



# HHS Public Access

Author manuscript

Reprod Toxicol. Author manuscript; available in PMC 2020 December 01.

Published in final edited form as:

Reprod Toxicol. 2019 December ; 90: 102–108. doi:10.1016/j.reprotox.2019.07.013.

## Python BMDS: A Python interface library and web application for the canonical EPA dose-response modeling software

Ly Ly Pham<sup>\*†</sup>, Sean Watford<sup>\*‡</sup>, Katie Paul Friedman<sup>\*</sup>, Jessica Wignall<sup>¶</sup>, Andrew J Shapiro<sup>§</sup>

<sup>\*</sup>National Center for Computational Toxicology, Office of Research and Development, United States Environmental Protection Agency, Research Triangle Park, NC USA

<sup>†</sup>ORISE Postdoctoral Research Participant

<sup>‡</sup>ORAU, contractor to U.S. Environmental Protection Agency through the National Student Services Contract

<sup>§</sup>National Toxicology Program at NIEHS, Research Triangle Park, NC

<sup>¶</sup>ICF, Burlington, Vermont

### Abstract

Several primary sources of publicly available, quantitative dose-response data from traditional toxicology study designs relevant to predictive toxicology applications are now available, including the redeveloped U.S. Environmental Protection Agency's Toxicity Reference Database (ToxRefDB v2.0), the Health Assessment Workspace Collaborative (HAWC), and the National Toxicology Program's Chemical Program's Chemical Effects in Biological Systems (CEBS). These resources provide effect level information but modeling these data to a curve may be more informative for predictive toxicology applications. Benchmark Dose Software (BMDS) has been recognized broadly and used for regulatory applications at multiple agencies. However, the current BMDS software was not amenable to modeling large datasets. Herein we describe development and use of a Python package that implements a wrapper around BMDS, a software that requires

---

**Corresponding Author:** Andrew Shapiro, shapiromatron@gmail.com. Current affiliation: Infinia ML, Durham, NC USA.

<sup>7.1</sup> Author's contributions

LLP, KPF, and SW wrote the manuscript, and used the software in the ToxRefDB v2.0. JW provided data and analysis for comparison to manual BMDS Wizard modeling and provided feedback on the re-analysis. AJS designed and implemented the Python BMDS interface and web application, portions of the manuscript, and integrated the BMDS interface in HAWC. All authors read and reviewed the final manuscript.

**Declaration of competing financial interests (CFI):** Andrew Shapiro has changed positions and is currently working at Infinia ML; however, work conducted at the current position is independent and unrelated to work in this publication. All other authors declare they have no actual or potential competing financial interests.

Declaration of interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

**Disclaimer:** The United States Environmental Protection Agency (U.S. EPA) through its Office of Research and Development has subjected this article to Agency administrative review and approved it for publication. Mention of trade names or commercial products does not constitute endorsement for use. The views expressed in this article are those of the authors and do not necessarily represent the views or policies of the US EPA.

**Publisher's Disclaimer:** This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

manual input in the dose-response modeling process (i.e., best-fitting model-selection, reporting, and dose-dropping). In the Python BMDS, users can select the BMDS version, customize model recommendation logic, and export summaries of the resultant BMDS output. Further, using the Python interface, a web-based application programming interface (API) has been developed for easy integration into other software systems, pipelines, or databases. Software utility was demonstrated via modeling nearly 28,000 datasets in ToxRefDB v2.0, re-creation of an existing, published large-scale analysis, and demonstration of usage in software such as CEBS and HAWC. Python BMDS enables rapid-batch processing of dose-response datasets using a modeling software with broad acceptance in the toxicology community, thereby providing an important tool for leveraging the publicly available quantitative toxicology data in a reproducible manner.

### Keywords

Benchmark dose modeling; software; dose-response; systematic review; *in vivo* toxicology

---

## 2 AVAILABILITY

Implemented in Python (available on the Python packaging index, <https://pypi.org/project/bmds/>). Source code is open-source and MIT licensed: Shapiro 2018a [1], Shapiro 2018b [2]; this code is available for anyone to use. A demo web application where one can model data is available at [bmds-python.com](https://bmds-python.com); this URL will be available for at least 2 years post-publication of this article.

## 4 INTRODUCTION

Dose-response analyses of hazard information are used to inform selection of a toxicologically-relevant dose for use in chemical risk assessments. Typically, a dose corresponding to observed effects from a repeat dose animal toxicity study is selected for each study to inform these assessments. The selected doses are called points-of-departure (PODs). Traditionally, PODs have been dose levels corresponding to the no or lowest observable adverse effect levels (NOAELs, LOAELs) from traditional animal studies. However, this traditional approach is limited because of dependencies on the dose selection, dose spacing, and sample size used in the original study. It is widely accepted that fitting a model to the dose-response data for observed effects can address some of the limitations of the traditional approach to POD selection [3-5]. Indeed, many modeling methods exist, including Benchmark Dose Software (BMDS) [6], PROAST [7], and the ToxCast Data Analysis (tcpl) and curve-fitting procedure [8]; however, one software program has predominately been used in toxicology: BMDS. It is important to implement a high-throughput functionality for BMDS because the model fitting procedure included in BMDS is familiar to toxicologists, risk assessors, and other stakeholders interested in defining a toxicologically-relevant dose based on a benchmark response (BMR), i.e., a user-defined value signifying a toxicologically significant change based on baseline recommendations or specific knowledge of the quantitative relationship between the specific effect and an adverse outcome. In many cases, the BMR has been based on the type of effect observed

(e.g., cancer versus non-cancer endpoints). Suggested defaults for BMR selection are available in EPA BMD guidance [6].

