treatment.

Analysis type. Both the SD and IT model are applied, fitting background hazard along with the other parameters (hence, 4 parameters are fitted).

Parameter space. The plots of parameter space are shown in Figures 30 and 31, and do not look very disturbing. For SD, we see fast kinetics ($k_d \to \infty$), but otherwise everything looks fine. More interesting are the fits in Figures 32 and 33. The fit for SD does not look very convincing (though model efficiency is quite reasonable at 0.884), whereas the fit for IT is almost perfect (model efficiency 0.997).

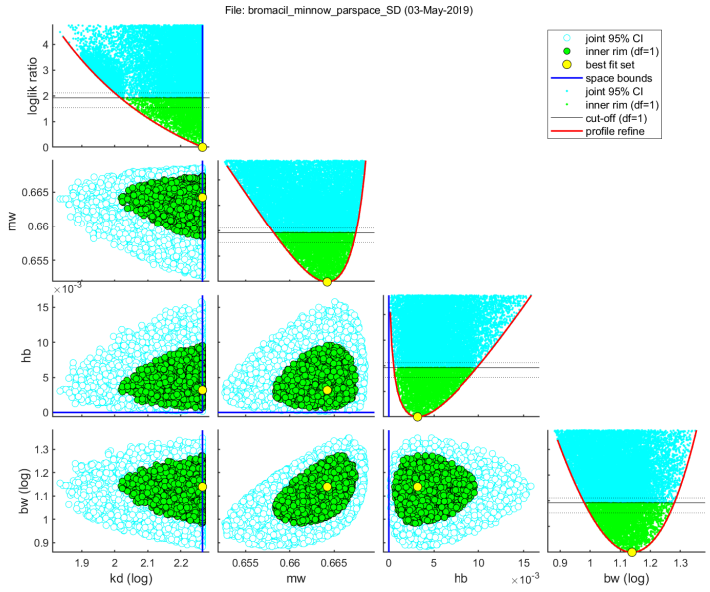

Figure 30: Final plot of parameter space for bromacil in fathead minnows, using the SD model and fitting h_b.

Cause of this behaviour. In this case, IT fits much better because the survival pattern over time shows a specific pattern: a decrease in mortality that stops before killing all individuals. This levelling off of mortality (even though exposure has been constant) is typical for IT, and cannot be matched by SD.

Consequences. In this case, it would be tempting to only use IT for model predictions and discard SD, as it clearly does not capture the patterns in the data set. However, this pattern can also occur when the animals gradually develop resistance to the chemical (e.g.,

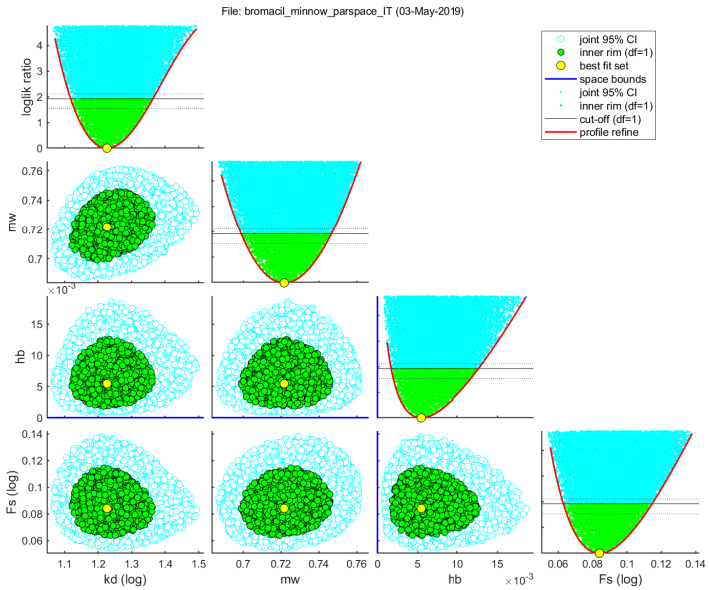

Figure 31: Final plot of parameter space for bromacil in fathead minnows, using the IT model and fitting h_b.

by inducing biotransformation enzymes). It would be impossible to settle this dispute using this data set alone, but the good news is that an IT analysis will generally be representative (possibly even worst case) in this situation (under IT, the surviving individuals will be the tolerant ones). Furthermore, an additional toxicity test with time-varying exposure will likely shed more light on the underlying mechanisms of toxicity.

