***Currently available SSD software tools***

It has become increasingly obvious to us that there is considerable overlap and duplication in the methodological development and practical application associated with the derivation of WQBs. This is no more apparent than the ‘cottage industry’ that has grown up around the development of computer tools and software to perform the complex calculations demanded by the SSD method. For example, the software tools in Table 1 all perform the same basic functions.

Table 1. SSD software tools

| **Tool** | **Description** | **Origin** | **Reference** |
| --- | --- | --- | --- |
| Burrlioz | Stand-alone compiled program for fitting and plotting SSDs; estimating HCx values, fraction affected and confidence intervals. | Australia and New Zealand | Barry, S and Henderson, B (2014) |
| ETX 2.0 | Stand-alone compiled program for estimating HC5 and HC50 and fraction affected by fitting a log-normal SSD. | RIVM, Netherlands | Van Vlaardingen, Traas, Wintersen ,and Aldenberg (2004) |
| SSD generator | Excel macros for estimation of fraction affected using single distribution (log-probit) | USEPA | USEPA (2004) |
| SSD Master | Excel macros for fitting SSDs and estimating fraction affected. | CCME / Intrinsik | CCME (2013) |
| MOSAIC | On-line tool for fitting and plotting SSDs; estimating HCx values, fraction affected and confidence intervals. Handles censored data. | University of Lyon, France | Kon Kam King et al. (2014) |
| ssdtools | R package for fitting and plotting SSDs; estimating HCx values, fraction affected and confidence intervals. | Canada (BC) | Thorley and Schwarz (2018) |
| ssdtools Shiny App | On-line tool for fitting and plotting SSDs; estimating HCx values, fraction affected and confidence intervals using mixture-modelling. | Canada (BC) | Dalgarno (2018) |
| Shinyssd  Shiny App | On-line tool for fitting and plotting SSDs and estimating HCx values. | Argentina | D’Andrea and Brodeur (2019) |
| SSD Toolbox | Stand-alone compiled program for fitting and plotting SSDs; estimating HCx values, fraction affected and confidence intervals. | USEPA | Center for Computational Toxicology and Exposure (2020). |

While we are not advocating adoption of a single standard approach or tool, we think there is a need for closer jurisdictional collaboration, greater harmonisation of methods, and development of at least some benchmark data sets and reference results. The last of these is particularly pressing given the frequency with which we have observed noticeably different HCx values for the same data set from the different tools in Table 1. This is to be expected if different estimation strategies are employed (for example maximum likelihood versus method of moments or single SSD versus a model-averaged SSD) but all things being equal, all tools should give the same point estimates to within some nominally small tolerance (e.g.1-2%). Certainly, differences of a factor of 2 or more are indicative of flawed coding and/or numerical instabilities and convergence issues. This is not a new idea and indeed ‘reference data sets’ were commonly used in the early days of statistical computing to allow both software developers and end-users to assess the adequacy of numerical routines underpinning routine analyses such as ANOVA, regression, and correlation. Even today, the National Institute of Standards and Technology still maintains a number of statistical reference data sets at <https://itl.nist.gov/div898/strd/index.html>, including the famous Longley data set (Longley 1967).

Also evident from Table 1 is the mixture of deployment modes. These encompass stand-alone compiled programs, open-source code, Excel macros and on-line GUIs attached to an underlying fitting program. Interestingly, at a time when Australia and New Zealand are contemplating a shift to open-source R in line with recent Canadian and French on-line tool development, the USEPA has just released SSD Toolbox which, at the present time, is only available as compiled MATLAB code (<https://www.epa.gov/chemical-research/species-sensitivity-distribution-ssd-toolbox>). There are pros and cons associated with the different deployment methods, although we believe there is a major dichotomy between stand-alone compiled code and open-source code. The former provides total control over all aspects of the software’s features, computational methods, and presentation of results. In a regulatory context, this may be desirable, but is anathema to the concepts of openness, transparency and user-participation. We are not in favour of this approach for the following reasons:

* compiled code does not encourage transparency, open science, nor foster greater public trust in science (Munafò et al., 2017; Mancini et al. 2019)
* compiled code makes it harder for the user to see what is going on ‘under the hood’ and does not allow the user to make modifications for their own personal use. Compiled code written in proprietary software adds an extra layer of cost and complexity. For example, SSD Toolbox is coded in MATLAB and the user must either pay a significant fee to purchase or lease the MATLAB software or alternatively install a free run-time compiler distributed by MATLAB. In the case of the latter, the continued ‘free’ use of compiled MATLAB code is at the discretion of a commercial third-party;
* even if the code is made publicly available, user modifications, adaptations and enhancements are only possible by: (a) purchasing the software the code was originally written in; and (b) having training and expertise in the proprietary code;
* the ability to collaborate widely is severely curtailed when software development is undertaken within a highly specialised environment that has a limited user base.