Adaptation of BMDS for automated analysis of large amounts of legacy data follows in the steps of previous work, briefly overviewed here. BMDS has been adapted for analysis of transcriptomic information with BMDExpress [9, 10], and now BMDExpress2.0 [11-13] which provides a command line interface to fit continuous, transcriptomic concentration response data in parallel. The BMDS Wizard [14] was designed to capture all required input and outputs for a BMDS session, defined as application of multiple models to a single dataset, in an Excel file. These software libraries or user-interfaces for BMDS are heavily customized for the specific use-cases for which they were designed; for instance, the BMDS graphical user interface and the BMDS Wizard are both useful in cases with limited numbers of datasets to be modeled, and BMDExpress constrains the input data to continuous data, where there are thousands of response (such as probes on a microarray) related to a single dose profile.

The work presented herein addresses a new use case: the need for BMD modeling of continuous and dichotomous data, integrated with other software and database systems used in computational toxicology. A new web application and software interface for BMDS was built using Python, which can be used for efficient modeling of large volumes of data. Unlike the BMDS Wizard where only one dataset can be modeled per Excel file, or BMD Express, which is designed specifically for continuous data from toxicogenomic studies, the Python BMDS interface library and web application can enable automation of the full suite of models available from the canonical EPA BMDS. This can facilitate the use of BMDS on large numbers of datasets, such as those now available in the US EPA's Toxicity Reference Database (ToxRefDB v2.0) [15], the National Toxicology Program's Chemical Effects in Biological Systems (CEBS) [16], or the Health Assessment Workspace Collaborative (HAWC), an open-source content management system for human health assessments [17, 18]. The application described herein enables easier software integration of BMD modeling, which may promote use of BMD modeling in toxicology research more broadly, as opposed to the NOAEL/LOAEL methods which are still most frequently used.

## 5 IMPLEMENTATION AND FEATURES

### 5.1 Programming languages

Concurrent with each release of BMDS, EPA releases both source code, developer documentation, and compiled versions of the software ([www.epa.gov/bmds](http://www.epa.gov/bmds); [6]). As discussed in Section 5.7, Future Work, the source code for the very recently released BMDS 3.0 was not available at the time of this publication. For previous versions of BMDS (2.7 and below), the published source-code includes two components: a user-interface for BMD modeling which is written in the C# programming language, and the individual dose-response models, where each executable represents a single model or a model family (e.g., Hill, multistage, exponential, logistic), written in C. Extensive developer documentation is provided for input files to the dose-response models.

Due to its common use, performance characteristics with regards to interoperability with C, and extensive existing software libraries in data science and application development, we selected the Python software language for development of the BMDS interface library. Python is one of the most widely used programming languages and has seen substantial growth in the last few years in the data-science domain, as well as general scripting and web-applications [19, 20]. Further, the Python programming language interfaces well with other programming languages such as C, Fortran, R, and Java; therefore; is an ideal language for a scientific software interface, as these languages are frequently used in scientific applications.

## 5.2 Benchmark dose modeling features

The interface library, i.e., the Python package which acts as an entry point to the underlying BMDS models, was engineered with three key design considerations. First, it was designed to be adaptable to analysis-specific modeling decisions, e.g. model recommendation and/or high dose-dropping, and flexible so that users of the software could implement modeling per the needs of their application. However, common use case patterns were also implemented, such as automated model recommendation, as previously implemented in the BMDS Wizard software [14]. Second, performance (i.e., speed) was also important when designing the interface library; by default modeling is highly concurrent with multiprocessor support. Finally, it was important to ensure that the EPA BMDS modeling library, instead of a surrogate, is used for all complex numerical evaluations of the dose-response data. This includes using the compiled code provided by EPA with no modifications. Since the BMD software has previously been applied in myriad research and regulatory applications, we did not want to change any features which have already gone through extensive peer-review. A comparison of four applications of BMDS are shown in Table 1.

The EPA BMDS package and associated EPA BMD tools implement dose-response modeling for many types of toxicological data, including continuous, dichotomous, nested-dichotomous, categorical (via Categorical Regression Analysis, or CatReg [21]), repeated response, etc. The Python BMDS interface library was implemented to include three standard widely-used data types, continuous, dichotomous, and cancer-dichotomous; however, the Python BMDS could be extended in the future to include new releases of BMDS (version 3.0, [www.epa.gov/bmbs](http://www.epa.gov/bmbs)), other data types, or even non-BMDS approaches, such as Bayesian benchmark dose estimation [22]. The Python BMDS workflow is depicted in Figure 1, starting with the (A) data input, working through the (B) Python BMDS processing, to (C) outputs from the software.

