STATISTICAL ECO(-TOXICO)LOGY

IMPROVING THE UTILIZATION OF DATA FOR ECOLOGICAL RISK ASSESSMENT

by

EDUARD SZÖCS
from ZĂRNEŞTI / ROMANIA

Submitted Dissertation thesis for the partial fulfillment of the requirements for a Doctor of Natural Sciences
Fachbereich 7: Natur- und Umweltwissenschaften
Universität Koblenz-Landau

11. November 2016

1 WEBCHEM: AN R PACKAGE TO RETRIEVE CHEMICAL INFORMATION FROM THE WEB

Eduard Szöcs[a] & Ralf B. Schäfer[a]

[a]Institute for Environmental Sciences, University Koblenz-Landau, Landau, Germany

Accepted in *Journal of Statistical Software*, 2016.

ABSTRACT

A wide range of chemical information is freely available online, including identifiers, experimental and predicted chemical properties. However, these data are scattered over various data sources and not easily accessible to researchers. Manual searching and downloading of such data is time-consuming and error-prone. We developed the open-source R package webchem that allows users to automatically query chemical data from currently 11 web sources. These cover a broad spectrum of information. The data are automatically imported into an R object and can directly be used in subsequent analyses. webchem enables easy, structured and reproducible data retrieval and usage from publicly available web sources. In addition, it facilitates data cleaning, identification and reporting of substances. Consequently, it reduces the time researchers need to spend on chemical data compilation.

INTRODUCTION

Before each statistical analysis, data cleaning is often required to ensure good data quality. Data cleaning is the process of detecting errors and inconsistencies in data sets (Chapman, 2005). In practice, the data cleaning step is often more time consuming than the subsequent statistical analysis, particularly, when the analysis relies on the joining of multiple data sources.

When dealing with chemical data sets (e.g. environmental monitoring data, toxicological data), a first step is often to validate the names of chemicals or to link them to unique codes that simplify subsequent querying and appending of compound-related physico-chemical or toxicological information. Several web sources provide chemical names or link them to unique codes (see also section *Data sources* below). However, manual searching for each compound, often through a graphical web interface, is tedious, error-prone and not reproducible (Peng, 2009).

To simplify, robustify and automate this task, i.e. to search and retrieve chemical information from the web, we created the webchem package for the free and open source R language (R Core Team, 2016; Wehrens, 2011). R is one of the most widely used software environments for data cleaning, analysing and visualising data, and supports full reproducibility of each step (Marwick, 2016).

In the following, we describe the basic functionality of the package and demonstrate with a few use cases how to clean and retrieve new data with webchem.

IMPLEMENTATION AND DESIGN DETAILS

The webchem package is written entirely in R and available under a MIT license. The development repository is hosted on GitHub, (2016) and a stable version is released on the official R repository (CRAN, 2016). webchem is part of the rOpenSci project (Boettiger et al., 2015), which aims at fully reproducible data analysis.

webchem follows best practices for scientific software (Poisot, 2015; Wilson et al., 2014), namely: (i) a public available repository with easy collaboration and an issue tracker (via GitHub), (ii) a non-restrictive license, version control (git), (iii) an elaborate test-suite covering more than 90% of the relevant lines of code (currently approximately 1500 lines, using testthat (Wickham, 2011)), (iv) continuous integration (via Travis-CI, (2016) and AppVeyor, (2016); testing on Linux & Windows with current and development R versions), (v) in-source documentation (using roxygen2 (Wickham et al., 2015)) and (vi) compliance with a style guide (Wickham, 2015a).

webchem builds on top of the following R packages: RCurl (Lang and Team, 2016) and httr (Wickham, 2016) for data transfer, stringr (Wickham, 2015c) for string handling, xml2 (Wickham, 2015d) and rvest (Wickham, 2015b) for parsing HTML and XML, jsonlite (Ooms, 2014) for parsing JSON, rcdk (Guha, 2007) for parsing SMILES. For parsing molfiles we use a lightweight implementation of (Grabner et al., 2012).

Some data sources provide application programming interfaces (API). Web APIs define functions that allow accessing services and data via http and return data in a specific way. webchem uses the API of a data source provider, where available. For sources where an API is lacking, data is directly searched and extracted from the web pages, analogous to manual interaction with a website.

