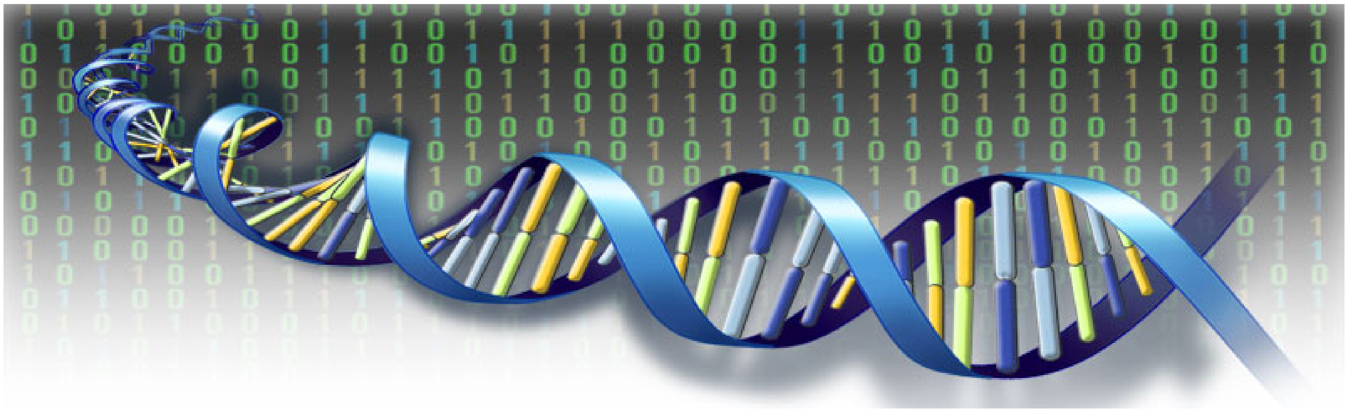


Toxicity Reference Database User Guide

by
Sean Watford, Ly Ly Pham, Katie Paul Friedman
US Environmental Protection Agency
Office of Research and Development
National Center for Computational Toxicology
Research Triangle Park, North Carolina

Version 1
February 2019

EPA United States Environmental Protection Agency

Purpose

The purpose of this document is to provide documentation on how to technically access and use ToxRefDB version 2.0. The latest data can be accessed through EPA's FTP site. More information about ToxRefDB version 2.0 and its development can be found in the recent publications below. Please use the contact us form for further questions.

Watford S, Pham LL, Wignall J, Shin R, Martin MT, Paul Friedman, K. (submitted). ToxRefDB version 2.0: Improved utility for predictive and retrospective toxicology analyses.

Pham, L.L., Watford S, Paul Friedman K, Fostel J, Wignall J, Shapiro A. (submitted). Python BMDS: A python interface and webserver for the canonical EPA dose-response modeling software.

This user guide does not necessarily reflect U.S. EPA policy.

Table of Contents

1 Overview

The Toxicity Reference Database (ToxRefDB) structures data from approximately 5,900 studies or study summaries based on *in vivo* experiments, conducted predominantly to guidelines and specifications from the US Environmental Protection Agency (US EPA) and the National Toxicology Program (NTP) headquartered at the National Institute of Environmental Health Sciences, into a public resource that has been used in the training and validation of predictive toxicology models. These data are useful because it serves as a resource for retrospective analysis and aids in the development of predictive models by allowing researchers to compare the results of high throughput *in vitro* experiments and predictive models to data from *in vivo* experiments.

Many of these studies (over 3,000 of them) come from registrant-submitted toxicity studies known as data evaluation records (DERs) from the U.S. EPA's Office of Pesticide Programs (OPP). The majority of chemicals in the database are therefore pesticides. Since 2009, our current and continued efforts in data collection and curation will have expanded ToxRefDB to include toxicity studies from additional sources, including the National Toxicology Program (NTP), peer-reviewed primary research articles, and pharmaceutical preclinical toxicity studies (Pfizer, Sanofi, GSK, Merck), among others (RIVM, PMRA, unpublished and unassigned sources)

The first version of ToxRefDB (ToxRefDB 1.0) was initially released as series of spreadsheets, which are still available on EPA's FTP site and referenced in FigShare (**https://doi.org/10.23645/epacomptox.6062545.v1**). Since the last update in 2014, ToxRefDB has undergone significant updates that are described in the recent publication (Watford et al., submitted) and has been released as ToxRefDB 2.0.

1.1 Summary Statistics

ToxRefDB version 2.0 contains summary information from 5,960 studies or study summaries for 1,142 chemicals. Table 1 shows both the number of chemicals and number of studies for each study type and species. As part of the ToxRefDB 2.0 update, quantitative (i.e. dose-response) data was extracted. Currently, this is completed for 3,882 studies with plans to extract and release the remaining data in subsequent data releases. Study reliability was assessed using ToxRTool for 528 studies that did not explicitly comply with a guideline or specification.

Table 1. Summary of study types by species, with the number of studies and number of chemicals associated.

