# Exploring Novel Ways of Using Species Sensitivity Distributions to Establish PNECs for Industrial Chemicals

**Final Report to Steering Group**
**Peter Craig**
**April 3 2013**

## Introduction

In all of what follows, we are concerned with the problem of estimating the $HC_5$ (concentration hazardous[1] to 5% of species) for a chemical based on the SSD (species sensitivity distribution) concept and using acute data. Data for the chemical will be a limited number of test results ($LC_{50}$s or appropriate $EC_{50}$s — hereafter all referred to as $EC_{50}$s). An $EC_{50}$ measures the toxicity of a chemical to a particular species or, equivalently, the sensitivity of the species to that chemical.

The intention is that parts I, II and III of this report should be accessible to scientists with an expertise in applying species sensitivity distributions while the technical material in the appendices is aimed at computationally-expert Bayesian statisticians.

[1] Hazardous here actually means that the concentration is greater than or equal to the $EC_{50}$ for a species

**Contents**

# Part I

# Science

## 1 Background: Aldenberg & Jaworska (A&J)

There is a large literature on ecotoxicological risk assessment and in particular on SSDs. See Posthuma et al (2002) for an introduction to the area and Craig et al (2012) for a more statistically focussed account. Aldenberg and Jaworska (2000) is a particularly important article as it provides a statistical model and $HC_5$ calculation which was subsequently made available for general use via the ETX software (ETX 2.0 , 2004) and adopted as part of the REACH guidance (EC , 2006; ECHA , 2008). One way of understanding this project is that it updates the A&J statistical model in several ways to account for features of available data which are inconsistent with their model and then makes the updated model and resulting hazardous concentration calculation available in software.

### 1.1 The Aldenberg and Jaworska $HC_5$ calculation

Everything works on logarithmic scale as is usual for SSDs; base 10 logarithms will be used throughout this document. Denote the $\log$-$EC_{50}$s for $n$ species by $y_1, \ldots, y_n$ and let $\overline{y}$ be the sample mean and $s$ the sample standard deviation.

Then the $\log$-$HC_5$ estimate is

$$\overline{y} - \kappa s/\sqrt{n}$$

where $\kappa$ is the 50th percentile of the non-central $t$-distribution with $n-1$ degrees of freedom and non-centrality parameter $1.645\sqrt{n}$

For upper and lower confidence/credibility limits on the $\log$-$HC_5$ use the appropriate percentiles of the same non-central $t$-distribution. For example, for a lower 10% confidence/credibility limit, take $\kappa$ to be the 90th percentile of the non-central $t$-distribution with $n-1$ degrees of freedom and non-centrality parameter $1.645\sqrt{n}$.

A numerical example:

- Suppose that the $\log$-$EC_{50}$s for $n = 8$ species are: 3.00, 3.15, 3.32, 2.74, 3.78, 0.94, 4.70 and 4.18 (based on $EC_{50}$s measured in $\mu$g/l).
- Then $\overline{y} = 3.23$ and $s = 1.13$.
- The values of $\kappa$ corresponding to the central estimate and upper and lower 10% limits are respectively 4.86, 3.10 and 7.79.
- The $\log$-$HC_5$ estimate is $3.23 - 4.86 \times 1.13/\sqrt{8} = 1.29$ (10% lower and upper limits 0.12 and 1.99). The $HC_5$ estimate is therefore 19.5$\mu$g/l (lower and upper limits 1.32 and 97.7).

The calculation is based on assuming a normal distribution for $\log$-$EC_{50}$s. The obvious normal distribution to fit to the data is one with mean $\overline{y}$ and standard deviation $s$. Figure 1 shows the fitted normal distribution (black curve), the data (black dots) and the

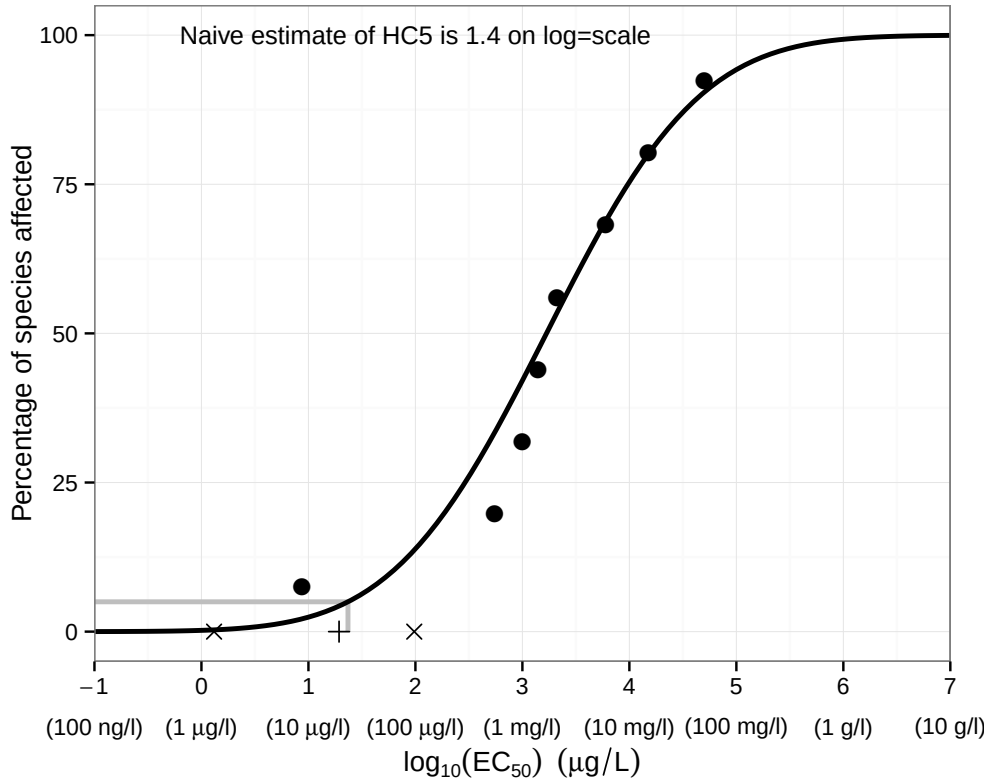


Figure 1: Example of the Aldenberg and Jaworska calculation

A&J estimate (plus symbol) and upper and lower 10% limits (crosses). It also shows the naive estimate of the $HC_5$, the 5th percentile of the fitted normal distribution, using grey lines. We see that the naive estimate is quite close to the A&J estimate for this sample size; however, the naive method gives no indication of the uncertainty attached to the estimate.

The data used in this example are actually the real data for one of the chemicals in our working database to be described later. However, two of the data values are actually censored and so in fact we should not be applying the A&J calculation to them anyway. These data will reappear in part III of this report which gives an example of how to use the software tool which we have developed.

### 1.2 Assumptions underlying the A&J calculation

The explicit assumptions made by A&J are:

- The SSD (of $\log$-$EC_{50}$s) for a chemical has a normal distribution.
- Measured $\log$-$EC_{50}$s may be considered to have been randomly sampled from the SSD.

In relation to the first of these, the statistical population (group of species) to which the SSD refers is rarely made explicit. Moreover, one does not choose species at random to test.

However, the mathematical argument based on random sampling is also justified if we can instead make the assumption of exchangeability: for a new chemical, sensitivities for all species are a priori exchangeable. What does this mean in practice? It means that if asked to predict a test result (say by giving a probability distribution over possible values), we would make the same prediction for each species. Moreover, for any two species which might be measured, knowing which species they were would not give us any information about the difference between their $\log$-$EC_{50}$s: we would think that either species was equally likely to have the higher $EC_{50}$ and the fact that species were or were not closely taxonomically related would have no bearing on the expected magnitude of the difference between their $EC_{50}s$.

There are also two further implicit assumptions underlying the A&J calculation:

- There is no benefit in looking at outcomes for other chemicals. From a Bayesian statistical perspective, this means that there is no information to incorporate in a prior distribution. From a frequentist perspective, it means that there is no point in extending the statistical model to include other chemicals.

  This assumption has been criticised, for example in EFSA (2006). A major source of instability/uncertainty for the A&J calculation for small sample sizes is that the sample standard deviation is then very variable between samples. However the data from other chemicals indicates that the per-chemical population standard deviation is not so variable between chemicals. That information can be used in a more sophisticated form of the A&J calculation, as proposed in EFSA (2006), to give a better estimate of the $HC_5$ for a new chemical. However, that calculation only addresses this single issue and does not address the other weaknesses of the A&J calculation.

- Measurements are exact (no extra variation). No distinction is made in A&J between a measured $EC_{50}$ and the true $EC_{50}$ for the same chemical and species.

  There is also no correct way to take censored data into account without changing the calculation.

## 2 RIVM database

In the rest of the report, we make use of a database of acute test results for a wide variety of chemicals and aquatic species.

Hickey et al (2012) give an account, including references, of the underlying database obtained from RIVM and of rules used to reduce that database to one suitable for the research described in that paper: the endpoint must be an $LC_{50}$ or an $EC_{50}$; the effect must be mortality or immobility; the minimum duration in the experiment must be 48 hours for crustacea or insecta and 96 hours for others and the measurement must be point-wise or censored but not approximate; the species tested must be identified fully to species level (many tested species were identified only to genus level or higher).

We start with the same database which has 30806 records involving 3448 chemicals and 1557 latin names; of the 30806, 10842 were $EC_{50}$s, 17451 $LC_{50}$s and 2076 NOECs — the 30369 mentioned by Hickey et al (2012). We apply essentially the same rules: Hickey et al (2012) erroneously excluded 28 measurements when restricting to data

with fully specified species. This leaves us with 9798 records. Hickey et al (2012) worked with 6576 of these obtained by restricting attention to chemicals for which there point-wise measurements for at least 5 distinct species. We do not make that restriction here as we are not attempting at any point to fit an SSD to data from a single chemical without reference to data from other chemicals. However, in the process, described below, of establishing a taxonomic classification for the species involved, 3 measurements were found to be for species not in the animalia kingdom and were excluded, leaving 9795 measurements in our *working database* involving 2047 chemicals and 631 species (7745 chemical-species combinations).

### 2.1 Taxonomic classification for the database

We wanted a reasonably full hierarchical taxonomic classification for the species in our working database. As provided by RIVM, there was a classification into major and minor taxonomic groups but this was not really suitable for our needs. Scott Dyer kindly provided a classification for a good many species which was derived from the US EPA's ECOTOX database and this was used as the basis for the process described below. That classification has the following taxonomic levels from high to low: kingdom, phylum division, sub-phylum division, super-class, class, order, family, genus and species.

It is worth noting that an attempt to construct a consistent classification directly for the original 30806 records, from sources such as ITIS and Catalogue of Life, foundered for a combination of reasons: issues with consistency of use of latin names with deprecated synonyms often being used; the effort involved in searching multiple taxonomic databases; the difficulties in finding any classification for some species; inconsistencies in classification systems used by different databases. It may well be worth the effort to complete that process in the future.

Of the 631 species in our working database, 607 were matched in the classification provided by Scott Dyer; of the remainder 22 were found in ITIS, 4 in Catalogue of Life and 1 in Uniprot. No attempt was made to investigate whether the same species was present using more than one synonym. In order to make the taxonomic hierarchy of the the working database completely hierarchically coherent, it was necessary to fill-in higher components of the classification for some species from more complete classifications for other species in the same genus or family. The resulting hierarchy is fully coherent in the sense that classifications at each level of the hierarchy below kingdom are a refinement of the classifications at the preceding higher level.

## 3 Data Analysis

We can examine the validity of the assumptions discussed in section 1 by appropriate data analysis. Except where stated otherwise, the analyses that follow are all based on the working database of aquatic acute toxicity tests described in section 2.

### 3.1 Intertest variation

Intertest variation (or "measurement error") means variability of test outcomes for a single chemical-species combination. Hickey et al (2012) give a detailed discussion

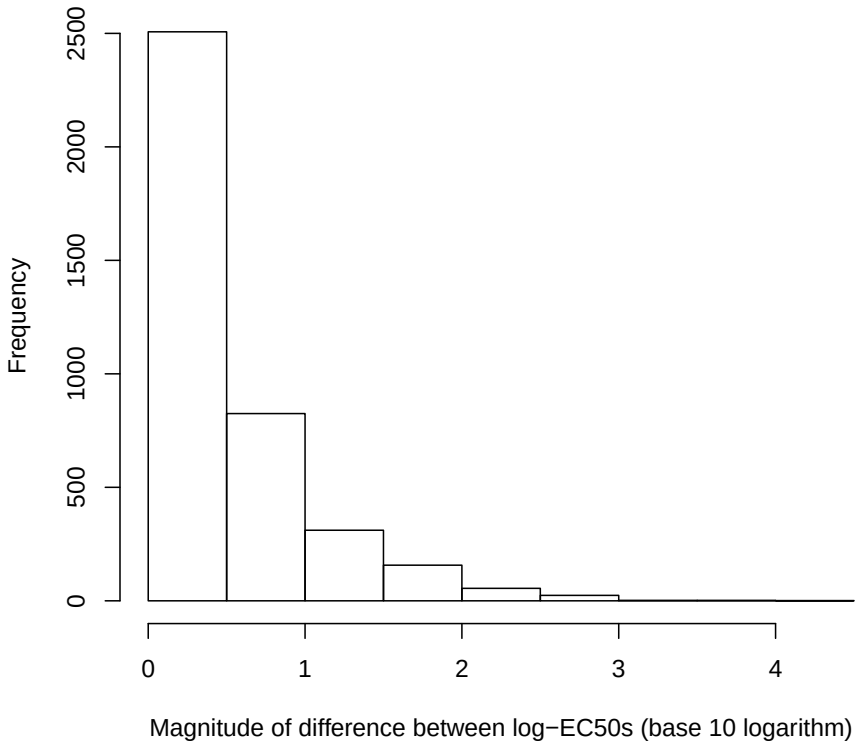


Figure 2: Differences between measured $\log$-$EC_{50}$s for the same chemical-species combination, pooling all combinations

of intertest variation, the empirical evidence for intertest variation, and consequences of adjusting the A&J calculation to incorporate it. Here we will consider just a simple graphical summary of intertest variation.

In the working database, there are many instances where the same chemical-species combination has been tested more than once. For each pair of tests on the same combination, we calculate the magnitude of the difference between the two measured $\log$-$EC_{50}$s. We display all those differences as a histogram in Figure 2. The median difference between measurements on the same combination is approximately 0.3 which corresponds to roughly one-third of an order of magnitude of difference in size between the corresponding $EC_{50}$s on original scale. This may or may not be considered to be large. However, there is a long tail to this distribution. Roughly 10% of differences are greater than 1 which corresponds to an order of magnitude difference between $EC_{50}$s. Clearly, intertest variation is not a tiny contribution.

It is also clear that this distribution is too heavy-tailed to be modelled by a homogeneous normal distribution for intertest variation. There may well be explanatory factors which would allow the use of a normal distribution model with standard deviation depending on those factors but we have not found such factors in the database and we will use a heavy-tailed t-distribution model in section 5.

### 3.2 Sensitivity tendency of a species

EFSA (2006) and Craig et al (2012) reported evidence that species sensitivities were not a priori exchangeable. In particular, they suggested that rainbow trout had a tendency to be more sensitive than average for a chemical. Figure 3 is similar to one in

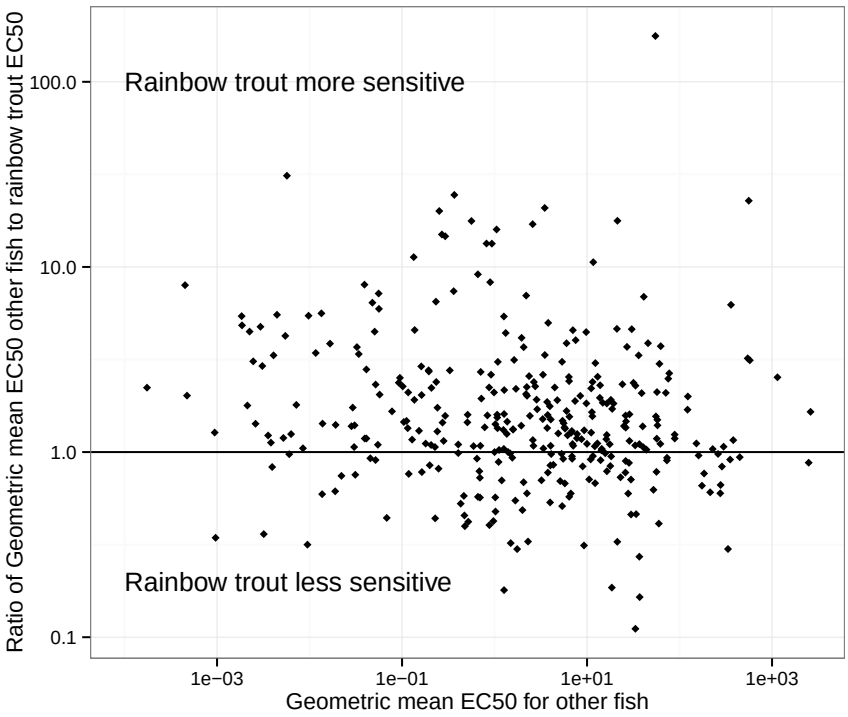


Figure 3: Informal tendency of rainbow trout to be more sensitive than average: each point corresponds to a single chemical; vertical axis shows ratio of average (geometric mean) $EC_{50}$ for other fish to $EC_{50}$ for rainbow trout; there are many more points above the line of equality than below.

EFSA (2006) and is based on the same data. Formal statistical evidence was provided in Craig et al (2012), quoting a P-value of $4 \times 10^{-15}$ for the null hypothesis of exchangeability. They also found that, when all rainbow trout data were excluded from the analysis, the null hypothesis was still strongly rejected with various other species being proposed as non-exchangeable. Further analyses carried in this project by Graeme Hickey (see various progress reports) suggest that there is no reason to consider a small number of species as the exceptions with the others being considered exchangeable; a more reasonable view is that each species has some tendency to be above or below average and that the magnitude of the tendency varies between species. It is likely that rainbow trout was singled out not because of an exceptionally strong tendency but because there is a lot of relevant data: it has been tested on many chemicals.

### 3.3 Interspecies correlation

There is a considerable body of literature indicating the existence of interspecies correlation (across chemicals) in sensitivity and suggesting that it might be used in a variety of ways, in particular to improve estimation of $HC_5$s. For example, see Dyer et al (2006) and Dyer et al (2008).

An example of an interspecies association is shown in Figure 4. It's clear that there is a strong interspecies correlation: the Pearson correlation is 0.955. However, we would expect to see some correlation due simply to the fact that chemicals themselves vary in overall toxicity. For a highly toxic chemical, most species will have a low $EC_{50}$ and for

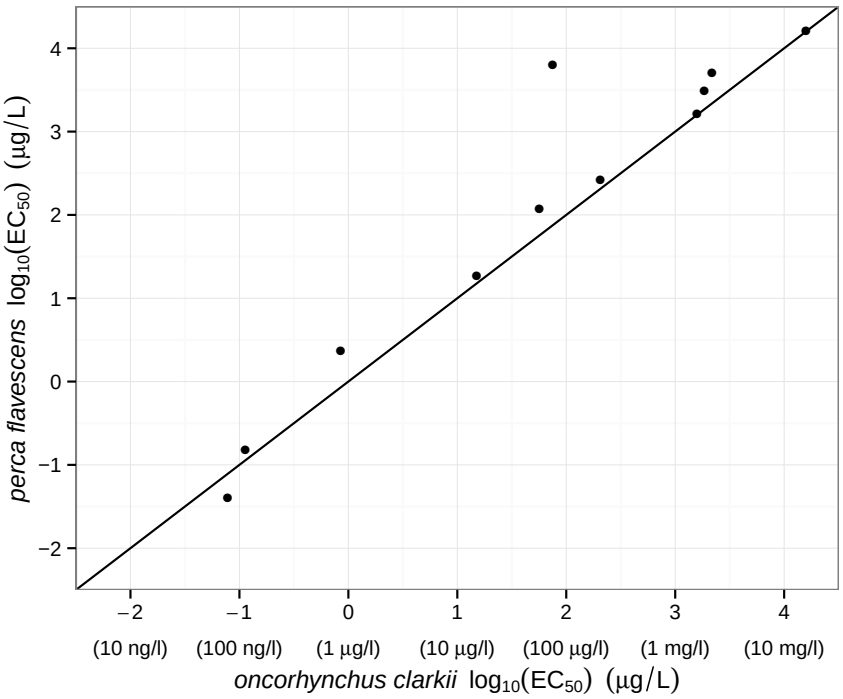


Figure 4: Example of an interspecies association

.

a relatively inert chemical most will have a high $EC_{50}$. To emphasise that each point in Figure 4 comes from a different chemical, Figure 5 shows the same data with the CAS number overlaid for each point.