Only few design decisions have been made: Each function name has a prefix and suffix separated by an underscore (Chamberlain and Szöcs, 2013). They follow the format of source_function, e.g. cs_compinfo uses ChemSpider as source (see next section) to retrieve compound information. Some functions require querying first a unique identifier from the data source and then use this identifier to query further information. The prefix get is used to denote these functions, e.g. get_csid to retrieve the identifier used in ChemSpider.

webchem is friendly to the resources of data providers. Between each request there is a time-out of 0.3 to 2 seconds depending on the data source. Therefore, processing of larger data sets can take some time, but still represents a major improvement compared to manual lookup. We provide a link to the *Terms of Use* of data providers in the documentation of each function and we encourage the users to read these before using webchem. Moreover, all functions return an URL of the source, which can be used for (micro-)attribution.

DATA SOURCES

The backbone of webchem are data sources providing their data and functionality to the public. Currently, data can be retrieved from 11 sources. These cover a broad spectrum of available data, like identifiers, experimental and predicted properties and regulatory information (Figure 1.1, a detailed overview of all sources is included as supplement):

NIH CHEMICAL IDENTIFIER RESOLVER (CIR) A web service that converts from and to various chemical identifiers (NIH, 2016).

CHEMICAL TRANSLATION SERVICE (CTS) A web service that converts from and to various chemical identifiers (Wohlgemuth et al., 2010).

ETOX Information System Ecotoxicology and Environmental Quality Targets by the German Federal Environmental Agency. Provides basic identifiers, synonyms, ecotoxicological data and quality targets for different countries (UBA, 2016).

PAN PESTICIDE DATABASE Information on pesticides - provides basic identifiers, ecotoxicological data and chemical properties (PAN, 2016).

SRC PHYSPROP Contains physical properties for over 41,000 chemicals. Physical properties collected from a wide variety of sources including experimental and modeled values(Howard and Meylan, 2016).

PUBCHEM PubChem is a public repository for information on chemical substances, providing identifiers, properties and synonyms (Kim et al., 2016). We use an interface to the PUG-REST web service (Kim et al., 2015).

WIKIDATA Wikipedia contains information for over 15,000 chemicals (Ertl et al., 2015; Wikipedia, 2016). Currently webchem can only query chemical identifiers.

COMPENDIUM OF PESTICIDE COMMON NAMES The compendium provides information on pesticide common names, identifiers and classification (Wood, 2016).

CHEMID*plus* is a large web-based database provided by the National Library of Medicine (NLM). It provides identifiers, synonyms, toxicological data and chemical properties (Tomasulo, 2002).

CHEMSPIDER is a free chemical structure database providing access to over 40 million structures. It provides identifiers, properties and can also be used to convert identifiers (Pence and Williams, 2010).

OPSIN The Open Parser for Systematic IUPAC nomenclature is a chemical name interpreter and provides InChI and SMILES identifiers (Lowe et al., 2011).

Though the data sources exhibit some overlap in the provided information, each has been selected because it also provides unique information and we encourage the interested reader to consult the related source for details. However, we provide a brief overview in the Supporting Information.

