bayesnec: An R Package for C-R Modelling and Estimation of No-Effect-Concentrations

Rebecca Fisher
Australian Institute of Marine Science

Diego Barneche
Australian Institute of Marine Science

Gerard Ricardo
The University of Queensland

David Fox
Environmetrics Australia

Abstract

The **bayesnec** package in R has been developed to fit concentration(dose) - response curves (C-R) to toxicity data for the purpose of deriving no-effect-concentration (*NEC*), no-significant-effect-concentration (*NSEC*), and effect-concentration (of specified percentage 'x', *ECx*) thresholds from non-linear models fitted using Bayesian MCMC fitting methods via **brms** (Bürkner 2017; Bürkner 2018) and **rstan** (Stan Development Team 2020). In **bayesnec** it is possible to fit a single model, custom model-set, specific model-set or all of the available models. When multiple models are specified the `bnec` function returns a model weighted average estimate of predicted posterior values. A range of support functions and methods are also included to work with the returned single, or multi-model objects that allow extraction of raw, or model averaged predicted, *ECx*, *NEC* and *NSEC* values and to interrogate the fitted model or model-set. The statistical methods used mean that the uncertainty in derived threshold values can be robustly quantified, including uncertainty in individual model fits to the underlying observed data, as well as uncertainty in the model functional form.

Keywords: ecotoxicology, effect concentration, threshold derivation, nec, `brms`.

1. Introduction

Concentration-response (C-R) modelling (also known as dose-response modelling or dose-response analysis) is a key tool for assessing toxicity and deriving the toxicity thresholds used in the risk assessments that underpin protection of human health and the environment. It is widely used in the disciplines of pharmacology, toxicology and ecotoxicology, where parametric

non-linear regression methods are used to model response data of interest, with the resulting fitted models used to estimate critical thresholds of concern. These thresholds may be used directly to assess risk (e.g see (Fisher, Walshe, Bessell-Browne, and Jones 2018)), or are subsequently incorporated into a broader population-level risk assessment framework (e.g. (Warne, Batley, van Dam, Chapman, Fox, Hickey, and Stauber 2015)). Typical thresholds derived from C-R modelling include the effect-concentration of defined percentage x (ECx) and the no-effect-concentration (NEC), the latter being the generally accepted preferred option (Fox 2008; Warne *et al.* 2015).

Typically C-R relationships are non-linear, which increases the complexity of model fitting. There are a wide array of packages for fitting non-linear models in R, including the `nls` function in **stats** which uses least squares minimisation and extensions in the **nlme** (Pinheiro, Bates, DebRoy, Sarkar, and R Core Team 2021) package based on maximum likelihood estimation that also allows random effects. While C-R modelling can be carried out using multi-purpose software packages, this can be cumbersome in practice, requiring extensive manual programming to obtain the necessary, often relatively standard outputs. The **drc** package (Ritz, Baty, Streibig, and Gerhard 2016) was developed as a user friendly frequentist based solution to C-R modelling in R, and is currently widely used across a range of disciplines. **drc** implements a broad range of non-linear C-R models, provides facilities for ranking model fit based on AIC (Burnham and Anderson 2002), joint modelling of multiple response curves, and supports a range of estimation procedures (Ritz *et al.* 2016).

Estimates of uncertainty in parameters and derived thresholds are critical for effective integration of threshold estimates into risk assessment and formal decision frameworks (Fisher *et al.* 2018). Bayesian methods allow robust quantification of uncertainty with intuitive and direct probabilistic meaning (Ellison 1996) and are therefore an excellent platform for C-R modelling in most settings. Furthermore, the posterior samples generated through Bayesian model fitting methods provides a rich resource that can be used to explore synergistic and antagonistic impacts (Fisher, Bessell-Browne, and Jones 2019a; Flores, F., Marques, J.A., Uthicke, S., Fisher, R., Patel, F., Kaserzon, S., Negri 2021), propagate endpoint uncertainty (Charles, Wu, and Ducrot 2020; Gottschalk and Nowack 2013), and for testing hypotheses of no-effect (Thomas 2006).

There are a wide array of packages available for Bayesian model fitting via Markov-Chain-Monte-Carlo (MCMC, or similar), including WinBUGS (Lunn, Thomas, Best, and Spiegelhalter 2000), jags (Plummer 2003) and stan (Carpenter, Gelman, Hoffman, Lee, Goodrich, Betancourt, Brubaker, Guo, Li, and Riddell 2017), that seamlessly interface with R through packages such as **R2WinBUGS**, **R2jags** (Su and Yajima 2015) and **rstan** (Stan Development Team 2020). These packages generally require many lines of code to fit even a single model, along with extensive debugging and optimisation that is often time consuming and requires specialist expertise. Several extension packages are becoming available, such as **brms** (Bürkner 2017), that allow a broad range of models to be easily fit using Bayesian methods using simple, **lme4**-like formula syntax. However, even with packages like **brms**, Bayesian model fitting can be difficult to automate across all potential usage cases, particularly with respect to specifying valid initial parameter values and appropriate priors. In addition, as was the motivation for the development of the **drc** package in the frequentist setting, the R code required for fitting non-linear models and extracting the relevant outputs (e.g. NEC, ECx) can be cumbersome in practice, and even more so in the Bayesian setting where model fits contain many posterior samples.

The greater complexity associated with Bayesian model fitting has likely hindered the uptake of Bayesian statistics for C-R threshold derivation across the broader ecotoxicology and toxicology communities, who may not have access to specialist statistical expertise (Fisher, van Dam, Batley, Fox, Harford, Humphrey, King, Menendez, Negri, Proctor, Shao, Stauber, van Dam, and Warne 2019b). The **bayesnec** package provides an accessible interface specifically for fitting *NEC* and other C-R models using Bayesian methods. A variety of models can be specified based on the known distribution of the "concentration" or "dose" variable (the predictor, x) as well as the "response" (y) variable. The model formula, including priors and initial values required to call **brms** are automatically generated based on information contained in the supplied data. A range of tools are supplied to aid the user in interrogating model fits, plotting and generating predicted values, as well as extracting the standard outputs, such as NEC and ECx - either as a full posterior sample or in summary form.

2. Technical details and Usage

This project started with an implementation of the *NEC* model based on that described in (Fox 2010; Pires, Branco, Picado, and Mendonca 2002) using **R2jags** (Su and Yajima 2015). Code for fitting this initial model was developed into the R package **jagsNEC** and expanded to include several C-R models and further generalised to allow a large range of response variables to be modelled using their appropriate statistical distribution. While the original **jagsNEC** implementation supported Gaussian, Poisson, Binomial, Gamma, Negative-binomial and Beta response data, **bayesnec** supports all of these in addition to the Beta-binomial family, implemented as a custom family in **brms**. Furthermore, the new structure implemented using **brms** means **bayesnec** can be readily extended to include any of the **brms** families. In addition to greater flexibility in the available response families, **bayesnec** includes a larger range of alternative *NEC* model types, as well as most typically used smooth C-R models (such as 4-parameter logistic and Weibull models) that have no *NEC* 'step' function but simply model the response as a smooth function of concentration. We have now incorporated most of the commonly used models in frequentist packages such as **drc** (Ritz *et al.* 2016) (please see the Model details vignette for more information on the full list of models currently available in **bayesnec**).