Figure 6 shows the same data as before but we have added to the plot the average sensitivity for each chemical of species other than the two for which the correlation is being investigated. We see quite clearly that the points with higher $EC_{50}$s for the two focal species also have higher average $EC_{50}$s for other chemicals. A substantial contribution to this particular inter-species correlation appears to be inter-chemical variation in overall toxicity. The important question is whether this is the dominant contribution or whether there is additional correlation because the positions within SSDs of two species are linked in some way: a positive correlation of this kind would mean that finding the first species to be in the upper (or lower) end of the SSD for a particular chemical would increase the likelihood that the second species would also appear in the upper (or lower) end of that SSD. In order to explore this issue, we need some way of assessing whether a particular $EC_{50}$ is at the upper or lower end of the corresponding SSD.

Of course, we don't know the SSD mean and standard deviation but we can ask how a particular $EC_{50}$ compares to the other $EC_{50}$s in our database for the same chemical. For each point in Figure 6, subtract from the $\log$-$EC_{50}$ measurements for both focal species (the axis coordinates) the mean toxicity for that chemical for all other species which have been tested (the larger-text number not in parentheses next to each point); we call this process "standardising" the measurements. The association between the standardised measurements is shown in Figure 7. We see that there is still a clear association, but it is weaker than originally: the Pearson correlation is now 0.710. This informal analysis suggests that there may well be a real interspecies correlation, even

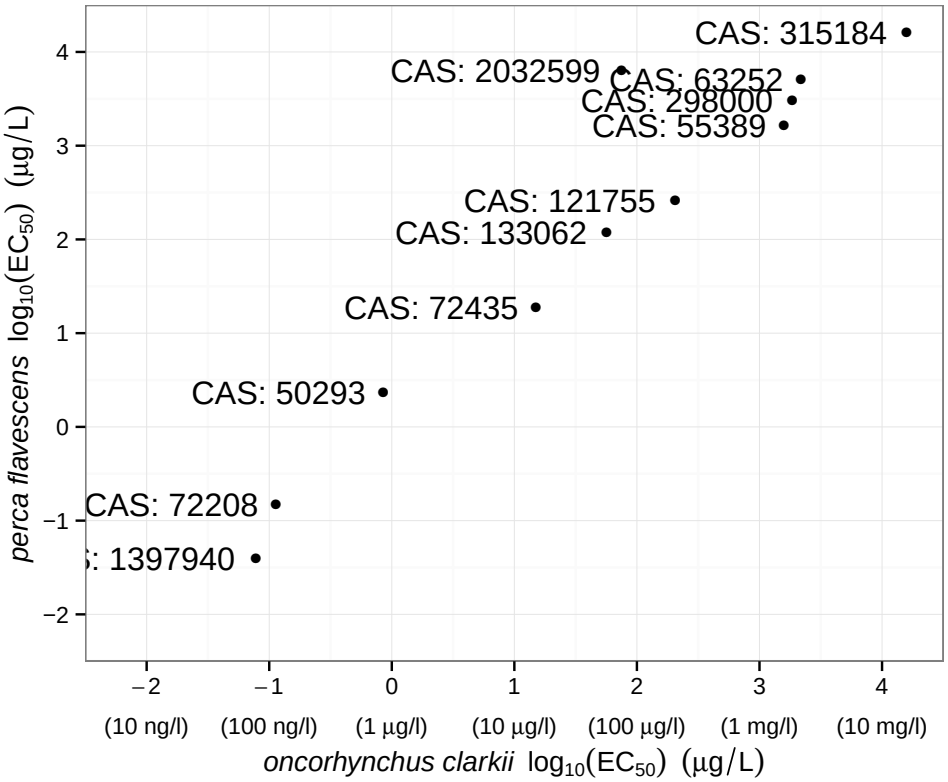


Figure 5: Example of an interspecies association showing that each point is a different chemical.

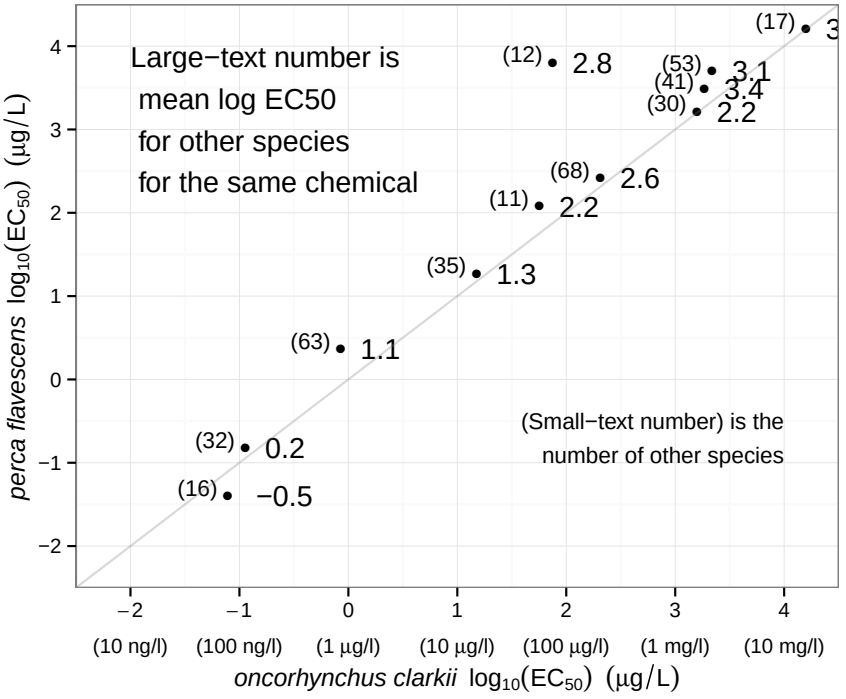


Figure 6: Example of an interspecies association showing the mean sensitivity of other species tested on each chemical.

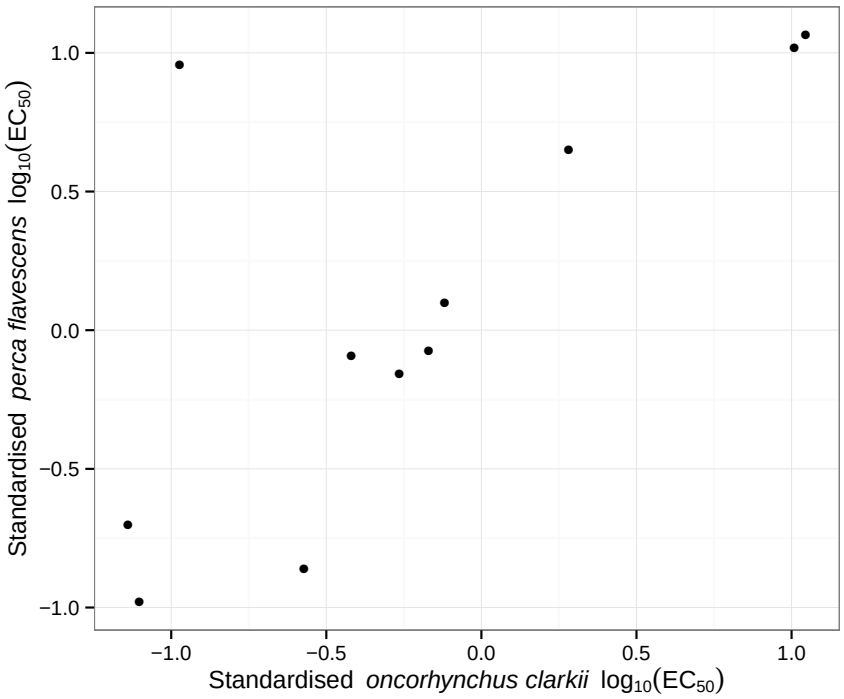


Figure 7: Example of an interspecies association after standardising each sensitivity relative to mean sensitivity of other species tested on the same chemical.

after adjusting for inter-chemical variation in overall toxicity. In what follows, we call the correlation between the standardised measurements the "residual correlation".

What we don't yet know is what drives the residual correlation (after standardising) and how the residual correlation varies depending on the pair of species in focus. A difficulty is that the number of chemicals involved is not usually very large and that the other species used to standardise each point vary from chemical to chemical for a particular pair of focal species and between pairs of focal species. Consequently, there is likely to be a considerable amount of noise attached to each estimate of residual correlation and we really need a way to see what happens for many pairs of species.

### 3.4 Hierarchical taxonomic structure of sensitivity

An obvious driver for the residual correlation is similarity between species and the most readily available measure of similarity of two species is through their taxonomic classifications. Figure 8 shows box-plots of the raw (no standardising) and residual correlations for all pairs of species in the database, restricting to cases where the pair of species have test results for at least 6 chemicals in common. The box-plots are divided according to the taxonomic similarity of the pair of species: we describe the taxonomic similarity as as, for example, "family" if the species are in the same family but not the same genus.

Looking at the solid-line box-plots (raw correlations), and in particular at the median lines, we see that the correlations are highest for species in the same genus and weaken as the taxonomic similarity decreases, being lowest for species which are not even in the same phylum. A similar pattern holds for the dashed-line box-plots (residual corre-

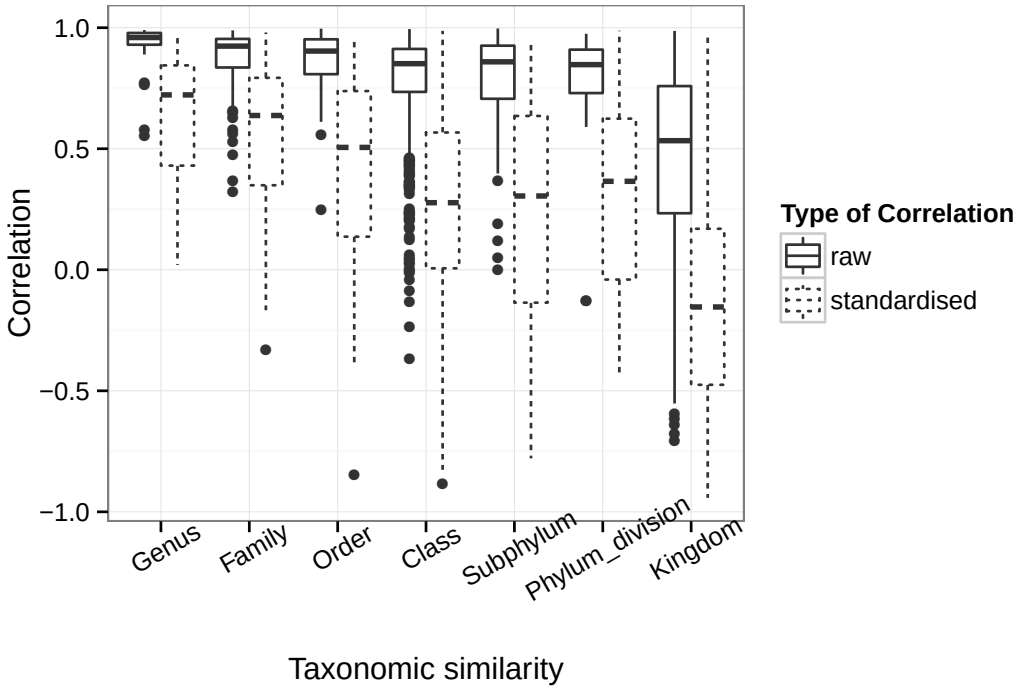


Figure 8: Raw and residual Pearson correlations for all pairs of species having test results for at least 6 chemicals in common, grouped according to taxonomic similarity: for example, a pair of species in the same family but different genera will appear in the two box-plots labelled “Family”.

lations) but the correlations are substantially weaker and in fact the median correlation is close to 0 for pairs of species which are not in the same phylum. It seems likely that much of the variation in residual correlation is sampling variation driven by the limited number of chemicals involved in each correlation and variations in the species used to standardise each measurement; for species from different phyla, we cannot rule out the possibility that all the variation is sampling variation, i.e. that there is actually no residual correlation for such species.

### 3.5 Conclusions from the data analysis

- Intertest variation is real and large enough that it should not be ignored.
- Species tendencies are real. It is likely that they vary. It is also plausible that they exhibit taxonomic structure.
- Interspecies correlation is real. It is not just due to variation in overall toxicity but is linked to taxonomic similarity.

# 4 Redefining the SSD — true sensitivities, taxonomic scenarios and scenario-based $HC_5$s

In what follows, the term "taxonomic scenario" is used to mean the choice of which species are included in the SSD, with particular emphasis on the taxonomic classification of those species.

## 4.1 Motivation for defining scenarios

One might well wonder why this is necessary.

The first part of the answer is that it's not necessary when one trusts the assumptions underlying the A&J calculation: if all species are exchangeable, it does not matter which species one measures and it does not matter which species are in the SSD; this is provided also that there are so many species in the SSD that we don't run into difficulties because we are modelling a finite number of sensitivities by a continuous distribution. One could even add intertest variation to the A&J model (see Hickey et al (2012)) and address the issue censored data without needing to introduce taxonomic scenarios.

The second part of the answer is that there is a conflict between the assumption of exchangeability and the existence of species sensitivity tendencies and residual inter-species correlations. If we accept the conclusions of our earlier data analysis, we should not continue to use A&J unless we can establish that modelling things correctly actually makes little difference to the final $HC_5$.

The third part of the answer is that the existence of species sensitivity tendencies and residual interspecies correlations means that inference about the $HC_5$ must depend on both which species are tested and which species are in the SSD:

- Known tendencies of tested species could in principle be handled by adjusting the test results.

  However, the existence of tendencies for species in the SSD implies that it matters which species are in the SSD. For example, if only less-sensitive species are in the SSD, the $HC_5$ must be higher.

- Taxonomically structured inter-species residual correlation affects uncertainty about the sensitivities of untested species; there will be less uncertainty for those which are taxonomically more similar to tested species.

- The inter-species residual correlation also affects

  - the magnitude of uncertainty about the relative positions in the SSD of two untested species;
  - the overall amount of information we get from the test data; usually we will receive less information by testing closely related species than more distant species; an exception would be if the SSD was taxonomically restricted when one might benefit from taxonomically restricting the choice of species to test.

- Taxonomically structuring tendencies affects estimates of sensitivity for species related to those tested and estimates of relative position of species in the SSD.

The effects of all of these on inference about the $HC_5$ may or may not be large; there seems no obvious way to find out without doing the modelling properly.

### 4.2 True versus measured sensitivities

As discussed earlier, the A&J model/calculation makes no distinction between true and measured sensitivities. On the other hand, we have seen that intertest variation is not negligible; in other words, it is not reasonable to ignore the distinction.

For risk assessment purposes, it seems clear that it is the true sensitivities of species which are really relevant to the assessment. Therefore, the conceptual model of the SSD should be applied to true sensitivities and it is the 5th percentile of true sensitivity about which we wish to make inferences from data; the actual calculations will of course be applied to measured sensitivities and it is part of the role of a statistical model to bridge the gap between measured and true sensitivities.

In an attempt to avoid ambiguity and/or confusion in what follows, we will continue to use $EC_{50}$ to refer to a measurement of sensitivity and we will write $TEC_{50}$ for the corresponding true sensitivity.

### 4.3 Scenario based SSDs and $HC_p$s

We have just argued that it is necessary to decide which species are to be included in the SSD. The "scenario specific SSD" for a chemical is then the distribution of true sensitivities for the species included in the scenario. Note that statistically this is now explicitly a distribution for a finite population.

It seems natural to define the scenario specific $HC_5$ as the 5th percentile of that distribution.

The complication is that a distribution for a finite population does not really have well-defined percentiles. If there are $N$ species in the SSD, there will be $N$ concentrations of interest: the $N$ $TEC_{50}$ values. Let us label them $x_{(1)} \leq x_{(2)} \leq .... \leq x_{(N)}$ so that $x_{(1)}$ is the lowest $TEC_{50}$ and $x_{(N)}$ the highest. For $J = 1, 2 \ldots, N-1$, *any* concentration greater than or equal to $x_{(J)}$ and less than $x_{(J+1)}$ is then hazardous[2] to $J$ species in the SSD; this can be expressed as a percentage of species: 100(J/N)%. For example, if there are $N = 8$ species in the scenario, then any concentration between the second and third lowest of the $TEC_{50}$s is hazardous to 2 out of the 8 species, i.e. to 25%.

This raises two issues in terms of defining an $HC_5$ or equivalent: only certain percentages are possible and there is a actually a range of concentrations corresponding to each of those percentages. However, in many situations risk managers end up defining an acceptable environmental concentration which should be the highest concentration which is safe enough. Setting aside the very important issues involved in extrapolating from single-species acute sensitivity to field ecosystems, any concentration between $x_{(J)}$ and less than $x_{(J+1)}$ is hazardous to 100(J/N)% of species and so the highest such concentration is just below $x_{(J+1)}$. We could therefore consider $x_{(J+1)}$ to be the $HC_{100(J/N)}$ provided that we make it clear that concentrations must be lower than this level.

[2]Same meaning of hazardous as in the phrase "concentration hazardous to 5% of species" defining the $HC_5$.

When $N$ is small, there are not many possible values of $100(J/N)$ and the software described in part II makes the user select one of them. For larger $N$, there are enough values of $100(J/N)$ that there will be one a little less than or equal to 5 (or other value of $p$ of interest) which can be used as though it was 5. In this case the software requires the user to choose a whole number $p$ to define an approximate $\mathrm{HC}_p$.

Finally, it is not obvious that this scenario-specific $\mathrm{HC}_5$ (or $\mathrm{HC}_\mathrm{p}$ for some other p) is a quantity of interest for risk assessment. Some of that doubt applies just as much to the A&J calculation and there is a literature discussing the appropriateness of using the A&J $\mathrm{HC}_5$. Making the SSD apply only to a finite number of species is likely to raise further questions, especially when $N$ is small. Computing a hazardous concentration is not the only possible way to exploit the posterior distribution of true sensitivities for the scenario species obtained using the new statistical model; it may be that SSDs for taxonomic scenarios need to be summarised in a different way altogether.

## 5 Hierarchical Statistical Modelling

Here we give a relatively non-technical outline of how to build a multivariate statistical model of sensitivity, for multiple chemicals and multiple species, which incorporates (i) intertest variation, (ii) inter-chemical variation of overall toxicity and of magnitude of interspecies sensitivity variability, (iii) species sensitivity tendencies and (iv) residual interspecies correlation. The modelling exploits the taxonomic classification of species to make tendencies likely to differ less for closely related species and to make interspecies correlation stronger for closely related pairs of species.

Full mathematical details of the resulting model are given in appendix A.

It is not really possible to convey a proper sense of the modelling approach without using some mathematical notation. To try to make what follows more digestible, it is broken into a sequence of steps each adding a bit of complexity.

### 5.1 The single chemical A&J model

For a single chemical, the A&J model is a normally distributed SSD for $\log$-$\mathrm{EC}_{50}$s with mean $\mu$ and standard deviation $\phi$. This can be written as:

$$y_j = \mu + \phi z_j$$

where

- $j$ indexes species;
- $y_j$ is the $\log$-$\mathrm{EC}_{50}$ for species $j$;
- $z_j$ is the standardised *chemical-species interaction*[3];
- the $z_j$ are independently sampled from the standard normal distribution (mean 0, standard deviation 1)

### 5.2 Multiple chemicals

When more than one chemical needs to be modelled, we need a notation to make explicit which chemical is being considered. We do this by adding an index $i$ to each

[3]The term 'interaction' is used here in a statistical rather than chemical sense.

quantity in the single chemical model:

$$< y_{ij} = \mu_i + \phi_i z_{ij} \tag{1}$$

where $i$ now indexes chemicals and $j$ continues to index species.

Note that there is an index $i$ on $\mu$ and $\phi$ which means that each chemical has a different mean and standard deviation for its SSD.

### 5.3 Adding features to A&J's model

#### 5.3.1 Feature 1: inter-test variation

We now need to distinguish measured from true sensitivity. In doing so, we also need to allow for the possibility of more than one measurement for a particular chemical-species combination.

We write

$$y_{ijk} = \mu_{ij} + \epsilon_{ijk}$$

where:

- as before, $i$ indexes chemicals and $j$ indexes species;
- the new index $k$ indexes (potential) multiple measurements for the same chemical-species combination;
- $y_{ijk}$ denotes a $\log$-$EC_{50}$; specifically, it is the $k$th $\log$-$EC_{50}$ measurement for chemical $i$ tested on species $j$.;
- $\mu_{ij}$ is the true sensitivity ($\log$-$TEC_{50}$) of species $j$ to chemical $i$.
- $\epsilon_{ijk}$ is the intertest variation component ("measurement error") for this particular $\log$-$EC_{50}$.