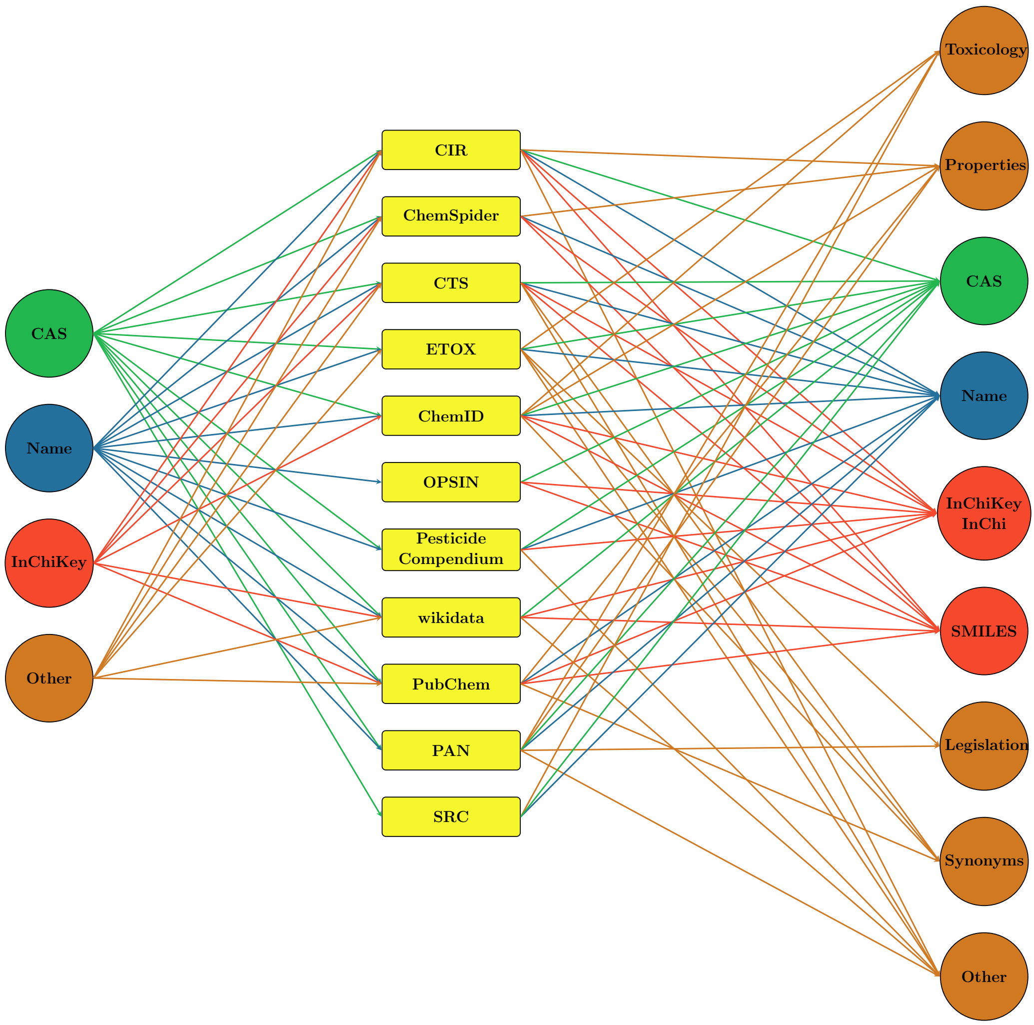

Figure 1.1: Overview of current data sources. Input and output possibilities currently implemented in the package.

USE CASES

Installation

webchem can be easily installed and loaded from CRAN:

```
R> install.packages("webchem")
R> library("webchem")
```

The package is under active development. The latest development version is available from GitHub and also permanently available at Zenodo, (2016). This document has been created using webchem version 0.1.

Sample data sets

To demonstrate the capabilities of webchem we use two small publicly available real world data sets. The data sets are only used for purpose of demonstration, have been slightly preprocessed (not shown) and are available through the package.

(i) jagst: This data set comprises environmental monitoring data of organic substances in the river Jagst, Germany, sampled in 2013. The data is publicly available and can be retrieved from LUBW, (2016). It comprises concentrations (in µg / L) of 34 substances on 13 sampling occasions. First we load the data set and inspect the first six rows:

```
R> data("jagst")
R> head(jagst)
```

```
##         date          substance value qual
## 1 2013-01-04 2,4-Dimethylphenol 0.006    <
## 2 2013-01-29 2,4-Dimethylphenol 0.006    <
## 3 2013-02-26 2,4-Dimethylphenol 0.006    <
## 4 2013-03-26 2,4-Dimethylphenol 0.006    <
## 5 2013-04-23 2,4-Dimethylphenol 0.006    <
## 6 2013-05-22 2,4-Dimethylphenol 0.006    <
```

This data set identifies substances only by substance names. Values below the limit of quantification (LOQ) are indicated by a qualifier column.

(ii) lc50: This data consists of median acute lethal concentration for the water flea *Daphnia magna* in 48 h tests ($LC_{50,D.magna,48h}$) of 124 insecticides. The data has been retrieved from the EPA ECOTOX database (U.S. EPA, 2016).

```
R> data("lc50")
R> head(lc50)
```

```
##          cas        value
## 4  50-29-3    12.415277
## 12 52-68-6     1.282980
## 15 55-38-9    12.168138
## 18 56-23-5 35000.000000
## 21 56-38-2     1.539119
## 36 57-74-9    98.400000
```

This data set identifies the substances only by CAS numbers.