A primary feature of using Python BMDS is the ability to run data modeling sessions concurrently. As in all versions of EPA BMDS, a single dataset can be added and modeled with multiple dose-response modeling curves in a session. However, there are limitations to the batch processing capabilities of BMDS (i.e., running multiple sessions). In contrast, the Python BMDS interface library, allows for dozens to thousands of sessions to be created and executed in parallel, thus dramatically increasing the possible use-cases for the software. Higher-throughput analyses that could take advantage of this include modeling of datasets

from large legacy databases, systematic reviews, or possibly even high-throughput screening modeling applications.

An additional strength of the Python BMDS interface is its ease in integrating benchmark dose modeling into other applications by providing a simple API (Figure 1B). For continuous datasets, the BMDS software includes two different variance models, i.e., constant or modeled variance [14] when using BMDS Wizard, models may need to be executed twice in order to use the correct variance model. The Python BMDS interface library implements the same test used in BMDS, and the appropriate variance model can be automatically selected, reducing model outputs and runtime by half for each dataset. Evaluating multiple BMRs is also simple; sessions can be cloned and evaluated with different BMRs. Model recommendation logic largely follows the logic used in the BMDS Wizard as described by Wignall and colleagues (2014) [14] (Figure 1), with some minor updates noted in Section 5.6 (details on default recommendation logic are in the online documentation) per more recent BMDS guidance. The logic used for model recommendation in Python BMDS is fully customizable, i.e., new rules could be created, or existing rules removed. The software could also be extended to use Bayesian model-averaging techniques such as those implemented in BMDS 3.0. Finally, the software includes an option to drop the highest dose-group when a model cannot be recommended, as long as there are a sufficient number of dose-groups for modeling. This is a common approach to dose-response modeling where higher dose groups may behave non-monotonically due to mortality or acute toxicity rather than the endpoint of interest. By default, this option is disabled; however, when applied, the BMDS output preserves the original dataset and notes the number of doses dropped to enable a model with a recommendation, rather than forcing a user to track the re-analysis manually.

### 5.3 Outputs

Reports can be generated in standard human readable formats (Word, Excel), or machine-readable formats (JSON, Python pickle serialization, etc.) (Figure 1C). Dose-response plots can be generated and customized using the popular matplotlib [23] Python plotting library and exported in standard formats (PNG, SVG, PDF, TIFF, etc.). The software was written in an object-oriented design pattern, e.g., classes for common idioms in dose-response modeling include descriptors such as Session, SessionBatch, Dataset, and Model, which can be composed by software developers or data analysts to build pipelines or integrate into other software.

### 5.4 Versioning, compatibility, and web application

The Python BMDS interface library uses the most recent version of EPA BMDS in the version 2 series by default (2.7), however prior versions of EPA BMDS are also available in the Python BMDS interface library. Generally, BMD analyses and assessments are done with a single version of EPA BMDS; the interface library was designed to allow forwards and backwards compatibility with BMDS as new versions are created, so that prior analyses can be reproduced. The Python BMDS interface library currently includes multiple versions of BMDS, including versions 2.3.1, 2.4, 2.6, and 2.7. The software will be updated to use the

newly released version 3 series of BMDS [24] when developer documentation becomes available, as described below in Section 5.7.

The US EPA releases both the source-code for dose-response models used in BMDS, as well as compiled-executables of these models for Microsoft Windows operating system. However, for BMDS 2.7 and below there can be minor differences in versions of the model executables based on various settings during compiling, which are operating-system specific (e.g., BMDS models compiled on Microsoft Windows may give slightly different answers than BMDS on Apple MacOS, or Linux web-servers). Therefore, while it is possible to compile and use the EPA models distributed prior to BMDS 3.0 on different operating systems, in order to make results fully reproducible with previous versions, it is important to rely exclusively on using the pre-compiled versions of the software which are distributed by EPA. Thus, for BMDS 2.7 and below, reproducibility is only ensured by the use of pre-compiled models executed in a Windows operating system environment. Restricting use of the Python BMDS interface library on a Windows operating system may pose problems in developing pipelines or integrating into other software systems, as Linux environments are frequently used for web-applications and high-performance computing. Therefore, in addition to developing a Python BMDS interface library, a BMDS web application was also developed that allows for the Python BMDS interface library to be used on any computer that can access the web. Users provide a dataset or list of datasets and BMDS modeling options. The web application converts the request to “jobs,” which are placed on a queue and are modeled as computing resources are available. BMDS modeling results can be downloaded and are deleted after a period of time (defaults to one week). While users can interact directly with the web application using a web browser, the web application was primarily designed for interfacing with other software platforms or pipelines where dose-response modeling was needed. The web application was implemented using the widely-used Python Django web framework [25], and is deployed on a Windows server so that the compiled executable files provided by EPA are used, ensuring results are fully reproducible in any environment. A Representational State Transfer (REST) application programming interface (API) was implemented for submitting/receiving data from the web application, a *de facto* standard in transferring data/requests on the internet [26]. A demonstration website, at [bmds-python.com](http://bmds-python.com), is available as described above.