We adapt the multiple-chemical version of the A&J model to be a model for true sensitivities ($\log$-$TEC_{50}$s) instead of measured sensitivities ($\log$-$EC_{50}$s). We simply replace $y_{ij}$ by $\mu_{ij}$ in (1) to give

$$\mu_{ij} = \mu_i + \phi_i z_{ij} \tag{2}$$

The intertest variation will always have zero mean. A normal distribution would be mathematically and computationally convenient but is unrealistic. A scaled t-distribution provides the necessary long-tailed distribution indicated by the earlier data analysis. The degrees of freedom might either be specified a priori or learned from the data along with other parameters.

#### 5.3.2 Feature 2: species' tendencies

For each species, we add a sensitivity tendency parameter to (2):

$$\mu_{ij} = \mu_i + \beta_j + \phi_i z_{ij}$$

Here, $\beta_j$ is the sensitivity tendency of species $j$. Since it has only a $j$ index, this value applies to all chemicals.

A simple model is that sensitivity tendencies follow a normal distribution.

### 5.3.3 Feature 3: interspecies correlations

We introduce interspecies correlation in two ways:

- correlating the tendencies for two species $j$ and $j'$ so that $\text{Corr}[\beta_j,\, \beta_{j'}] = \rho_{jj'}$ means that pairs of species with higher $\rho_{jj'}$ are likely to have smaller differences between the corresponding pair of tendencies.
- correlating the SSD variability (chemical-species interactions) so that $\text{Corr}[z_{ij},\, z_{ij'}] = \gamma_{jj'}$ means that the difference between two species' interactions for the same chemical is likely to be smaller when $\gamma_{jj'}$ is greater.
  Note that the correlation here only depends on the two species and is the same for all chemicals and only applies to interactions for the same chemical.

At this stage we seem to be introducing a very large number of correlation parameters (two for every pair of species).

### 5.3.4 Feature 4: taxonomic structure for correlations

To avoid the problem of having too many correlation parameters and to take into account what we learned from the earlier data analysis, we make the correlations actually depend only on the *taxonomic distance* between a pair of species.

The detail depends on the precise taxonomic classification scheme we choose to use. Here we will consider the hypothetical situation where we use just three levels of classification: phylum, family and species. Then

- Species in the same family are at distance 1 from each other
- Species in the same phylum but different families are at distance 2
- Species in different phyla are at distance 3

The correlations will be zero for pairs of species at distance 3. So we now needs just four correlation parameters: $\gamma_1$ and $\rho_1$ will be used for pairs of species at distance 1 and $\gamma_2$ and $\rho_2$ for pairs at distance 2.

More generally, in a classification scheme with $L$ levels, there will be $2(L-1)$ correlation parameters.

### 5.3.5 Feature 5: model variation of per-chemical SSD parameters

Each chemical has an SSD mean parameter $\mu_i$. We suppose that the inter-chemical variation in these follows a normal distribution.

Similarly, we suppose that the inter-chemical variation in the SSD variation parameter $\phi_i$ follows a gamma distribution (on the scale of $1/\phi^2$)

## 5.4 The final model

For computational purposes it is easier to represent the correlation structure above by breaking the species sensitivities and the chemical-species interactions into sums of independent components of variation. The result would be a classical mixed model (also known as random-effects or multi-level models) except for two features: the presence of the per-chemical scaling $\phi_i$ and the use of t-distribution to describe intertest variation.

## 6 Computation

In practice, there are five aspects to computation:

- Specify a prior distribution for the hyper-parameters (variances of tendency and interaction components, variability of per-chemical SSD parameters etc). We propose the use of relatively diffuse individual prior distributions which do not require elicitation of expert judgements.

- Choose a suitable database of ecotoxicity data. We currently use the working database described in section 2

- Fit a variety of versions of the model to the database to decide which levels of taxonomic classification to include in the the model.

  As is usual with statistical models, there is a trade-off between complexity, computation time and quality of prediction. Overly simple models make poor predictions but so too do overly complex ones.

  A good deal of time has been spent in early 2013 looking at this issue from a variety of perspectives including informal data analysis (Figure 8), the Akaike and Bayes information criteria AIC and BIC (Burnham and Anderson , 2002), the deviance information criterion DIC (Spiegelhalter et al , 2002) and recent work (Plummer , 2008) addressing known weaknesses in AIC, BIC and DIC for hierarchical random-effects models.

  No conclusive answer has yet been reached about which levels of taxonomic classification to incorporate in building the hierarchical structure. Provisionally, it would seem that phylum, order, family, genus and species are all contenders for inclusion with the greatest doubt about the inclusion of order and/or genus.

- Obtain the initial posterior distribution of all the model parameters which are relevant to the database.

  The initial posterior will be represented by a Monte Carlo sample obtained using Markov Chain Monte Carlo implemented in R (R Core Team , 2012); see appendix D for the code. That sample will then be used in the software tool described later to make inferences for new chemicals.

- In the software tool, for each new chemical, the initial posterior will then be used as the prior distribution for a version of the model specific to the new chemical.

  That prior will be updated by the software tool using test data for the new chemical to obtain the posterior distribution of sensitivities to the new chemical of all species in the user's chosen taxonomic scenario.

  Again the posterior will be represented by a Monte Carlo sample obtained by Markov Chain Monte Carlo but this time implemented in Matlab (Matlab , 2012) which also provides the graphical user interface for the software tool.

- Use the posterior for the new chemical to compute the posterior distribution of summary quantities such as the scenario-specific $HC_5$.

Details of computation are provided in appendix B.

# Part II
# Software Tool (hSSD)

In addition to the science described earlier, an important output of the project is an open-source freely accessed software tool which makes the model and algorithms available for general use. The name of the tool is "hSSD" which stands for "hierarchical species sensitivity distribution".

## 7 Structure of the tool

The tool provides a graphical user interface to the methodology. It is written in Matlab (Matlab , 2012) and the source code is included with the software. However, Matlab is not freely available and so the software is distributed as a "compiled Matlab" program which does not require the user to have a Matlab license. Instead, the user must download and install the "Matlab compiler runtime" which is available for free and which enables the user to run compiled Matlab programs.

To reduce the burden of installation for users, Microsoft Visual Studio 10 is used to build a Microsoft Installer for the tool, resulting in a standard Microsoft `.msi` file. The user simply downloads an executable zip file and executes and installation takes place, checking for problems as it proceeds.

## 8 Using the tool

To use the tool, the user must explicitly provide

- Test data for a new chemical. These are measured $EC_{50}$s and can be point values or censored (upper, lower or interval) values.
- The chosen taxonomic scenario: the list of species to be included in the SSD.
- Full taxonomic classification for any species which appears in the test data or the taxonomic scenario and which is not already listed in the file of taxonomic classifications which accompanies the software.

  The user can add classifications using the graphical user interface and save the extended file of taxonomic classifications for future use.

Implicitly, the user is also specifying an HDF5 file (see section B.4) to be used. The HDF5 contains the details of the model being used and the sample from the initial posterior distribution described in sections 5 and 6. An HDF5 file is distributed with the software and is used by default; it is based on the model described here, a particular choice of taxonomic levels, and the database described in section 2.

It is not currently envisaged that users will create other HDF5 files for themselves. Allowing the user to choose an HDF5 file to use in the software provides flexibility for the future by effectively allowing some changes to software without having to distribute an entirely new version. For example, we might want to change the database or to assist

users in using different databases or to allow users to experiment with different choices of taxonomic levels.

The user then "runs the model" for the new chemical: the algorithm from section B.5. Because this is a Monte Carlo algorithm, one must specify the number of samples to be taken from the posterior and because it is an Markov Chain Monte Carlo algorithm, the user must also specify how much the output should be "thinned" and how much initial output should be discarded ("burn-in"). The sample size, thinning rate and burn-in period are all options under the control of the user but the software is distributed with default values so that the user does not need immediately to consider these issues. The user's preferences can be saved to be used automatically when the software is run again.

Once the model run is complete, the user can explore various aspects of the output:

- The main output "screen" shows the estimated SSD and uncertainty attached to the estimate. The horizontal axis is concentration and the vertical access is the fraction of species "affected": a species is considered affected at a given concentration if its true $EC_{50}$ is lower than the given concentration[4]

  Test results are overlaid for those tested species which are also in the scenario. This can be done in two distinct ways. In both cases, the vertical coordinate of a species is given by the rank of that species in the scenario-SSD, based on the posterior median estimate of true sensitivity for each species: each time through the MCMC loop we sample a value for the true sensitivity for each species in the scenario and at the end can compute the median of that value for each species in the scenario. The choices for the horizontal coordinate are

  - Each species is plotted as point with horizontal coordinate given by the $EC_{50}$ measurement: lower/upper bound is used for upper/lower censored data and mid-point of interval for interval-censored data. Censored data are plotted with a greyed-out symbol.
  - Each species is plotted as point with horizontal coordinate given by the posterior median estimate of true sensitivity of the species. Optionally, a horizontal line is plotted showing a credible interval for the true sensitivity of the species.

  Optionally, the user can show the estimated $HC_p$ together with upper and lower credible bounds on the estimate. The value of $p$ is chosen using a pop-up menu which offers a list of possible values. When the number $N$ of species in the scenario is small, the choices offered are all the allowable ones (see section 4.3); for larger $N$, the user is offered whole number values of $p$ and the software uses the largest allowable value less than or equal to the user's choice.

  Optionally, the user can show the estimated "fraction affected" (FA) at a concentration specified by the user. The fraction affected is the percentage of scenario species for which the true $EC_{50}$ is lower than the given concentration. The central estimate is the posterior mean of FA and is accompanied by upper and lower credibility estimates.

  Various further options exist to control symbols, colours and fonts used and to adjust locations of species names or to display them as higher levels of classification or even to omit them altogether.

[4] This matches the use of the term "hazardous".


Figure 9: Splash screen for hSSD software

- A second output screen provides access to information about each individual species in the scenario.

  The user selects a species from a pop-up menu and the software shows the posterior distribution of true sensitivity ($\log$-$\mathrm{TEC}_{50}$) for that species. This screen has two intended purposes: to facilitate the user who is interested in the risk to individual species as well as in the $\mathrm{HC}_p$; to enable the user to develop insight, by examining detailed output, into what the model and algorithms actually do.

- A third output screen provides access to diagnostic information about the MCMC process itself.

  The user can examine trace and autocorrelation plots for the MCMC sample of true sensitivity for each species in the taxonomic scenario. The species to be considered is chosen from a pop-up menu.

## 9 User Interface

When the user starts the software, it immediately displays a "splash screen" (Figure 9) to show that something is happening; the splash screen remains visible until the main user interface starts up and is ready to use, at which point the splash screen disappears.

The opening screen for the main user interface is shown in Figure 10.

Further information about the interface will be conveyed via the example in part III of this report; the example includes a number of additional screen-shots.

## 10 Documentation

A small amount of documentation is currently included with the software. It is intended simply to get the user started; the software interface is meant subsequently to be self-

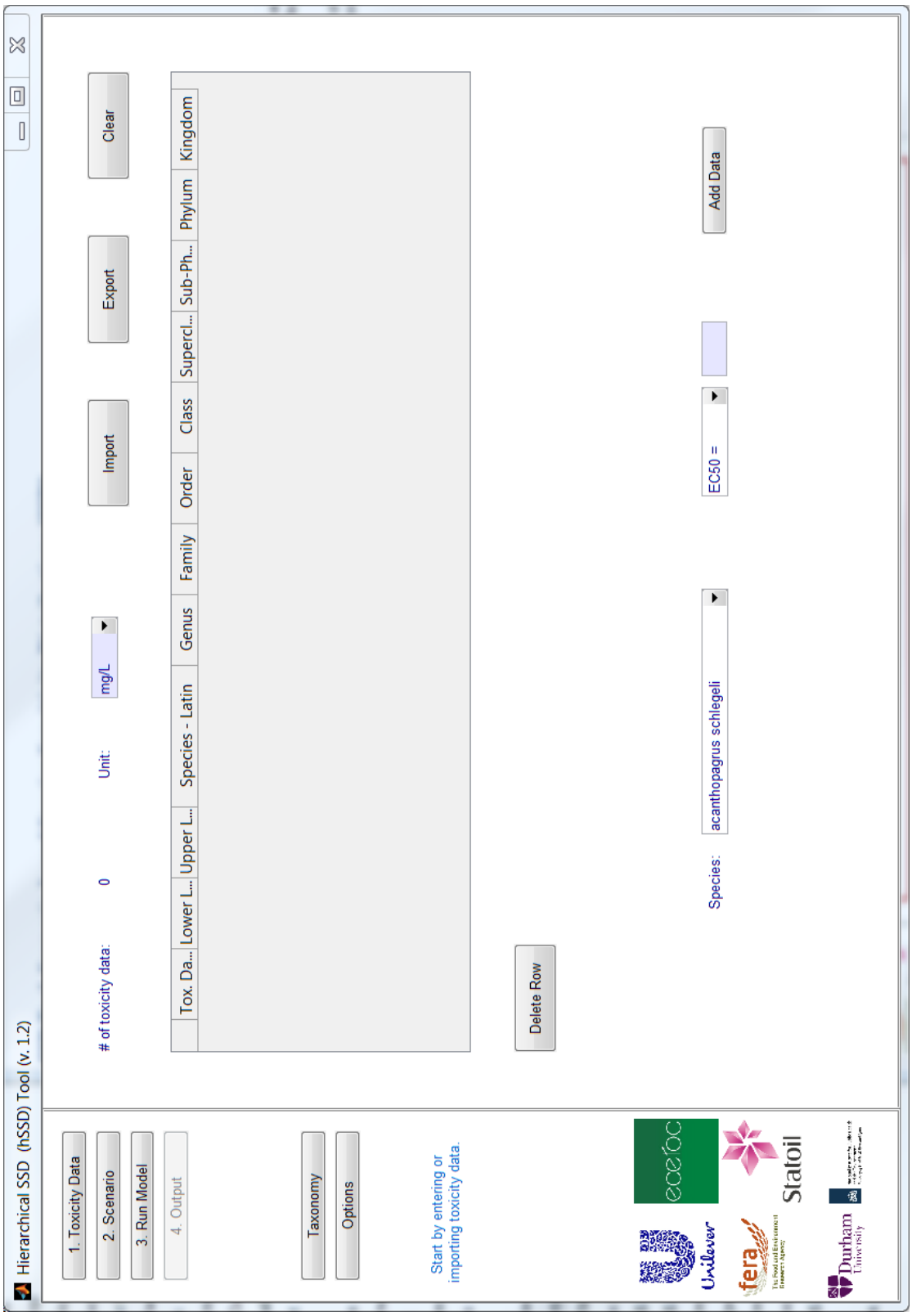


Figure 10: Initial window for the main user interface of the hSSD software

explanatory to users who understand the basic science. In practice, we do not yet know whether the documentation is sufficient nor whether the interface is self-explanatory; we also need to publish the science to make it more available.

| $EC_{50}$ | Lower-limit on $EC_{50}$ | Upper-limit on $EC_{50}$ | Species |
|---|---|---|---|
| — | 0·0 | 1000·0 | corophium salmonis |
| 1400·0 | 1400·0 | 1400·0 | crassostrea gigas |
| 2100·0 | 2100·0 | 2100·0 | daphnia magna |
| 550·0 | 550·0 | 550·0 | daphnia pulex |
| 6000·0 | 6000·0 | 6000·0 | macoma balthica |
| 8·7 | 8·7 | 8·7 | metapenaeus monoceros |
| — | 50000·0 | — | mytilus edulis |
| 15000·0 | 15000·0 | 15000·0 | poecilia reticulata |

Table 1: Data used in example: $EC_{50}$s are expressed in $\mu$g/l; censored data have only an upper or lower limit on the $EC_{50}$.

# Part III

# Example

We now give a worked example of basic use of the software and interpretation of the results. We use as data for a new chemical the data for sodium sulfide (CAS=1313822) from the working data base described in section 2. These data were chosen because they are real data and include both lower-censored and upper-censored data. Although these data are for a chemical in the database, the software treats them as though they were new data and no connection is made in the software to the fact that they derive from sodium sulfide. The data are presented in Table 1. Note that these data were also used in section 1 to illustrate the A&J calculation; however the censoring was ignored in that calculation.

Data can be entered individually into hSSD or provided as an Excel (`.xls`) file: the example data are available as `demo2.xls` distributed with hSSD from version 1.3 onwards. Note that the units used for $EC_{50}$s may be specified in the Excel file by an integer entered in column F of row 1: 1=ng/l, 2=$\mu$g/l, 3=mg/l etc.

By clicking the `Import` button (see Figure 10) and selecting `demo2.xls` in the resulting file dialog, we arrive at Figure 11.

We now have to choose the species to be included in the scenario. To do so, we click on `2. Scenario` and then on `Add All Species`. The result is shown in Figure 12.

Now we need to run the model. To do so with default MCMC settings, we simply click `3. Run Model`. Initially we get a "progress bar" screen like in Figure 13 and then, when the calculation is finished, we are taken directly to the output screen shown in Figure 14. The figure shows the output screen after three choices have been made by the user: (i) the range of concentrations for the plot has been set to run from 0.01$\mu$g/l to $10^7\mu$g/l using the text-entry boxes at the top-right of the plot; (ii) for the $HC_p$ output, $p = 5$ has been selected using the drop-down menu and approximated in the software by the nearest available value $p = 4.91$; and (iii) a concentration for the fraction affected (FA) output has been set to 10$\mu$g/l.

The plot shows the median estimate of the SSD together with upper and lower bounds; as indicated by Aldenberg and Jaworska (2000), the bounds can be interpreted in two ways: (a) horizontally to give a range representing uncertainty about the concentration

| | Tox. Da... | Lower L... | Upper L... | Species - Latin | Genus | Family | Order | Class | Supercl... | Sub-Ph... | Phylum | Kingdom |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | NaN | 0 | 1000 | corophium salmonis | corophium | corophiid... | amphipoda | malacos... | Missing ... | crustacea | arthropoda | animalia |
| 2 | 1400 | 1400 | 1400 | crassostrea gigas | crassost... | ostreidae | ostreoida | bivalvia | Missing ... | Missing ... | mollusca | animalia |
| 3 | 2100 | 2100 | 2100 | daphnia magna | daphnia | daphniidae | diplostraca | branchio... | Missing ... | crustacea | arthropoda | animalia |
| 4 | 550 | 550 | 550 | daphnia pulex | daphnia | daphniidae | diplostraca | branchio... | Missing ... | crustacea | arthropoda | animalia |
| 5 | 6000 | 6000 | 6000 | macoma balthica | macoma | tellinidae | veneroida | bivalvia | Missing ... | Missing ... | mollusca | animalia |
| 6 | 8.7 | 8.7 | 8.7 | metapenaeus mono... | metapen... | penaeidae | decapoda | malacos... | Missing ... | crustacea | arthropoda | animalia |
| 7 | NaN | 50000 | Inf | mytilus edulis | mytilus | mytilidae | mytiloida | bivalvia | Missing ... | Missing ... | mollusca | animalia |
| 8 | 15000 | 15000 | 15000 | poecilia reticulata | poecilia | poeciliidae | cyprinod... | actinopt... | osteicht... | vertebrata | chordata | animalia |

Figure 11: Screen shot of the hSSD software after the data for the example have been loaded.