Query identifiers

The jagst data set covers 34 substances that are identified by (German) names. Merging and linking these to other tables is hampered by differences and ambiguity in compound names.

One possibility to resolve this, is to use different chemical identifiers allowing easy identification. There are several identifiers available, e.g. registry numbers like CAS or EC, database identifiers like PubChemCID (Kim et al., 2016) or ChemSpiderID (Pence and Williams, 2010), line notations like SMILES (Weininger, 1990), InChI and InChiKey (Heller et al., 2015). In this first example we query several identifiers to create a table that can be used as (i) supplemental information to a research article or (ii) to facilitate subsequent matching with other data.

As we are are dealing with German substance names we start to query ETOX for CAS registry numbers. A common work flow when dealing with web resources is to 1) query a unique identifier of the source, 2) use this identifier to retrieve additional information and 3) extract the parts that are needed from the R object (Chamberlain and Szöcs, 2013).

First we search for ETOX internal ID numbers using the substance names:

```
R> subs <- unique(jagst$substance)
R> ids <- get_etoxid(subs, match = 'best')
R> head(ids)
```

```
##   etoxid                            match distance                 query
## 1   8668      2,4-Dimethylphenol ( 8668 )        0     2,4-Dimethylphenol
## 2   8494 4-Chlor-2-methylphenol ( 8494 )        0 4-Chlor-2-methylphenol
## 3   <NA>                             <NA>     <NA>     4-para-nonylphenol
## 4   8397                 Atrazin ( 8397 )        0                Atrazin
## 5   7240                  Benzol ( 7240 )        0                 Benzol
## 6   7331        Desethylatrazin ( 7331 )        0        Desethylatrazin
```

Only three substances could not be found in ETOX. Here we specify that only the *'best'* match (in terms of the Levenshtein distance between query and results) is returned. A manual check confirms appropriate matches. Other options include: *'all'* - returns all matches; *'first'* - returns only the first match (not necessarily the best match); *'ask'* - this enters an interactive mode, where the user is asked for a choice if multiple matches are found and *'na'* which returns NA in case of multiple matches.

We use these data to retrieve basic information on the substances.

```
R> etox_data <- etox_basic(ids$etoxid)
```

webchem always returns a named list (one entry for each substance) and the available information content can be very voluminous. Therefore, we provide extractor functions for the common identifiers: CAS, SMILES and InChIKeys.

```
R> etox_cas <- cas(etox_data)
R> head(etox_cas)
```

```
##        8668         8494         <NA>         8397         7240         7331
##   "105-67-9" "1570-64-5"           NA "1912-24-9"    "71-43-2" "6190-65-4"
```

A variety of data are available and we cannot provide extractor functions for each of those. Therefore, if users need to extract other data, they have to write simple extractor functions (see following examples).

In the same manner, we can now query other identifiers from another source using these CAS numbers (Figure 1.1), like PubChem

```
R> cids <- get_cid(etox_cas)
R> pc_data <- pc_prop(cids, properties = c('CanonicalSMILES'))
R> pc_smiles <- smiles(pc_data)
```

or ChemSpider

```
R> csids <- get_csid(etox_cas, token = token)
R> cs_data <- cs_compinfo(csids, token = token)
R> cs_inchikey <- inchikey(cs_data)
```

Finally, we combine the queried data into one data.frame

```
R> res <- data.frame(name = subs, cas = etox_cas, smiles = pc_smiles,
  cid = pc_data$CID, inchikey = cs_inchikey, csid = cs_data$csid,
  stringsAsFactors = FALSE)
```

Note that in order to use the ChemSpider functions, a personal authentication key (token) is needed, which can be retrieved from the ChemSpider web page. Finally, we obtain a compound table containing many different identifiers (Table 1.1), allowing easy identification and merging with other data sets, e.g. the lc50 data set based on CAS.