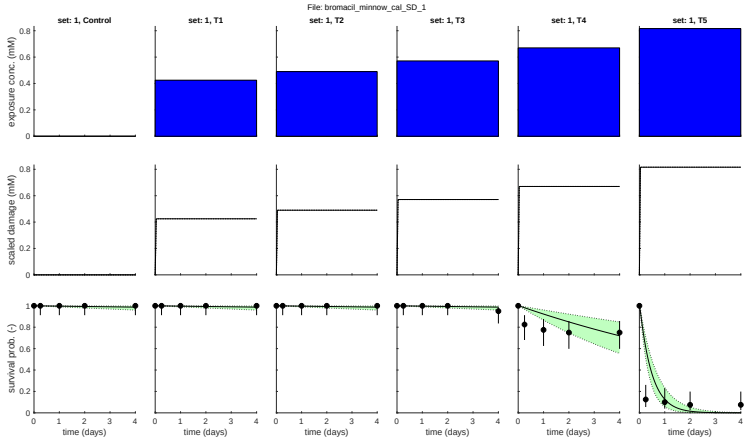

Figure 32: Model fit for bromacil in fathead minnows, using the SD model and fitting h_b.

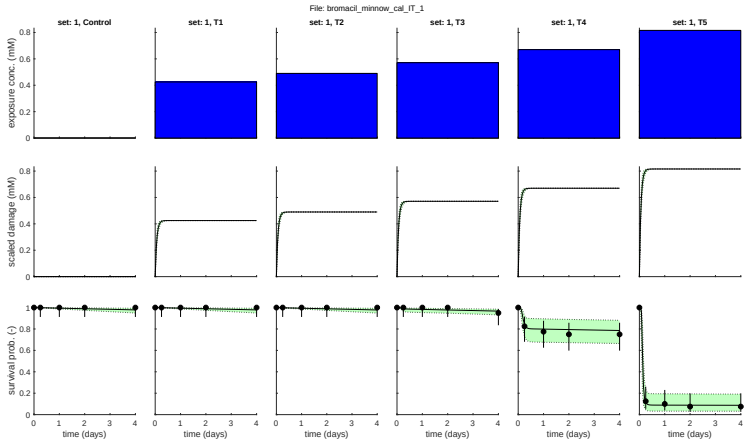

Figure 33: Model fit for bromacil in fathead minnows, using the IT model and fitting h_b.

4.7 The Maltese-cross anomaly

Data set. The data set here is for dieldrin in Guppy as used in [1], and available from the example files for the openGUTS software. The original data set, with a regular analysis, will not show this particular anomaly. However, I will modify things a bit to make it come out. I will use the unmodified data set while fitting only 2 parameters, and a modified one where all survivor numbers are multiplied by 1000 to mimic a situation where uncertainty is very low (and the relevant cloud in parameter space very small).

Analysis type. The IT model is applied for the regular data set, keeping h_b fixed to the value fitted on the controls, and fixing k_d to its best value in an unconstrained fit (hence, 2 parameters are fitted). The SD model is applied to the modified data set (many more individuals per treatment), fitting background hazard on the controls (hence, 3 parameters are fitted).

Parameter space. The plots of parameter space are shown in Figures 34 and 35, showing some weird Maltese-cross-like patterns. Other shapes are possible, but they will all be characterised by bands in the plots, and large chunks of parameter space apparently missing. In the text output from the optimisation, you will see that the regular sampling rounds end with very few accepted parameters (for 3 fitted parameters, the algorithm wants to have at least 5000 parameters in the total cloud, and 2000 in the inner). This trigger lots of additional sampling (from very few points). The output to screen belonging to Figure 35, after the basic sampling rounds (and before profiling) reads:

```
We have now done 12 rounds without reaching the stopping criterion ... we stop here!
  Finished sampling, running a simplex optimisation ...
  Status: 26 sets within total CI and 14 within inner. Best fit: 163416.694
```

Cause of this behaviour. If you look at the axes in Figures 34 and 35, it is clear that the final parameter cloud is very small, relative to the starting cloud. This causes problems in the genetic algorithm of openGUTS. The algorithm is not geared towards such small clouds, as it is optimised for larger (possible oddly-shaped) parameter clouds. For small clouds, it does not contract fast enough in the initial rounds, which implies that all of the 'filling in' of the cloud happens in extra sampling rounds (which are optimised to fill gaps and not large areas). The cause of this behaviour is thus with the genetic algorithm used in openGUTS, and not an inherent problem with the data set.