2.1. Model specification

The main working function in **bayesnec** is `bnec`. We have attempted to make the `bnec` function as easy to use as possible, targeting the novice R user. While the package is available on CRAN, here we use the development version by installing directly from github using the package **devtools**:

```
R> devtools::install_github("https://github.com/open-AIMS/bayesnec")
```

A **bayesnec** model can be fit as simply as:

```
R> library(bayesnec)
R> data(nec_data)
R> exmp_fit <- bnec(data = nec_data,
```

```
+                  x_var = "x",
+                  y_var = "y",
+                  model = "decline")
```

Only three arguments must be supplied, including: `data`—a `data.frame` containing the data to use for the model; `x_var`—a `character` indicating the column heading containing the concentration (x) variable; and `y_var`—a `character` indicating the column heading containing the response (y) variable. The next argument `model` is a `character` indicating the desired model (or model-set, see details below) and currently defaults to "all" models available. A large range of arguments to be passed to **brms** or model weighting via `loo` can be specified manually by the user as required for more advanced users.

While the statistical distribution of the response variable (y) can be specified directly, **bayesnec** will automatically try and predict the most appropriate distribution (family) to use based on the characteristics of the provided data, as well as build the relevant model priors (see more details below) and initial values for the **brms** model. Initial values are selected such that they yield predicted values in the range of the observed response (`y_var`) data.

If not supplied, the appropriate family to use will be determined based on the characteristics of the input data. This includes evaluation of class (integer or numeric) as well as the observed range via the function `check_data`. Family predictions include all of the available families (see above) except Negative-binomial and Beta-binomial because these require knowledge on whether the data are overdispersed, which are not assessed prior to model fitting. By default **bayesnec** will use the default link function for the predicted family. Note that in `bnec` the default link for the Gamma family has been set to `log`. If other link functions are required for any of the implemented families, such as `identity` in the case of a Beta family for example, this will need to be specified manually using:

Data that are numeric and scaled from 0 to 1, or 0 to ∞ are assumed to be Beta- and Gamma-distributed respectively. Because both distributions are naturally truncated at 0, if necessary a value that is 0 is increased by $1/10^{\text{th}}$ of the next smallest non-zero value. Similarly, because Beta is also truncated at 1, ones are substituted with 0.999. Data scaled from or $-\infty$ to ∞ are modelled using a Gaussian distribution.

If `y_var` data are integers and a `trials_var` argument is supplied (must also be integer) a Binomial distribution is assumed. If no `trials_var` argument is supplied the `y_var` data are assumed to be Poisson if they are integers. Similar checks are used to assign a likely family to the `x_var` (concentration) data. In this case of `x_var` data, the assigned family only influences the priors on `x_var`-based model parameters (*NEC*), which can be overridden by setting manual priors (see more information on priors below).

Specific models can be fit directly using `bnec`, in which case an object of class `bayesnecfit` is returned. **bayesnec** includes a set of `bayesnecfit` methods for predicting, plotting and extracting effect concentrations and related threshold values. Alternatively it is possible to fit a custom model-set, specific model-set or all of the available models. The default in **bayesnec** is to use `model = "all"` which fits all of the available models.

When multiple models are fitted using `bnec`, an object of class `bayesmanecfit` is returned that includes a model weighted estimate of predicted posterior values. **bayesnec** also includes a set of `bayesmanecfit` methods for predicting, plotting and extracting effect concentrations and related threshold values that return model weighted estimates.

Multi-model inference can be useful where there are a range of plausible models that could be used (Burnham and Anderson 2002) and has been recently adopted in ecotoxicology for Species Sensitivity Distribution (SSD) model inference (Thorley and Schwarz 2018; Fox, van Dam, Fisher, Batley, Tillmanns, Thorley, Schwarz, Spry, and McTavish 2021; Dalgarno 2021). The approach may have considerable value in C-R modelling because there is often no a priori knowledge of the functional form that the response relationship should take. In this case model averaging can be a useful way of allowing the data to drive the model selection process, with weights proportional to how well the individual models fit the data. Well-fitting models will have high weights, dominating the model averaged outcome. Conversely, poorly fitting models will have very low model weights and will therefore have little influence on the outcome. Where multiple models fit the data equally well, these can equally influence the outcome, and the resultant posterior predictions reflect that model uncertainty. It is possible to specify the "stacking" method (Yao, Vehtari, Simpson, and Gelman 2018) for model weights if desired (through the argument `loo_controls`) which aims to minimise prediction error. We do not currently recommend using stacking weights given the typical sample sizes associated with most C-R experiments, and because the main motivation for model averaging within the **bayesnec** package is to properly capture model uncertainty rather than reduce prediction error. By default **bayesnec** uses the "pseudobma" method using the Bayesian bootstrap through `loo_model_weights` (Vehtari, Gabry, Magnusson, Yao, Bürkner, Paananen, and Gelman 2020; Vehtari, Gelman, and Gabry 2017). These are analogous to the way model weights are generated using AIC or AICc (Burnham and Anderson 2002), although more research on their behaviour and suitability for model averaging in the current non-linear modelling setting is warranted.

Model diagnostics

A range of tools are available to assess model fit, including an estimate of overdispersion (for relevant families), an extension of the **brms rhat** function that can be applied to both `bayesnecfit` and `bayesmanecfit` model objects, and a function `check_chains` that can be used to visually assess chain mixing and stability.

All diagnostic functions available in **brms** and **rstan** can be used on the underlying **brm** model fit by extracting the fitted **brms** model from the `bayenecfit` or `bayesmanecfit` model object. For example, we can use the default **brms** plotting method to obtain a diagnostic plot of the individual fit of the **nec4param** model using:

```
R> plot(exmp_fit$mod_fits$nec4param$fit)
```

which yields a plot of the posterior densities and chains plot for each parameter in the specified model Figure 1.

The default number of total iterations in **bayesnec** is 10,000 per chain, with 9,000 of these used as warm-up (or burn-in) across 4 chains. If the `bnec` call returns **brms** warning messages the number of iterations and warm-up samples can be adjusted through arguments `iter` and `warmup`. A range of other arguments can be further adjusted to improve convergence, see the rich set of Resources available for the **brms** package for further information.