Name	CAS	SMILES	CID	InChIKey	CSID
2,4-Dimethylphenol	105-67-9	CC1=CC(...	7771	KUFFULV...	13839123
4-Chlor-2-methylphenol	1570-64-5	CC1=C(C...	14855	RHPUJHQ...	14165
4-para-nonylphenol	-	-	-	-	-
Atrazin	1912-24-9	CCNC1=N...	2256	MXWJVTO...	2169
Benzol	71-43-2	C1=CC=C...	241	UHOVQNZ...	236
Desethylatrazin	6190-65-4	CC(C)NC...	22563	DFWFIQK...	21157

Table 1.1: Identifiers for the jagst data sets as queried with webchem. Only the first 6 entries are shown. For SMILES and InChIKey only the first 7 characters are shown. - = not found.

Toxicity of different pesticide groups

Another question we might ask is *How does toxicity vary between insecticide groups?* Answering this question would require tedious lookup of insecticide groups for each of the 124 CAS numbers in the lc50 data set. The Compendium of

Pesticide Common Names (Wood, 2016) contains such information and can be easily queried using CAS numbers with webchem:

```
R> aw_data <- aw_query(lc50$cas, type = 'cas')
```

To extract the chemical group from the retrieved data set, we write a simple extractor function and apply this to the retrieved data:

```
R> igroup <- sapply(aw_data, function(y) y$subactivity[1])
R> igroup[1:3]
```

```
##                                  50-29-3
##             "organochlorine insecticides"
##                                  52-68-6
##                "phosphonate insecticides"
##                                  55-38-9
## "phenyl organothiophosphate insecticides"
```

Figure 1.2 displays the result after additional data cleaning (see supplement for full code). Overall, it took only 5 R statements to retrieve, clean and plot the data using ggplot2 (Wickham, 2009).

Querying partitioning coefficients

Some data sources also provide data on chemical properties that can be queried. Here we query for the lc50 data the log $P_{oct/wat}$ from the SRC PHYSPROP database to build a simple quantitative structure–activity relationship (QSAR) to predict toxicity.

```
R> pp_data <- pp_query(lc50$cas)
```

The database contains predicted and experimental values. Extracting log $P_{oct/wat}$ from the data object is slightly more complicated, because i) for some compounds no data could be found and ii) the data-object has a more complex structure (a data frame within a list).

```
R> lc50$logp <- sapply(pp_data, function(y) {
  if (length(y) == 1 && is.na(y))
```

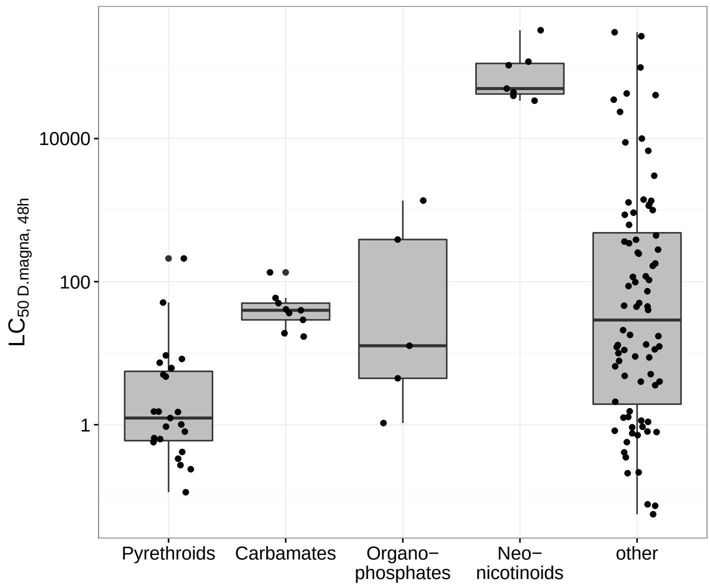

Figure 1.2: Toxicity of different pesticide groups. LC_{50} values have been retrieved from EPA ECOTOX database, chemical groups from the Compendium of Pesticide Common Names.

```
    return(NA)
  y$prop$value[y$prop$variable == 'Log P (octanol-water)']
})
```

We opted for this more complex approach, because the information available is very diverse and we cannot provide an extractor function for each purpose. Moreover, it provides users with high flexibility regarding organisation of their data. Nevertheless, in the documentation of each function we provide examples on how to extract more complicated parts of the data. The resulting data and model is displayed in Figure 1.3.