This behaviour can also occur when attempting to fit parameters like k_d and b_w on normal scale, while their ranges span multiple orders or magnitude. The well-behaved propiconazole data set of Section 3.1 will produce a very nice example of this anomaly for SD when all parameters are fitted on normal scale (hence modifying the default settings for the algorithm). In this case, the pattern is even more extreme, and more flower, or fireworks, shaped (Fig. 36).

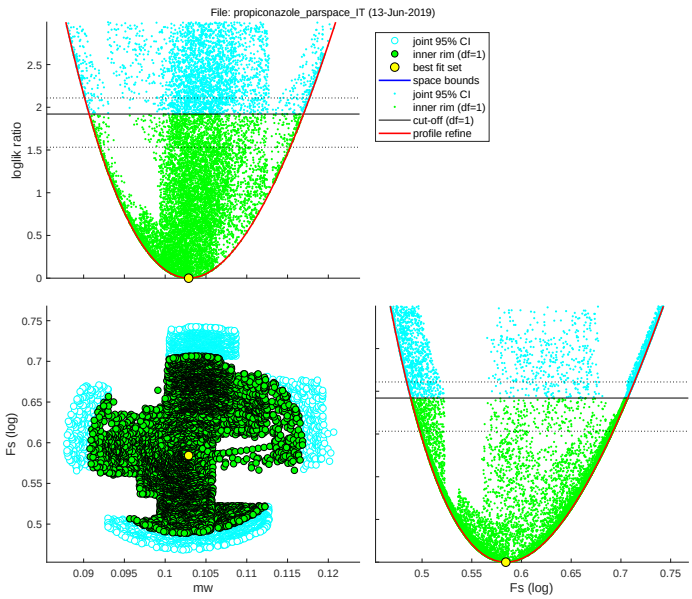

Figure 34: Final plot of parameter space for dieldrin in guppies, using the IT model keeping h_b and k_d fixed to their best value.

Consequences. In this case, the best parameter set can be trusted, as well as the CIs on the parameters resulting from profiling the likelihood function. However, some areas of parameter space will be ill represented in the sample for error propagation. Therefore, the CIs on model predictions could be less reliable (though they would be very narrow anyway). To obtain a more reliable sample, the best way to proceed is to redo the calibration with a smaller initial parameter grid, i.e., tighter min-max ranges on the parameters, which can be based on the information from the previous parameter-space plots.[14]

For the propiconazole case study in Figure 36, the morale of the story is to *not* change the default settings unless you know what you are doing. Even seemingly innocent changes can lead to a poor sample.

For an update of the model, the code could be amended to capture this situation better. However, given that it only occurs in rather extreme and unrealistic cases, and given that there is an effective workaround, this should not receive priority.

Complete failure of the optimisation In the most extreme version of this case, the 'Maltese cross' is not produced but the optimisation routine will fail completely. Generally, this is caused by some property of the data set that allows an artificial 'excellent fit' in a tiny portion of parameter space (and pretty good fits in a larger part elsewhere in parameter space). This tiny part of parameter space may be missed during the main rounds of mutation. When it is then spotted during profiling, the new optimum may be so much better than the old one that none of the parameter sets from the main sample is part of the new confidence set.

[14]This is one of the very few cases where changing the default parameter ranges is acceptable, and actually advisable, to produce a representative sample from parameter space.

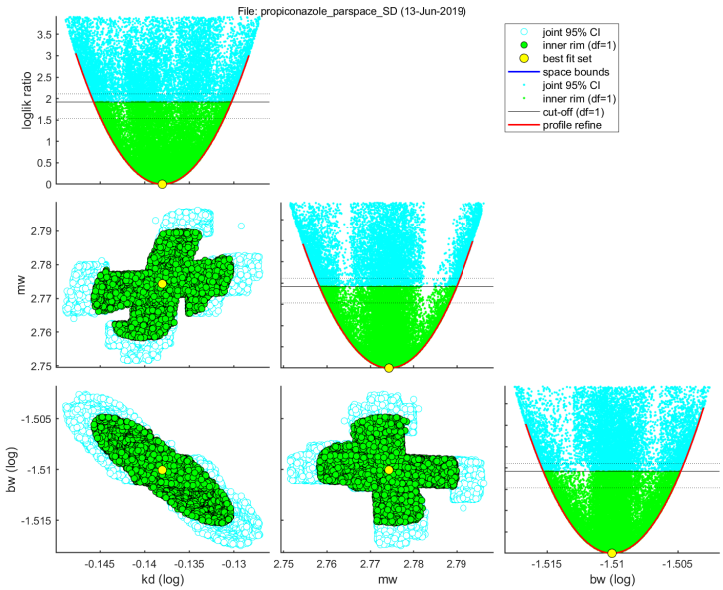

Figure 35: Final plot of parameter space for dieldrin in guppies, using the SD model keeping h_b fixed. Note that the dataset is manipulated by multiplying all survivor numbers by 1000.