## 5.5 Large scale batch processing using the Python BMDS interface

Wignall *et al.* (2014) [14] first demonstrated the ability to use BMDS with a set of standardized BMD settings on a database of toxicological data. The original analysis was conducted using the BMDS Wizard, an Excel-based wrapper of the EPA BMDS modeling software version 2.3.1, where each dataset along with model settings and results are saved in a single Excel file. The modeling was conducted in a semi-automated fashion; each Excel file was automatically created but manually executed. Doses may have been dropped as needed and saved in separate Excel files, a manual process, and results were aggregated manually for subsequent analysis.

We referenced the analysis from Wignall *et al.* (2014) [14] to test concordance of the Python BMDS interface library with a previous analysis. After preparing data, modeling setup and

execution was conducted in 18.5 minutes, or approximately 1.0 second per dataset (using a 2012 laptop with 4-core hyperthreaded Intel i7-3610 processors and 8GB RAM). In total, over 7,000 individual dose-response models were executed, a throughput of approximately 390 models per minute. The total time taken for model executions was not captured in Wignall *et al.* 2014 [14], but assuming 1.5 minutes per dataset for execution and human review, selection of model recommendation, manual dose-dropping, and data transfer into summary spreadsheets, it would take approximately 22 hours of time. Note also that all pre- and post-processing using the Python BMDS interface is captured in code, whereas the prior study required substantial manual editing and aggregation of tables, which limits reproducibility. Detailed results for the analysis, including input and output files, as well as the source code for the analysis, are available in Supplemental Material 1.

Overall, high similarity was found in results between the prior analysis and the reanalysis using this software library (Figure 2). In terms of the number and type of datasets fit, 87.7% of the 896 datasets reanalyzed demonstrated a concordant result (either recommended or did not recommend a best-fitting model), while in 12.3% of the cases a different result was achieved (83.2% matching continuous, 86.9% dichotomous, and 90.2% cancer dichotomous). In many cases the Python BMDS interface recommended a model when the BMDS wizard analysis was unable to (24, 37, and 38 datasets, for continuous, dichotomous, and cancer dichotomous, respectively). The opposite case was also observed (3, 6, and 2 datasets, respectively).

For cases where both Python BMDS and BMDS Wizard calculated a BMD value, there is a high degree of concordance between recommended BMD values; a linear regression of the Python BMDS and BMDS Wizard BMD values produced an  $R^2$  0.941 (Figure 2B). Over 75% of datasets with BMD values in both cases were within 0.01 relative dose-units equivalence; in absolute terms 84% of datasets were within 1 dose-unit. Discordances between analyses could be a byproduct of BMDS version (2.3.1 vs 2.7), workflow implementation and manual processing in the prior analysis, additional model-options in Python BMDS, or mapping old summary files to the new analysis. There are also slight differences in workflow implementation; for example, previously in BMDS Wizard, only a single multistage model was executed with a polynomial degree equal to the number of dose groups (n) minus 1; in the Python BMDS interface, multiple multistage models are executed from order 1 to n-1 by default, and therefore additional models are available for possible model-fit. As previously stated, all recommendation logic can be modified, but the more stringent logic can result in differences in model recommendations. A fully automated workflow as implemented with the Python BMDS interface improves on limitations with manual processing, including facilitating and maintaining ease of reproducibility, standardizing the approach, and limiting user-error in configuration or manipulation of input and output data. It should be noted that even repeating an older analysis conducted in a different platform (BMDS Wizard) demonstrates the utility of the Python BMDS interface; it allows flexibility in modeling, model recommendation logic, and analysis of results.

The application of Python BMDS in ToxRefDB v2.0 is currently the largest automated implementation of BMDS (1,170 cancer datasets; 17,318 non-cancer dichotomous datasets; and 9,268 continuous datasets), thus serving as a proof-of-concept for large batch-processing

of dose-response data using BMDS. Using Python BMDS to analyze data in ToxRefDB v2.0, models were built for effect data from five different types of study designs, including chronic, subchronic, subacute, multi-generation reproduction, and developmental study designs, and over 300 categories of observed effects, e.g. the tissue or effect of interest [15]. Based on BMDS guidance [27], default BMRs were selected for each dose response-type, i.e. cancer (5% and 10%), non-cancer dichotomous (5% and 10%), continuous body/organ weight (10% relative deviation), and continuous non-body/organ weight (1 standard deviation), though these could be expanded in the future. Approximately one-third of the available quantitative dose-response datasets in ToxRefDB v2.0 passed quality control and minimum dataset requirements imposed (e.g., N, three dose groups and a control, and an estimate of variance such as standard deviation) for batch processing by Python BMDS. Of the 27,756 total datasets, 73-86.7% had at least one recommended model that yielded a BMD and BMDL value for the chemical-effect pair, depending if 5 or 10% BMR is used, respectively. However, specific use cases may necessitate the need for expert judgement and evaluation before use. The datasets and models are available in the ToxRefDB v2.0 release ([15]; GitHub link <https://github.com/USEPA/CompTox-ToxRefDB>).