Several helper functions have been included that allow the user to add or drop models from a `bayesmanecfit` object, or change the model weighting method (`amend`); extract a single or
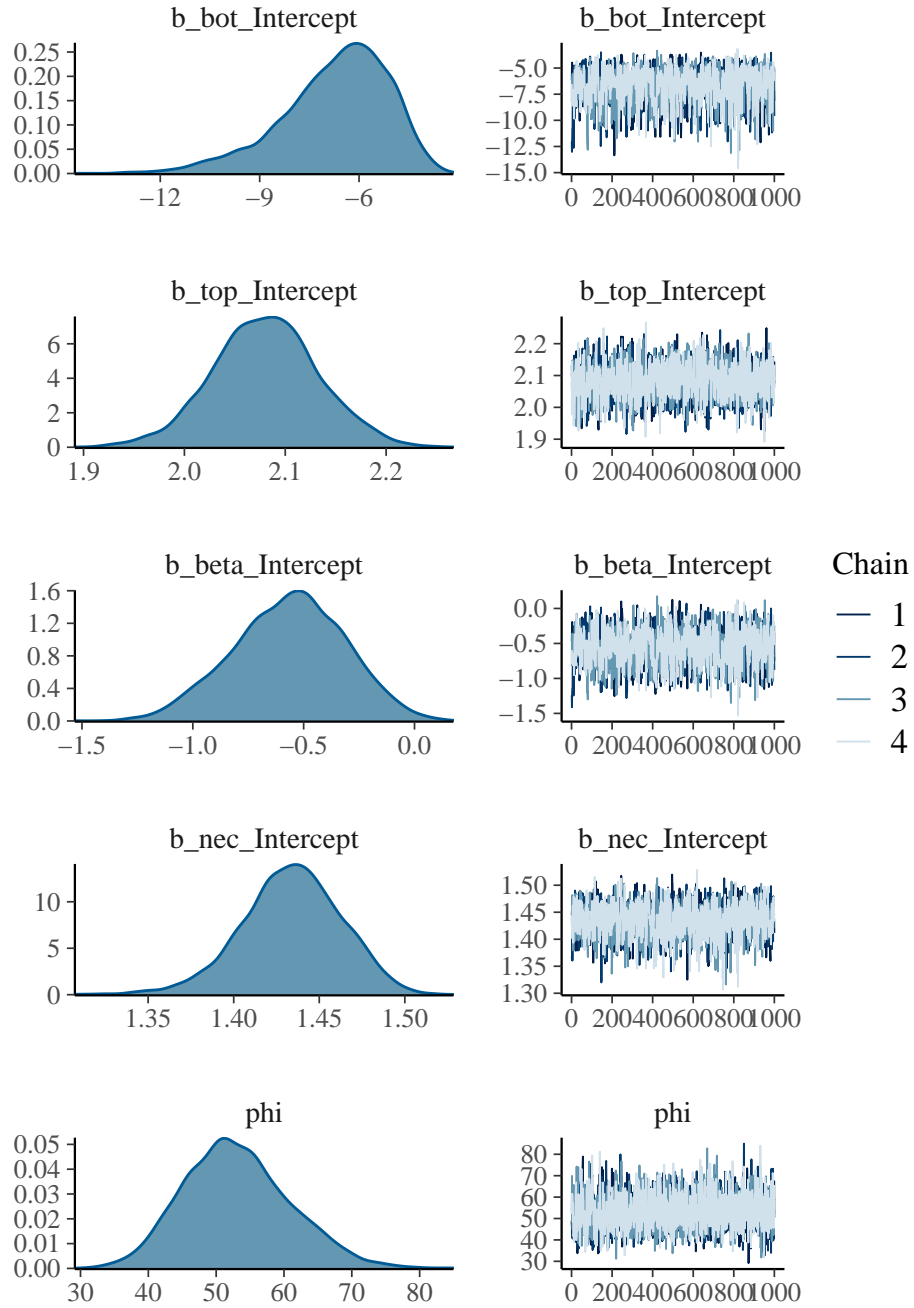
Figure 1: Default brms plot of the nec4param model showing the posterior probability densities and chain mixing for each of the included parameters.

subset of models from the `bayesmanecfit` object (`pull_out`); and examine the priors used for model fitting (`pull_prior`, `sample_priors` and `check_priors`).

Model inference

bayesnec includes a summary method for both `bayesnecfit` and `bayesmanecfit` model objects, providing the usual summary of model parameters and any relevant model fit statistics as returned in the underlying `brm` model fits. This includes a list of fitted models, their respective model weights, and a model-averaged *NEC*—which is reported with a warning when it contains NSEC values (see more information below). A warning message also indicates that the **ecxll5** model may have convergence issues according to the default **brms** $\hat{R}$ criteria:

```
R> summary(exmp_fit)
```

```
Object of class bayesmanecfit containing the following non-linear models:
  - nec4param
  - neclin
  - ecxlin
  - ecx4param
  - ecxwb1
  - ecxwb2
  - ecxll5
  - ecxll4


Distribution family: beta
Number of posterior draws per model:  4000


Model weights (Method: pseudobma_bb_weights):
            waic   wi
nec4param -333.37 0.50
neclin    -332.62 0.41
ecxlin    -188.23 0.00
ecx4param -302.53 0.00
ecxwb1    -294.57 0.00
ecxwb2    -316.99 0.00
ecxll5    -329.55 0.09
ecxll4    -302.34 0.00



Summary of weighted NEC posterior estimates:
NB: Model set contains the ECX models: ecxlin;ecx4param;ecxwb1;ecxwb2;ecxll5;ecxll4; weigh
    Estimate Q2.5 Q97.5
NEC     1.40 1.28  1.48
```

Base R (`plot`) and ggplot2 (`ggbnec`) plotting methods, as well as predict methods have also been developed for both `bayesnecfit` and `bayesmanecfit` model classes. In addition, there are method-based functions for extracting *ECx* (`ecx`), *NEC* (`nec`) and *NSEC* (`nsec`) threshold
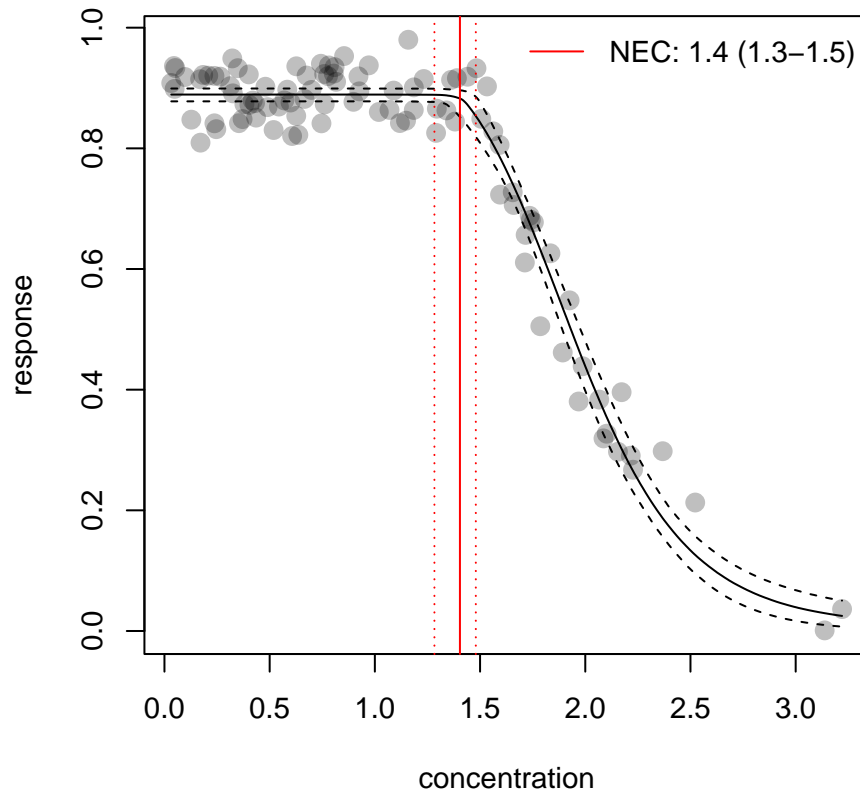
Figure 2: Base plot of the example fit model averaged curve. The solid black line is the fitted median of the posterior prediction, dashed black lines are the 95 percent credible intervals, and the red vertical lines show the estimated NEC value.

values. In all cases the posterior samples that underpin these functions are achieved through `posterior_epred` from the **brms** package. An example base plot of a `bayesmanecfit` model fit can be seen in Figure 2.

```
R> plot(exmp_fit)
```

By default the plot shows the fitted posterior curve with 95% credible intervals, along with an estimate of the $\eta = \text{NEC}$ value. Please see the vignettes for more examples using **bayesnec** models for inference.