Study type	Study source	Species	Number of studies	Number of chemicals
Acute (ACU)	OpenLit	rat	6	6
	OPP DER	rabbit	1	1
		rat	4	4
Total ACU:			11	10
Chronic (CHR)	NTP	mouse	188	183
		rat	189	183
	OpenLit	dog	11	11
		mouse	45	41
		rabbit	1	1

		rat	67	62
	OPP DER	dog	333	299
		hamster	5	4
		mouse	344	304
		primate	1	1
		rat	401	330
	Other	dog	2	2
		mouse	3	3
		rat	7	7
	Pharma	dog	28	20
		mouse	7	7
		primate	4	4
		rat	33	28
Total CHR:			1669	663
Developmental (DEV)	NTP	mouse	14	11
		rabbit	8	8
		rat	14	13
	OpenLit	dog	1	1
		hamster	2	2
		mouse	37	31
		rabbit	44	39
		rat	157	117
	OPP DER	mouse	22	18
		rabbit	448	378
		rat	552	452
	Other	mouse	1	1
		rabbit	7	6
		rat	17	14
	Pharma	mouse	2	1
		rabbit	87	50
		rat	93	53
Total DEV:			1506	710
Developmental Neurotoxicity (DNT)	OpenLit	hamster	1	1
		mouse	13	13
		rat	90	55
	OPP DER	rat	80	64
	Other	rat	1	1
Total DNT:			185	124
Multigeneration Reproductive (MGR)	NTP	mouse	13	11
		rat	6	6
	OpenLit	dog	1	1
		hamster	1	1
		mink	2	2
		mouse	18	17

		rat	68	58
	OPP DER	mouse	3	3
		rat	401	346
	Other	rat	33	32
	Pharma	rat	1	1
Total MGR:			547	458
Neurotoxicity (NEU)	OpenLit	mouse	1	1
		rat	10	10
	OPP DER	rat	5	5
	Other	rat	2	2
Total NEU:			18	18
Other (OTH)	NTP	mouse	1	1
	OpenLit	rat	2	2
	OPP DER	dog	2	2
		mouse	1	1
		rat	5	5
	Pharma	dog	2	2
		mouse	1	1
		primate	1	1
		rabbit	1	1
		rat	9	6
Total OTH:			25	18
Reproductive (REP)	NTP	rat	9	8
	OpenLit	mouse	4	4
		rat	25	22
	OPP DER	rat	3	3
	Pharma	mouse	3	2
		rat	73	40
Total REP:			117	77
Subacute (SAC)	NTP	mouse	37	33
		rat	35	34
	OpenLit	dog	5	5
		mouse	2	2
		rabbit	1	1
		rat	26	22
	OPP DER	dog	2	2
		mouse	5	5
		rabbit	9	9
		rat	23	20
	Other	rabbit	1	1
		rat	4	4
	Pharma	dog	84	62
		mouse	20	11
		primate	57	26

		rabbit	28	22
		rat	187	94
Total SAC:			526	191
Sub-chronic (SUB)	NTP	hamster	1	1
		mouse	182	163
		rat	183	160
	OpenLit	dog	4	4
		mouse	21	18
		primate	1	1
		rabbit	2	2
		rat	91	75
	OPP DER	dog	216	196
		guinea-pig	2	2
		hamster	5	4
		mouse	123	112
		primate	3	3
		rabbit	6	5
		rat	426	337
	Other	dog	3	3
		mouse	2	2
		rat	10	10
	Pharma	dog	24	21
		mouse	16	16
		primate	10	9
		rat	25	22
Total SUB:			1356	659
Database totals:			5960	1142

2 Accessing information in ToxRefDB

ToxRefDB 2.0 is available as spreadsheets and a MySQL database ((**https://doi.org/10.23645/epacomptox.6062545.v2**). All files and resources can be found on the FTP site. The spreadsheets contain information in a format that many users are familiar with from ToxRefDB 1.0. Study-level information is also available as spreadsheets that contain all information that has been extracted and calculated, i.e. effect levels including lowest effect levels (LELs) and lowest observable adverse effect levels (LOAELS), for a given study. The entire MySQL database is available for customized applications. The database schema is also available on the FTP site along with a data dictionary. Below is documentation on how to install MySQL, load ToxRefDB, and access the data using both SQL and programmatic access using either Python or R. Another useful tool to access the data is MySQL Workbench, which provides a user interface to interact with any MySQL database.

2.1 Installing MySQL and loading ToxRefDB

- **Download the ToxRefDB MySQL database**
- **Download the latest version of the MySQL community server**
    - **Select the appropriate installer for your operating system**
        - **For Windows, download the MSI installer**
        - **For MAC, download the DMG installer**
    - **The installer will walk you through the installation**
    - **During the installation, be sure to copy the temporary root password. You will need it later.**
    - **For Windows, MySQL should automatically be added to your PATH**
    - **For MAC, if MySQL was not added to your PATH automatically you will have to add it manually**
        - **Open the terminal and type:**

```
>> echo 'export PATH=/usr/local/mysql/bin:$PATH'
>> ~/.bash_profile
```

- **Open the command line (Windows) or terminal (MAC) to login to the MySQL server with the command**

```
>> mysql -u root -p
```

- **Enter the temporary root password when prompted for a password**
    - **Change the root password**
        - **Full documentation can be at the MySQL website**

- **Create the ToxRefDB database, select it as the default database, and load the dump file:**

```
mysql> CREATE DATABASE IF NOT EXISTS toxrefdb_2_0;
mysql> USE toxrefdb_2_0;
mysql> source toxrefdb_2_0.sql
```

  - **Further documentation can be found at the MySQL website**

2.2 Example queries

2.2.1 Get number of studies per study type

```
SELECT study_type, COUNT(study_id) FROM study GROUP BY study_type;
```

2.2.2 Get number of studies per study type and species

```
SELECT study_type,species, COUNT(study_id) FROM study GROUP BY
study_type,species;
```

2.2.3 Get number of studies per source

```
SELECT study_source, COUNT(study_id) FROM study GROUP BY study_source;
```

2.2.4 Get all study information for chronic studies

```
SELECT * FROM study WHERE study_type="CHR";
```

2.2.5 Get all treatment group and dosing information for a single chemical

```
SELECT *
FROM
    chemical INNER JOIN study ON chemical.chemical_id=study.chemical_id
        INNER JOIN tg ON tg.study_id=study.study_id
            INNER JOIN dose ON dose.study_id=study.study_id
                INNER JOIN dtg ON dtg.tg_id=tg.tg_id AND dose.dose_id=dtg.dose_id
WHERE casrn="42509-80-8";
```