## 5.6 Application integrations using the Python BMDS web application

Here, we discuss additional examples early adoption of the publicly available Python BMDS code and/or web application by other data aggregation projects. The Python BMDS web application has been adopted for use in the NTP CEBS database [16]. BMDS was used to model data from the National Toxicology Program's chronic carcinogenicity studies stored in CEBS, with BMD and BMDL values calculated for significant histopathological findings using a default BMR. In the future, CEBS is planning to calculate BMD values for all treatment-related findings in NTP public data (from toxicology and carcinogenicity studies, five-day dose-response transcriptomics studies, genetic toxicology assessments, etc.). Currently, internal NTP users can access data in CEBS and run the Python BMDS service for these data; however, it is anticipated that these features in CEBS will become available to the public in the future. Another project utilizing Python BMDS is the HAWC project (<https://hawcproject.org/>), a free, open-source content-management designed for systematic reviews of human health assessments [17]. The software has been used for a number of reports by state, federal, and international agencies, including a National Academy of Sciences report [28]. Users can create an assessment where bioassay, epidemiological, and *in vitro* data can be extracted and integrated in summary visualizations. Bioassay data can be modeled using BMDS in HAWC, which is implemented using the Python BMDS library. BMD analyses can be created by the user with full control of endpoint selection, BMR(s), and model settings. An example analysis is shown here: <https://hawcproject.org/ani/endpoint/33/>.

## 5.7 Future work

The Python BMDS interface library is dependent on the official EPA BMDS software releases. Thus, when new versions of the EPA software are released, Python BMDS will also be updated. A new version of BMDS (version 3.0) was released in October of 2018. This release was a major update to the software, including completely rewriting the existing API for how models are executed; instead of executing models via a command-line interface,



they are executed via function calls using shared-object libraries. At the time of writing, the software was released to the public, but source code and developer documentation were not yet available. However, when these become available, the Python BMDS interface library will be updated to include new BMDS versions. The authors believe that in addition to improvements in modeling approaches (e.g., Bayesian model averaging for dichotomous datasets, etc.), there will likely be faster performance of modeling. Further, should shared-object libraries be released in operating systems other than Windows (i.e., 64-bit Linux, Mac), these could also be used to allow for platform agnostic model execution, thus relaxing the Windows-only execution restriction of the current implementation. As guidance and user experience will BMDS version 3.0, these can shape the evolution of Python BMDS implementation.

## 6 CONCLUSIONS

From a regulatory and scientific perspective, EPA BMDS has provided dramatic advances in the ability to build a unified approach to dose-response modeling, which is a foundational principal in human health risk assessment and toxicology research applications. EPA has publicly provided source code and extensive documentation on how to use the software both at a user-interface level, as well as for software developers. However, from a software integration perspective, BMDS was difficult to integrate into other software systems due to operating system dependencies and input/output formatting constraints. Herein we present a new Python interface for BMDS and a web-application for executing BMD modeling. We have demonstrated the utility of this system in implementing the gold-standard BMDS modeling methods on medium to large dose-response datasets in an automated fashion. The robustness and flexibility of the Python BMDS interface to be incorporated into automated pipeline have been demonstrated in multiple use cases, such as during systematic review where data are extracted from literature or using large datasets in existing databases. The Python BMDS interface was designed in such a way that the software can be used in the future for further integrations with other data systems or data pipelines, using modern software interfaces and development techniques. Finally, it is open-source and publicly available with a permissive software license, which enables re-use and future collaborations. Python BMDS provides a tool that advances reproducibility in modeling of large toxicology databases, substantially extending the functionality of BMDS for predictive toxicology applications.

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## Acknowledgements

The authors would like to thank Jeff Gift, Reeder Sams, R. Woodrow Setzer, and Imran Shah (U.S. EPA) and Scott Auerbach and Jennifer Fostel (U.S. National Toxicology Program) for their insightful comments on previous versions of this manuscript.

### 7.3 Funding

L. L. P. was supported by appointment to the Research Participation Program of the U.S. Environmental Protection Agency, Office of Research and Development, administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and the U.S. EPA.