2.2. Models in bayesnec

The argument `model` in the function `bnec` is a character string indicating the name(s) of the desired model. If a recognised model name is provided, a single model of the specified type is fit, and `bnec` returns a model object of class `bayesnecfit`. If a vector of two or more

of the available models are supplied, `bnec` returns a model object of class `bayesmanecfit` containing Bayesian model averaged predictions for the supplied models, providing they were successfully fitted.

The `model` argument may also be one of "all", meaning all of the available models will be fit; "ecx" meaning only models excluding the η = NEC step parameter will be fit; or "nec" meaning only models with the η = NEC step parameter (see below **Model parameters**) will be fit. There are a range of other pre-defined model groups available. The full list of currently implemented model groups can be seen using the function `models()`.

Please see the Model details vignette or `?model("all")` for more information on all the models available in **bayesnec** and their specific formulation.

Parameter definitions

Where possible we have aimed for consistency in the interpretable meaning of the individual parameters across models. Across the currently implemented model-set, models contain from two (basic linear or exponential decay, see **ecxlin** or **ecxexp**) to five possible parameters (e.g. **nechorme4**, **ecxhormebc5**), including:

- τ = top, usually interpretable as either the y-intercept or the upper plateau representing the mean concentration of the response at zero concentration;

- η = NEC, the no-effect-concentration value (the x concentration value where the break-point in the regression is estimated at, see **Model types for *NEC* and *ECx* estimation** and (Fox 2010) for more details on parameter based *NEC* estimation);

- β = beta, generally the exponential decay rate of response, either from 0 concentration or from the estimated η value, with the exception of the **neclinhorme** model where it represents a linear decay from η because slope (α) is required for the linear increase;

- δ = bottom, representing the lower plateau for the response at infinite concentration;

- α = slope, the linear decay rate in the models **neclin** and **ecxlin**, or the linear increase rate prior to η for all hormesis models;

- ec50 notionally the 50% effect concentration but may be influenced by scaling and should therefore not be strictly interpreted; and

- ϵ = d, the exponent in the **ecxsigm** and **necisgm** models.

In addition to the model parameters, all `nec...` models have a step function used to define the breakpoint in the regression, which can be defined as:

$$f(x_i, \eta) = \begin{cases} 0, & x_i - \eta < 0 \\ 1, & x_i - \eta \geq 0 \end{cases}$$

Model types for NEC and ECx estimation

All models provide an estimate of the no-effect-concentration (*NEC*). For model types with "nec" as a prefix, the *NEC* is directly estimated as parameter η = NEC in the model, as

per Fox (2010). Models with "ecx" as a prefix are continuous curve models, typically used for extracting *ECx* values from C-R data. In this instance the *NEC* reported is actually the no-significant-effect-concentration (*NSEC*), defined as the concentration at which the predicted response is now "significantly" lower than the values observed at the lowest treatment concentration (typically the 'control'). The desired level of "significance" can be controlled by the user through the argument `sig_val`, which is passed as a probability to `quantile` and used to calculate the lower bound of the posterior predictions at the lowest treatment concentration. The default value for `sig_val` is 0.01, which is analogous to an alpha value (Type 1 error rate) of 0.01 for a one-sided test of significance. See `?nsec` for more details. We currently recommend only using the "nec" model-set for estimation of *NEC* values, as the *NSEC* concept has yet to be formally peer-reviewed, and likely suffers from at least some of the same issues as the No-observed-effect-concentration (*NOEC*, (Warne and van Dam 2008; Fox 2008)).

ECx estimates can be equally validly obtained from both "nec" and "ecx" models. *ECx* estimates will usually be lower (more conservative) for "ecx" models fitted to the same data as "nec" models (see the Comparing posterior predictions vignette for an example. However, we recommend using "all" models where *ECx* estimation is required because "nec" models can fit some datasets better than "ecx" models and the model averaging approach will place the greatest weight for the outcome that best fits the supplied data. This approach will yield *ECx* estimates that are the most representative of the underlying relationship in the dataset.

There is ambiguity in the definition of *ECx* estimates from hormesis models (these allow an initial increase in the response (see Mattson 2008) and include models with the character string `horme` in their name), as well as those that have no natural lower bound on the scale of the response (models with the string `lin` in their name, in the case of Gaussian response data). For this reason, the `ecx` function has arguments `hormesis_def` and `type`, both character vectors indicating the desired behaviour. For `hormesis_def = "max"` *ECx* values are calculated as a decline from the maximum estimates (i.e. the peak at $\eta = \text{NEC}$); and `hormesis_def = "control"` (the default) indicates *ECx* values should be calculated relative to the control, which is assumed to be the lowest observed concentration. For `type = "relative"` *ECx* is calculated as the percentage decrease from the maximum predicted value of the response ($\tau = \text{top}$) to the minimum predicted value of the response (ie, 'relative' to the observed data). For `type = "absolute"` (the default) *ECx* is calculated as the percentage decrease from the maximum value of the response ($\tau = \text{top}$) to 0 (or $\delta = \text{bottom}$ for models with that parameter). For `type = "direct"`, a direct interpolation of y on x is obtained.

Model suitability for response types

Models that have an exponential decay (most models with parameter $\beta = \text{beta}$) with no $\delta = \text{bottom}$ parameter are 0 bounded and are not suitable for the Gaussian family, or any family modelled using a logit or log link because they cannot generate predictions of negative y (response). Conversely models with a linear decay (containing the string `lin` in their name) are not suitable for modelling families that are 0 bounded (Gamma, Poisson, Negative-binomial, Beta, Binomial, Beta-binomial) using an identity link. These restrictions do not need to be controlled by the user as a call to `bnec` with `models = "all"` will simply exclude inappropriate models, albeit with a message.

Strictly speaking, models with a linear hormesis increase are not suitable for modelling data

from the Binomial, Beta and Beta-binomial families, however they are currently allowed in **bayesnec**, with a reasonable fit achieved through a combination of the appropriate family being applied to the response, and the **bayesnec** `make_inits` function that ensures initial values passed to **brms** yield response values within the range of the observed `y_var` data.

2.3. Priors on model parameters

In Bayesian inference, model parameters and their inherent uncertainty are estimated as statistical probability distributions. This is achieved by combining an a-prior understanding of each parameter's probability density function (the 'priors') with the likelihood of the observed information (data) given the model parameters, to yield a so-called posterior probability distribution. Regardless of whether this is done mathematically via Bayes' theorem or through Monte Carlo simulation, to carry out a Bayesian analysis the prior probability densities must be defined. Sometimes there may be substantial prior knowledge, for example when pilot data or data from a previous experiment exist for a given response curve. In this case the prior probability distribution may be quite narrow (highly "informative") and can have a strong influence on the posterior, especially when subsequent data are scarce or highly variable. In our experience however, such prior knowledge is generally the exception. Where no quantitative prior information exists, it is common in Bayesian statistics to use "vague" or "weakly" informative priors. The use of "vague", "diffuse", "flat" or otherwise so-called "uninformative" priors is no longer recommended (Banner, Irvine, and Rodhouse 2020). Such priors generally form the default for many Bayesian packages, and are often used in practice without critical thought or evaluation, possibly as a result of fear of being too 'subjective' (Banner *et al.* 2020).