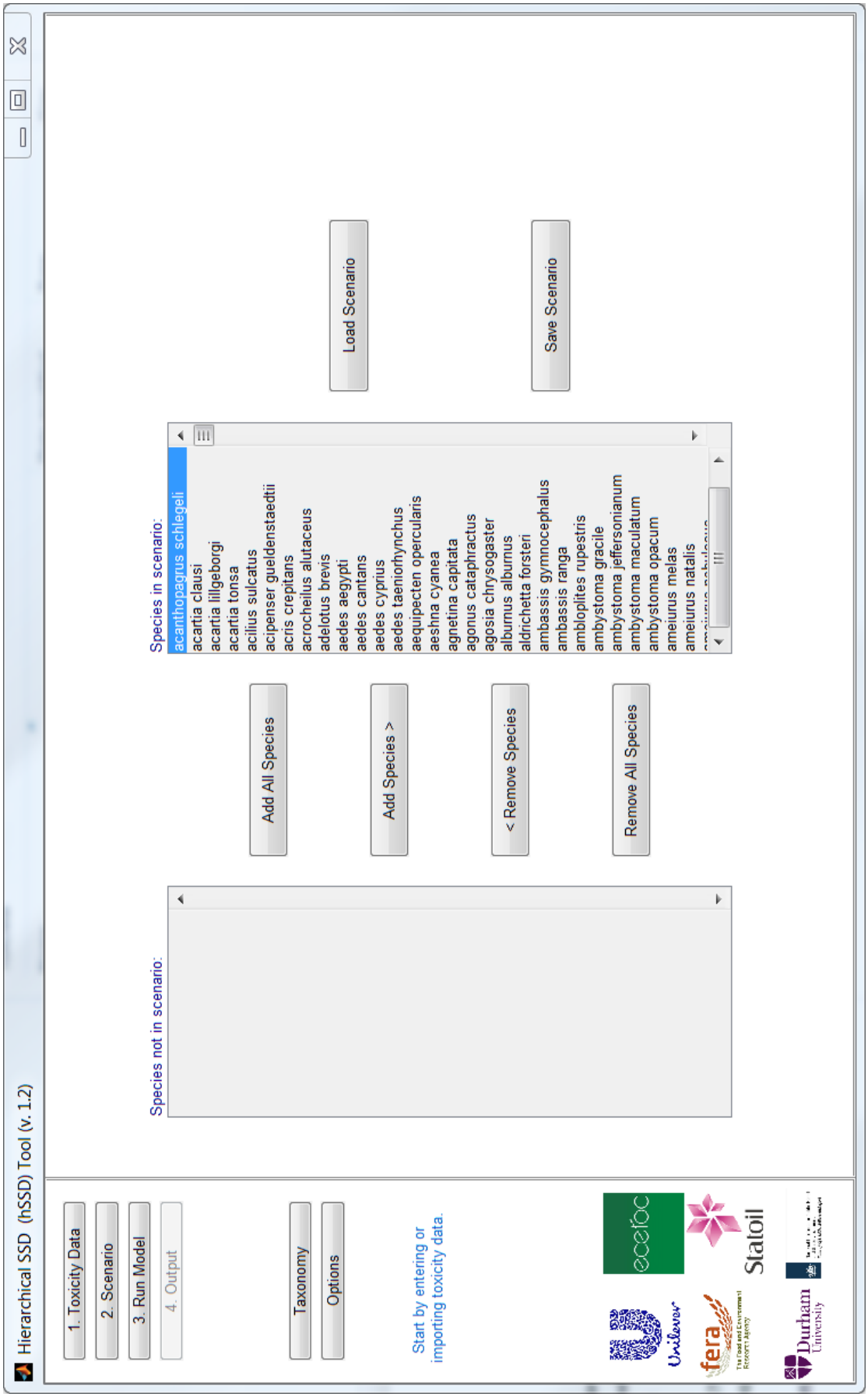


Figure 12: Screen shot of the hSSD software after scenario has been chosen — all species are in the scenario.

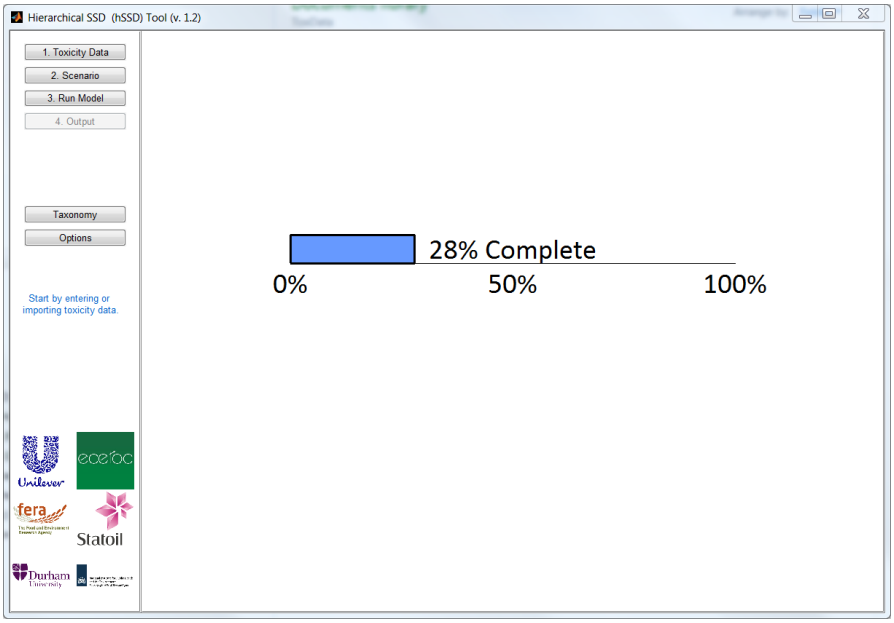


Figure 13: The hSSD software "progress bar" screen.

corresponding to a particular percentage of species affected; or (b) vertically to give a range representing uncertainty about the percentage of species affected at a particular concentration. For each tested species which is included in the scenario, a point is plotted: the horizontal coordinate is the posterior median estimate of the $TEC_{50}$ and the vertical coordinate is the position of that species in the scenario-specific SSD, i.e. the estimate of the percentage of scenario species which are more sensitive than the tested species (based on tghe posterior median estimate of sensitivity of each species). The median estimate of the $HC_5$ and upper and lower bounds (90% credibility) are shown underneath the plot at the bottom-left while the mean estimate of the fraction affected and uppper and lower bounds (at least 90% credibility) are shown to the bottom-right.

Figure 15 shows the effect of selecting the `Plot HCx` tick-box which visually highlights the interval for the $HC_5$ reported underneath. Similarly, Figure 16 shows the effect of selecting the `Plot FA`. tick-box to visually highlight the estimate of the FA and associated uncertainty.

Figure 17 shows the SSD overlaid with the uncertainty attached to each estimate of true sensitivity for the tested species; the species names have been omitted this time.

Figure 18 shows the SSD with the data points overlaid and plotted using + symbols. Species names are positioned at the right of the graph this time.

We can also look at the posterior uncertainty relating to the true sensitivity ($TEC_{50}$) of an individual species from the scenario: Figures 19 and 20 show the uncertainty about two species: one tested and the other not tested.

To conclude this part of the report, it may be interesting to compare the earlier A&J $HC_5$ estimate of 19.5$\mu$g/l (10% lower and upper limits 1.32 and 97.7) with the estimate provided for the scenario in this example: 15.6 (5% lower and upper limits 0.63 and 424.). A more valid comparison arises from making the limits more comparable: 5% lower and upper limits for the A&J estimate are 0.42 and 140. In making any comparison, bear in mind that the A&J calculation completely ignored the fact that highest

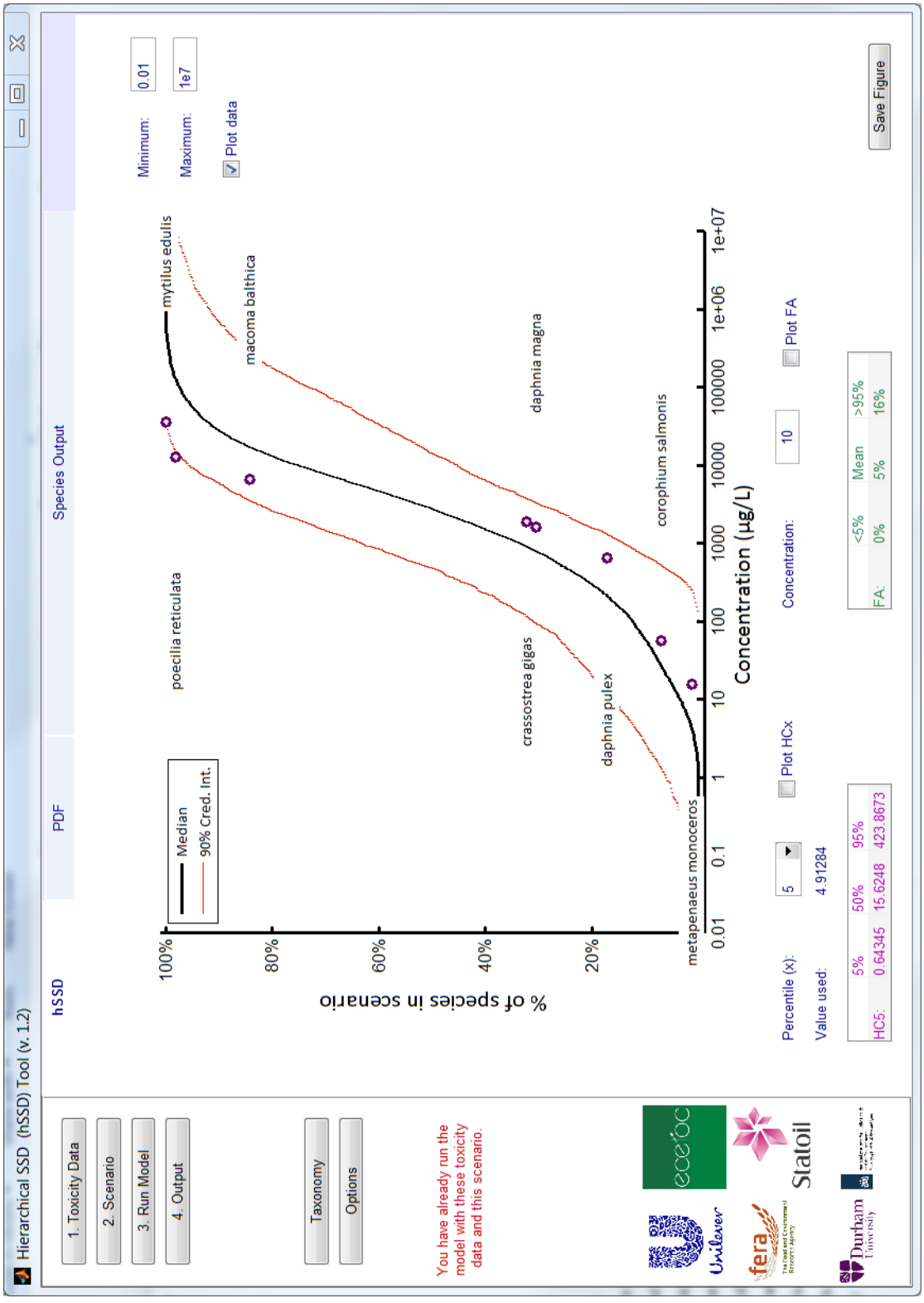


Figure 14: Screen shot of the hSSD software showing the basic output screen.

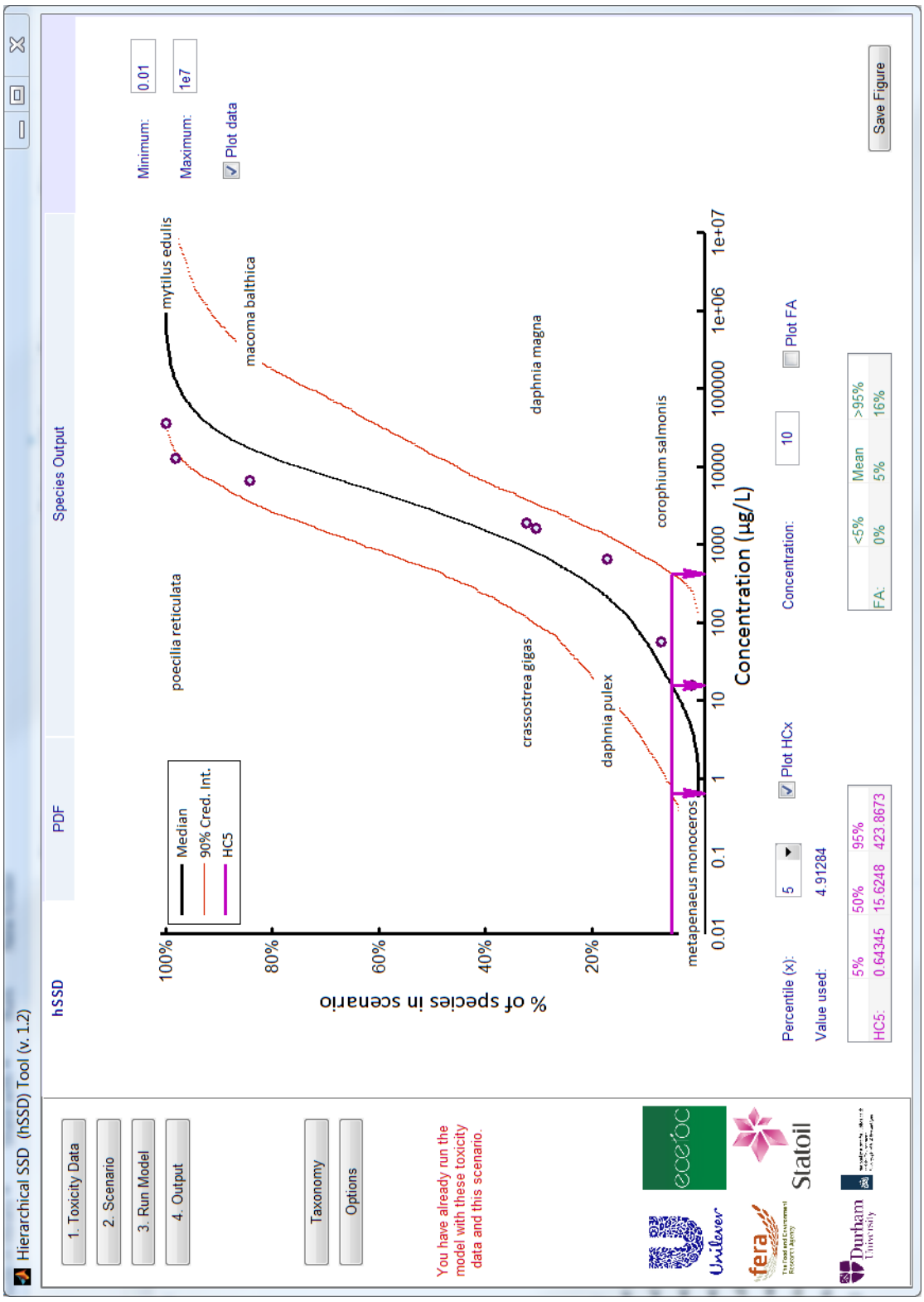


Figure 15: Screen shot of the hSSD software showing the output emphasising the $HC_5$ estimate and uncertainty summarised underneath the plot.

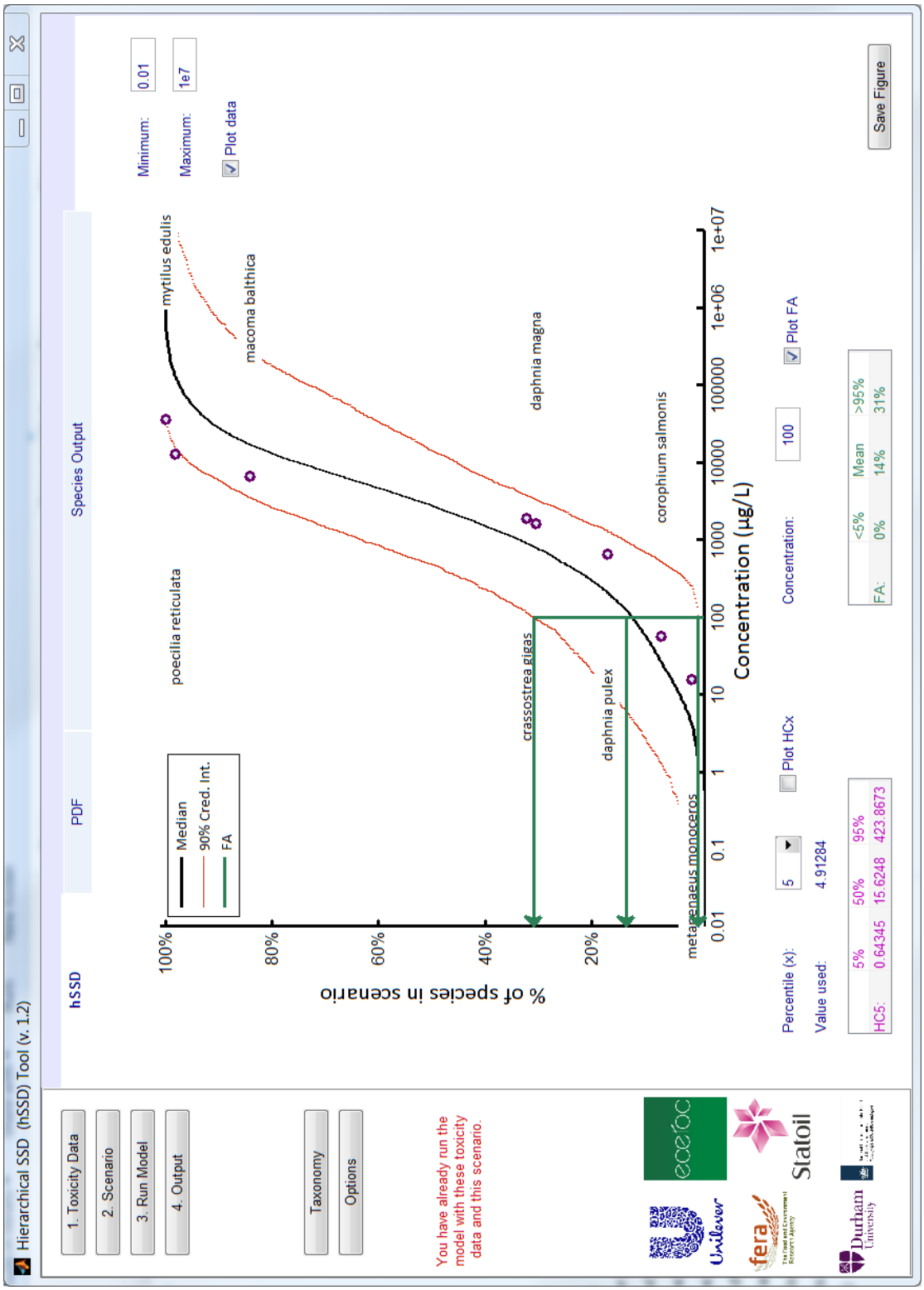


Figure 16: Screen shot of the hSSD software showing the output emphasising the $HC_5$ estimate and uncertainty summarised underneath the plot.

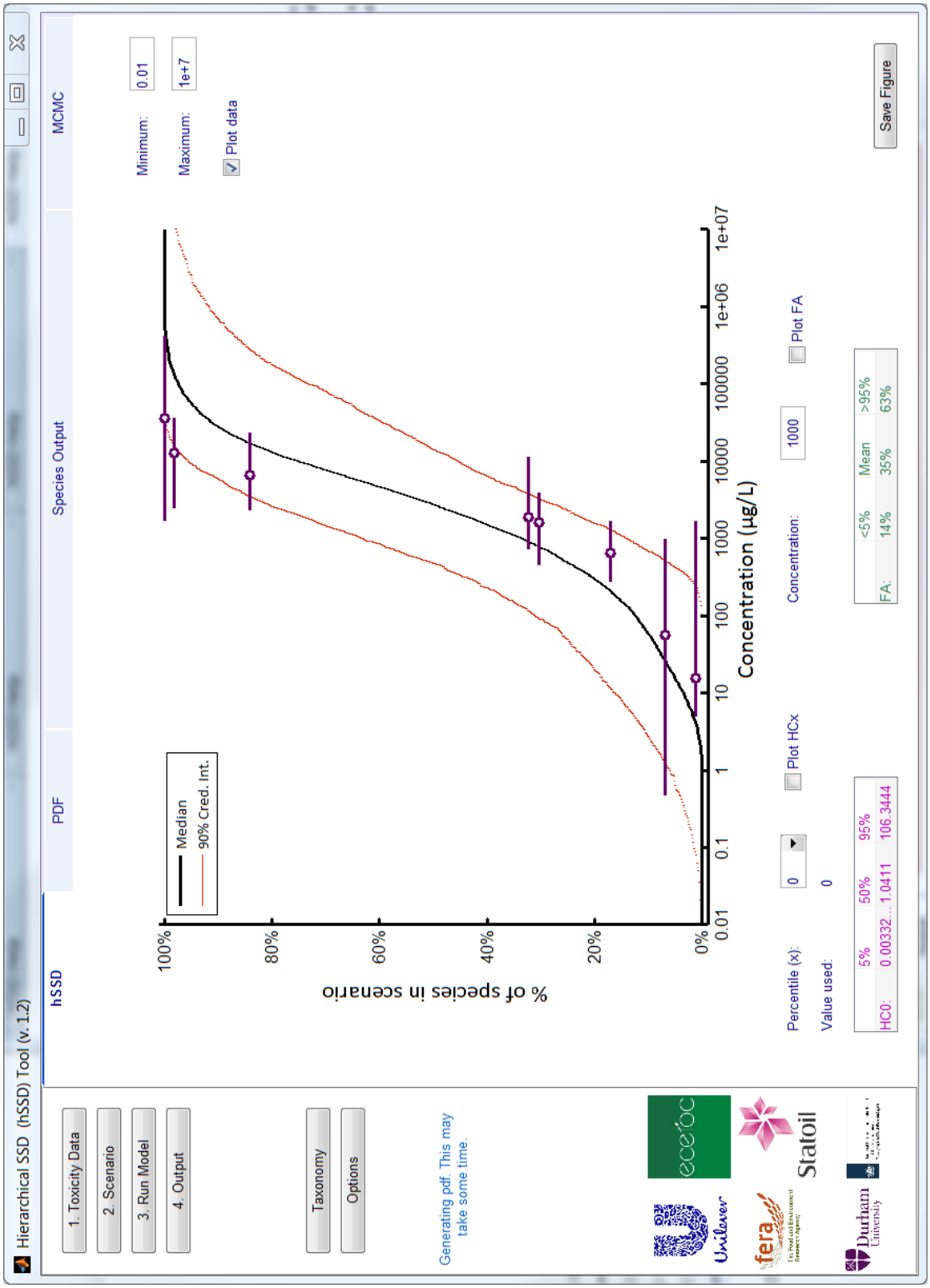


Figure 17: Screen shot of the hSSD software showing the output, emphasising the uncertainty attached to the true sensitivity of each tested species.