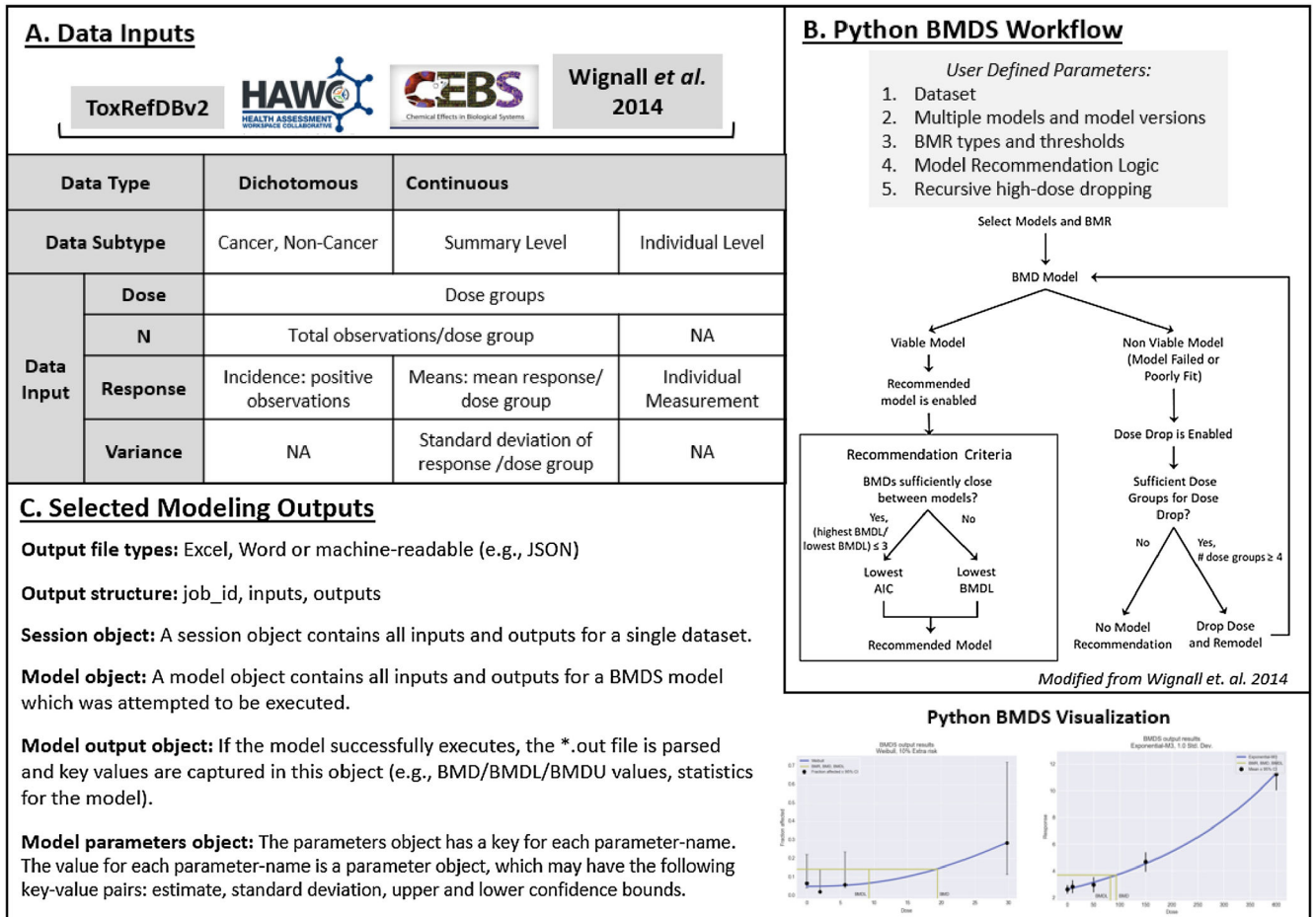
## 9 References

- [1]. Shapiro A, BMDS python source code. 2018a, 10.5281/zenodo.1419333 Zenodo, 2008.
- [2]. Shapiro A, BMDS python server source code. 2018b, 10.5281/zenodo.1419828 Zenodo, 2018.
- [3]. Slob W, Benchmark dose and the three Rs. Part I. Getting more information from the same number of animals, *Critical reviews in toxicology* 44(7) (2014) 557–67. [PubMed: 25000332]
- [4]. Slob W, Benchmark dose and the three Rs. Part II. Consequences for study design and animal use, *Critical reviews in toxicology* 44(7) (2014) 568–80. [PubMed: 25000331]
- [5]. Slob W, Setzer RW, Shape and steepness of toxicological dose-response relationships of continuous endpoints, *Critical reviews in toxicology* 44(3) (2014) 270–97. [PubMed: 24252121]
- [6]. Davis JA, Gift JS, Zhao QJ, Introduction to benchmark dose methods and U.S. EPA's benchmark dose software (BMDS) version 2.1.1, *Toxicology and applied pharmacology* 254(2) (2011)181–91. [PubMed: 21034758]
- [7]. Hardy A, Benford D, Halldorsson T, Jeger MJ, Knutsen KH, More S, Mortensen A, Naegeli H, Noteborn H, Ockleford C, Ricci A, Rychen G, Silano V, Solecki R, Turck D, Aerts M, Bodin L, Davis A, Edler L, Gundert-Remy U, Sand S, Slob W, Bottex B, Abrahantes JC, Marques DC, Kass G, Schlatter JR, Update: use of the benchmark dose approach in risk assessment, *EFSA Journal* 15(1) (2017) e04658.
- [8]. Filer DL, Kothiyi P, Setzer RW, Judson RS, Martin MT, tcpl: the ToxCast pipeline for high-throughput screening data, *Bioinformatics* 33(4) (2017) 618–620. [PubMed: 27797781]
- [9]. Yang L, Allen BC, Thomas RS, BMDEExpress: a software tool for the benchmark dose analyses of genomic data, *BMC Genomics* 8(1) (2007) 387. [PubMed: 17961223]