2.2.6 Get number of studies per endpoint

```
SELECT
    endpoint_category,
    endpoint_type,
    endpoint_target,
    COUNT(DISTINCT study.study_id) AS "number of studies"
FROM
    study INNER JOIN tg ON study.study_id=tg.study_id
        INNER JOIN tg_effect ON tg.tg_id=tg_effect.tg_id
            INNER JOIN effect ON effect.effect_id=tg_effect.effect_id
                INNER JOIN endpoint ON endpoint.endpoint_id=effect.endpoint_id
GROUP BY endpoint_category,endpoint_type,endpoint_target;
```

2.2.7 Get all study-level LELs and LOAELs for effect profile 2

```
SELECT * FROM pod WHERE effect_profile_id=2 AND study_id IS NOT NULL AND pod_type
IN("loael","lel");
```

2.2.8 Get chemical-level PODs for effect profile 2

```
SELECT * FROM pod WHERE effect_profile_id=2 AND study_id IS NULL;
```

2.2.9 Get study-level PODs for effect profile 2 and for a specific endpoint

```
SELECT DISTINCT pod.*
FROM
    pod INNER JOIN pod_tg_effect ON pod.pod_id=pod_tg_effect.pod_id
        INNER JOIN tg_effect ON tg_effect.tg_effect_id=pod_tg_effect.tg_effect_id
            INNER JOIN effect ON effect.effect_id=tg_effect.effect_id
                INNER JOIN endpoint ON endpoint.endpoint_id=effect.endpoint_id
WHERE effect_profile_id=2 AND study_id IS NOT NULL AND endpoint_target LIKE
"thyroid%";
```

2.2.10 Get all BMD models for a chemical

```
SELECT *
FROM
    bmd_models INNER JOIN study ON study.study_id=bmd_models.study_id
        INNER JOIN chemical ON chemical.chemical_id=study.chemical_id
WHERE casrn="42509-80-8";
```

2.2.11 Get all dose-response data for a study

```
SELECT *
FROM
    chemical INNER JOIN study ON study.chemical_id=chemical.chemical_id
        INNER JOIN tg ON tg.study_id=study.study_id
            INNER JOIN dose ON dose.study_id=study.study_id
                INNER JOIN dtg ON dtg.tg_id=tg.tg_id AND dose.dose_id=dtg.dose_id
                    INNER JOIN tg_effect ON tg.tg_id=tg_effect.tg_id
                        INNER JOIN effect ON effect.effect_id=tg_effect.effect_id
                            INNER JOIN endpoint ON
endpoint.endpoint_id=effect.endpoint_id
                                INNER JOIN dtg_effect ON
tg_effect.tg_effect_id=dtg_effect.tg_effect_id AND dtg.dtg_id=dtg_effect.dtg_id
WHERE study.study_id=687;
```

2.3 Programmatic access

You are not limited to only directly querying the database to access ToxRefDB. You can also programmatically access the data with a number of languages. Below are examples of accessingthe data into datasets for further work in Python and R. You will still have to connect to the database through the language specific connector.

2.3.1 Python

In the example below, the python packages sqlalchemy, pandas, and pymysql are required. You can, however, use any type of connector. Any query can replace the one provided in this example.

```
import sqlalchemy as sa
import pandas as pd

username = "<username>"
password = "<password>"
host = "<host>"
database = "<database>"

engine =
sa.create_engine(f"""mysql+pymysql://{username}:{password}@{host}/{database}""")
writer = pd.ExcelWriter("guideline_profiles.xlsx")

results = pd.read_sql("""
SELECT
    guideline.guideline_id,
    guideline.guideline_number,
    guideline.name,
    guideline.profile_name,
    guideline.description,
    guideline_profile.guideline_profile_id,
    guideline_profile.obs_status,
    guideline_profile.description,
    endpoint.endpoint_id,
    endpoint.endpoint_category,
    endpoint.endpoint_type,
    endpoint.endpoint_target
FROM
    guideline INNER JOIN guideline_profile ON
guideline.guideline_id=guideline_profile.guideline_id
        INNER JOIN endpoint ON endpoint.endpoint_id=guideline_profile.endpoint_id
""",engine)

results.to_excel(writer,index=False,merge_cells=False)
writer.save()
```

2.3.2 R

In the example below, the R package RMySQL is required. Any query can replace the one provided in this example.