Considerable thought has gone into development of an algorithm (`define_prior`) to build "weakly" informative priors for fitting models in **bayesnec**. The priors are "weakly" informative in that in addition to specifying the relevant statistical family that appropriately captures the parameters likely theoretical statistical distribution, we also use information contained within the observed data to centre the probability density near the most likely parameter space and/or constrain priors to sensible bounds. These weakly informative priors are used to help constrain the underlying routines so that they are less likely to consider what the researcher would deem highly improbable estimates, that also cause the routines to become unstable. Weakly informative priors can be particularly helpful in complex non-linear modelling to ensure reliable convergence. These types of priors specify the general shape and bounds of the expected probability distribution for a given parameter, whilst remaining sufficiently broad so as not to influence the parameter's estimated posterior distribution (given a reasonable amount of observed data). In this sense appropriately weak priors should yield analytical outcomes that share the same level of *objectivity* as equivalent frequentist approaches, whilst yielding robust parameter estimates with probabilistically interpretable uncertainty bounds. To ensure a **bayesnec** analysis retains the same level of objectivity as an equivalent frequentist approach we recommend using the function `check_priors` to compare the prior and posterior probability densities and check that the priors used by **bayesnec** are sensible and are not exerting an undesirable influence over the analysis.

bayesnec's `define_prior` *algorithm*

Priors are constructed in **bayesnec** for each parameter of each model being fitted based on

the characteristics of either the input `x_var` or `y_var` data, depending on which is relevant to the specific parameter scaling. In the case of parameters that scale with `y_var` (the response), priors are constructed based on the relevant link scaling, whether that be identity, the default (for that family), or user specified link function for a specific family. The priors are constructed by `bnec` internally calling the function `define_prior`, which takes the arguments `model`, `family` (including the relevant link function), `predictor` (`x_var` data), and `response` (`y_var` data).

Priors for response (`y_var`) scaled parameters   Only the parameters $\tau = $ top and $\delta = $ bottom scale specifically with the response (`y_var` data) family. For Gaussian `y_var` data (or any `y_var` data for which the link ensures valid values of the response can take from $-\infty$ to ∞, including `log` and `logit`) priors are Gaussian with a standard deviation of 2.5 and a mean set at the 90th and 10th quantiles for $\tau = $ top and $\delta = $ bottom respectively. In this way **bayesnec** attempts to construct a prior that scales appropriately with the observed data, with greatest density near the most likely region of the response for the $\tau = $ top and $\delta = $ bottom parameters, whilst remaining broad enough to have little influence on each parameter's posterior density.

For Poisson-, Negative-binomial- and Gamma-distributed `y_var` data, the response is bounded by 0 and therefore Gaussian priors are unsuitable. Instead we use Gamma priors, with a mean scaled to correspond to the 75th and 25th quantiles for $\tau = $ top and $\delta = $ bottom respectively. The mean (μ) is linked mathematically to the shape (s) and rate parameters (r) by the equation

$$\mu = s * (1/r)$$

with the shape parameter being set to 2 by default.

For the Binomial, Beta, and Beta-binomial families, estimates for $\tau = $ top and $\delta = $ bottom must necessarily be constrained between 0 and 1 when modelled on the identity link. Because of this constraint there is no need to adjust scaling based on the response. In this case **bayesnec** uses `beta(5, 1)` and `beta(1, 5)` priors to provide a broad density centred across the upper and lower 0 to 1 range for the $\tau = $ top and $\delta = $ bottom parameters respectively.

Priors for predictor (`x_var`) scaled parameters   The parameters $\eta = $ NEC and $\eta = $ ec50 scale with respect to the predictor (`x_var` data), because both of these are estimated in units of the predictor (`x_var`, usually concentration). To stabilise model fitting, the $\eta = $ NEC and $\eta = $ ec50 parameters are bounded to the upper and lower observed range in the predictor, under the assumption that the range of concentrations in the experiment were sufficient to cover the full range of the response outcomes. The priors used reflect the characteristics of the observed data that are used to predict the appropriate family. If the `x_var` data are bounded to 0 and >1 a Gamma prior is used, with maximum density (μ, see above) at the median value of the predictor, and a shape parameter of 5. If the `x_var` data are bounded to 0 and 1 a `beta(2, 2)` prior is used. For `x_var` data ranging from $-\infty$ to ∞, a Gaussian prior is used, with a mean set at the median of the `x_var` values and a standard deviation of 2.5.

Priors for other parameters   For the parameters $\beta = $ beta, $\alpha = $ slope and $\epsilon = $ d we first ensured any relevant transformations in the model formula such that theoretical values with the range $-\infty$ to ∞ are allowable, and a `normal(0, 2)` prior is used. For example in the
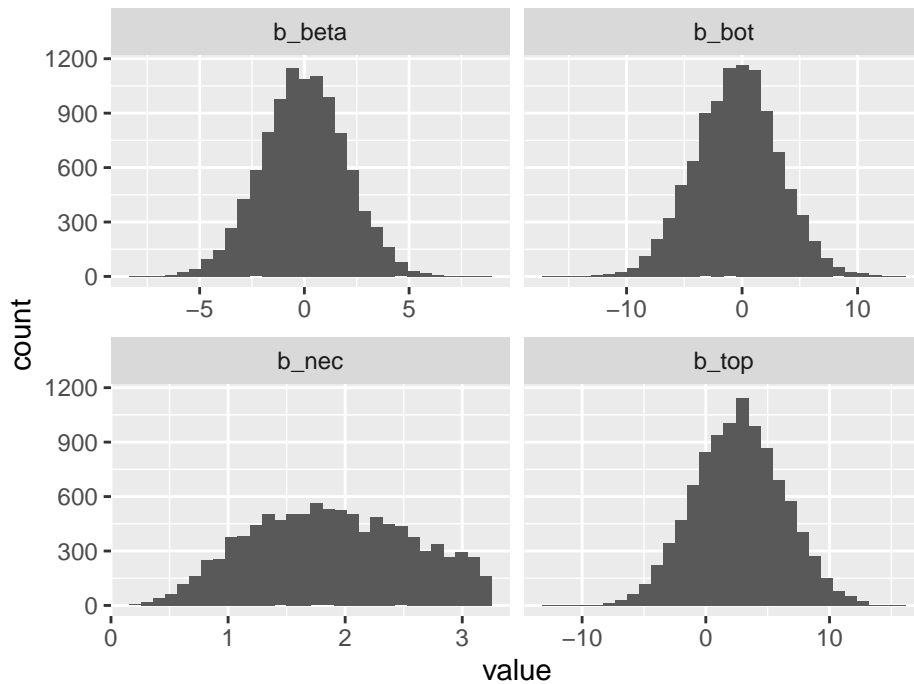
Figure 3: Frequency histograms of samples of the default priors used by bnec for fitting the nec4param model to the example data.

