Identifying Benchmarks for Thresholdless Pollutants

John Inman

June 16, 2023 (Updated November 7, 2023)

State Water Resources Control Board (Water Board) Integrated Report staff (staff) apply water quality standards to water quality data to assess attainment of designated beneficial uses of the state's waterbodies. Staff use numeric thresholds to evaluate whether water quality standards are met. These thresholds are established by a variety organizations and programs, including the United States Environmental Protection Agency's (USEPA) Aquatic Life Benchmarks and Ecological Risk Assessments for Registered Pesticides (benchmarks). Staff have identifed thresholds for most of the pollutants for which water quality data has been collected and regularly review threshold sources for new thresholds to apply to the remaining few without identified thresholds.

This script automates the review of the benchmarks for previously unidentified thresholds that can be applied to pollutants in the CEDEN dataset. This script comprises three sections. The first section loads the relevant data and prepares it for analysis. The second section searches the thresholdless pollutants for matches to a benchmark using CAS numbers. The third and final section searches the remaining thresholdless pollutants for *possible* matches to a benchmark using chemical name synonyms.

Section 1: Loading and Transforming Data

```
library(tidyverse)
```

1. Load the benchmarks. If necessary, download the benchmarks from USEPA.

```
benchmarks <- read_csv("./input/benchmarks.csv",
                       col_types = cols(.default = "c")) %>%
              rename(benchmark_name = Pesticide) %>%
              rename(cas_number = "CAS number")
```

2. Load the thresholdless pollutants list. If necessary, and if you are Jacob, create a list of thresholdless pollutants in CalWQA using an SQL query. Otherwise, ask Jacob to create this list. Jacob used the following SQL query to create the list of thresholdless pollutants for this analysis:

```
WITH CTE AS
  (SELECT a.analyte_name,
          CASE
```

```
            WHEN b.summing_name IS NULL THEN a.analyte_name
            ELSE b.summing_name
        END AS Objective_Name,
        listagg(DISTINCT (a.data_source_type),',') Data_Source,
        listagg(DISTINCT (a.matrix_name),',') Matrix,
        count(*) AS Number_Of_Rows
   FROM calwqa2.ir_combine_data_summary a
   LEFT JOIN calwqa2.IR_RELEP_SUMMING_POLLUTANTS b ON
a.analyte_Name=b.Analyte_Name
   WHERE a.data_filter_Type ='Query'
     AND a.data_Source_type!='Habitat'
     AND a.data_source_type!='Toxicity'
   GROUP BY a.Analyte_Name,
            b.summing_Name)
SELECT c.Analyte_Name,
       c.Objective_Name,
       c.Data_Source,
       c.Matrix,
       c.number_of_rows
FROM cte c
LEFT JOIN Calwqa2.IR_RELEP_Analytes d ON c.Objective_Name=d.Analyte_Name
WHERE d.IR_RELEP_Analyte_ID IS NULL
  AND c.objective_Name not in ('Aroclor',
                               'Coliform, Total',
                               'Enterococcus',
                               'Coliform, Fecal',
                               'E. coli',
                               'ElectricalConductivity') --AND
contains(c.objective_name, 'PBDE')>0
ORDER BY NUMBER_OF_ROWS DESC
```

```
        ceden_thresholdless <- read_csv("./input/ceden-
thresholdless.csv",
                                        col_types = cols(.default =
"c")) %>%
        rename(ceden_name = ANALYTE_NAME) %>%
        select(ceden_name) %>%
        distinct() # "Hydroxycarbofuran, 3-" is duplicated
```

3. Load a complete list of CEDEN pollutants with associated CAS numbers. Download a complete list of pollutants from CEDEN.
   The list of thresholdless pollutants created by the SQL query above is stripped down to just pollutant names and lacks associated CAS numbers, which we want to use to match to benchmarks. We will regain this information by joining the CAS number from the complete list to the thresholdless list later. The complete list will also be used later when we search for updated benchmarks. Note that a complete list of pollutants with CAS numbers may be more correctly sourced from CalWQA but I did not know how to do that at the time I wrote this script.

```
        ceden_all <- read_csv("./input/ceden.csv",
                              col_types = cols(.default = "c")) %>%
```

```
  rename(ceden_name = AnalyteName) %>%
  rename(cas_number = "CASNumber")
```