- [10]. Auerbach S, The NTP Proposed Approach to Genomic Dose-Response Modeling - Scott Auerbach, Ph.D., NIH/NIEHS/DNTP, 2017 <https://ntp.niehs.nih.gov/about/org/sep/ntpexpertpanel/meetings/2017/videos/index.html>.
- [11]. Sciome, BMDEExpress 2 software, National Toxicology Program, <https://www.sciome.com/bmdexpress/>, 2018.
- [12]. Auerbach S, BMDEExpress-2 Wiki, 2018 <https://github.com/auerbachs/BMDEExpress-2/wiki/>.
- [13]. Phillips JR, Svoboda DL, Tandon A, Patel S, Sedykh A, Mav D, Kuo B, Yauk CL, Yang L, Thomas RS, Gift JS, Davis JA, Olyszk L, Merrick BA, Paules RS, Parham F, Saddler T, Shah RR, Auerbach SS, BMDEExpress 2: Enhanced transcriptomic dose-response analysis workflow, *Bioinformatics* (2018) bty878–bty878.
- [14]. Wignall JA, Shapiro AJ, Wright FA, Woodruff TJ, Chiu WA, Guyton KZ, Rusyn I, Standardizing benchmark dose calculations to improve science-based decisions in human health assessments, *Environmental health perspectives* 122(5) (2014) 499–505. [PubMed: 24569956]
- [15]. Watford S, Pham LL, Wignall JA, Shin R, Martin MT, Paul Friedman K, ToxRefDB version 2.0: Improved utility for predictive and retrospective toxicology analyses, *Reproductive Toxicology* (submitted).
- [16]. Lea IA, Gong H, Paleja A, Rashid A, Fostel J, CEBS: a comprehensive annotated database of toxicological data, *Nucleic acids research* 45(D1) (2017) D964–d971. [PubMed: 27899660]
- [17]. HAWC, HAWC: Health Assessment Workplace Collaborative, 2013 <https://hawcproject.org/>. (Accessed August 23, 2018).
- [18]. Shapiro AJ, Antoni S, Guyton KZ, Lunn RM, Loomis D, Rusyn I, Jahnke GD, Schwingl PJ, Mehta SS, Addington J, Guha N, Software Tools to Facilitate Systematic Review Used for Cancer Hazard Identification, *Environmental health perspectives* 126(10) (2018) 104501. [PubMed: 30392397]
- [19]. Robinson D, The Incredible Growth of Python <https://stackoverflow.blog/2017/09/06/incredible-growth-python/>, 2017.