Regulatory information

Regulatory information is of particular interest if concentrations exceed national thresholds. In the European Union (EU) the Water Framework Directive (WFD, EU-WFD, (2000)) defines Environmental Quality Standards (EQS). Similarly, the U.S. and Canadian EPA and the WHO define Quality Standards. Information on

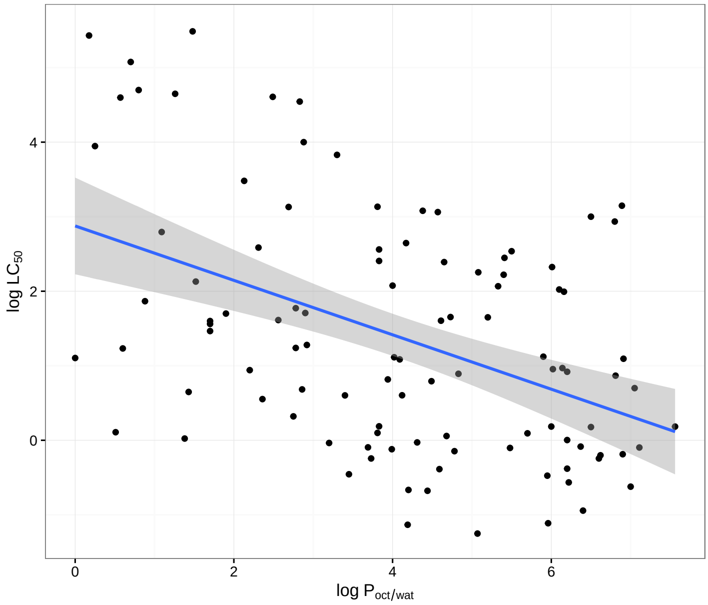

Figure 1.3: Simple QSAR for predicting log LC_{50} of pesticides by log P. Log P values have been retrieved from SRC Physprop database (97 experimental data, 9 estimated data and 18 substances without data). Blue line indicates the regression model ($\log LC_{50} = 2.88 - 0.37 \log P$, RMSE = 1.45).

these standards can be queried with webchem from the PAN Pesticide Database (using pan_query()) and from ETOX (using etox_targets()).

In this example we search for the minimum EQS for the EU for the compounds in the jagst data set, join these with measured concentrations and evaluate wether exceedances occurred..

We re-use the above queried ETOX-IDs to obtain further information from ETOX, namely the MAC-EQS:

```
R> eqs <- etox_targets(ids$etoxid)
R> ids$mac <- sapply(eqs, function(y){
  if (length(y) == 1 && is.na(y)) {
    return(NA)
  } else {
    res <- y$res
    min(res[res$Country_or_Region == 'EEC / EU' &
               res$Designation == 'MAC-EQS', 'Value_Target_LR'])
  }
```

```
})
```

Again, the returned information is humongous and we encourage users to study the returned objects and description of the data source. Here, the column Designation defines the type of EQS and Value_Target_LR contains the value. Unfortunately, we only found MAC-EQS values for 5 substances:

```
R> (mac <- with(ids, ids[!is.na(mac) & is.finite(mac),
                      c('etoxid', 'query', 'mac')]))

##    etoxid       query    mac
## 4    8397     Atrazin  2.000
## 5    7240      Benzol 50.000
## 11   8836     Irgarol  0.016
## 12   7442 Isoproturon  1.000
## 29   8756   Terbutryn  0.034
```

The get_etoxid() function used to search ETOX-IDs returns also the original substance name (query), so that we can easily join the table with MAC values with the measurements table :

```
R> jagst_eqs <- merge(jagst, mac, by.x = 'substance', by.y = 'query')
R> head(jagst_eqs)

##   substance       date  value qual etoxid mac
## 1   Atrazin 2013-09-10 0.0068    =   8397   2
## 2   Atrazin 2013-10-08 0.0072    =   8397   2
## 3   Atrazin 2013-03-26 0.0040    =   8397   2
## 4   Atrazin 2013-04-23 0.0048    =   8397   2
## 5   Atrazin 2013-11-05 0.0036    =   8397   2
## 6   Atrazin 2013-07-16 0.0052    =   8397   2
```

Finally, we can compare the measured value to the MAC, which reveals that there have been no exceedances of these 5 compounds.