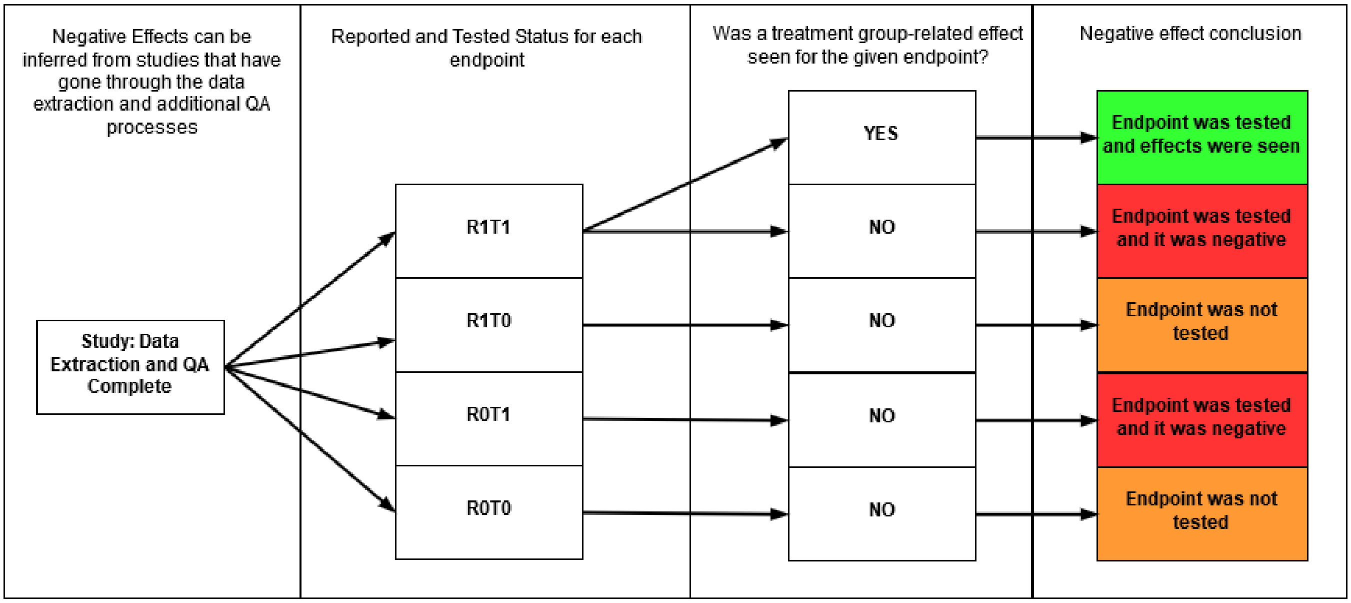
```
library(RMySQL)

con <- dbConnect(drv = RMySQL::MySQL(), user="<username>",
                 password = "<password>",
                 host = "<host>", database =”<database>”)

ToxRefv2 <- dbGetQuery(con,
"SELECT
          chemical.casrn,
          chemical.preferred_name,
          study.study_id,
          study.study_type,
          study.study_year,
          study.study_source,
          study.species,
          study.strain_group,
          study.admin_route,
          study.admin_method,
          endpoint.endpoint_category,
          endpoint.endpoint_type,
          endpoint.endpoint_target,
          endpoint.endpoint_id,
          tg_effect.life_stage,
          tg_effect.tg_effect_id,
          effect.effect_id,
          effect.effect_desc,
          tg.sex,
          tg.generation,
          dose.dose_level,
          dtg.dose_adjusted,
          dtg.dose_adjusted_unit,
          dtg_effect.treatment_related,
          dtg_effect.critical_effect,
          tested_status,
          reported_status
  FROM
          chemical INNER JOIN study ON chemical.chemical_id=study.chemical_id
          LEFT JOIN dose ON dose.study_id=study.study_id
          LEFT JOIN tg ON tg.study_id=study.study_id
          LEFT JOIN dtg ON tg.tg_id=dtg.tg_id AND dose.dose_id=dtg.dose_id
          LEFT JOIN tg_effect ON tg.tg_id=tg_effect.tg_id
          LEFT JOIN dtg_effect ON tg_effect.tg_effect_id=dtg_effect.tg_effect_id AND
dtg.dtg_id=dtg_effect.dtg_id
          LEFT JOIN effect ON effect.effect_id=tg_effect.effect_id
          LEFT JOIN endpoint ON endpoint.endpoint_id=effect.endpoint_id
          LEFT JOIN obs ON obs.study_id=study.study_id AND
obs.endpoint_id=endpoint.endpoint_id
 WHERE
study_type='SUB' ")
```

3 Negative endpoints and effects

As part of the latest update to ToxRefDB, negative endpoints and effects can be inferred from guideline profiles and the testing and reporting statuses of endpoints. Information on the data collected to make this inference can be found in the recent publication (link tbd). This section will focus on how to access the current inferred negatives and how to calculate inferences for a specific subset.

The MySQL database has inferred study-level negative effects and negative endpoints available in two tables: “negative_effect” and “negative_endpoint”. These tables were created from stored procedures (repopulate_negative_effect and repopulate_negative_endpoint) that are also available with the full MySQL database. The logic for the stored procedures follows the inference workflow seen in Figure 1. In these tables, an effect is negative if the study has gone through the data extraction process, the effect was tested (regardless of being reported), and no effect was seen in the study. An endpoint is negative for a study if all effects for that endpoint are also negative in the study.

Figure 1: Decision tree for identification of negative endpoints and effects.
Negative endpoints and effects can only be identified in studies that have gone through data extraction and any subsequent QA processes because this ensures confidence in decisions made about the adherence and/or deviations from the corresponding guideline profiles. We can infer negatives based on whether or not an endpoint was tested and no treatment group-related effects were seen.


Negative Effects can be inferred from studies that have gone through the data extraction and additional QA processes
Reported and Tested Status for each endpoint
Was a treatment group-related effect seen for the given endpoint?
Negative effect conclusion
Study: Data Extraction and QA Complete
R1T1
R1T0
R0T1
R0T0
YES
NO
NO
NO
NO
Endpoint was tested and effects were seen
Endpoint was tested and it was negative
Endpoint was not tested
Endpoint was tested and it was negative
Endpoint was not tested