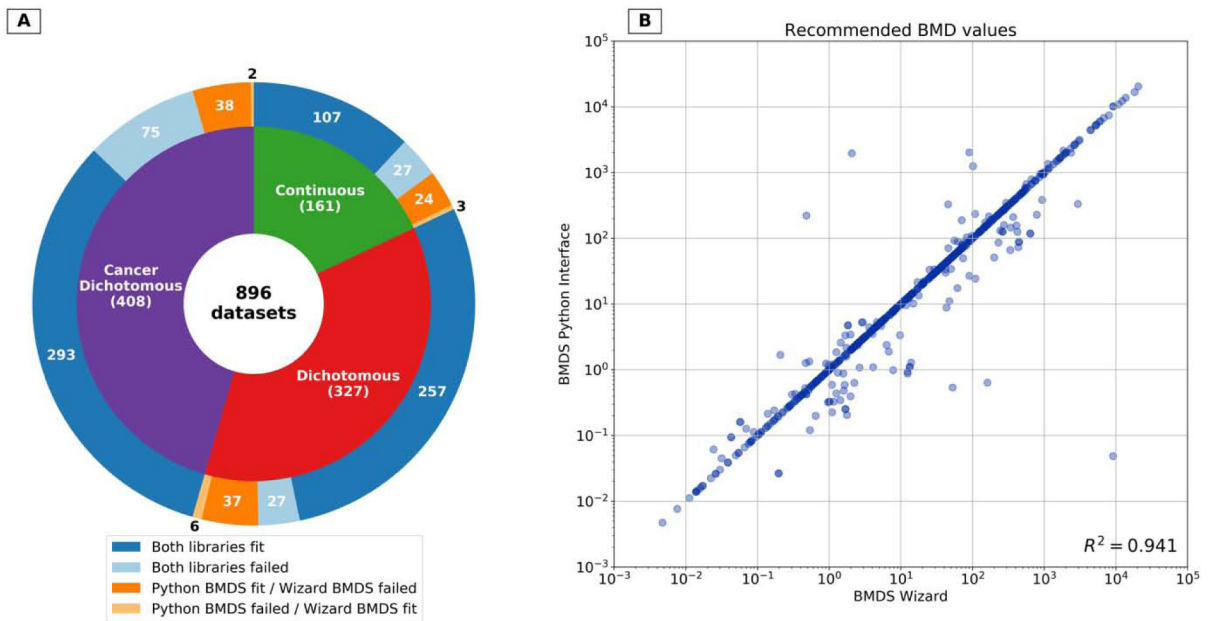
- [20]. Cass S, The 2017 Top Programming Languages Python jumps to No. 1, and Swift enters the Top Ten, 2017 <https://spectrum.ieee.org/computing/software/the-2017-top-programming-languages-2018>).
- [21]. U. EPA, Categorical Regression (CatReg) Software version 3.1.0.7, <https://www.epa.gov/bmds/catreg>, 2017.
- [22]. Shao K, Shapiro AJ, A Web-Based System for Bayesian Benchmark Dose Estimation, *Environmental health perspectives* 126(1) (2018) 017002. [PubMed: 29329100]
- [23]. Hunter JD, Matplotlib: A 2D graphics environment, *Comput Sci Eng* 9(3) (2007) 90–95.
- [24]. USEPA, Benchmark Dose Software (BMDS) - Version 3.0, 2018.
- [25]. "DjangoSoftwareFoundation", Django, <https://www.djangoproject.com/>, 2018, p. [Computer Software].
- [26]. Fielding RT, Taylor RN, Principled design of the modern Web architecture, *ACM Trans. Internet Technol* 2(2) (2002) 115–150.
- [27]. Benchmark Dose Technical Guidance, <https://www.epa.gov/risk/benchmark-dose-technical-guidance>, 2012, pp. 1–99.
- [28]. E. National Academies of Sciences, and Medicine, *Application of Systematic Review Methods in an Overall Strategy for Evaluating Low-Dose Toxicity from Endocrine Active Chemicals*, National Academies Press (US), Washington, D.C., 2017.

### Highlights

- Python BMDS and web application enable automation of models available from BMDS.
- Large public datasets can now be efficiently modeled for predictive toxicology.
- Python BMDS users can customize BMDS version and model recommendation logic.
- Python BMDS and previously published BMD values were highly concordant.
- Python BMDS was used to model nearly 28,000 datasets in ToxRefDB version 2.0.



**Figure 1. Python BMDS workflow.** The workflow, including necessary inputs (A), the logic used in the Python BMDS for fitting (B), and the stored output (C) are illustrated.



**Figure 2: Comparison of BMD estimates from BMDS Wizard and Python BMDS Interface.** Re-analysis using the Python BMDS library was compared for 896 datasets from Wignall *et al.* 2014 [14]. (A) 87.7% of the reanalyzed datasets demonstrated a concordant result, with either both libraries fitting (dark blue) or failing (light blue). In some cases (for 24, 37, and 38 datasets, for continuous, dichotomous, and cancer dichotomous, respectively), the Python BMDS interface recommended a model when the BMDS wizard analysis did not (dark orange). Conversely, for 3, 6, and 2 datasets, Python BMDS failed and the BMDS Wizard fit (light orange). (B) When BMD estimates were recommended in both analyses, a linear regression demonstrates high concordance, with an R-squared of 0.941.

**Table 1.**

Comparison of software libraries using EPA BMDS

Feature	U.S. EPA BMDS 2.7	U.S. EPA BMDS 3.0	BMDS Wizard	BMDExpress2.0	Python BMDS interface and web application
User interface	Standalone GUI + command line	Excel + programmatic API	Excel	GUI or command line	Web-based REST API or Python programmatic API
Language(s)	C+.NET	C + Excel + VBA	Excel + VBA	Java	Python
Data structure	Single dose-response dataset	Single/multiple dose-response dataset(s)	Single dose-response dataset	Toxicogenomic dataset	Single/multiple dose-response dataset(s)
Data input	Manual	Manual	Manual or Scripted	Manual or Scripted	Manual or Scripted
Throughput	Low	Moderate	Low	High	High
Data types	Continuous, Dichotomous, Nested-dichotomous, Categorical, etc.	Continuous, Dichotomous, Nested-dichotomous, Categorical, etc.	Continuous, dichotomous	Continuous	Continuous, dichotomous
Model recommendation	No automated recommendation	Automated; customizable decision logic or model-averaging	Automated; customizable decision logic	Automated; predefined decision logic	Automated; customizable decision logic
Continuous data variance model selection	Manual	Automatic	Manual	Scripted	Automatic
Output formats	Text files, EMF	Excel, Word, native excel plots	Excel, Word	TSV, JSON, PNG, SVG	Word, Excel, JSON, Text, PNG, SVG
Supported Operating Systems	Windows	Windows	Windows	Windows, Mac Linux, (BMDS models recompiled)	Windows with library; any with web application (using native Windows BMDS binaries)