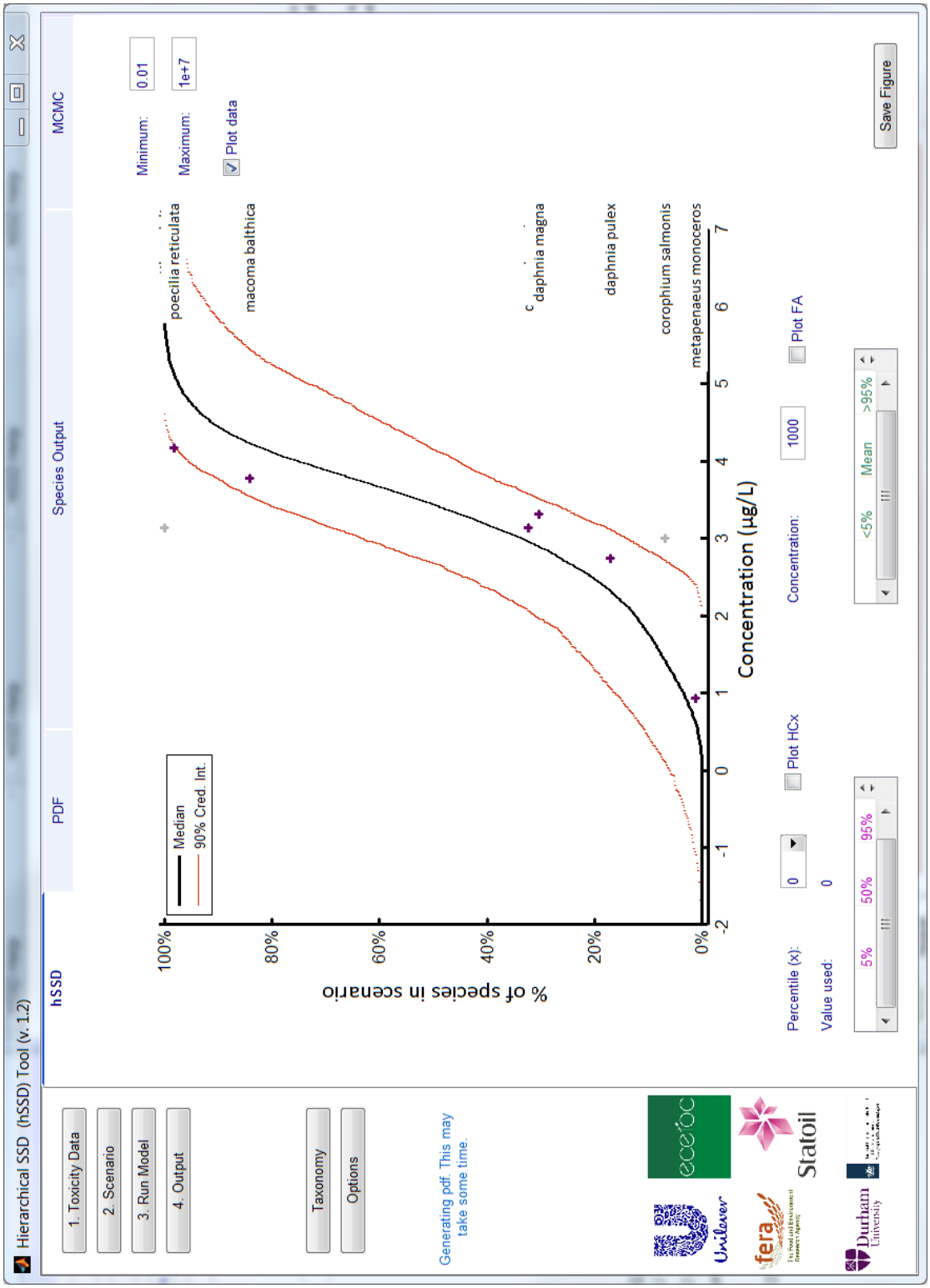


Figure 18: Screen shot of the hSSD software showing the output; the SSD is overlaid with the data for the tested species; censored values are indicated by greyed-out symbols.

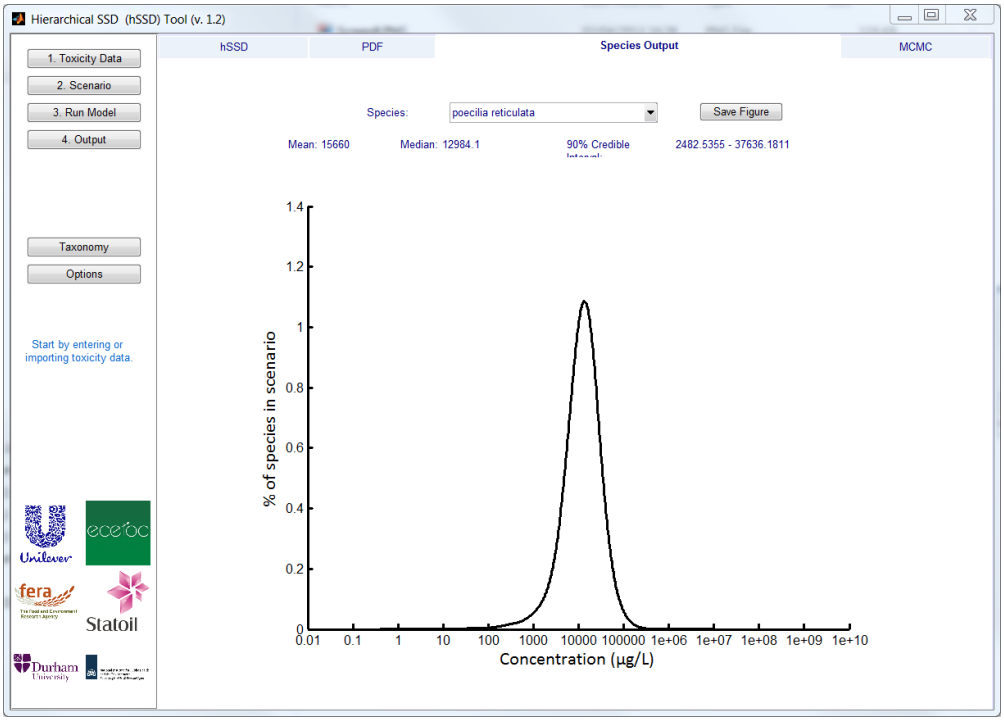


Figure 19: Posterior uncertainty about the true sensitivity ($TEC_{50}$) to the example chemical of a tested species (poecilia reticulata).

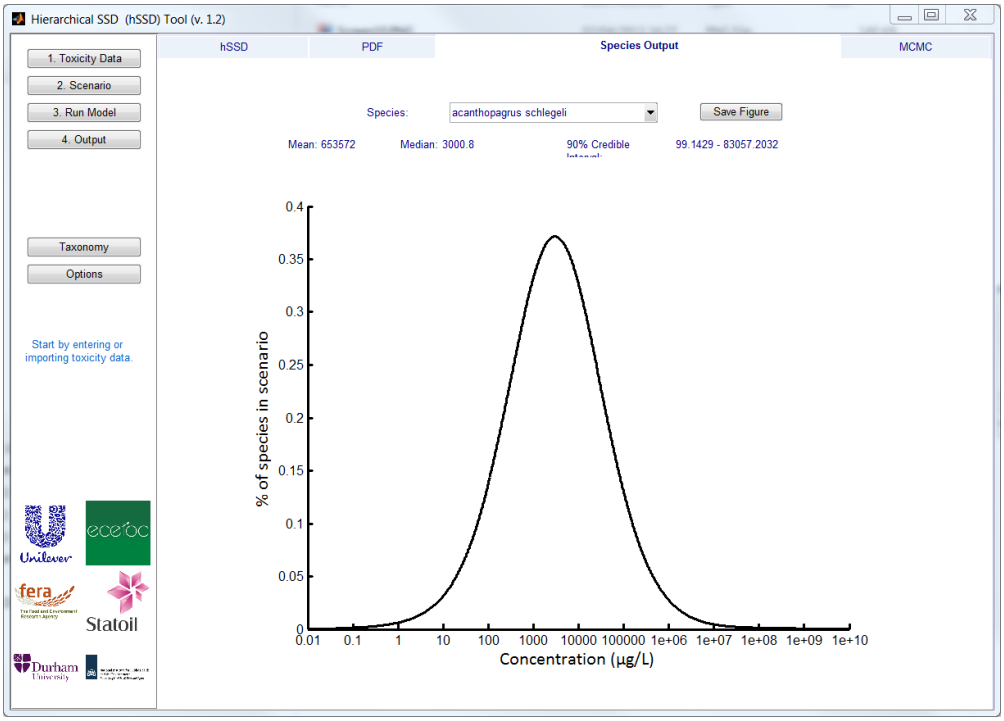


Figure 20: Posterior uncertainty about the true sensitivity ($TEC_{50}$) to the example chemical of an un-tested species (acanthopagrus schlegeli).

$EC_{50}$ was in fact censored and could in fact be any value above the recorded 50000 $\mu$g/l while one of lower $EC_{50s}$ was censored in the other direction. We are of course comparing chalk and cheese, as there are many other fundamental differences beteween the calculations, but one interesting question which arises is: how much difference would there be between A&J and hSSD across a range of scenarios, chemicals and sample sizes/structures?

# References


Aldenberg T and Jaworska JS (2000), Uncertainty of the hazardous concentration and fraction affected for normal species sensitivity distributions, *Ecotoxicol. Environl. Saf.*, **46**, pp 1–18.

Bates D, Maechler M and Bolker B (2012), *lme4: Linear mixed-effects models using S4 classes*. URL: `http://CRAN.R-project.org/package=lme4`.

Burnham KP and Anderson DR (2002), *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach (2nd ed.)*, Springer-Verlag.

Craig P, Hickey GL, Luttik R and Hart A (2012), On species non-exchangeability in probabilistic ecological risk assessment, *J Roy Stat Soc Stat Soc A (Statistics in Society)*, **175**, pp 243–262.

Davis TA (2006), *Direct Methods for Sparse Linear Systems: Fundamentals of Algorithms*, SIAM, Philadelphia.

Dyer SD, Versteeg DJ, Belanger SE, Chaney JG and Mayer FL (2006), Interspecies correlation estimates predict protective environmental concentrations, *Environ Sci Technol*, **40**, pp 3102–3111.

Dyer SD, Versteeg DJ, Belanger SE, Chaney JG, Raimondo S and Barron MG (2008), Comparison of species sensitivity distributions derived from interspecies correlation models to distributions used to derive water quality criteria, *Environ Sci Technol*, **42**, pp 3076–3083.

European Chemicals Agency (2008), *Uncertainty analysis*, in *Guidance for the Implementation of REACH: Guidance on Information Requirements and Chemical Safety Assessment, ch R.19*, European Chemicals Agency, Helsinki.

European Commission (2006), Regulation (EC) No. 1907/2006 of the European Parliament and of the Council of 18, December 2006, *Off. J. Eur. Un.*, L 396/1.

European Food Safety Authority (2005), Opinion of the Scientific Panel on Plant Health, Plant Protection Products and their Residues on a request from EFSA related to the assessment of the acute and chronic risk to aquatic organisms with regard to the possibility of lowering the uncertainty factor if additional species were tested, *The EFSA Journal*, **301**, pp 1–45.

Hadfield JD (2010), MCMC Methods for multi-response generalized linear mixed models: the MCMCglmm R package, *J Stat Soft*, **33**, pp 1–22. URL: `http://www.jstatsoft.org/v33/i02/`.

Hickey GL, Craig P, Luttik R and De Zwart D (2012), On the quantification of measurement error in ecotoxicity data with application to species sensitivity distributions, *Environ Toxicol Chem*, **31**, pp 1903–1910.

MATLAB and Statistics Toolbox Release 2012b, The MathWorks, Inc., Natick, Massachusetts, United States.

Plummer M (2008), Penalized loss functions for Bayesian model comparison, *Biostatistics* **9**, pp 523–539.

Posthuma L, Suter GW, Traas TP (Eds) (2002). *Species Sensitivity Distributions in Ecotoxicology.* Boca Raton: Lewis Publishers.

R Core Team (2012), *R: A language and environment for statistical computing* R Foundation for Statistical Computing, Vienna, Austria. URL `http://www.R-project.org/`.

Spiegelhalter DJ, Best NG, Carlin BP and Van der Linde A, Bayesian Measures of Model Complexity and Fit (with Discussion), *Journal of the Royal Statistical Society, Series B*, **64**, pp 583–616.

Van Vlaardingen PLA, Traas RP, Wintersen AM and Aldenberg T (2004), *ETX 2.0: A program to calculate hazardous concentrations and fraction affected, based on normally distributed toxicity data*, Bilthoven, the Netherlands: National Institute for Public Health and the Environment (RIVM). Report no. 601501028/2004. URL: `http://www.rivm.nl/rvs/Risicobeoordeling/Modellen_voor_risicobeoordeling/ETX_2_0`.

# Acknowledgements

There are many people and organisations to thank:

- Statoil and Unilever for jointly funding this research project and Unilever for funding the research consultancy which preceded it, in particular Peter Chapman, Stuart Marshall, Oliver Price and Mathijs Smit.
- ECETOC for providing support for steering group travel, meetings and teleconferences, in particular Malyka Galay-Burgos.
- Graeme Hickey for his outstanding contributions to this research area, in the build-up to this project as PhD student, and during the first half of this project as post-doctoral researcher.
- All the members of the core steering group for their significant scientific and other contributions throughout the project: Peter Chapman (Unilever/Tecsolve), Malyka Galay-Burgos (ECETOC), Mick Hamer (Syngenta), Andy Hart (FERA), Robert Luttik (RIVM), Stuart Marshall (Unilever), Oliver Price (Unilever), Mathijs Smit (Statoil), Paul Whitehouse (UK Environment Agency).
- Peter Chapman for his work in bringing the project into being and managing the project.
- Willem Roelofs (FERA) for software development and support.
- Robert Luttik and Dick De Zwart (RIVM) for provision of toxicity data and information about scenarios.
- Scott Dyer (Procter & Gamble) for providing taxonomic classification data.
- EFSA (through Andy Hart, Tony Hardy and Robert Luttik) for giving me the opportunity to become interested in this area of science as an *ad hoc* expert providing input to EFSA (2006).

# Part IV

# Appendices

## A Hierarchical Statistical Modelling

In what follows, we give details of our multivariate statistical model of sensitivity.

### A.1 Notation

- $i$ indexes chemicals.
- $j$ indexes species.
- $k$ indexes measurement(s) for the same chemical-species combination.
- $y_{ijk}$ is the $k$-th measured log-sensitivity for chemical $i$ tested on species $j$. This will be a $\log$-$EC_{50}$ for some acceptable end-point, probably mortality or in some cases immobility.
- Taxonomic structure in the model may use a subset of the levels in use in standard taxonomic classification systems. $\ell$ indexes taxonomic levels in the model. It ranges from 1 to $L$ where $L$ is the number of levels in the classification being used in the chosen model.
- $t_\ell(j)$ is then the taxonomic classification of species $j$ at level $\ell$ for $\ell = 1, \ldots, L$.

### A.2 Model structure

The structure of model we consider for now is:

$$y_{ijk} = \mu_{ij} + \epsilon_{ijk} \tag{3}$$

and

$$\mu_{ij} = \mu + \alpha_i + \beta_j + \psi_{ij} \tag{4}$$

In (3) and (4):

- $\mu_{ij}$ is the true log-sensitivity for species $j$ exposed to chemical $i$;
- $\epsilon_{ijk}$ is "measurement error" or, more pedantically, intertest variation;
- $\mu$ is the overall central value of log-sensitivity across all chemicals and species;
- $\alpha_i$ is difference between the central value of log-sensitivity for chemical $i$ and $\mu$; thus $\mu + \alpha_i$ is the central value of log-sensitivity for chemical $i$.
- $\beta_j$ is the (log-sensitivity) "tendency" of species $j$; this applies to all chemicals.
- $\psi_{ij}$ is the (log-sensitivity) "interaction" between chemical $i$ and species $j$; this is what allows the position of a species in the SSD to vary between chemicals.

However, we want to be able to incorporate both chemical-specific variability and taxonomically-related structure while at the same maintaining as much exchangeability as possible. Therefore, we write

$$\psi_{ij} = \phi_i \xi_{ij} \tag{5}$$

where

- $\phi_i > 0$ scales (log-sensitivity) variation for chemical $i$; this allows some chemicals to exhibit more variation in sensitivity than others.
- $\xi_{ij}$ is directly comparable between different chemicals whereas $\psi_{ij}$ is not; this is the quantity we want to use as the basis for (partial) exchangeability between interactions;

We introduce taxonomically related structure by

$$\beta_j = \beta_{1t_1(j)} + \cdots + \beta_{Lt_L(j)} \tag{6}$$

and

$$\xi_{ij} = \xi_{i1t_1(j)} + \cdots + \xi_{iLt_L(j)} \tag{7}$$

so that the tendency of a species is made up of tendency components corresponding to its classifications at the levels of the taxonomic classification system being used and a similar decomposition applies to the interaction between each chemical and species. In this notation, $\beta_{\ell t}$ is the tendency component at level $\ell$ for a species whose classification at level $\ell$ is $t$.

We complete the structure with statements about exchangeability and independence.

- All $\epsilon_{ijk}$ values are are a priori exchangeable.

  This assumption could easily be relaxed, for example by making them exchangeable only within taxonomically defined groups; we could also make the $\epsilon$ distribution depend on covariates should they be available.

  So that the decomposition in (3) is meaningful, the $\epsilon_{\cdots}$ distribution is centered at zero.

- All $\alpha_i$ values are a priori exchangeable. The $\alpha_{\cdot}$ distribution is located at zero.
- At each taxonomic level $\ell$, all $\beta_{\ell t}$ values are exchangeable.

  Each $\beta_{\ell \cdot}$ distribution is located at zero.

- All $\phi_i$ values are exchangeable. The $\phi_{\cdot}$ distribution has scale one.
- At each taxonomic level $\ell$, all $\xi_{i\ell t}$ values are exchangeable.

  Each The $\xi_{\cdot \ell \cdot}$ distribution is located at zero.

- Conditional on any parameters for the various distributions representing exchangeability, we specify that the following (blocks) are a priori independent:

$$\mu, \{\alpha_i\}, \{\beta_{1t}\}, \ldots, \{\beta_{Lt}\}, \{\phi_i\}, \{\xi_{i1t}\}, \ldots, \{\xi_{iLt}\}, \{\epsilon_{ijk}\}$$

### A.3 Parameterisation

We now make specific distribution family choices for all the components of the model:

- Write
$$\epsilon_{ijk} = \sigma_\epsilon \frac{z_{ijk}}{\sqrt{\kappa_{ijk}}}$$
where $z_{ijk}$ are iid (independent and identically distributed) $\mathrm{N}(0,1)$ and $\kappa_{ijk}$ are iid $\Gamma(\frac{1}{2}\nu_\kappa, \frac{1}{2}(\nu_\kappa - q_\kappa))$. The $z_{...}$ and $\kappa_{...}$ blocks are independent of each other and of all the other blocks of parameters of which $\epsilon_{...}$ was previously declared to be independent.

  This construction gives $\epsilon_{...}$ a scaled t-distribution with $\nu_\kappa$ degrees of freedom. The degrees of freedom may be fixed or may be sampled; either way it will be included in the sample from the posterior for the database. The parameter $q_\epsilon$ will be fixed; its purpose is to improve independence in the sample from the posterior obtained by MCMC.

