Exposure RMarkdown Form Documentation

Created by: Jonathan Taylor Wall

Last Updated: 10/29/2019

Contents

[Overview 2](#_Toc25156716)

[R Libraries 2](#_Toc25156717)

[Create DTXSID Forms 2](#_Toc25156718)

[Scripts 3](#_Toc25156719)

[Detection in Media by DTXSID 3](#_Toc25156720)

[Functions 3](#_Toc25156721)

[Chemical Score Assessment 5](#_Toc25156722)

[Minnesota Form Template 5](#_Toc25156723)

[Running the Scripts to Create Forms 6](#_Toc25156724)

[inputDTXSIDList.csv 6](#_Toc25156725)

[R Session 6](#_Toc25156726)

[Console Command (In Development) 6](#_Toc25156727)

# Overview

This package of files and scripts is used to produce chemical exposure forms based on DTXSID. This can be completed by running the Create DTXSID Forms script and passing a list of DTXSID values either directly or through an input CSV file.

# R Libraries

1. tidyverse
   1. Use of dplyr and stringr
2. DBI
   1. Accessing external database
3. Readxl, xlsx
   1. Read/Write .xlsx files
4. sjPlot
   1. Creating output tables
5. magrittr
   1. Use of T-operator pipe
6. jsonlite
   1. Pull and parse JSON files
7. knitr, rmarkdown
   1. RMarkdown rendering
8. data.table
   1. Creation of data table objects
9. pander
   1. Table outputs for Word Docs

# Create DTXSID Forms

* Name: create\_DTXSID\_forms.R
* Description: This is the primary script used to create the output DTXSID forms. It uses an input list of DTXSID values and creates the chemical exposure forms using the Detection in Media by DTXSID and Minnesota Form Template scripts. It can create forms for all DTXSID values or can run batches if the input has “Group” information. See “DTXSID\_List.csv” for more information.

|  |  |
| --- | --- |
| **Input** | **Description** |
| inputDTXSIDList.csv | A CSV file a user creates with grouped lsits of DTXSID values under category headings (see below). Can be passed from command line as a file path or directly hardcoded |
| chemList | A list of DTXSID values in the form of DTXSID# from the inputDTXSIDList.csv |

* Output: DTXSID Exposure Monitoring Forms
  + Word document forms are created and stored in a new output folder created in the project/session directory. If users specified multiple groupings, they will be stored in separate folders named after these groupings.

# Scripts

## Detection in Media by DTXSID

* Name: DTXSID\_media\_detected.R
* Description: A standalone script with functions to pull chemical data from the Factotum database by DTXSID

### Functions

#### getProductData

* 1. Description: Function which pulls all paged data for a single DTXSID from the factotum database using the [Factotum Web Services API](http://api.factotum.epa.gov/). The function is set to randomly rest 10 seconds between consecutive requests 10% of the time.
  2. Returns: A dataframe parsed from the paged JSON data the Factotum API returns based on the input DTXSID. A cached .CSV file to prevent unnecessary data pulls between identical runs.
  3. Note: This function is called internally by the get\_DTXSID\_Media\_Detected function, unless a user wishes to implement it directly to retrieve parsed data from the Factotum API on a single DTXSID. Certain fields are returned nested and must be further processed/flattened.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Default** |
| page\_size | Positive integer value of entries to pull at a time from the API ([see API documentation](http://api.factotum.epa.gov/) for parameter limits) | 100 |
| dtxsid | A single DTXSID string value in the form of “DTXSID#” | “” |
| filePath | A string value that represents the desired file path and name for saving a CSV copy of the pulled data | NULL |
| overwrite | A logical switch to determine if data should be pulled if a file with the same filePath exists. Set to FALSE is you wish to pull from cached data. | FALSE |
| Use Example | data <- getProductData(page\_size=500, dtxsid=DTXSID#, filePath = "input/cache/productDat.xlsx", overwrite = TRUE) | |

**Note:** If a file with the same filepath exists, new data will not be pulled unless the overwrite parameter is passed a TRUE. Instead, that file will be returned.

#### getTrueChem

* 1. Description: Function which pulls all paged data of verified CASRN and chemical names for every DTXSID within the factotum database using the [Factotum Web Services API](http://api.factotum.epa.gov/). The function is set to rest after each pull so that no more than 4 pulls happen per second.
  2. Returns: A dataframe parsed from the paged JSON data the Factotum API returns based on the input DTXSID. The user may also save a CSV copy of the data if they pass a filepath.
  3. Note: This function is called internally by the get\_DTXSID\_Media\_Detected function, unless a user wishes to implement it directly to retrieve parsed data from the Factotum API on a single DTXSID. Certain fields are returned nested and must be further processed/flattened.