The use of an open source licencing and development platform overcomes all these deficiencies while fostering a collaborative environment that enhances harmonization of approaches and facilitates the timely dissemination of the most up-to-date methodologies.

---------------------------------------------------------------------------------------------------------------------

***Implications for fitting SSDs***

Computations associated with fitting and using SSDs are invariably complex and best handled by purpose-built software such as those listed in Table 1. Some of these software tools have been in existence for over 20 years and are both used and endorsed by regulatory agencies for the purpose of setting WQBs for marine and freshwater systems. It is not our intention to provide a comprehensive review of all these tools, but rather to highlight new additions and features. Accordingly, we focus on two products: the ssdtoolsR package (Thorley and Schwarz 2018) together with the associated shinyssdtools app (Dalgarno 2018); and the recently released SSD Toolbox (Center for Computational Toxicology and Exposure 2020).

**ssdtools Shiny app:** ssdtools is an R software package developed for the British Columbia Ministry of Environment and Climate Change Strategy (Thorley and Schwarz 2018). shinyssdtools is a web-based graphical user interface to ssdtools which uses the R shiny package (Chang et al. 2019).

Web deployment of apps is becoming increasingly popular and has several advantages over standalone software. In particular, the user is guaranteed of using the most up-to-date version of the software as well as being able to run analyses from any device that supports browsing. Furthermore, being an R package means the ssdtools source code is completely transparent and available for local modification. As noted in the current status section, issues such as statistical consistency and transparency need to be considered when using SSDs for various purposes, and there is likely to be demand for both modifiable and “locked” (i.e. compiled) code.

shinyssdtools is currently hosted at <https://bcgov-env.shinyapps.io/ssdtools/> although shinyssdtools is itself an R package (https://github.com/bcgov/shinyssdtools) that can be run locally. The interface is clean and simple and allows the user to either cut and paste data directly into the app or upload from a local csv file. Although individual distributions can be used to obtain HCxvalues, the focus and strength of ssdtools is its intrinsic use of model-averaging. The R package ssdtools and the accompanying Shiny app (Dalgarno 2018) currently fits the log-normal, log-logistic and gamma by default and optionally, the Burr III, log-Gumbel, and Gompertz. The default distributions were selected on the basis of having a compact candidate set of distributions that have sufficient capability to model a variety of tail behaviours.

The log-normal distribution was selected as the starting distribution given the data are for effect concentrations. The log-normal distribution does have a couple of characteristics that need to be considered when fitting species sensitivity data. First, on the logarithmic scale, the normal distribution is symmetrical and there are no a priori grounds on which to make any assumption about an SSDs shape or scale whether that be on the original or log-transformed scale. Second, the log-normal distribution decays quickly in the tails giving narrow tails that may not adequately fit the data.

The log-logistic distribution was selected as it is often used as a candidate SSD primarily because of its analytic tractability (Aldenderg and Slob 1993). It was included because it has wider tails than the log-normal and because it is a specific case of the more general Burr family of distributions (Burr 1942, Shao 2000).

The gamma distribution is a two-parameter distribution commonly used to model failure times or time to events. For use in modelling species sensitivity data, the gamma distribution has two key features that provide additional flexibility when added to the log-normal distribution: (i) it is non-symmetrical on the logarithmic scale; and (ii) it has wider tails. The Weibull distribution was also considered as a default distribution but the gamma distribution is generally more flexible whilst capturing similar shaped distributions to the Weibull.

**SSD Toolbox:** The SSD Toolbox is a US Environmental Protection Agency product. It is made available as a Windows executable file and can be downloaded from <https://epa.figshare.com/articles/Species_Sensitivity_Distribution_SSD_Toolbox/11971392>

Before using SSD Toolbox, the user must also download and install version 9.5 of the Matlab Runtime Compiler (MCR) from Mathworks. The MCR software enables the compiled code to execute without having to purchase the Matlab product. It is however a resource-hungry piece of software with its 88,000+ files consuming 3.75GB of hard disk space.

Overall, SSD Toolbox is a competent piece of software and essentially performs the same functions as ssdtools. It has a graphical user interface (GUI) which is adequate, but not as aesthetically appealing as the ssdtools Shiny app. There are 6 theoretical distributions for SSD fitting log-transformed data (normal; logistic; triangular; Gumbel; Weibull; Burr) using up to 4 fitting methods (maximum likelihood; moment matching; *cdf* linearization; and Bayesian methods). Although formally used by the USEPA for deriving ambient water quality criteria, the triangular distribution is a curious inclusion given that it has tail characteristics that are not generally encountered in practice and therefore not widely used as a realistic SSD. The *cdf* linearization method is also an unusual choice as this is a relatively crude way of fitting distributions and provides SSD parameter estimates that do not necessarily share desirable statistical properties enjoyed by other methodologies such as maximum likelihood estimation.