- the chemical central values $\alpha_i$ are iid $\mathrm{N}(0, \sigma_\alpha^2)$;
- the species tendency components at each level $\ell$ are iid $\mathrm{N}(0, \sigma_{\beta\ell}^2)$;
- the interaction components at each level $\ell$ are iid $\mathrm{N}(0, \sigma_{\xi\ell}^2)$;
- $\lambda_i = 1/\phi_i^2$, the reciprocal squares of the chemical scalings, are iid $\Gamma(\frac{1}{2}\nu_\phi, \frac{1}{2}(\nu_\phi - q_\phi))$. Again $\nu_\phi$ may be sampled or fixed and $q_\phi$ is fixed.

The reason for choosing a scaled t-distribution for measurement error rather than a normal distribution is the long-tailed behaviour observed in Figure 2.

Where normal distributions were chosen, the intent was to provide enough flexibility to infer variability of the model component involved whilst making MCMC reasonably straightforward to implement.

The single-parameter gamma distribution for the $\lambda_i$ was chosen to provide the capacity to infer inter-chemical relative variation in per-chemical SSD variability; the typical magnitude of SSD variability is determined by the $\sigma_{\xi\ell}^2$ parameters.

Conditional on the hyper-parameters $\sigma_\alpha$, $\sigma_{\beta 1}, \ldots, \sigma_{\beta L}$, $\sigma_{\xi 1}, \ldots, \sigma_{\xi L}$, $\nu_\phi$, $\sigma_\epsilon$ and $\nu_\kappa$, we specify that the following (blocks) are a priori independent:

$$\mu, \{\alpha_i\}, \{\beta_{1t}\}, \ldots, \{\beta_{Lt}\}, \{\phi_i\}, \{\xi_{i1t}\}, \ldots, \{\xi_{iLt}\}, \{\kappa_{ijk}\}, \{z_{ijk}\}$$

## B Details of computation

For a summary of the overall computational approach, see section 6.

### B.1 Prior distribution

The prior on the hyper-parameters consists of a number of independent components:

$$p(\mu, \sigma_\alpha, \sigma_{\beta 1}, \ldots, \sigma_{\beta L}, \sigma_{\xi 1}, \ldots, \sigma_{\xi L}, \nu_\phi, \sigma_\epsilon, \nu_\kappa) \propto \\ p(\mu)p(\sigma_\alpha)p(\sigma_{\beta 1}, \ldots, \sigma_{\beta L})p(\sigma_{\xi 1}, \ldots, \sigma_{\xi L})p(\nu_\phi)p(\nu_\kappa)/\sigma_\epsilon$$

where, at present,

- $\mu \sim \mathrm{N}(0, 10^2)$ which is a fairly diffuse distribution assigning 95% probability to a range covering 40 orders of magnitude for central sensitivity.
- $p(\sigma_\alpha) \propto 1$.
- $p(\sigma_{\beta 1}, \ldots, \sigma_{\beta L}) \propto 1$.
- $p(\sigma_{\phi 1}, \ldots, \sigma_{\phi L}) \propto 1$.
- $p(\nu_\phi)$ is uniform on $1/\nu_\phi$ subject to $\nu_\phi > \max(1, q_\phi)$.
- $p(\nu_\kappa)$ is uniform on $1/\nu_\kappa$ subject to $\nu_\kappa > \max(1, q_\kappa)$.

### B.2 Ecotoxicity database

We use the database and taxonomic classifications described in detail in section 2.

Additional notation specific to the database:

- The set of all chemicals $i$ in the database is $\mathcal{I}$.
- The set of all species $j$ in the database is $\mathcal{J}$.
- The set of all chemical-species combinations $(i, j)$ in the database is $\mathcal{J}_\mathcal{I}$. The set of species tested on chemical $i$ is $\mathcal{J}_i$.
- For the database, $k$ ranges from 1 to $K_{ij}$ for $(i, j) \in \mathcal{J}_\mathcal{I}$.
- $\mathcal{L}_\ell$ is the set of classifications at level $\ell$ for species in the database and $T_\ell$ is the number of such classifications. Thus, for species in the database, $t_\ell(j) \in \mathcal{L}_\ell$. Similarly, $\mathcal{L}_{i\ell}$ is the set of classifications at level $\ell$ for species in the database tested on chemical $i$.

### B.3 Algorithm for sampling from the initial posterior

In what follows, I am going to assume that a decision has already been made about which levels of taxonomic classification to include in the model.

I will also assume initially that there are no censored data and then subsequently explain how to adapt the algorithm to cope with censored data.

The initial posterior will be represented by a Markov Chain Monte Carlo sample. The rest of this section describes an algorithm for MCMC sampling from the posterior distribution of all the parameters given the data in the database.

The parameters to be inferred, collectively denoted by $\theta$, are:

$$\mu, \sigma_\alpha, \{\sigma_{\beta\ell} : \ell = 1, \ldots, L\}, \nu_\phi, \{\sigma_{\xi\ell} : \ell = 1, \ldots, L\}, \sigma_\epsilon, \nu_\kappa,$$
$$\{\alpha_i : i \in \mathcal{I}\}, \{\lambda_i : i \in \mathcal{I}\}, \{\beta_{\ell t} : \ell = 1, \ldots, L; t \in \mathcal{L}_\ell\}, \{\psi_{i\ell t} : i \in \mathcal{I}; \ell = 1, \ldots, L; t \in \mathcal{L}_{i\ell}\},$$
$$\{\kappa_{ijk} : (i,j) \in \mathcal{J}_\mathcal{I}; k = 1, \ldots, K_{ij}\}$$

where we define $\psi_{i\ell t} = \phi_i \xi_{i\ell t}$. Note that the $z_{\ldots}$ do not appear in this list as their values are determined by the data together with the other parameters.

For MCMC we choose to work with the $\psi_{\ldots}$ rather than the $\xi_{\ldots}$ as it leads to a more straightforward update. To do so, we need to note that $\psi_{i\ell t} \,|\, \phi_i, \sigma_{\xi\ell} \sim \mathrm{N}(0, \phi_i^2 \sigma_{\xi\ell}^2)$ and that the $\psi_{\ldots}$ are conditionally independent given $\phi_\cdot$ and $\sigma_{\xi 1}, \ldots, \sigma_{\xi L}$.

The posterior pdf is

$$\begin{aligned}
p(\theta \,|\, y) \propto \quad & p(\nu_\phi) p(\nu_\kappa) p(\sigma_\epsilon) \\
& \times \prod_{i \in \mathcal{I}} p_{\text{normal}}(\alpha_i; 0, \sigma_\alpha) \\
& \times \prod_{\ell=1}^{L} \prod_{t \in \mathcal{L}_\ell} p_{\text{normal}}(\beta_{\ell t}; 0, \sigma_{\beta\ell}) \\
& \times \prod_{i \in \mathcal{I}} p_{\text{gamma}}(\lambda_i; \tfrac{1}{2}\nu_\phi, \tfrac{1}{2}(\nu_\phi - q_\phi)) \\
& \times \prod_{i \in \mathcal{I}} \prod_{\ell=1}^{L} \prod_{t \in \mathcal{L}_{i\ell}} p_{\text{normal}}(\psi_{i\ell t}; 0, \phi_i \sigma_{\xi\ell}) \\
& \times \prod_{(i,j) \in \mathcal{I}\mathcal{J}} \prod_{k=1}^{K_{ij}} p_{\text{gamma}}(\kappa_{ijk}; \tfrac{1}{2}\nu_\kappa, \tfrac{1}{2}(\nu_\kappa - q_\kappa)) p_{\text{normal}}(y_{ijk}; \mu_{ij}, \sigma_\epsilon / \kappa_{ijk}^{\frac{1}{2}})
\end{aligned}$$

where

$$\mu_{ij} = \mu + \alpha_i + \sum_{\ell=1}^{L} \beta_{\ell t_\ell(j)} + \sum_{\ell=1}^{L} \psi_{i\ell t_\ell(j)}$$

Here, $p_{\text{normal}}(x; \mu, \sigma)$ is the probability density at $x$ of the normal distribution with mean $\mu$ and standard deviation $\sigma$ and $p_{\text{gamma}}(x; a, b)$ is the probability density at $x$ of the gamma distribution with shape $a$ and rate $b$.

We will use a Metropolis within block Gibbs approach. The sampling/updating blocks are:

- $\{\kappa_{ijk}\}$
- $\sigma_\alpha, \{\sigma_{\beta\ell}\}, \{\sigma_{\xi\ell}\}, \sigma_\epsilon$
- $\{\phi_i\}$
- $\nu_\phi$ and $\nu_\kappa$
- $\mu, \{\alpha_i\}, \{\beta_{\ell t}\}, \{\psi_{i\ell t}\}$ (the linear predictor)

### B.3.1 Updating the $\kappa$ block

We can see that $\kappa_{ijk}$ only appears in

$$p_{\text{gamma}}(\kappa_{ijk}; \tfrac{1}{2}\nu_\kappa, \tfrac{1}{2}(\nu_\kappa - q_\kappa)) p_{\text{normal}}(y_{ijk}; \mu_{ij}, \sigma_\epsilon/\kappa_{ijk}^{\frac{1}{2}})$$

so that

$$\begin{aligned} p(\kappa_{ijk} \mid \dots) &\propto \kappa_{ijk}^{\frac{1}{2}\nu_\kappa - 1} \exp\{-\tfrac{1}{2}(\nu_\kappa - q_\kappa)\kappa_{ijk}\} \kappa_{ijk}^{\frac{1}{2}} \exp\{-\tfrac{1}{2}\kappa_{ijk}(y_{ijk} - \mu_{ij})^2/\sigma_\epsilon^2\} \\ &= \kappa_{ijk}^{\frac{1}{2}(\nu_\kappa+1)-1} \exp\{-\tfrac{1}{2}\kappa_{ijk}[\nu_\kappa - q_\kappa + (y_{ijk} - \mu_{ij})^2/\sigma_\epsilon^2]\} \end{aligned}$$

Thus the conditional distribution of $\kappa_{ijk}$ is $\Gamma(\frac{1}{2}(\nu_\kappa + 1), \frac{1}{2}[\nu_\kappa - q_\kappa + (y_{ijk} - \mu_{ij})^2/\sigma_\epsilon^2])$.

The $\kappa_{\dots}$ values are conditionally independent and it is computationally more efficient in a language such as R to sample them in a single line of code so as to avoid explicit looping.

### B.3.2 Updating the $\sigma$ block

Except for $\sigma_\epsilon$, each $\sigma$ appears only in the normal pdfs for the quantities whose variability it controls, and even the case of $\sigma_\epsilon$ requires only a minor change. Therefore the updates are all essentially the same, depending only on how many quantities are controlled by the parameter and the sum of squares of of those quantities.

Consider the $\sigma_\epsilon$ case. We have

$$\begin{aligned} p(\sigma_\epsilon \mid \dots) &\propto p(\sigma_\epsilon) \prod_{(i,j)\in\mathcal{J}_\mathcal{I}} \prod_{k=1}^{K_{ij}} p_{\text{normal}}(y_{ijk}; \mu_{ij}, \sigma_\epsilon/\kappa_{ijk}^{\frac{1}{2}}) \\ &\propto \sigma_\epsilon^{-1} \sigma_\epsilon^{-K} \exp\{-\frac{1}{2}\sigma_\epsilon^{-2} \sum_{(i,j)\in\mathcal{J}_\mathcal{I}} \sum_{k=1}^{K_{ij}} \kappa_{ijk}(y_{ijk} - \mu_{ij})^2\} \end{aligned}$$

where $K = \sum_{(i,j)\in\mathcal{J}_\mathcal{I}} K_{ij}$.

Changing variable to $\tau_\epsilon = 1/\sigma_\epsilon$ with Jacobian $\tau_\epsilon^{-3/2}$, we see that the conditional distribution of $\tau_\epsilon$ is $\Gamma(\frac{1}{2}K, \frac{1}{2}\sum_{k=1}^{K_{ij}} \kappa_{ijk}(y_{ijk} - \mu_{ij})^2)$.

What's really happening here is that $\sigma_\epsilon$ is the standard deviation of the normally distributed quantity $e_{ijk} = \epsilon_{ijk}/\kappa_{ijk}^{\frac{1}{2}}$ and the resulting gamma distribution is really $\Gamma(\frac{1}{2}n_e, \frac{1}{2}S_e)$ where $n_e$ is the number of $e_{\dots}$ and $S_e$ is the sum of their squares.

Similarly, we find the following conditional distributions:

- $\tau_\alpha \,|\, ... \sim \Gamma(\frac{1}{2}(n_\alpha - 1), \frac{1}{2}S_\alpha)$ where $n_\alpha = |\mathcal{I}|$ and $S_\alpha = \sum_{i\in\mathcal{I}} \alpha_i^2$.
- $\tau_{\beta\ell} \,|\, ... \sim \Gamma(\frac{1}{2}(n_{\beta_\ell} - 1), \frac{1}{2}S_{\beta_\ell})$ where $n_{\beta_\ell} = |\mathcal{L}_\ell|$ and $S_{\beta_\ell} = \sum_{t\in\mathcal{L}_\ell} \beta_{\ell t}^2$.
- $\tau_{\xi\ell} \,|\, ... \sim \Gamma(\frac{1}{2}(n_{\xi_\ell} - 1), \frac{1}{2}S_{\xi_\ell})$ where $n_{\xi_\ell} = \sum_{i\in\mathcal{I}} |\mathcal{L}_{i\ell}|$ and $S_{\xi_\ell} = \sum_{i\in\mathcal{I}} \sum_{t\in\mathcal{L}_{i\ell}} \xi_{i\ell t}^2 = \sum_{i\in\mathcal{I}} \sum_{t\in\mathcal{L}_{i\ell}} \psi_{i\ell t}^2/\phi_i^2$.

Unlike $\tau_\epsilon$, each of these has one less degree of freedom than the number of quantities because the prior on the corresponding standard deviation was uniform.

Note that the $\sigma$-parameters are conditionally independent given everything else.

### B.3.3 Updating the $\phi$ block

$\lambda_i$ only appears in

$$p_{\text{gamma}}(\lambda_i; \tfrac{1}{2}\nu_\phi, \tfrac{1}{2}(\nu_\phi - q_\phi)) \prod_{\ell=1}^{L} \prod_{t\in\mathcal{L}_{i\ell}} p_{\text{normal}}(\psi_{i\ell t}; 0, \phi_i\sigma_{\xi\ell})$$

so that

$$p(\lambda_i \,|\, \dots) \propto \lambda_i^{\frac{1}{2}\nu_\phi - 1} \exp\{-\tfrac{1}{2}(\nu_\phi - q_\phi)\lambda_i\} \lambda_i^{\frac{1}{2}n_{\lambda_i}} \exp\{-\tfrac{1}{2}\lambda_i \sum_{\ell=1}^{L} \sum_{t\in\mathcal{L}_{i\ell}} \psi_{i\ell t}^2/\sigma_{\xi\ell}^2\}$$

$$= \lambda_i^{\frac{1}{2}(\nu_\phi + n_{\lambda_i}) - 1} \exp\{-\tfrac{1}{2}\lambda_i[\nu_\phi - q_\phi + \sum_{\ell=1}^{L} \sum_{t\in\mathcal{L}_{i\ell}} \psi_{i\ell t}^2/\sigma_{\xi\ell}^2]\}$$

where $n_{\lambda_i} = \sum_{\ell=1}^{L} |\mathcal{L}_{i\ell}|$

Thus the conditional distribution of $\lambda_i$ given everything else is $\Gamma(\frac{1}{2}[\nu_\phi + n_{\lambda_i}], \frac{1}{2}[\nu_\phi - q_\phi + \sum_{\ell=1}^{L} \sum_{t\in\mathcal{L}_{i\ell}} \psi_{i\ell t}^2/\sigma_{\xi\ell}^2])$. Again, the $\lambda_i$ are conditionally independent and so it may be more efficient to sample them all in a a single line of code.

### B.3.4 Updating the $\nu$ block

The $\nu_\phi$ and $\nu_\kappa$ updates have essentially the same structure. First, consider the situation where we observe $x_1, \dots, x_n$ where $x_i \,|\, \nu \sim p_{\text{gamma}}(x_i; \frac{1}{2}\nu, \frac{1}{2}(\nu - q))$. Then the conditional pdf of $\nu$ is proportional to

$$p(\nu) \left[\frac{(\frac{1}{2}[\nu - q])^{\nu/2}}{\Gamma(\nu/2)}\right]^n (\prod_i x_i)^{\nu/2} \exp\{-\tfrac{1}{2}\nu \sum x_i\}$$

which on taking the (natural) logarithm becomes

$$\log p(\nu) + n\left\{\tfrac{1}{2}\nu \log(\tfrac{1}{2}[\nu - q]) - \log\Gamma(\tfrac{1}{2}\nu) + \tfrac{1}{2}\nu\overline{\log x} - \tfrac{1}{2}\nu\overline{x}\right\}$$

We can easily use a Metropolis random walk to update $\nu$. We choose to so by making appropriately sized normally distributed random steps on $1/\nu$ restricted[5] to some appropriate interval. This corresponds to putting a uniform prior on $1/\nu$ on that interval.

For $\nu_\kappa$, the equivalent of $x_i$ is $\kappa_{ijk}$ and $n = K$ (defined earlier).

For $\nu_\phi$, the equivalent of $x_i$ is $\lambda_i$ and $n = |\mathcal{I}|$.

[5]To be precise, we make proposals outside the interval but those are always rejected.

### B.3.5 Updating the linear predictor block

We are now left with the problem of updating the parameters of the linear predictor: $\mu$, $\alpha_\cdot$, $\beta_{\cdot\cdot}$, $\psi_{\cdots}$.

Fortunately, when the other parameters are known, the model is a standard linear mixed model with heterogeneous errors having known heterogeneity. Many algorithms have been proposed but the algorithm in MCMCglmm (2010) is highly efficient, making clever use of sparse matrix computations. See appendix C for the details.

We need to formulate our linear predictor update problem as a linear mixed model. Writing $y$ and $\epsilon$ for the vectors formed respectively from all the $y_{ijk}$ and all the $\epsilon_{ijk}$, and $\vartheta$ for the vector formed from all the linear predictor parameters, we have

$$y = W\vartheta + \epsilon$$

where $W$ is a constant matrix implementing (4) for the database, $\vartheta$ has a multivariate normal distribution prior with known mean $\vartheta_0$ and known variance $\Sigma$, and $\epsilon$ is a vector of independent mean zero normal components having known standard deviations. In this framework, we can efficiently obtain a sample from the distribution of $\vartheta$ given $y$ by the method described in detail in appendix C. An important contributing factor to the efficiency is that the matrix $W$ is sparse (many zeroes) and the pattern of sparseness does not change between MCMC iterations.

$\Sigma$ and $W$ may be constructed as follows:

- Let us order the parameters according to the blocks listed at the start of this section and assume that we have a specified ordering for the elements of $\mathcal{I}$ and for each $\mathcal{L}_\ell$ for $\ell = 1, \ldots L$. Then we will order: (i) $\alpha_\cdot$ to correspond to the ordering of $\mathcal{I}$; (ii) $\beta_{\cdot\cdot}$ in order first of increasing $\ell$ and then by the specified ordering of $\mathcal{L}_\ell$; and (iii) $\psi_{\cdots}$ first by $\ell$, then by $\mathcal{I}$ and then within each $\mathcal{L}_{i\ell}$ to be consistent with the ordering on $\mathcal{L}_\ell$ (uniquely defined since $\mathcal{L}_{i\ell} \subseteq \mathcal{L}_\ell$).
- For this ordering, $\Sigma^{1/2}$ has a very simple form: it is diagonal with the following sequence of elements: (i) prior standard deviation of $\mu$; (ii) $\sigma_\alpha$ repeated $|\mathcal{I}|$ times; (iii) $\sigma_{\beta\ell}$ repeated $T_\ell$ times for $\ell = 1, \ldots, L$ and arranged in order of increasing $\ell$; (iv) $\phi_i\sigma_{\xi\ell}$ repeated $|\mathcal{L}_{i\ell}|$ times for $i \in \mathcal{I}$ and $\ell = 1, \ldots, L$ and arranged in order of increasing $\ell$ and within each $\ell$ in the order of the elements of $\mathcal{I}$.
- We can order the components of $y$ in any order; we just need to be consistent and to be able to determine $i$, $j$ and $k$ for each element. The matrix $W$ is then determined by the orderings of the components of $y$ and $\vartheta$. The standard deviation of each $\epsilon_{ijk}$ is $\sigma_\epsilon/\kappa_{ijk}^{\frac{1}{2}}$.