There are other scenarios in which a user may want to identify negatives that do not fall within the available criteria. For example, identifying chemicals that are negative for a cancer endpoint. Below is a Python example to find chemicals that have no cancer-related effects for thyroid.

```
# get all tested results (tested_status=1) from mouse and rat and CHR and SUB
# studies that have been gone through the extraction workflow (processed=1)
tested_results = pd.read_sql(f"""SELECT
    DISTINCT chemical.casrn,study.study_id,study.species,study.study_type,
    endpoint.*,effect.effect_id,effect.effect_desc
FROM
  study INNER JOIN chemical ON chemical.chemical_id=study.chemical_id
    INNER JOIN (SELECT
                DISTINCT obs.study_id,obs.endpoint_id
                FROM obs WHERE study_id IN(
                  SELECT study_id FROM study
                  WHERE study_type IN("CHR","SUB")
                     AND species IN("mouse","rat"))
                     AND tested_status=1) AS tbl1 ON study.study_id=tbl1.study_id
    INNER JOIN endpoint ON endpoint.endpoint_id=tbl1.endpoint_id
      INNER JOIN effect ON effect.endpoint_id=endpoint.endpoint_id AND
tbl1.endpoint_id=effect.endpoint_id
        AND cancer_related=1
        AND processed=1 AND endpoint.endpoint_id IN(140,269)
        AND study_type IN("CHR","SUB") AND species IN("mouse","rat")""",engine)

# get all positive results (treatment_related=1) from mouse and rat and CHR and
SUB studies
positive_results = pd.read_sql("""SELECT DISTINCT chemical.casrn,
           study.study_id,study.species,study.study_type,
           endpoint.*,effect.effect_id,effect.effect_desc
FROM
    dose INNER JOIN dtg ON dose.dose_id=dtg.dose_id
      INNER JOIN tg ON tg.tg_id=dtg.tg_id
        INNER JOIN tg_effect ON tg.tg_id=tg_effect.tg_id
          INNER JOIN effect ON effect.effect_id=tg_effect.effect_id
            INNER JOIN dtg_effect ON dtg_effect.dtg_id=dtg.dtg_id AND
            dtg_effect.tg_effect_id=tg_effect.tg_effect_id
              INNER JOIN study ON study.study_id=tg.study_id AND
              dose.study_id=study.study_id
                INNER JOIN chemical ON chemical.chemical_id=study.chemical_id
                  INNER JOIN endpoint ON endpoint.endpoint_id=effect.endpoint_id
WHERE treatment_related=1 AND
study_type IN("CHR","SUB") AND species IN("mouse","rat")""", engine)

# find rows that are in tested_results but not in positive_results
negative_chemicals =
tested_results[~tested_results.isin(positive_results)].dropna().casrn.unique()
```

4 Calculating PODs and effect levels

4.1 Effect profiles

Effect profiles are specific groupings of effects to calculate PODs and effect levels. Currently, there are two effect profiles available in the database. Effect profile 1 groups effects based on study type, endpoint category, and life stage, while effect profile 2 generally groups effects by endpoint target (i.e. organ-level groupings). These effect profiles are further described in the recent publication (link tbd). This section will walk through creating a new effect profile, effect profile 3, and how to calculate PODs and effect levels for the new effect profile.

4.1.1 Creating a new effect profile

The best way to create a new effect profile is to develop standard criteria so the effect groups can be automatically generated using a query. A more manual approach can also be taken by manually assigning group numbers to specific effects. However, this leaves a lot of room for error if you are managing a lot of effect groups. This section will review the example for a cancer-related effect profile that is available in the full MySQL database as effect profile 3.

- Create the effect profile in the table "effect_profile"

```
INSERT INTO effect_profile (effect_profile_name,effect_profile_description)
VALUES("cancer-related endpoints","Endpoints are grouped according to
endpoint_category, but only with cancer-related effects")
```

- Create the effect groups and insert them into "effect_profile_group"

```
INSERT INTO effect_profile_group (group_id,group_name,effect_profile_id)
SELECT group_id, effect_desc AS group_name, 3 AS effect_profile_id FROM
(SELECT endpoint_id, (@group_id:=@group_id + 1) AS group_id
FROM
(SELECT DISTINCT endpoint_id
FROM effect WHERE cancer_related=1) AS tmp1, (SELECT @group_id:=0) AS tmp2)
AS tmp3
    INNER JOIN endpoint ON tmp3.endpoint_id=endpoint.endpoint_id
        INNER JOIN effect ON effect.endpoint_id=endpoint.endpoint_id AND
tmp3.endpoint_id=effect.endpoint_id
WHERE cancer_related=1;
```

- Assign ToxRefDB data to groups in table "effect_profile_group_toxrefdb"

```
INSERT INTO effect_profile_group_toxrefdb
(group_id,effect_profile_id,toxrefdb_id,toxrefdb_table)
SELECT group_id, 3 AS effect_profile_id, effect_id AS toxrefdb_id, "effect"
AS toxrefdb_table FROM
(SELECT endpoint_id, (@group_id:=@group_id + 1) AS group_id
FROM
(SELECT DISTINCT endpoint_id
FROM effect WHERE cancer_related=1) AS tmp1, (SELECT @group_id:=0) AS tmp2)
AS tmp3
    INNER JOIN endpoint ON tmp3.endpoint_id=endpoint.endpoint_id
        INNER JOIN effect ON effect.endpoint_id=endpoint.endpoint_id AND
tmp3.endpoint_id=effect.endpoint_id
WHERE cancer_related=1;
```

4.2 Calculating PODs and effect levels from effect profiles

With a new effect profile, PODs and effect levels can be calculated according to that effect profile. The script to do this is available as a Python notebook on the github repo Comptox-ToxRefDB. The two parameters to consider when running the script are the effect_profile_id and the pod level (study-level or chemical-level). Both parameters are described as comments in the script.

4.3 Bench Mark Dose (BMD) modeling

BMD modeling results are now available when possible. Data were batch-processed through BMDS (Pham *et al*., in prep) following data preparation steps as briefly described here. ToxRefDB was filtered to include only studies in which the data usability was labeled “acceptable,” meaning that sufficient study design parameter information was available, data entry level included “all effects,” and data entry status was “complete.” An additional filter was applied to only include studies with more than 2 treatment doses. For each effect in the filtered dataset, the dose-response data were extracted and grouped according to response type (continuous, continuous organ/body weight, dichotomous, or dichotomous cancer). The response type guided the type of models and BMRs used. A 10% and 5% BMR was used for dichotomous dataset, a 10 % relative deviation was used for all body weight and organ weight response type, and 1 standard deviation from the mean was used for all other continuous response. These data are stored in the “bmd_models” table. For a full description of this work, please see Pham *et al*. (in prep) for more information on the BMD pipeline for ToxRefDB and Watford *et al*. (in prep) for more information on the modeled data in ToxRefDB version 2.0.