4. Load a complete list of CEDEN pollutant synonyms. If necessary, create a list of pollutant synonyms using the PubChem Identifier Exchange Service. PubChem is an online database of chemical names and attributes maintained by the National Institutes of Health (NIH). The Identifier Exchange Service translates between various chemical identifiers. We are using it to create lists of synonyms that will be used to search for *potential* matches between pollutants and benchmarks in section 3. Here are the steps to create a list of pollutant synonyms:

    1. Create a simple list of pollutant names with one pollutant per line to use as identifier exchange service input. We use the complete list of pollutants (rather than the thresholdless list) because the resulting list will be used in another script to identify pollutants with existing thresholds that need to be updated.

    ```
    ceden_names <- ceden_all %>%
      select(ceden_name) %>%
      write_csv("./input/ceden-names.csv")
    ```

    2. Go to the PubChem Identifier Exchange Service.

    3. Under **Input ID List** select `Synonyms` from the Input ID List dropdown menu, click `Browse,` and navigate to the file you created in step 1.

    4. Under **Operator Type** select `Same CID.` This will search for synonyms that PubChem considers identical to the pollutant. Future staff may consider exploring other options to match isomers, congeners, and metabolites.

    5. Under **Output IDs** select `Synonyms.`

    6. Under **Output Method** select `Two column file showing each input-output correspondence.`

    7. Under **Compression** select whatever file format you are able to extract (zip is not an option though you can download and install 7zip to extract the available options) or `No compression.`

    8. Click `Submit job` and download the resulting list.

    ```
    ceden_synonyms <- read_csv("./input/ceden-synonyms.csv",
                               col_types = cols(.default = "c"))
    ```

5. Add CEDEN pollutant names themeselves to the synonyms list. The synonyms list loaded above has pollutant names in the first column repeating for each corresponding synonym in the second column. We want to add the pollutant names themselves to the second column to cover cases where pollutant names match benchmark names directly and not via a shared synonym. Alternatively, we could add an additional section that searches benchmarks names for exact matches to pollutant names to accomplish the same thing with more code.

```
ceden_identities <- ceden_synonyms %>%
  mutate(pubchem_synonym = ceden_name) %>%
```

```
  distinct()

ceden_synonyms <- ceden_synonyms %>%
  bind_rows(ceden_identities) %>%
  filter(! is.na(pubchem_synonym)) %>%
  distinct()
```

6. Load a list of benchmark synonyms. If necessary, create a list of benchark synonyms using the same procedure outline above for pollutant synonyms, including creating a simple list of benchmark names to use as PubChem Identifier Exchange Service input.

```
benchmarks_synonyms <- read_csv("./input/benchmarks-
synonyms.csv", col_types =
cols(.default = "c"))
```

7. Add benchmark names themeselves to the synonyms list for the same reason we added pollutant names themselves to their respective synonyms list.

```
benchmarks_identity <- benchmarks_synonyms %>%
  mutate(pubchem_synonym = benchmark_name) %>%
  distinct()

benchmarks_synonyms <- benchmarks_synonyms %>%
  bind_rows(benchmarks_identity) %>%
  filter(! is.na(pubchem_synonym)) %>%
  distinct()
```

8. Join the CEDEN pollutant synonyms to the benchmark synonyms, resulting in a table with `ceden_name`, `pubchem_synonym`, and `benchmark_name` columns. Note that we created two lists of pubchem synonyms—one using pollutant names and another using benchmark names—and then joined them by the synonyms. This would be overkill if we were certain that PubChem infers all possible connections between pollutants based on synonyms, which is to say if synonym C corresponded with both pollutant A and benchmark B, then PubChem would automatically include synonym B in the synonym list for pollutant A and synonym A in the synonym list for benchmark B. Since we are not certain of this we pull synonym lists for both pollutants and benchmarks.

```
synonyms <- inner_join(ceden_synonyms, benchmarks_synonyms,
                       relationship = "many-to-many")
```