### B.3.6 Censored data

Censoring is easily handled: we sample values for censored $y_{ijk}$ once in each MCMC loop. Denoting the lower and upper bounds (possibly infinite) from censoring by $L_{ijk}$ and $U_{ijk}$ respectively, we sample $y_{ijk}$ from the normal distribution $\mathrm{N}(\mu_{ij}, \sigma_\epsilon^2/\kappa_{ijk})$ truncated to the interval $[L_{ijk}, U_{ijk}]$.

### B.3.7 Initial values

For initial values when working in R, I made the following choices:

- In order to avoid a complicated calculation, I simply initialised each $\sigma$ to 1 with the exception of $\sigma_\epsilon$ which I set to 0.5. These choices mean that enough early output (burn-in) from the chain must be discarded to be sure that the chain has reached equilibrium.
- All $\phi_i$ were initialised to 1.
- All $\kappa_{ijk}$ were initialised to 1.
- More recent code starts with $\nu_\phi = 1$ and $\nu_\kappa = 2$.
- Initial values of $y_{ijk}$ for censored measurements were taken to be the censoring value for left or right censored data and the interval mid-point for interval censored data.

My main goal was to avoid having to provide initial values for the linear predictor parameters; in principle these could have been initialised by output from fitting a version of the model without $\phi_i$ by REML (restricted maximum likelihood) using the `lme4` package for R.

### B.3.8 Ordering of blocks

The initial values described are sufficient to enable the linear predictor block to be sampled. After that, many orderings are possible and I made the following arbitrary choice of order: censored data values, $\kappa$ block, $\sigma_\epsilon$, $\nu_\kappa$, remainder of the $\sigma$ block, $\phi$ block, $\nu_\phi$.

## B.4 Output from analysing the database

We use the HDF5 format for transferring data. HDF5 is a widely used system for handling/storing hierarchical numerical data and has working interfaces in both R and Matlab on both Linux and Windows.

Entries in HDF5 files are named rather like files on a UNIX system (or Windows except that "/" replaces "\").

Here is the current structure which can evolve easily by changing version numbers if we run into difficulties.

| HDF path | Type of object | Value |
|---|---|---|
| /Version | single integer | currently 1 |
| /MetaScenario | group | |
| /Model | group | |
| /Posterior | group | |

### B.4.1 Meta-scenario details

The meta-scenario is a collection of species which must include at least all the species in the database but may include others if available. In the MetaScenario group in the HDF5 file, we provide a list of those and their full taxonomic classification.

By the term full classification, we do not mean to make a judgement about correct classification practice. Its practical meaning is that all versions of the model fitted to the database, when choosing the classification levels to include in the final model, uses a subset of the levels in the full classification.

Meta-scenario notation:

- $L^*$ is the number of levels in the full classification being used in the meta-scenario.
- $\ell^*$ indexes levels in the classification. It ranges from 1 (coarsest, probably kingdom) to $L^*$ (finest, probably species). Each higher level must be a (possibly trivial) refinement of the previous level.
- $T^*_{\ell^*}$ is the number of different classifications found in the meta-scenario at level $\ell^*$.

The MetaScenario group in the HFD5 file has the following structure:

| HDF path | Type of object | Value | Notation |
|---|---|---|---|
| /Version | single integer | currently 1 | |
| /ScenarioType | single integer | currently 1 | |
| /ScenarioName | character string | | |
| /ScenarioDescription | character string | | |
| /NumberOfSpecies | single integer | | $M$ |
| /LatinNames | vector of $M$ character strings | | |
| /NumberOfClassificationLevels | single integer | | $L^*$ |
| /NamesOfClassificationLevels | vector of $L^*$ character strings | | |
| /NamesOfClassifications | group with $L^*$ entries | | |
| /NamesOfClassifications/1 | vector of $T^*_1$ character strings | | |
| /NamesOfClassifications/2 | vector of $T^*_2$ character strings | | |
| ⋮ | | | |
| /CodedClassification | $M$ by $L^*$ matrix | encoded so that the $i$th row gives the classification for the $i$th latin name and the $j$th column in that row is the position of its classification in "/NamesOfClassifications/*j*" | |

### B.4.2 Model description

This defines the model being used and names and describes it. It is "aware" of the meta-scenario. For the current model, it also defines $L$ and how the levels in the model correspond to levels in the meta-scenario and gives values for $q_\kappa$ and $q_\nu$. The correspondence of taxonomic levels in the model to levels in the full classification in the meta-scenario is determined by $\ell_1^* < \cdots < \ell_L^*$ where $\ell_\ell^*$ is the index in the full classification of level $\ell$ used in the model.

The Model group in the HFD5 file has the following structure:

| HDF path | Type of object | Value | Notation |
|---|---|---|---|
| /Version | single integer | currently 1 | |
| /ModelType | single integer | currently 1 (the model described above) | |
| /ModelName | character string | | |
| /ModelDescription | character string | | |
| /NumberOfTaxonomicLevels | single integer | | $L$ |
| /TaxonomicLevels | vector of $L$ increasing values in $1, \ldots, L^*$ | | $\ell_1^* < \cdots < \ell_L^*$ |
| /ClassificationsUsed | group with $l$ entries | | |
| /ClassificationsUsed/1 | vector of $T_1$ integers | values selecting entries from classification level $\ell_1^*$ in the meta-scenario | |
| /ClassificationsUsed/2 | vector of $T_2$ integers | values selecting entries from classification level $\ell_2^*$ in the meta-scenario | |
| ⋮ | | | |
| /Qkappa | number | | |
| /Qphi | number | | |

### B.4.3 Posterior distribution

This defines the type of posterior being provided (currently an MCMC sample). For the current type of posterior, it then indicates the number of samples and gives the sample data. It is "aware" of the model being used. The current description of the "Samples" group is for model type 1.

We need to output all parameters which may be needed in when dealing with a new chemical. These are $\mu$, $\sigma_\alpha$, $\sigma_{\beta 1}, \ldots, \sigma_{\beta L}$, $\{\beta_{1.}\}, \ldots, \{\beta_{L.}\}$, $\nu_\phi$, $\sigma_{\xi 1}, \ldots, \sigma_{\xi L}$, $\sigma_\epsilon$ and $\nu_\kappa$. The reason we need the actual $\beta$-values and not the actual $\phi$-values is that the former apply to all chemicals whereas the latter are chemical-specific. The reason we might also need the $\sigma_\beta$ parameters is that they provide the distribution of tendencies for

species which are not in the database but which for which test data are available for a new chemical.

The Posterior group in the HFD5 file has the following structure:

| HDF path | Type of object | Value | Notation |
|---|---|---|---|
| /Version | single integer | currently 1 | |
| /PosteriorType | single integer | currently 1 (denotes MCMC sample) | |
| /NoOfSamples | single integer | | $n$ |
| /Samples | group | | |
| /Samples/mu | vector of $n$ numbers | | $\mu$ |
| /Samples/nu | group | | |
| /Samples/nu/kappa | vector of $n$ numbers | | $\nu_\kappa$ |
| /Samples/nu/phi | vector of $n$ numbers | | $\nu_\phi$ |
| /Samples/beta | group | | |
| /Samples/beta/1 | $n$ by $T_1$ matrix | | $\beta_{1.}$ |
| /Samples/beta/2 | $n$ by $T_2$ matrix | | $\beta_{2.}$ |
| ⋮ | | | |
| /Samples/sigma | group | | |
| /Samples/sigma/epsilon | vector of $n$ numbers | | $\sigma_\epsilon$ |
| /Samples/sigma/alpha | vector of $n$ numbers | | $\sigma_\alpha$ |
| /Samples/sigma/beta | $n$ by $L$ matrix | | $\sigma_{\beta 1}, \ldots, \sigma_{\beta L}$ |
| /Samples/sigma/xi | $n$ by $L$ matrix | | $\sigma_{\xi 1}, \ldots, \sigma_{\xi L}$ |

### B.5 Algorithm for MCMC sampling for a new chemical

#### B.5.1 More notation

- The collection of all the database data is now $y$ and all the parameters in the model for the database are $\theta$ (as before).
- The data for the new chemical are $y_{0jk}$ (collectively $y_0$). The values of $(j, k)$ involved will be referred to as "measured".
- Any extra parameters for the new chemical are $\theta_0$. $\theta_0$ contains two groups of parameters: those needed for the construction of $y_{0jk}$ and those needed for the construction of $\mu_{0j}$ for all species in the scenario; in both cases, we omit those parameters which already appear in $\theta$.
- Let $\zeta$ be the subset of $\theta$ given which $y_0, \theta_0$ and $y$ are independent. Denote the remainder of $\theta$ by $\gamma$.
- Let $\zeta_0$ be the subset of $\theta_0$ which actually appear in $p(y_0 \mid \theta_0, \zeta)$ and let $\gamma_0$ be the rest of the parameters in $\theta_0$.
- "Relevant" $(\ell, t)$ are those from the database which actually appear for species in the scenario or the data for the new chemical.

- "Novel" $(\ell, t)$ are those which appear in the scenario or data for the new chemical but which are not in the database.
- "Active" $(\ell, t)$ are all those which are either relevant or novel.
- "Measured" $(\ell, t)$ are all $(\ell, t_\ell(j))$ for measured $j$.

### B.5.2 The parameter groups in detail

- What's in $\zeta$? $\mu$, all the $\sigma$ and $\nu$ parameters and $\beta_{\ell t}$ for all relevant $(\ell, t)$.
- What's in $\gamma$? $\beta_{\ell t}$ for irrelevant $(\ell, t)$ in the database and all $\kappa_{\ldots}$, $\phi_i$ and $\phi_{i\ell t}$ for the database.
- What's in $\zeta_0$? $\kappa_{0jk}$ for measured $(j, k)$, $\alpha_0$, $\phi_0$, $\psi_{0\ell t}$ for all measured $(\ell, t)$ and $\beta_{\ell t}$ for all measured novel $(\ell, t)$.
- What's in $\gamma_0$? $\psi_{0\ell t}$ for all unmeasured active $(\ell, t)$ and $\beta_{\ell t}$ for all unmeasured novel $(\ell, t)$

From these parameter groups, we can obtain the following decomposition: (corresponding to a DAG):

$$p(\zeta, \gamma, y, \zeta_0, \gamma_0, y_0) = p(y \mid \gamma, \zeta)p(\gamma \mid \zeta)p(\zeta)p(\gamma_0 \mid \zeta)p(\zeta_0 \mid \zeta)p(y_0 \mid \zeta_0, \zeta) \tag{8}$$

### B.5.3 Structure of algorithm

To make inference about the scenario-specific SSD, in particular the scenario-specific $HC_p$ for an allowable value of $p$, we need the joint posterior distribution of all the true sensitivities $\mu_{0j}$ to the new chemical for species in the scenario. This can be obtained by marginalisation from $p(\theta_0, \zeta \mid y, y_0)$. Taking a sample from $p(\theta_0, \zeta \mid y, y_0)$ would suffice.

Moreover, integrating out $\gamma$ from (8) and rewriting $p(y \mid \zeta)p(\zeta)$ as $p(\zeta \mid y)p(y)$, we arrive at

$$p(\theta_0, \zeta \mid y, y_0) = p(\gamma_0 \mid \zeta)p(\zeta_0, \zeta \mid y, y_0)$$

and

$$p(\zeta_0, \zeta \mid y, y_0) \propto p(y_0 \mid \zeta_0, \zeta)p(\zeta_0 \mid \zeta)p(\zeta \mid y)$$

Looking closely at $p(\gamma_0 \mid \zeta)$, we can see all the components of $\gamma_0$ are conditionally independent and sampled from normal distributions. If we can obtain a sample from $p(\zeta_0, \zeta \mid y, y_0)$, it is then trivial to extend it to a sample from $p(\theta_0, \zeta \mid y, y_0)$

Now, from stage 3 we have a random sample from $p(\theta \mid y)$ and so, simply by omitting some variables, that is a random sample from $p(\zeta \mid y)$ which can therefore be used directly as a mechanism in an MCMC algorithm for making proposals for $\zeta$ from $p(\zeta \mid y)$ as part of a Metropolis-Hastings update step. Applying the usual formula, the acceptance ratio would then be

$$\frac{p(y_0 \mid \zeta_0, \zeta_{\text{proposed}})p(\zeta_0 \mid \zeta_{\text{proposed}})}{p(y_0 \mid \zeta_0, \zeta_{\text{old}})p(\zeta_0 \mid \zeta_{\text{old}})}$$

and we shall see that this is easily computed. There is a possibility that the acceptance rate will be low, in which case a more intelligent proposal mechanism may be needed — perhaps something which looks a little like a random walk over key variables.

We shall also need a mechanism for making updates of $\zeta_0$. This is essentially the same problem as updating $\theta$ in the MCMC algorithm used in stage 3 for sampling from the posterior for the database.

### B.5.4 Nitty-gritty

Then

$$p(y_0 \mid \zeta_0, \zeta) = \prod_{\text{measured } (j,k)} p_{\text{normal}}(y_{0jk}; \mu_{0j}, \sigma_\epsilon / \kappa_{0jk}^{\frac{1}{2}})$$

where $\mu_{0j}$ may be calculated using (4), (5), (6) and (7), and

$$\begin{aligned} p(\zeta_0 \mid \zeta) = & p_{\text{normal}}(\alpha_0; 0, \sigma_\alpha) \\ & \times p_{\text{gamma}}(\lambda_0; \tfrac{1}{2}\nu_\phi, \tfrac{1}{2}(\nu_\phi - q_\phi)) \\ & \times \prod_{\text{measured } (j,k)} p_{\text{gamma}}(\kappa_{0jk}; \tfrac{1}{2}\nu_\kappa, \tfrac{1}{2}(\nu_\kappa - q_\kappa)) \\ & \times \prod_{\text{measured } (\ell,t)} p_{\text{normal}}(\psi_{0\ell t}; 0, \phi_0 \sigma_{\xi\ell}) \\ & \times \prod_{\text{measured novel } (\ell,t)} p_{\text{normal}}(\beta_{\ell t}; 0, \sigma_{\beta\ell}) \end{aligned}$$

How to update $\zeta_0$ in an MCMC algorithm?

- $\kappa_{0jk}$ — this is just the same as for the main algorithm: sample from $\Gamma(\frac{1}{2}(\nu_\kappa + 1), \frac{1}{2}[\nu_\kappa - q_\kappa + (y_{0jk} - \mu_{0j})^2/\sigma_\epsilon^2])$.
- $\phi_0$ — this is just the same as for the main algorithm: sample from $\Gamma(\frac{1}{2}[\nu_\phi + n_{\lambda_0}], \frac{1}{2}[\nu_\phi - q_\phi + \sum_{\text{measured } (\ell,t)} \psi_{0\ell t}^2/\sigma_{\xi\ell}^2])$ where $n_{\lambda_0}$ is the total number of measured $(\ell, t)$.
- $\alpha_0$, measured novel $\beta_{\ell t}$, measured $\psi_{0\ell t}$ — again similar for the main algorithm. Write

  $$y_0 = W_1\vartheta_1 + W_2\vartheta_2 + \epsilon_0$$

  where: (i) $\vartheta_2$ consists of $\alpha_0$, measured novel $\beta_{\ell t}$ and measured $\psi_{0\ell t}$; (ii) $\vartheta_1$ consists of those components of $\zeta$ needed to compute $\mu_{0j}$ for measured $j$, i.e. $\mu$ and measured relevant $\beta_{\ell t}$; (iii) $\epsilon_0$ is the vector of all $\epsilon_{0jk}$ for measured $(j, k)$.

  Then $\vartheta_1$ is known and we can apply the algorithm from appendix C taking $y = y_0 - W_1\vartheta_1$, $W = W_2$, $\vartheta = \vartheta_2$ and $\varepsilon = \epsilon_0$.

  There may or may not be much benefit in exploiting sparseness $W_2$. Given that Matlab support for CHOLMOD is currently incomplete[6], we just use ordinary Cholesky factorisation and back-solving.

[6]However, examination of the scipy interface suggests that extended the current Matlab interface should be fairly straightforward.

Having updated $\zeta$ and $\zeta_0$, we should then sample $\gamma_0$ and compute $\mu_{0j}$ for all species in the scenario. We then compute the percentiles of these $\mu_{0j}$ values as the values of $HC_p$ for various values of $p$; see section 4.3 for a discussion of the issue. Iterating the MCMC loop many times, we obtain a collection of $HC_p$ values which represents the posterior uncertainty about the scenario-specific $HC_p$ for the new chemical.

### B.5.5 Censoring

The fore-going assumes that there are no censored data values for the new chemical. As with the database analysis, a censored data value $[L_{0jk}, U_{0jk}]$ simply adds an extra node to the Bayesian network which has $y_{0jk}$ has its only parent. Consequently we can proceed by data augmentation and including a step to sample all censored $y_{0jk}$ in the MCMC loop.

As for the main algorithm, we initialise each censored $y_{0jk}$ to be the mid-point between $L_{0jk}$ and $U_{0jk}$ if both are finite or to be the unique finite value if only one is finite.

In order to sample $y_{0jk}$, we need to know $\mu_{0jk}$, $\sigma_\epsilon$ and $\kappa_{0jk}$. A convenient point in the MCMC sequence to do so is when we have just computed all the true sensitivities to the chemical for the purpose of calculating the $HC_5$.

## C Hadfield's problem/algorithm

This appendix is a restatement (with correction) of the method described in appendix A.2 of MCMCglmm (2010)

Suppose that $y = W\vartheta + \varepsilon$ where (prior distribution) $\vartheta \sim \mathrm{N}(\vartheta_0, \Sigma)$ and $\varepsilon \sim \mathrm{N}(0, R)$ are independent and the matrix $W$ is of full rank. We assume also that $\vartheta_0$ is known and that $\Sigma$ and $R$ are known positive definite symmetric matrices

The question is how to simulate (efficiently) from $p(\vartheta \mid y)$.

### C.1 The algorithm

A standard result for the multivariate normal shows that $\vartheta \mid y \sim \mathrm{N}(C^{-1}[\Sigma^{-1}\vartheta_0 + W^{\mathrm{T}}R^{-1}y], C^{-1})$ where $C = W^{\mathrm{T}}R^{-1}W + \Sigma^{-1}$.

Hence $C\vartheta \mid y \sim \mathrm{N}(\Sigma^{-1}\vartheta_0 + W^{\mathrm{T}}R^{-1}y, C)$ since $C$ is symmetric. If we can simulate from this distribution and then (effectively) pre-multiply by $C^{-1}$ we are done.

Hadfield uses a method for simulating $\vartheta \mid y$ which he attributes to Garcia-Cortes and Sorensen (2001) and then gets slightly wrong in his own text. Here's my account of his method:

- Simulate $\vartheta^*$ and $\varepsilon^*$ from the prior.
- Set $y^* = W\vartheta^* + \varepsilon^*$
- Compute $\tilde{\vartheta} = C^{-1}W^{\mathrm{T}}R^{-1}(y - y^*)$.
- Set $\vartheta^\dagger = \tilde{\vartheta} + \vartheta^*$ to be the simulation of $\vartheta \mid y$

Since, it is clearly multivariate normal, we just need to show that $C\vartheta^\dagger$ has the right mean and variance.

But

$$\begin{aligned} C\vartheta^\dagger &= C\tilde{\vartheta} + C\vartheta^* \\ &= W^{\mathrm{T}}R^{-1}y - W^{\mathrm{T}}R^{-1}y^* + W^{\mathrm{T}}R^{-1}W\vartheta^* + \Sigma^{-1}\vartheta^* \\ &= W^{\mathrm{T}}R^{-1}y - W^{\mathrm{T}}R^{-1}W\vartheta^* - W^{\mathrm{T}}R^{-1}\varepsilon^* + W^{\mathrm{T}}R^{-1}W\vartheta^* + \Sigma^{-1}\vartheta^* \\ &= W^{\mathrm{T}}R^{-1}y + \Sigma^{-1}\vartheta_0 + \Sigma^{-1}(\vartheta^* - \vartheta_0) - W^{\mathrm{T}}R^{-1}\varepsilon^* \end{aligned}$$

The first two terms in the final expression are the required mean and have zero variance, the third term is $\mathrm{N}(0, \Sigma^{-1})$ and the final term is $\mathrm{N}(0, W^{\mathrm{T}}R^{-1}W)$ so that the whole expression has the required mean and variance.

### C.2 Exploiting the algorithm

What makes the algorithm nice is that everything is very easy and computationally efficient except for the apparent need to multiply by $C^{-1}$: in our (and Hadfield's) framework, $W$ is sparse; $\Sigma$ and $R$ are both diagonal.