When a 'Maltese cross' is produced the main sampling stage also fails to sample the confidence set as it is too small. However, a few parameters sets were found, which could then be used as basis for additional sampling rounds. However, if there is only one acceptable parameter set (the new optimum), there is nothing to base additional sampling on.

These cases are flagged by the software with the advise to seek expert assistance. With a careful selection of the initial parameter ranges, this global optimum can be mapped effectively. However, the 'excellent fit' from this global optimum will probably always be biologically unrealistic; real-world data sets do not yield almost-perfect identification of the model parameters.

It is good to note that this case may also occur as a consequence of improper data entry (e.g., entering time points in hours rather than days) or improper modification of the 'expert settings' (initial parameter ranges and log settings).

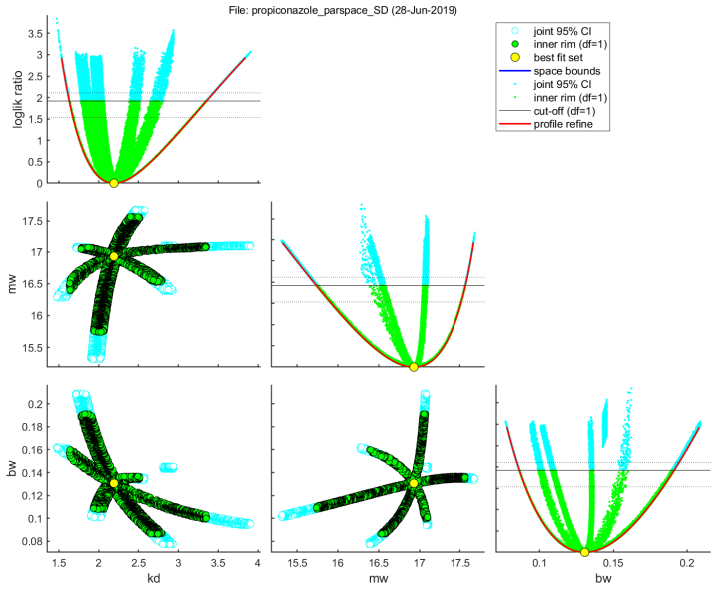

Figure 36: Final plot of parameter space for propiconazole in *Gammarus pulex*, using the SD model keeping h_b fixed, and all parameters fitted on normal scale (as opposed to the default settings where k_d and b_w are on log-scale).

References

[1] J. J. M. Bedaux and S. A. L. M. Kooijman. Statistical analysis of bioassays based on hazard modelling. *Environmental and Ecological Statistics*, 1:303–314, 1994.

[2] EFSA. Scientific opinion on the state of the art of toxicokinetic/toxicodynamic (TKTD) effect models for regulatory risk assessment of pesticides for aquatic organisms. *EFSA journal*, 16(8):5377, 2018.

[3] T. Jager. Reconsidering sufficient and optimal test design in acute toxicity testing. *Ecotoxicology*, 23(1):38–44, 2014.

[4] T. Jager and R. Ashauer. *Modelling survival under chemical stress. A comprehensive guide to the GUTS framework*. Toxicodynamics Ltd., York, UK. Available from Leanpub, https://leanpub.com/guts_book, Version 2.0, 8 December 2018, 2018.

[5] A. M. Nyman, K. Schirmer, and R. Ashauer. Toxicokinetic-toxicodynamic modelling of survival of *Gammarus pulex* in multiple pulse exposures to propiconazole: model assumptions, calibration data requirements and predictive power. *Ecotoxicology*, 21(7):1828–1840, 2012.

[6] C. L. Russom, S. P. Bradbury, S. J. Broderius, D. E. Hammermeister, and R. A. Drummond. Predicting modes of toxic action from chemical structure: acute toxicity in the fathead minnow (*Pimephales promelas*). *Environmental Toxicology and Chemistry*, 16(5):948–967, 1997.