`nec3param` model β = beta is an exponential decay parameter, which must by definition be bounded to 0 and ∞. Calling `exp(beta)` in the model formula ensures the exponent meets these requirements. Note also that a mean of 0 and standard deviation of 2 represents a relatively broad prior on this exponential scaling, so this is still generally a weakly informative prior in practice.

User specified priors

There may be situations where the default **bayesnec** priors do not behave as desired, or the user wants to provide informative priors. For example, the default priors may be too informative, yielding unreasonably tight confidence bands (although this is only likely where there are few data or unique values of the `x_var` data). Conversely, priors may be too vague, leading to poor model convergence. Alternatively, the default priors may be of the wrong statistical family if there was insufficient information in the provided data for **bayesnec** to correctly predict the appropriate ones to use. The priors used in the default model fit can be extracted using `pull_prior`, and a sample or plot of prior values can be obtained from the individual **brms** model fits through the function `sample_priors` which samples directly from the `prior` element in the `brm` model fit (see Figure 3).

```
R> sample_priors(exmp_fit$mod_fits$nec4param$fit$prior)
```

We can also use the function `check_priors` (based on the `hypothesis` function of **brms**) to assess how the posterior probability density for each parameter differs from that of the
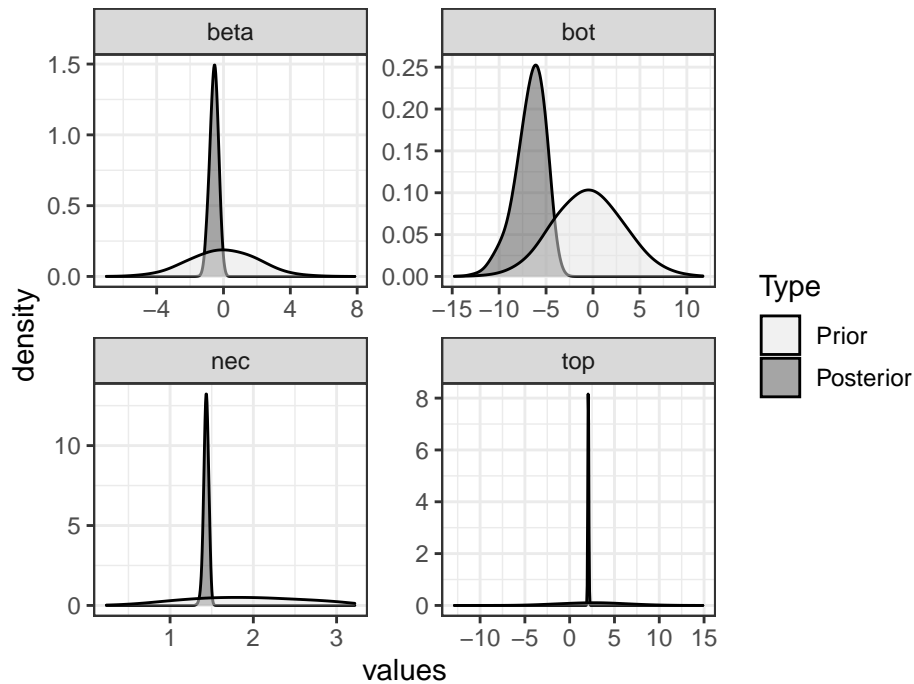
Figure 4: A comparison of the prior and posterior parameter probability densities for the nec4param model fit to the example data.

prior. Here we show the prior and posterior probability densities for the parameters in the **nec4param** model, extracted from our example fit (see Figure 4). There is also a class `bayesmanecfit` method that can be used to sequentially view all plots in a `bnec` call with multiple models, or write to a pdf as in `check_chains`.

This can be done for a single model using:

```
R> exmp_fit_nec4param <- pull_out(exmp_fit, model = "nec4param")
R> check_priors(exmp_fit_nec4param)
```

or for all models, writing to a pdf file named Check_priors_plots.pdf:

```
R> check_priors(exmp_fit, filename = "Check_priors_plots")
```

2.4. Model comparison

With **bayesnec** we have included a function (`compare_posterior`) that allows bootstrapped comparisons of posterior predictions. This function allows the user to fit several different `bnec` model fits and compare differences in the posterior predictions. Comparisons can be made across the model fits for individual endpoint estimates (e.g. *NEC*, *NSEC* or *ECx*) or across a range of predictor (x) values. Usage is demonstrated in the relevant vignette by comparing different types of models and model-sets using a single dataset. However, the intent of this function is to allow comparison across different datasets that might represent,

for example, different levels of a fixed factor covariate. For example, this function has been used to compare toxicity of herbicides across three different climate scenarios, to examine the cumulative impacts of pesticides and global warming on corals (Flores, F., Marques, J.A., Uthicke, S., Fisher, R., Patel, F., Kaserzon, S., Negri 2021).

At this time `bnec` does not allow for an inclusion of an interaction with a fixed factor. Including an interaction term within each of the non-linear models implemented in **bayesnec** is relatively straightforward, and may be introduced in future releases. However, in many cases the functional form of the response may change with different levels of a given factor. The substantial complexity of defining all possible non-linear model combinations at each factor level means it unlikely this could be feasibly implemented in **bayesnec** in the short term. In the meantime the greatest flexibility in the functional form of individual model fits can be readily obtained using models fitted independently to data within each factor level.

2.5. Random effects

Most ecotoxicological and toxicology experiments include a range of grouping elements, such as tanks, vials or batches of samples that contain multiple measurements that cannot be considered strictly independent (aka they are pseudo-replicates). To avoid criticism around potential issues with pseudo-replication, it is often the practice for ecotoxicologists to pool such observations and carry out modelling using, for example, the group mean. Where the number of within group observations varies substantially across groups, this will have the undesirable effect of equally weighting the group means even though some may be based on far fewer observations than others. In addition there are often instances of ecotoxicology data from multiple experiments or other grouping factors within an experiment (such as genotype) that cover the full range of x concentrations that cannot be averaged prior to modelling, resulting in the ecotoxicologist either ignoring the potential non-independence, or fitting many independent datasets and subsequently needing to aggregate the endpoint estimates. Carrying out multiple fits on separate datasets is undesirable because each fit is based on fewer data and will have greater uncertainty.

The current version of **bayesnec** harnesses the powerful mixed (hierarchical) modelling capabilities of **brms** for accommodating hierarchical designs and other forms of non-independence. This is achieved by allowing a list of random terms to be passed through the argument *random*, which are then used to update the **brms** model formula. Random effects can be in the form of a random offset, which effectively allows different mean response levels across groups, and is achieved through specifying the random term "ost" in the named list. Random effects can also be added to any or all of the non-linear parameters in the model. Note that implementing random effects in a non-linear modelling setting is non-trivial and considerable thought and testing should go into selecting an appropriate random structure.

3. Discussion

In order to be accessible to a broad community of varying statistical capabilities, we have simplified fitting a **bayesnec** model as much as possible, whilst retaining the ability to modify a wide range of arguments as necessary. Where possible we have tried to set default values to align with those in **brms**. Wherever we deviate, this is generally towards being more conservative and/or we have clearly explained our reasoning. Specific examples include: 1)

`iter`, which we increased from the **brms** default of 2,000 to 10,000 as we found that a higher number of iterations are generally required for these non-linear models; 2) the use of pseudo-BMA rather than stacking weights to better reflect **bayesnec**'s goals with respect to model averaging (discussed in more detail above); and 3) the use of `pointwise = TRUE` (where possible) and `sample_prior = "yes"` to avoid excessive R crashes when used in the Windows operating system and allow the use of the `hypothesis` function respectively. We welcome constructive criticism of our selections and users must expect that default settings may change accordingly in later releases.