Moreover, a good way to calculate $\tilde{\vartheta} = C^{-1}x$ is to numerically solve $C\tilde{\vartheta} = x$ in an efficient manner. When $C$ is positive definite and symmetric, Cholesky factorisation

of $C$ followed by back-solving generally works well. It is even more efficient when $C$ is sparse as it tends then also to have a sparse Cholesky factor for a good choice of pivoting order. Finally, Cholesky factorisation of many such matrices requires much less effort if the the sparseness of $C$ is known to be the same each time.

The `Matrix` package for R contains support for sparse matrices, repeated Cholesky factorisation of the same sparseness structure and back-solving using sparse factors. It is in fact based on CHOLMOD by Tim Davis (as in Davis (2006)).

In fact, because of an implementation detail, I slightly adapt the algorithm. Write $C_* = \Sigma^{1/2} C \Sigma^{1/2} = \Sigma^{1/2} W^{\mathrm{T}} R^{-1} W \Sigma^{1/2} + I$. Then $C_* \Sigma^{-1/2} \tilde{\vartheta} = \Sigma^{1/2} W^{\mathrm{T}} R^{-1} (y - W\vartheta^* - \epsilon^*)$. The most fundamental object here is $H_* = \Sigma^{1/2} W^{\mathrm{T}} R^{-1/2}$ so that $C_* = H_* H_*^{\mathrm{T}} + I$ and we find $\tilde{\vartheta}$ by first solving $C_* \tilde{\vartheta}_* = H_* R^{-1/2} (y - W\vartheta^* - \epsilon^*)$ equation to find $\tilde{\vartheta}_*$ and then setting $\tilde{\vartheta} = \Sigma^{1/2} \tilde{\vartheta}_*$.

The reason for doing this is that CHOLMOD provides support for Cholesky factorisation and back-solving for matrices of the form $AA^{\mathrm{T}} + bI$ where one just passes in the matrix $A$: for us $A = H_*$ above and $b = 1$.

## D R code to obtain initial posterior

```
deviance.hetpsi4 = function(
  sigma.epsilon, nu.kappa, mu.y,
  y, have.censored, censored, yL, yU,
  q.kappa) {

  nu.kappa.scale.factor = sqrt((nu.kappa-q.kappa)/nu.kappa)
  scale.epsilon = sigma.epsilon * nu.kappa.scale.factor
  loglld1 = -log(scale.epsilon)+dt(
    (y-mu.y)[!censored]/scale.epsilon,
    nu.kappa,
    log=TRUE)
  deviance = numeric(length(y))
  deviance[!censored] = -2*loglld1
  if(have.censored) {
    loglld2 = pintervalt(
      yL, yU,
      nu.kappa,
      center=mu.y[censored],
      scale = scale.epsilon,
      log.p=TRUE
      )
    deviance[censored] = -2*loglld2
  }
  deviance
}

iterate.hetpsi4 = function(
  state0, N, thin=10, burn=0,
  detailed=FALSE, shout=NULL,
  switchtosingle=Inf,
  doublechol=FALSE
  ){

  ## Unpack the state of the chain
  for(n in names(state0)) assign(n, state0[[n]])

  verydetailed = switchtosingle<N
  if(verydetailed) {
    detailed=TRUE
    Nsingle = N-switchtosingle
  }

  ## Storage for results
  deviance.out = numeric(N)
  mu.out = numeric(N)
  sigma.epsilon.out = numeric(N)
```

```
sigma.alpha.out = numeric(N)
sigmas.beta.out = matrix(numeric(N*L), nrow=N)
colnames(sigmas.beta.out) = taxlevels
sigmas.xi.out = matrix(numeric(N*L), nrow=N)
colnames(sigmas.xi.out) = sprintf("%s:CAS", taxlevels)
if(nu.phi.free) nu.phi.out = numeric(N)
if(nu.kappa.free) nu.kappa.out = numeric(N)
if(detailed) {
  beta.out = matrix(numeric(N*sum(ns.beta)), nrow=N)
  # colnames(beta.out) =
}
if (verydetailed) {
  vartheta.out = matrix(numeric(Nsingle*nrow(Wt)), nrow=nrow(Wt))
  rtilde.out = matrix(numeric(Nsingle*n.y), nrow=n.y)
  kappa.out = matrix(numeric(Nsingle*n.y), nrow=n.y)
  ycens.out = matrix(numeric(Nsingle*n.censored), nrow=n.censored)
}

## Needed to support DIC calculation
mu.y.bar = rep(0, n.y)
Dbar.y = rep(0, n.y)

## The baseline Cholesky factor of C_* which is (relatively)
## expensive to compute and which we then update with less effort
## each time round the MCMC loop
basechol = Cholesky(tcrossprod(Wt), Imult=1)

cat("Starting main loop\n")

for(t in seq(-burn+1,N)) {
  if(!is.null(shout))
    if (t%%shout==0)
      cat(sprintf("Iteration %d\n", t))
  for(dummy in 1:thin) {
    sigmatilde = c(
      100,
      rep(sigma.alpha, n.CAS),
      rep(sigmas.beta, ns.beta),
      sigmas.xi[ell.for.psi]*phi[i.for.psi]
      )

    rtilde = sigma.epsilon/sqrt(kappa)

    varthetastar = rnorm(n.vartheta, 0, sigmatilde)
    estar = rnorm(n.y, 0, rtilde)
    SighWtRnegh =
      Diagonal(x=sigmatilde) %*% Wt %*% Diagonal(x=1/rtilde)
    RHS = SighWtRnegh %*%
```

```
  ((y-crossprod(Wt, varthetastar)-estar)/rtilde)
Cstarchol = update(basechol, parent=SighWtRnegh, mult=1)
if(doublechol) {
  Cstarchol2 = update(basechol, parent=SighWtRnegh, mult=1)
  stopifnot(identical(Cstarchol,Cstarchol2))
}
varthetatilde = sigmatilde*solve(Cstarchol, RHS)
vartheta = varthetatilde+varthetastar

## Compute the true sensitivity for each datum
mu.y = as(crossprod(Wt, vartheta), "numeric")
## Sample values for the censored data (if any)
if(have.censored)
  y[censored] = rcensnorm(
     n.censored,
     yL,
     yU,
     mu.y[censored],
     sigma.epsilon/sqrt(kappa[censored])
     )
## Added to debug problem with MCMC failure
stopifnot(all(is.finite(y)))
## Compute epsilon as pre-cursor to deviance and kappa sampling
epsilon = y - mu.y
## Compute the deviance now
deviance.y = deviance.hetpsi4(
  sigma.epsilon, nu.kappa, mu.y,
  y, have.censored, censored, yL, yU,
  q.kappa
  )
## sample kappa values followed by sigma.epsilon
kappa = rgamma(
  n.y,
  (nu.kappa+1)/2,
  (nu.kappa-q.kappa+(epsilon/sigma.epsilon)^2)/2
  )
## Added to debug problem with MCMC failure
stopifnot(all(is.finite(kappa)) && all(kappa>0))
sigma.epsilon = 1/sqrt(rgamma(1, n.y/2, sum(kappa*epsilon^2)/2))

## Update nu.kappa if varying
if (nu.kappa.free)
  nu.kappa = nu.update(
    nu.kappa,
    n.y,
    mean(kappa),
    mean(log(kappa)),
    delta = 0.1,
```

```
      q = q.kappa,
      nu.min = 1
      )

  sigma.alpha = 1/sqrt(rgamma(
    1,
    (n.CAS-1)/2,
    sum(as(vartheta[alpha.lookup]^2, "numeric")/2)
    ))

  sigmas.beta = 1/sqrt(rgamma(
    L,
    (ns.beta-1)/2,
    as((do.sum.beta.by.ell %*% vartheta[beta.lookup]^2)/2,
       "numeric")
    ))

  psi2 = vartheta[psi.lookup]^2

  sigmas.xi = 1/sqrt(rgamma(
    L,
    (ns.psi.by.ell-1)/2,
    as((do.sum.psi.by.ell %*% (psi2/phi[i.for.psi]^2))/2,
       "numeric")
    ))

  phi.sums = do.sum.psi.by.i %*% (psi2/sigmas.xi[ell.for.psi]^2)
  phi = 1/sqrt(rgamma(
    n.CAS,
    (nu.phi+ns.psi.by.i)/2,
    (nu.phi-q.phi+as(phi.sums, "numeric"))/2
    ))

  ## Now should sample nu.phi here

  if (nu.phi.free)
    nu.phi = nu.update(
      nu.phi,
      n.CAS,
      mean(1/phi^2),
      -2*mean(log(phi)),
      delta = 0.2,
      q = q.phi,
      nu.min = 1
      )
}
if(t==switchtosingle) thin = 1
if(t<=0) next
```

```
    ## Save results
    deviance.out[t] = sum(deviance.y)
    mu.out[t] = vartheta[1]
    sigma.epsilon.out[t] = sigma.epsilon
    sigma.alpha.out[t] = sigma.alpha
    sigmas.beta.out[t,] = sigmas.beta
    sigmas.xi.out[t,] = sigmas.xi
    if (nu.phi.free) nu.phi.out[t] = nu.phi
    if (nu.kappa.free) nu.kappa.out[t] = nu.kappa
    if(detailed) beta.out[t,] = vartheta[beta.lookup]
    if(verydetailed && (t>switchtosingle)) {
      vartheta.out[,t-switchtosingle] = as.vector(vartheta)
      rtilde.out[,t-switchtosingle] = rtilde
      kappa.out[,t-switchtosingle] = kappa
      if(have.censored)
      ycens.out[,t-switchtosingle] = y[censored]
    }
    ## Save for computing D(thetabar)
    mu.y.bar = mu.y.bar + (mu.y-mu.y.bar)/t
    Dbar.y = Dbar.y + (deviance.y-Dbar.y)/t
  }

  state = list()
  for(n in names(state0)) state[[n]] = get(n)

  result = cbind(
    mu=mu.out, sigma.epsilon=sigma.epsilon.out,
    sigma.alpha=sigma.alpha.out,
    sigmas.beta.out,sigmas.xi.out,
    deviance=deviance.out
    )
  if (nu.phi.free) result = cbind(result, nu.phi=nu.phi.out)
  if (nu.kappa.free) result = cbind(result, nu.kappa=nu.kappa.out)
  if(detailed) result = cbind(result, beta.out)

  ret = list(state=state, result=result, mu.y.bar=mu.y.bar,
    Dbar.y=Dbar.y)
  if(verydetailed) {
    ret$extradetail = list(
      vartheta = vartheta.out,
      rtilde = rtilde.out,
      kappa = kappa.out)
    if(have.censored)
      ret$extradetail$ycens = ycens.out
  }
  ret
}
```

```
MCMC.hetpsi4 = function(
  N, data, metascenario, taxlevels, thin=10, burn=10,
  Nblock=200,
  nu.phi=NULL, q.phi=NULL,
  nu.kappa=NULL, q.kappa=NULL,
  detailed=FALSE, shout=NULL,
  h5filename=NULL, h5name, h5description,
  switchtosingle=Inf,
  doublechol=FALSE
  ) {

  require(Matrix)
  require(doBy)

  nu.phi.free = is.null(nu.phi)
  if(nu.phi.free && is.null(q.phi)) q.phi = 1
  nu.kappa.free = is.null(nu.kappa)
  if(nu.kappa.free && is.null(q.kappa)) q.kappa = 1

  ## I managed to verify that, omitting the first row, the Wt matrix
  ## is the same as that from lmer in mixed77@Zt. To do so, I had to
  ## construct Wt at the end using the following code. I might also
  ## have needed to be careful about the merge of data and
  ## metascenario messing things up.

  ## Wt = rBind(
  ## Matrix(1, nrow=1, ncol=n.y),
  ## do.call("rBind", rev(Wt.psi)),
  ## Wt.alpha,
  ## do.call("rBind", rev(Wt.beta))
  ## )

  if(!is.null(h5filename)) {
    detailed = TRUE
    require(rhdf5)
    if (file.exists(h5filename)) file.remove(h5filename)
    h5createFile(h5filename)
    h5write(1, h5filename, "Version")
    write.h5.metascenario(metascenario, h5filename)
  }
  ## Get rid of data for species not in the metascenario. Very
  ## important that they have compatible taxonomic classification,
  ## i.e. from same run of encode.
  data = merge(data, metascenario)
  ## Only allow point or censored data
  stopifnot(all(data$conc.ind %in% c("P", "L", "U", "I")))

  ## Consider censoring
```

```
censored = data$conc.ind!="P"
have.censored = any(censored)
n.censored = sum(censored)
yL = data$lconc.low[censored]
yU = data$lconc.upp[censored]
if(have.censored) {
  yL[is.na(yL)] = -Inf
  yU[is.na(yU)] = Inf
  stopifnot(all(yL<=yU))
  stopifnot(all(is.finite(yL)|is.finite(yU)))
}

## Get rid of redundant factor levels now
CAS = factor(data$CAS)
classifications = droplevels(data[taxlevels])

L = length(taxlevels)
ellstar = match(taxlevels, names(metascenario))
taxlevels = taxlevels[order(ellstar)]
names(taxlevels) = taxlevels

if(!is.null(h5filename)) {
  h5model = list(
    Version=1,
    ModelType=1,
    ModelName=h5name,
    ModelDescription=h5description,
    NumberOfTaxonomicLevels=L,
    TaxonomicLevels=ellstar,
    ClassificationsUsed = local({
      x = lapply(
        1:L,
        function(ell) match(levels(classifications[[ell]]),
                            levels(metascenario[[taxlevels[ell]]]))
        )
      names(x) = 1:L
      x
    }),
    Qkappa = q.kappa,
    Qphi = q.phi
    )
  h5write(h5model, h5filename, "Model")
}

n.y = nrow(data)
Wt.alpha = as(CAS, "sparseMatrix")
Wt.beta = list()
Wt.psi = list()
```

```
i.for.psi = list()
#t.for.psi = list()
for(ell in 1:L) {
  Wt.beta[[ell]] = as(classifications[[ell]], "sparseMatrix")
  ti = interaction(classifications[[ell]], CAS, drop=TRUE)
  ti.data.frame = data.frame(t=classifications[[ell]], i=CAS, ti=ti)
  ti.data.frame = unique(ti.data.frame)
  ## Next two lines just make sure that the psi levels are ordered
  ## as described in the algorithms document
  ti.data.frame = orderBy(~i+t, ti.data.frame)
  stopifnot(all(sort(ti.data.frame$ti)==ti.data.frame$ti))
  Wt.psi[[ell]] = as(ti, "sparseMatrix")
  i.for.psi[[ell]] = ti.data.frame$i
  #t.for.psi[[ell]] = ti.data.frame$t
}
n.CAS = nrow(Wt.alpha)
ns.beta = sapply(Wt.beta, nrow)
ns.psi.by.ell = sapply(Wt.psi, nrow)
Wt = rBind(
  Matrix(1, nrow=1, ncol=n.y),
  Wt.alpha,
  do.call("rBind", Wt.beta),
  do.call("rBind", Wt.psi)
  )
# Number of linear predictor parameters
n.vartheta = nrow(Wt)
# Next three lines create vectors for pulling different kinds of
# variance component out of vartheta


alpha.lookup = 1+seq(n.CAS)
beta.lookup = 1+n.CAS+seq(sum(ns.beta))
psi.lookup = 1+n.CAS+sum(ns.beta)+seq(sum(ns.psi.by.ell))

## Used to find matching phi for psi/xi values
i.for.psi = do.call("c", i.for.psi)
# t.for.psi = do.call("c", t.for.psi)
do.sum.psi.by.i = as(factor(i.for.psi), "sparseMatrix")
ns.psi.by.i = rowSums(do.sum.psi.by.i)
stopifnot(sum(ns.psi.by.i)==sum(ns.psi.by.ell))
ell.for.psi = factor(rep(1:L, ns.psi.by.ell))
do.sum.psi.by.ell = as(ell.for.psi, "sparseMatrix")
do.sum.beta.by.ell = as(factor(rep(1:L, ns.beta)), "sparseMatrix")

## return(list(Wt=Wt, i.for.psi=i.for.psi, ell.for.psi=ell.for.psi,
##             t.for.psi = t.for.psi,
##             alpha.lookup=alpha.lookup, beta.lookup=beta.lookup,
##             psi.lookup=psi.lookup, ns.beta = ns.beta,
```

```
##             CAS=CAS, classifications=classifications,
##             do.sum.psi.by.ell = do.sum.psi.by.ell,
##             do.sum.psi.by.i = do.sum.psi.by.i,
##             do.sum.beta.by.ell = do.sum.beta.by.ell))

## Initial values
kappa = rep(1, n.y)
y = data$lconc.low
if(have.censored)
  y[censored] = ifelse(
     is.finite(yL),
     ifelse(is.finite(yU), (yL+yU)/2, yL),
     yU
     )
sigma.alpha = 1
sigma.epsilon = 0.5
sigmas.beta = rep(1, L)
sigmas.xi = rep(1, L)
phi = rep(1, n.CAS)
if(nu.phi.free) nu.phi=1
if(nu.kappa.free) nu.kappa=2

state = list()
for(n in  c("y", "taxlevels", "L",
            "have.censored", "n.censored",
            "censored", "yL", "yU",
            "ns.beta", "n.y", "n.CAS", "n.vartheta",
            "ell.for.psi", "i.for.psi", "Wt",
            "alpha.lookup", "beta.lookup", "do.sum.beta.by.ell",
            "psi.lookup", "do.sum.psi.by.ell", "ns.psi.by.ell",
            "do.sum.psi.by.i", "ns.psi.by.i",
            "sigma.alpha","sigmas.beta", "sigmas.xi", "sigma.epsilon"
            "kappa", "phi",
            "nu.kappa.free", "nu.kappa", "q.kappa",
            "nu.phi.free", "nu.phi", "q.phi"))
  state[[n]] = get(n)

## Perform the burn-in
if (burn>0) {
  iter.result = iterate.hetpsi4(
    state, burn, thin=thin, burn=0, detailed=FALSE, shout=shout,
    doublechol=doublechol)
  state = iter.result$state
}

## And now the main iteration in lumps of Nblock at a time
Ndone = min(N, Nblock)
iter.result = iterate.hetpsi4(
```

```
  state,
  Ndone, thin=thin, burn=0,
  detailed=detailed, shout=shout,
  doublechol=doublechol)
for(n in names(iter.result)) assign(n, iter.result[[n]])
while(N>Ndone) {
  Nnow = min(N-Ndone, Nblock)
  iter.result = iterate.hetpsi4(
    iter.result$state,
    Nnow, thin=thin, burn=0,
    detailed=detailed, shout=shout,
    doublechol=doublechol)
  Ndone = Ndone+Nnow
  result = rbind(result, iter.result$result)
  mu.y.bar = mu.y.bar + Nnow*(iter.result$mu.y.bar-mu.y.bar)/Ndone
  Dbar.y = Dbar.y + Nnow*(iter.result$Dbar.y-Dbar.y)/Ndone
}

# for(n in names(state)) assign(n, state[[n]])

## Save the h5 file if needed
if(!is.null(h5filename)) {
  h5sample = list(
    Version = 1,
    PosteriorType = 1,
    NoOfSamples = N,
    Samples = list(
      mu = mu.out,
      nu = list(
        kappa=nu.kappa.out,
        phi = nu.phi.out
        ),
      beta = local({
        x = lapply(
          1:L,
          function(ell)
            beta.out[,as.logical(do.sum.beta.by.ell[ell,])]
          )
        names(x) = 1:L
        x
      }),
      sigma = list(
        epsilon=sigma.epsilon.out,
        alpha=sigma.alpha.out,
        beta=sigmas.beta.out,
        xi=sigmas.xi.out
        )
      )
```

```
      )
    h5write(h5sample, h5filename, "Posterior")
  }

  list(result=result, state=iter.result$state,
       fit = data.frame(mu.y.bar=mu.y.bar, Dbar.y=Dbar.y),
       y=y, censored=censored, yL=yL, yU=yU, q.kappa=q.kappa
       )
}

deviance.plugin = function(MCMCout, onerun=TRUE) {
  result = MCMCout$result
  sigma.epsilon.bar = exp(mean(log(result[,"sigma.epsilon"])))
  nu.kappa.bar = mean(result[,"nu.kappa"])
  if (onerun)
    mu.y = MCMCout$fit$mu.y.bar
  else
    mu.y = summaryBy(mu.y.bar~.Sequence, MCMCout$fit)$mu.y.bar.mean
  deviance = deviance.hetpsi4(
    sigma.epsilon.bar, nu.kappa.bar, mu.y,
    MCMCout$y, any(MCMCout$censored), MCMCout$censored,
    MCMCout$yL, MCMCout$yU,
    MCMCout$q.kappa
    )
  deviance
}
```