5 Ongoing work

Extraction of quantitative data will continue for developmental, multigenerational reproductive, subacute, acute, and other study types, with updates to the ToxRefDB release. ToxRefDB information will be integrated into the CompTox Chemicals Dashboard and available in subsequent releases.

6 Data dictionary

The data dictionary is also available as a spreadsheet on the FTP site. The entity-relationship diagram (ERD) is also available on the FTP site for more information on other constraints and understanding how all the tables are connected.

ToxRefDB table name	ToxRefDB column name	description
bmd_continuous_input	id	A concatination of study it, endpoint id, tg_effect id, and sex
	endpoint_id	FK: A unique numeric identifier for each endpoint_id in the table.
	study_id	FK: A unique numeric identifier for each study in the database.
	tg_effect_id	FK: A unique numeric identifier for each treatment group-effect combination in the table.
	data_type	continuous_BW: Body weight or organ weight effect endpoint; continuous_notBW: Non body weight or organ weight effect endpoint

	doses	The doses used in the study
	means	The mean response at each dose group
	ns	The total observation at each dose group
	stdevs	The standard deviation of response at each dose group
bmd_dichotomous_input	id	A concatination of study it, endpoint id, tg_effect id, and sex
	endpoint_id	FK: A unique numeric identifier for each endpoint_id in the table.
	study_id	FK: A unique numeric identifier for each study in the database (auto-increment primary key).
	tg_effect_id	FK: A unique numeric identifier for each treatment group-effect combination in the table.
	data_type	cancer: Cancer endpoint ; dichotomous: Non-cancer endpoint
	doses	The doses used in the study
	incidences	The total positive observation made at each dose group
	ns	The total observation at each dose group
bmd_models	dataset_id	A concatination of study it, endpoint id, tg_effect id, and sex
	endpoint_id	FK: A unique numeric identifier for each endpoint_id in the table.
	study_id	FK: A unique numeric identifier for each study in the database.
	tg_effect_id	FK: A unique numeric identifier for each treatment group-effect combination in the table.
	AIC	Akaike's Information Criterion
	BMD	Benchmark Dose
	BMDL	Benchmark Dose Lower Confidence Limit(95%)
	BMDU	Benchmark Dose Upper Confidence Limit (95%)
	bmr	Benchmark response setting used
	bmr_type	Type of BMR used (For dichotomous data, Extra risk was used)
	Chi2	Chi2 value
	CSF	Cancer slow factors (dichotomous-cancer only)
	df	degrees of freedom
	doses_dropped	Number of dose dropped to get a viable model
	has_output	A boolean to determine if an .out file was created
	logic_bin	0: No serious warnings; 1: Serious warnings ; 2: Unusable
	logic_cautions	Cautions to consider with model
	logic_failures	Failures with the model
	logic_warnings	Warnings with the model

	model_name	Name of the model (are dataset-type specific)
	model_version	Version of the executed model
	pvalue1	For (continuous), pvalue 1
	pvalue2	For (continuous), pvalue 2
	pvalue3	For (continuous), pvalue 3
	pvalue4	For (continuous), pvalue 4; pvalue for dichtomous
	recommended	A recommended best-fitting model, using guidance from Wignall et al. 2014
	recommended_variable	If the model is recommend, the basis for the recommendation
	residual_of_interest	The residual closest to the estimated BMD
	warnings	An array of textual warnings in the output file
chemical	chemical_id	PK: Autoincemented unique identifer for a chemical
	dsstox_substance_id	Unique identifier from DSSTox
	casrn	The casrn of a chemical
	preferred_name	The name of a chemical
dose	dose_id	PK: Autoincemented unique identifer for a dose
	study_id	FK: A unique numeric identifier for each study in the database.
	conc	Concentration of a test chemical, typically reported in ppm within the exposure matrix (e.g., feed or water).
	conc_unit	Unit associated with a concentration of a test chemical, typically reported as ppm (almost 29,000 rows in this table as of now). Other units reported include: % (18), mg/kg/day (10), mg/L (3), mg/mL (20), mg/m^3 (4), mL/kg/day (4), None (4), not reported (13).
	dose_comment	NULL if no additional comment needed; explains any differences in dosing over the dosing interval and/or clarifying comments on how the dose was administered.
	dose_level	Numeric rank indicating the level of dose administered to test animals, with lower dose levels indicating lower concentrations of a chemical (e.g., 0 = vehicle, 1 = lowest dose, etc.). The dose level for some studies may be staggered since concentrations may vary by sex (e.g, male treatment group: 0 = vehicle, 1 = lowest dose, 3 = second lowest dose, etc.).
	vehicle	Deprecated. The media used in administration of chemical
dtg	dtg_id	PK: Autoincemented unique identifer for a dosed-treatment group

	dose_id	FK: A unique numeric identifier for each dose in the database.
	tg_id	FK: A unique numeric identifier for each treatment group in the database.
	dose_adjusted	The amount of the chemical administered in mg/kg of body weight/day (mg/kg/day). This value is typically different between male and female groups receiving the same dose concentration (conc) due to differences in bodyweight. If dose_adjusted values were not provided in a study, then they were calculated using species scaling factors (FAO/WHO, 2000).
	dose_adjusted_unit	Unit associated with the adjusted dose of a chemical, typically reported in mg/kg/day.
	dtg_comment	NULL if no additional comment needed; explains any difference in the dose-treatment-group over the course of the study (i.e., interim sacrifice or changes due to toxicity and/or morbidity); quality assurance (QA) flags indicate discrepancies between the reported and correct values for the study; differences in any dose_adjusted calculations are provided.
	mg_kg_day_value	The mg/kg/day species-specific, converted value from ppm concentration
dtg_effect	dtg_effect_id	PK: Autoincemented unique identifer for a dosed-treatment group effect
	dtg_id	FK: A unique numeric identifier for each dosed treatment group in the database.
	tg_effect_id	FK: A unique numeric identifier for each treatment group effect in the database.
	critical_effect	Binary description of a dose level for an effect by dose-treatment-group-effect (dtg_effect_id). 1 indicates that the dtg_effect_id corresponds to an effect that occurred at the lowest observable adverse effect level (LOAEL) in the study. The lowest dose at which the critical effect was observed is the lowest effect level (LOAEL),
	dtg_effect_comment	NULL if no additional comment needed; provides additional explanation of the dose-treatment-group-effect row in the table, including statistical significance.
	effect_val	Numeric value of a measured effect, can be continuous or dichotomous (incidence) data.
	effect_val_unit	Unit associated with the effect value.