We have made considerable effort to ensure that **bayesnec** makes a sensible prediction for the appropriate family, constructs appropriate weakly informative priors, and generates sensible initial values. However, this is a difficult task across such a broad range of non-linear models, and across the potential range of ecotoxicological data that may be used. The user must interrogate their model fits using the wide array of helper functions, and use their own judgement regarding the appropriateness of model inferences for their own application. Of particular importance are examination of model fit statistics through the `summary` and `rhat` methods, visual inspection of all model fits in `bayesmanecfit` objects (via `plot(...,  all_models = TRUE)` and `check_chains(..., all_models = TRUE)`) and an assessment of the posterior versus prior probability densities to ensure default priors are appropriate (using `check_priors`).

The model averaging approach implemented in **bayesnec** is widely used in a range of settings (in ecology for example, see Dormann, Calabrese, Guillera-Arroita, Matechou, Bahn, Bartoń, Beale, Ciuti, Elith, Gerstner, Guelat, Keil, Lahoz-Monfort, Pollock, Reineking, Roberts, Schröder, Thuiller, Warton, Wintle, Wood, Wüest, and Hartig 2018, for a thorough review). However, model averaging is relatively new to ecotoxicology (but see, for example, Shao and Gift 2014; Thorley and Schwarz 2018; Fox *et al.* 2021; Wheeler and Bailer 2009). In **bayesnec** we have implemented a broad range of potential models, and the default behaviour is to fit them all, although we discuss above situations where this is clearly not recommended (for example, in the estimation of **NEC**). More research is required to understand how model-set selection influences model inference. While some studies suggest using a broad range of models may be optimal (Wheeler and Bailer 2009), others indicate that including multiple models of similar shape may overweight the representation of that shape in model averaged predictions (Fox *et al.* 2021). In addition, it is important to understand that when models are added or removed from the model-set, this can sometimes have a substantial influence on model predictions (potentially changing estimated **ECx** values, for example). As the model-set in **bayesnec** may change through time it is important to keep a record of the models that were actually fitted in a given analysis, in the event it is necessary to reproduce a set of results. A potentially better strategy is to build a Docker container, an emerging approach representing one strategy towards overcoming the reproducibility crisis (Baker and Penny 2016). Considerations of analytical reproducibility are particularly relevant to C-R modelling, where the model outcomes can often have far reaching management implications.

4. Future directions

The **bayesnec** package is a work in progress, and we welcome suggestions and feedback that will improve the package performance and function. Our goal is to make **bayesnec** as user friendly as possible, and capable of dealing with most real world C-R modelling applications in the

hope that Bayesian statistics will become more widely used in applied risk assessment. Please submit requests through the package Issues on github. Some suggested future enhancements include:

- The addition of other custom families, such as the tweedie distribution and ordered beta model. Currently **bayesnec** implements adjustments away from 0 (Gamma, Beta) or 1 (Beta) as a strategy for allowing modelling with these types of data using the closest most convenient statistical distribution. There are no readily available distributions able to model data that includes 0 and 1 on the continuous scale in **brms** and **bayesnec** currently does 0 and 1 adjustments followed by modelling using a Beta distribution. The ordered beta model has been suggested as a better method for modelling continuous data with lower an upper bounds (see (Kubinec, Barcel'o, Goldszmidt, Grujic, Model, Schenk, Cheng, Hale, Hartnett, Messerschmidt, Petherick, and Thorvaldsdottir 2021)) that could be readily implemented in the **brms** customs families framework. For data that are 0 to ∞ on the continuous scale the Tweedie distribution may prove a much better option than the current zero bounded Gamma, and has been used extensively in fisheries research for biomass data (Shono 2008). As this family is not currently available in **brms** this would also need to be implemented as a custom family, which for the Tweedie is not trivial.

- A hypothesis method for testing against toxicity thresholds. The **brms** package includes a `hypothesis` function that allows for testing parameter estimates against specified criteria. This is used in **bayesnec** in the `check_prior` function, which is a wrapper that examines the deviation of each parameter in the given model relative to 0 as a means of generating posterior and prior probability density plots for comparison. However, an additional wrapper function could be developed that allows toxicity to be assessed, as measured through *NEC*, or *ECx* for example, against a required pre-defined threshold. Such a feature may be useful where toxicity testing is used as a trigger in risk management (for example, using whole-effluent-toxicity (WET) testing, (Karman and Smit 2019)).

5. Acknowledgements

References

Baker M, Penny D (2016). "Is there a reproducibility crisis?" `doi:10.1038/533452A`.

Banner KM, Irvine KM, Rodhouse TJ (2020). "The use of Bayesian priors in Ecology: The good, the bad and the not great." ISSN 2041210X. `doi:10.1111/2041-210X.13407`.

Bürkner PC (2017). "brms: An R package for Bayesian multilevel models using Stan." *Journal of Statistical Software.* ISSN 15487660. `doi:10.18637/jss.v080.i01`.

Burnham KP, Anderson DR (2002). *Model Selection and Multimodel Inference; A Practical Information-Theoretic Approach.* 2nd edition. Springer, New York.

Bürkner PC (2018). "Advanced Bayesian Multilevel Modeling with the R Package brms." *The R Journal*, **10**(1), 395–411. `doi:10.32614/RJ-2018-017`.

Carpenter B, Gelman A, Hoffman MD, Lee D, Goodrich B, Betancourt M, Brubaker MA, Guo J, Li P, Riddell A (2017). "Stan: A probabilistic programming language." *Journal of Statistical Software.* ISSN 15487660. `doi:10.18637/jss.v076.i01`.

Charles S, Wu D, Ducrot V (2020). "How to account for the uncertainty from standard toxicity tests in species sensitivity distributions: An example in non-target plants." `doi:10.1101/2020.07.02.183863`.

Dalgarno S (2021). "shinyssdtools: A web application for fitting Species Sensitivity Distributions (SSDs)." *Journal of Open Source Software*, **6**(57), 2848. `doi:10.21105/joss.02848`. URL `https://doi.org/10.21105/joss.02848`.

Dormann CF, Calabrese JM, Guillera-Arroita G, Matechou E, Bahn V, Bartoń K, Beale CM, Ciuti S, Elith J, Gerstner K, Guelat J, Keil P, Lahoz-Monfort JJ, Pollock LJ, Reineking B, Roberts DR, Schröder B, Thuiller W, Warton DI, Wintle BA, Wood SN, Wüest RO, Hartig F (2018). "Model averaging in ecology: a review of Bayesian, information-theoretic, and tactical approaches for predictive inference." `doi:10.1002/ecm.1309`.

Ellison AM (1996). "An introduction to bayesian inference for ecological research and environmental decision-making." *Ecological Applications.* ISSN 10510761. `doi:10.2307/2269588`.

Fisher R, Bessell-Browne P, Jones R (2019a). "Synergistic and antagonistic impacts of suspended sediments and thermal stress on corals." *Nature Communications.* URL `https://doi.org/10.1038/s41467-019-10288-9`.