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Default** |
| page\_size | Positive integer value of entries to pull at a time from the API ([see API documentation](http://api.factotum.epa.gov/) for parameter limits) | 100 |
| filePath | A string value that represents the desired file path and name for saving a CSV copy of the pulled data | NULL |
| overwrite | A logical switch to determine if data should be pulled if a file with the same filePath exists. Set to FALSE is you wish to pull from cached data. | FALSE |
| Use Example | data <- getTrueChem(page\_size=500, dtxsid="DTXSID1020560") | |

**Note:** If a file with the same filepath exists, new data will not be pulled unless the overwrite parameter is passed a TRUE. Instead, that file will be returned.

#### get\_DTXSID\_Media\_Detected

* 1. Description: Pulls chemical data by DTXSID from the factotum API
  2. Returns: A list of three elements
     1. **tbl1**: A dataframe of factotum data by DTXSID that is reformatted using the RMarkdown script
     2. **missingDTX**: A list of DTXSID’s that were not found within the factotum database and are thus missing from the overall RMarkdown form output
     3. **noPucDTX**: A list of DTCSID’s that have not been assigned a PUC in factotum and will be missing this data in the overall RMarkdown form output
  3. Use: This function is called internally by the RMarkdown, unless a user wishes to implement it separately to just pull the DTXSID data into a dataframe

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Default** |
| d | A vector or list of DTXSID values in the form of “DTXSID#” | “” |
| page\_size | Positive integer value of entries to pull at a time from the API ([see API documentation](http://api.factotum.epa.gov/) for parameter limits) | 100 |
| Use Example | data <- get\_DTXSID\_Media\_Detected(d=list("DTXSID1020560","DTXSID4020458"),  page\_size=500) | |

#### writePagedXLSX

* 1. Description: Writes a list of input data frames to a .xlsx file with labeled sheets for each data frame
  2. Returns: Creates a .xlsx file with the specified name and file path
  3. Use: Used within the getProductData function to cache pulled data between runs

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Value** | **Default** |
| data | A list of data frames to be written to the .xlsx sheet | list() |
| sheetNames | A list of names to label each .xlsx sheet | list() |
| filePath | A string file path/name of where to store the output .xlsx file | NULL |
| Use Example | writePagedXLSX(data=list(productDat, missingDTX$DTXSID, noPucDTX$DTXSID),  sheetNames=list("data", "missingDTX", "noPucDTX"), filePath=“input/cache/data”) | |

## Chemical Score Assessment

* Name: scoreExposureTable.R
* Description: Calculates exposure scores based on a scoresheet which recodes field values to a scale. See the accompanying “*Exposure Worksheet Scoring Documentation*” documentation for more details
* Note: This script can also be run separately either in an R session or from the console.

## Minnesota Form Template

* Name: Minnesota\_Form\_Template.Rmd
* Description: This is the main RMarkdown script which formats the input data from the Detection in Media by DTXSID script and other data inputs into the desired output DTXSID Exposure Monitoring forms. It sets up templated tables with specified rows from tableRows.xlsx, which are then populated by the corresponding datapoints, before the final table is printed using pandocs. Once finished, it outputs a single worksheet for a single DTXSID.

|  |  |
| --- | --- |
| **Input** | **Description** |
| params$data | A list of data tables passed from Create DTXSID Forms which provides data pulled from Factotum and the MMDB database |
| params$dtxsid | A DTXSID value passed from Create DTXSID Forms which determines which DTXSID data to filter input data by to create the final worksheet output |
| Minnesota-Form-styles.docx | A stylesheet template to help format the output Word document. It specifies how headers, tables, and other features should look using custom Word Styles |
| tableRows.xlsx | A file which specifies the rows of each of the Rmarkdown tables, specified in separate sheets. These rows are parsed into tables and paired with datapoints by the Rmarkdown script |
| mediaList.csv | Categorized list of unique harmonized media arbitrarily grouped together (E.g. {“sediment”, “soil”, “indoor dust”} = “soil”) |
| batchOverall.csv | An output of Detection in Media by DTXSID with mapped DTXSID, CASRN, and chemical name values used to create tables for DTXSID values missing from factotum or have unassigned PUCs. |

# Running the Scripts to Create Forms

## inputDTXSIDList.csv

|  |  |
| --- | --- |
| **Variable** | **Definition/Use** |
| DTXSID | A list of DTXSID values in the form of “DTXSID#” |
| Group | A string value to group/categorize DTXSID values together. This creates a separate output subfolder per group entry by this group variable value |

* **Note:** The “Group” field can be left blank. They will be assigned to an “Ungrouped” output folder

## R Session

* Open “create\_DTXSID\_forms.R”
* Set your file directory to the appropriate location to access the files (setwd(“pathToFiles”))
* Ensure the file name for your DTXSID list is correct for the initial args[1] variable as it is imported into the session (i.e. args[1] = "input/inputDTXSIDList3.csv")

## Console Command (In Development)

* Open your command line either for your computer or another platform (e.g. Anaconda)
* Set your file directory to the appropriate location to access the files: $ cd /pathToFiles
* Enter the following command: $ Rscript create\_DTXSID\_forms.R DTXSIDList.xlsx
  + Do not pass your file name/path to the DTXSID List with quotations