Section 2: Search Benchmarks by CAS Number

1. Prepare benchmark data. Remove benchmark rows that do no have CAS numbers and remove dashes from CAS numbers so they match the CAS number format of the CEDEN pollutants.

```
benchmarks <- benchmarks %>%
  filter(cas_number != "NR") %>%
  mutate(cas_number = str_replace_all(cas_number, "-", ""))
```

2. Search thresholdless pollutants for corresponding benchmarks that match my CAS number. Join CAS numbers to thresholdless pollutants. Remove rows that do not have CAS numbers. Remove all columns other that `ceden_name` and `cas_number`. Join benchmarks to thresholdless pollutants by CAS number. Order columns with `ceden_name` first, `benchmark_name` second, then everything else.

```
thresholdless_benchmarks_by_cas <- ceden_thresholdless %>%
  inner_join(ceden_all, by = c("ceden_name")) %>%
  filter(cas_number != "0") %>%
  select(ceden_name, cas_number) %>%
  inner_join(benchmarks, by = c("cas_number")) %>%
  select(ceden_name, benchmark_name, everything())
```

3. Save resulting table. Remove the non-breaking white space characters, which are an artifact of copy/pasting the benchmarks table from a web page. This process identified 30 new benchmarks for thresholdless pollutants for the 2026 Integrated Report cycle.

```
thresholdless_benchmarks_by_cas %>%
  mutate(across(everything(), ~ gsub("\ua0", NA, .))) %>%
  write_csv("output/thresholdless_benchmarks_by_cas.csv",
            na = "")
```

Section 3: Search Remaining Benchmarks by Synonyms

Note that searching by synonym results in *potential* matches, each of which must be reviewed and verified.

1. Remove the benchmarks that matched to a thresholdless pollutant by CAS number so they do not result in additional rows that need to be reviewed here. Join the synonyms list to the benchmarks by `benchmark_name`. Remove `ceden_name` column brought in by synonyms list so it does not conflict with the `ceden_name` in the pollutants table when we join it with the benchmarks in the next step.

```
benchmarks_remaining <- benchmarks %>%
  anti_join(thresholdless_benchmarks_by_cas, by =
c("benchmark_name")) %>%
  inner_join(synonyms, by = c("benchmark_name")) %>%
  select(-ceden_name)
```

2. Search thresholdless pollutants for corresponding benchmarks that *potentially* match by synonym. Remove pollutants that matched to a benchmark by CAS number. Join the synonyms list pollutants by `ceden_name`. Remove `benchmark_name` so it does not conflict with the `benchmark_name` column in the benchmarks table when we join it with the pollutants. Join the pollutants to the benchmarks by `pubchem_synonym`. Remove the `pubchem_synonym` column so that multiple rows of the same pollutant-benchmark combination that match by different synonyms can be reduced to a single row. Remove duplicate rows (that became dupliates after removing the `pubchem_synonym` column). Order the columns with `ceden_name` first, `benchmark_name` second, then everything else.

```
thresholdless_benchmarks_by_synonym <- ceden_thresholdless %>%
  anti_join(thresholdless_benchmarks_by_cas, by =
c("ceden_name")) %>%
  inner_join(synonyms, by = c("ceden_name")) %>%
  select(-benchmark_name) %>%
  inner_join(benchmarks_remaining, by = c("pubchem_synonym"))
%>%
  select(-pubchem_synonym) %>%
  distinct() %>%
  select(ceden_name, benchmark_name, everything())
```

3. Save resulting table. Remove the non-breaking white space characters, which are an artifact of copy/pasting the benchmarks table from a web page. This process *potential* identified 11 new benchmarks for review for thresholdless pollutants for the 2026 Integrated Report cycle.

```
thresholdless_benchmarks_by_synonym %>%
  mutate(across(everything(), ~ gsub("\ua0", NA, .))) %>%
  write_csv("output/thresholdless_benchmarks_by_synonym.csv",
            na = "")
```