	effect_var	Measurement of the variance for a set of data associated with a measured effect, generally reported as the standard deviation (SD) or standard error (SE).
	effect_var_type	Name of the variance metric used to determine the effect variance, typically the standard deviation (SD) or standard error (SE). Other effect_var types include: interquartile range, 95% confidence limit, and none.
	sample_size	Number of animals used for an examination for a particular effect.
	time	Numeric value associated with the duration of the exposure at which a particular effect was measured or observed, typically reported in hours, days, weeks, or months.
	treatment_related	Binary description of a dose level for an effect by dose-treatment-group-effect (dtg_effect_id); 1 indicates there was a difference from the control group for the effect and 0 indicates there was no difference from control. The highest dose at which no significant observable adverse effect level was observed corresponds to the no effect level(NEL), and this dose level as well as higher doses above this NEL are identified as treatment related effects in the database.
effect	effect_id	PK: Autoincemented unique identifer for an effect
	endpoint_id	FK: A unique numeric identifier for each endpoint in the database.
	effect_desc	More specific description for an effect than endpoint_category, usually detailing a specific condition associated with an endpoint_target (e.g. dysplasia, atrophy, necrosis, etc.).
effect_profile	effect_profile_id	PK: Autoincemented unique identifer for an effect profile
	effect_profile_description	Description of the effect profile
	effect_profile_name	Name of the effect profile
effect_profile_group	effect_profile_id	FK: A unique numeric identifier for each effect profile in the database.
	group_id	Unique idenfier for a group
	group_description	The description of a group
	group_name	The name of a group
endpoint	endpoint_id	PK: Autoincemented unique identifer for an endpoint

	endpoint_category	The broadest descriptive term for an endpoint. Possible endpoint categories include: systemic, developmental, reproductive, and cholinesterase.
	endpoint_target	Describes more specific information than endpoint_type, indicating where/how the sample was collected to supply data for a particular endpoint. Typically describes an organ/tissue or metabolite/protein measured.
	endpoint_type	The subcategory for endpoint_category, which is more descriptive for a particular endpoint (e.g. pathology gross, clinical chemistry, reproductive performance, etc.)
guideline	guideline_id	PK: Autoincemented unique identifer for a guideline
	description	Information pertinent to a studyâ€™s guideline. For example, MGR studies conducted post-1998 required the testing of developmental landmarks, which is notable for observation status.
	guideline_number	Number associated with the particular Office of Chemical Safety and Pollution Prevention (OCSPP) guideline that a study adheres to or most closely adheres to. Guideline numbers are differentiated by the distinct number proceeding 870, as dictated by the
	name	Name of the particular Office of Chemical Safety and Pollution Prevention (OCSPP) guideline that a study adheres to or most closely adheres to.
	profile_name	Abbreviated name of the particular Office of Chemical Safety and Pollution Prevention (OCSPP) guideline that a study adheres to or most closely adheres to. See abbreviations section for profile_name list.
guideline_profile	guideline_profile_id	PK: Autoincemented unique identifer for a guideline profile
	endpoint_id	FK: A unique numeric identifier for each endpoint in the database.
	guideline_id	FK: A unique numeric identifier for each guideline in the database.
	description	Provides a description of the rationale for an endpoint observation status.
	obs_status	Indicates whether or not an endpoint is required to be tested according to the particular guideline a study adheres to. The observation status for an endpoint can be required, not required, or triggered.
ontology	ontology_id	PK: Autoincemented unique identifer for an ontology class

	description	The associated description for the identifier
	label	The associated label for the idenfier
	uid	Unique identifier from respective terminology resource
	uid_type	Type of identifier
	uri	Uniform resource identifier
ontology_toxrefdb	ontology_toxrefdb_id	PK: Autoincemented unique identifer for an ontology class associated with a concept in ToxRefDB
	ontology_id	FK: A unique numeric identifier for each ontology class in the database.
	toxrefdb_table	The associated table in ToxRef
	toxrefdb_field	The associated field from toxrefdb_table" linked to a term"
	toxrefdb_id	Primary key from associated toxrefdb_table""
pod	pod_id	PK: Autoincemented unique identifer for a point of departure or associated effect level
	chemical_id	FK: A unique numeric identifier for each chemical in the database.
	effect_profile_id	FK: A unique numeric identifier for each effect profile in the database.
	group_id	FK: A unique numeric identifier for each effect profile group in the database.
	study_id	FK: A unique numeric identifier for each study in the database.
	dose_level	Dose level at which the POD was seen
	max_dose_level	Maximum dose level tested with relation to where the POD was captured
	mg_kg_day_value	Converted mg/kg/day value
	qualifier	<, <=, >, >=, =
	pod_type	LEL, NEL, LOAEL, or NOAEL
	pod_value	Value of the POD or associated effect level
	pod_unit	Corresponding unit of the POD or associated effect level
pod_tg_effect	pod_tg_effect_id	PK: Autoincemented unique identifer for a POD associated with a treatment group effect
	pod_id	FK: A unique numeric identifier for each POD or associated effect level in the database.
	tg_effect_id	FK: A unique numeric identifier for each treatment group effect in the database.
study	study_id	PK: Autoincemented unique identifer for a study
	chemical_id	FK: A unique numeric identifier for each chemical in the database.
	guideline_id	FK: A unique numeric identifier for each guideline in the database.