Fisher R, van Dam R, Batley G, Fox D, Harford A, Humphrey C, King C, Menendez P, Negri A, Proctor A, Shao Q, Stauber J, van Dam J, Warne M (2019b). "Key issues in the derivation of water quality guideline values: a workshop report. Australian Institute of Marine Science Report, Crawley, WA, Australia. 57 pp." *Technical report.* `doi:10.13140/RG.2.2.29774.82241`.

Fisher R, Walshe T, Bessell-Browne P, Jones R (2018). "Accounting for environmental uncertainty in the management of dredging impacts using probabilistic dose–response relationships and thresholds." *Journal of Applied Ecology*, **55**(1), 415–425. ISSN 13652664. `doi:10.1111/1365-2664.12936`.

Flores, F, Marques, JA, Uthicke, S, Fisher, R, Patel, F, Kaserzon, S, Negri A (2021). "Combined effects of climate change and the herbicide diuron on the coral <italic>Acropora millepora</italic>." *Mar Pollut Bull.*

Fox D (2008). "NECS, NOECS AND THE ECx." *Australasian Journal of Ecotoxicology*, **14**, 7–9.

Fox DR (2010). "A Bayesian approach for determining the no effect concentration and hazardous concentration in ecotoxicology." *Ecotoxicology and environmental safety*, **73**(2), 123–131.

Fox DR, van Dam RA, Fisher R, Batley GE, Tillmanns AR, Thorley J, Schwarz CJ, Spry DJ, McTavish K (2021). "Recent developments in Species Sensitivity Distribution Modeling." *Environmental Toxicology and Chemistry*, **40**(2), 293–308. `doi:https://doi.org/10.1002/etc.4925`. URL `https://setac.onlinelibrary.wiley.com/doi/abs/10.1002/etc.4925`.

Gottschalk F, Nowack B (2013). "A probabilistic method for species sensitivity distributions taking into account the inherent uncertainty and variability of effects to estimate environmental risk." *Integrated environmental assessment and management*, **9**(1), 79–86.

Karman CC, Smit MG (2019). "Whole Effluent Toxicity Data and Discharge Volumes to Assess the Likelihood that Environmental Risks of Offshore Produced Water Discharges Are Adequately Controlled." *Integrated Environmental Assessment and Management*, **15**(4), 584–595. `doi:10.1002/ieam.4139`. URL `www.ecfr.gov,`.

Kubinec R, Barcel'o J, Goldszmidt R, Grujic V, Model T, Schenk C, Cheng C, Hale T, Hartnett AS, Messerschmidt L, Petherick A, Thorvaldsdottir S (2021). "Statistically Validated Indices for COVID-19 Public Health Policies." `doi:10.31235/osf.io/rn9xk`.

Lunn DJ, Thomas A, Best N, Spiegelhalter D (2000). "WinBUGS - A Bayesian modelling framework: Concepts, structure, and extensibility." *Statistics and Computing*, **10**(4), 325–337. ISSN 09603174. `doi:10.1023/A:1008929526011`. URL `http://www.mrc-bsu.cam.ac.uk/bugs/`.

Mattson MP (2008). "Hormesis defined." `doi:10.1016/j.arr.2007.08.007`.

Pinheiro J, Bates D, DebRoy S, Sarkar D, R Core Team (2021). *nlme: Linear and Nonlinear Mixed Effects Models*. R package version 3.1-152, URL `https://CRAN.R-project.org/package=nlme`.

Pires AM, Branco JA, Picado A, Mendonca E (2002). "Models for the estimation of a 'no effect concentration'." *Environmetrics*. ISSN 11804009. `doi:10.1002/env.501`.

Plummer M (2003). "JAGS: A program for analysis of Bayesian graphical models using Gibbs sampling. http://citeseer.ist.psu.edu/plummer03jags.html."

Ritz C, Baty F, Streibig JC, Gerhard D (2016). "Dose-Response analysis using R." *PLoS ONE*, **10**(12), e0146021. `doi:10.1371/journal.pone.0146021`. URL `http://dx.doi.org/10.1371%2Fjournal.pone.0146021`.

Shao K, Gift JS (2014). "Model uncertainty and bayesian model averaged benchmark dose estimation for continuous data." *Risk Analysis*. ISSN 02724332. `doi:10.1111/risa.12078`.

Shono H (2008). "Application of the Tweedie distribution to zero-catch data in CPUE analysis." *Fisheries Research*, **93**(1), 154–162.

Stan Development Team (2020). "RStan: the R interface to Stan." R package version 2.21.3, URL `http://mc-stan.org/`.

Su YS, Yajima M (2015). "R2jags: Using R to Run 'JAGS'. R package version 0.5-6. http://CRAN.R-project.org/package=R2jags."

Thomas N (2006). "Hypothesis testing and Bayesian estimation using a sigmoid Emax model applied to sparse dose-response designs." *Journal of Biopharmaceutical Statistics*. ISSN 10543406. doi:10.1080/10543400600860469.

Thorley J, Schwarz C (2018). "ssdtools: Species Sensitivity Distributions. R package version 0.0.3. https://CRAN.R-project.org/package=ssdtools."

Vehtari A, Gabry J, Magnusson M, Yao Y, Bürkner PC, Paananen T, Gelman A (2020). "loo: Efficient leave-one-out cross-validation and WAIC for Bayesian models." R package version 2.3.1, URL https://mc-stan.org/loo.

Vehtari A, Gelman A, Gabry J (2017). "Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC." *Statistics and Computing*, **27**, 1413–1432. doi:10.1007/s11222-016-9696-4.

Warne M, Batley GE, van Dam RA, Chapman JC, Fox DR, Hickey CW, Stauber JL (2015). "Revised Method for Deriving Australian and New Zealand Water Quality Guideline Values for Toxicants." *Prepared for the Council of Australian Government's Standing Council on Environment and Water (SCEW). Department of Science, Information Technology and Innovation, Brisbane, Queensland. 43 pp.*

Warne MSJ, van Dam R (2008). "NOEC and LOEC data should no longer be used." *Australasian Journal of Ecotoxicology*, **14**, 1–5.

Wheeler MW, Bailer AJ (2009). "Comparing model averaging with other model selection strategies for benchmark dose estimation." *Environmental and Ecological Statistics*. ISSN 13528505. doi:10.1007/s10651-007-0071-7.

Yao Y, Vehtari A, Simpson D, Gelman A (2018). "Using Stacking to average Bayesian predictive distributions (with discussion)." *Bayesian Analysis*. ISSN 1936-0975. doi:10.1214/17-ba1091.

Affiliation:

Rebecca Fisher
The Indian Ocean Marine Research Centre,
University of Western Australia,
Crawley, WA, Australia
Australian Institute of Marine Science,
Crawley, WA, Australia
E-mail: r.fisher@aims.gov.au

Diego Barneche
The Indian Ocean Marine Research Centre,
University of Western Australia,
Crawley, WA, Australia
Australian Institute of Marine Science,
Crawley, WA, Australia

Gerard Ricardo
Australian Institute of Marine Science,
Townsville, Qld, Australia
The University of Queensland,
St Lucia, Qld, Australia

David Fox
University of Melbourne,
Parkville, Victoria, Australia
Environmetrics Australia,
Beaumaris, Victoria, Australia