	admin_method	Describes specifically how the chemicals were administered via the route (e.g. capsule, diet, gavage, topical, etc.)
	dose_end	Time during an animalâ€™s life that the administration of a test substance stopped.
	dose_end_unit	Unit of time associated with the end of the dose (dose_end).
	dose_start	Time during an animalâ€™s life that the administration of a test substance began.
	dose_start_unit	Unit of time associated with the start of the dose (dose_start).
	species	Species of the animal test subject used in a study.
	strain	Describes a group of animals at the intraspecific level; generally, a stock of animals that share a uniform morphological or physiological character, or group that is genetically uniform.
	strain_group	Descriptive category for a group of test animals that is more general than the strain.
	study_comment	Pertinent information the data extractor/curator deemed helpful to be noted about the study in general.

	study_type	ACU (acute): Dose period typically a day or less. Excludes developmental and neurological studies.; SAC (subacute): Dose period is typically 21-28 days. Excludes developmental and neurological studies.; SUB (subchronic): Dose period is typically 13 weeks, but may be as long as 6 months. Excludes developmental and neurological studies.; CHR (chronic): Dose period is typically 12, 18, or 24 months (generally any dosing lasting a year or longer). Excludes developmental and neurological studies.; DEV (developmental): Gestational (in utero) dose period. Sacrificed prior to delivery.; MGR (multigenerational reproductive): Dose period begins in adolescent F0 males and females and continues until terminal generation. At least some of the litters deliver their pups, some may be sacrificed prior to delivery.; NEU (neurological): Study contains functional observation battery or other battery of behavioral testing that occurs during or after dosing. Pathology has specific interest in the brain (i.e. regions, morphology, biochemistry, et cetera). excludes developmental studies; DNT (developmental neurotoxicity): dose period occurs anytime during development (i.e. in utero, lactational, adolescent [after weaning, before adulthood]). Study contains functional observation battery or other battery of behavioral testing that occurs during or after dosing, typically during adulthood. Pathology has specific interest in the brain (i.e. regions, morphology, biochemistry, etc.)
	study_type_guideline	Description that combines the study_type and guideline name for a study.
	substance_comment	Pertinent information regarding a substanceâ€™s origin (generally the corporation/organization that produced the substance), purity, or other notable information about the substance in general.
	substance_lot_batch	Identifier specific to the origin of a batch of the test substance used in a study.
	substance_purity	Percentage of the administered solution that is composed of the chemical to be tested after dilution.
	substance_source_name	Name of the organization/facility that provided the chemical substance for testing during the study.
tg	tg_id	PK: Autoincemented unique identifer for a treatment group
	study_id	FK: A unique numeric identifier for each study in the database.

	dose_duration	Amount of time a group is dosed. This varies within studies depending on the dose period of a particular treatment group.
	dose_duration_unit	Unit of time associated with the dose duration. Typically in days or months.
	dose_period	Period within a groupâ€™s lifetime that the animals were dosed and the sample for an endpointâ€™s data was taken (when the animals were sacrificed).
	generation	Generation a test animal belongs to. The F0 generation is the first generation mating group for MGR studies and is the default for non-reproductive studies (CHR, SUB, SAC). F1 is the second generation mating group, selected from either F1a or F1b litters. F2 is the third generation mating group, selected from either F2a or F2b litters. F1a and F1b are the first and second litter groups produced by F0 matings, F2a and F2b are the first and second litter groups produced by F1 matings, and F3a and F3b are the first and second litter groups produced by F2 matings. The fetal generation is the group produced by F0 matings in DEV studies, and are typically removed from a female via cesarean section in DEV studies.
	sex	Gender of a test animal. The gender of fetal groups is denoted as MF for both males and females.
	tg_comment	NULL if no additional comment needed; contains information that the extractor/curator found helpful in describing issues related to a treatment-group (e.g. animals dosed via capsule so concentration not reported, added recovery groups, etc.).
tg_effect	tg_effect_id	PK: Autoincemented unique identifer for a treatment group effect
	effect_id	FK: A unique numeric identifier for each effect in the database.
	tg_id	FK: A unique numeric identifier for each treatment group in the database.
	direction	Description of the net change across all doses that indicates whether the numerical data increased, decreased, or stayed the same. Also can be used to describe effects that did not have numerical data, but were still described in the study source.

	effect_comment	NULL if no additional comment needed; contains information that the extractor/curator found helpful in describing issues related to a treatment-group-effect (e.g. units not reported, effect only reported for certain treatment groups, etc.).
	effect_desc_free	Brief verbatim text from study file that was entered if the effect description differed from predetermined endpoint terminology.
	life_stage	Stage of life that a measurement was taken. CHR, SUB, and SAC studies typically only have adult for life_stage, whereas DEV and MGR studies will always be characterized by multiple life stages. The different life stages in the database include: fetal, juvenile, adult, adult-pregnancy and pregnancy.
	target_site	A more specific description than effect_target. Can describe a specific tissue within an organ, type of cell, etc.
toxrtool	toxrtool_id	PK: Autoincemented unique identifer for a toxrtool question
	criteria	The ToxRTool comprises a list of evaluation criteria to assess study reliability that are subdivided into five groups: test substance identification, test system characterization, study design description, study results documentation, and plausibility of study design and data.
	question	Question used as part of the ToxRTool evaluation criteria to assess study reliability.
	question_number	Number indicating the question as part of the ToxRTool evaluation criteria to assess study reliability.
study_toxrtool	study_toxrtool_id	PK: Autoincemented unique identifer for a ToxRTool question associated with a study
	toxrtool_id	FK: A unique numeric identifier for each ToxRTool question in the database.
	study_id	FK: A unique numeric identifier for each study in the database.
	score	The associated score for the ToxRTool question
	toxrtool_comment	The corresponding comment further describing the score