Utility functions

Furthermore, webchem provides also basic functions to check identifiers that can be used for data quality assessment. The functions either use simple formatting rules,

```
R> is.inchikey('BQJCRHHNABKAKU-KBQPJGBKS-AN')

## Hyphens not at position 15 and 26.

## [1] FALSE

R> is.cas('64-17-6')

## Checksum is not correct!  5 vs.  6

## [1] FALSE
```

or web resources like ChemSpider

```
R> is.inchikey('BQJCRHHNABKAKU-KBQPJGBKSA-5',
  type = 'chemspider')

## [1] FALSE
```

DISCUSSION

Related software

Within the R ecosystem, there are only a few similar projects: rpubchem (Guha, 2014) provides an interface to PubChem. Similarly, ChemmineR (Cao et al., 2008), a mature chemo-informatics package, provides an interface to Pubchem. webchem does not provide any chemo-informatic functionality, but integrates access to many data sources. WikidataR (Keyes and Graul, 2016) provides an interface to wikidata that could be used to retrieve chemical data from Wikipedia. However, it does not provide predefined methods for chemical data like webchem. Within the Python ecosystem the libraries PubChempy (Swain, 2015b), ChemSpiPy (Swain, 2015a) and CIRpy (Swain, 2016) are available for similar tasks as those outlined here. webchem is not specialized and tries to integrate many data sources and for some of these it provides a unique programmatic interface. The Chemical Translation Service (Wohlgemuth et al., 2010), which is also one of the sources that can be queried, allows batch conversion of chemical identifiers. However, it does not provide access to other data (experimental, modeled or regulatory data).

Open Science

An increasing number of scientific data is becoming publicly available (Gewin, 2016; O'Boyle et al., 2011; Reichman et al., 2011), either in public data repositories or as supplement to publications. To be usable for other researchers chemical compounds should be properly identified, not only by chemical names but also with accompanying identifiers like InChIKey, SMILES and authority-assigned identifiers. webchem provides an easy way to create such meta tables as shown in Table 1.1 and facilitates chemical data availability to researchers. However, good quality of data is crucial for every analysis (Stieger et al., 2014) and additional effort and methods are needed to validate data quality.

Further development

We have outlined only a few use cases that will likely be useful for many researchers. Given the huge amount of publicly available information, many other possibilities can be envisioned. webchem is currently under active development and several other data sources have not been implemented yet but may be in the future. GitHub makes contributing easy and we strongly encourage contribution to the package. Moreover, comments, feedback and feature requests are highly welcome.

CONCLUSIONS

Researchers need to have easy access to global knowledge on chemicals. webchem can save *"hundreds of working hours"* gathering this knowledge (Münch and Galizia, 2016), so that researchers can focus on other tasks.

REFERENCES

AppVeyor (2016). URL: https://www.appveyor.com/.

Boettiger, C., S. Chamberlain, E. Hart, and K. Ram (2015). "Building Software, Building Community: Lessons from the ROpenSci Project". *Journal of Open Research Software* 3 (1).

Cao, Y, Charisi, A, Cheng, L. C, Jiang, T, Girke, and T (2008). "ChemmineR: A Compound Mining Framework for R". *Bioinformatics* 24 (15), 1733–1734.

Chamberlain, S. A. and E. Szöcs (2013). "taxize: Taxonomic Search and Retrieval in R". *F1000Research* 2 (191).

Chapman, A. (2005). *Principles and Methods of Data Cleaning*. Report for the Global Biodiversity Information Facility, Copenhagen. GBIF. URL: http://www.gbif.org/orc/?doc_id=1262.

CRAN (2016). *webchem: Retrieve Chemical Information from the Web*. URL: https://CRAN.R-project.org/package=webchem.

Ertl, P., L. Patiny, T. Sander, C. Rufener, and M. Zasso (2015). "Wikipedia Chemical Structure Explorer: Substructure and Similarity Searching of Molecules from Wikipedia". *Journal of Cheminformatics* 7 (1).

Gewin, V. (2016). "Data sharing: An Open Mind on Open Data". *Nature* 529 (7584), 117–119.

GitHub (2016). *webchem: Retrieve Chemical Information from the Web*. URL: https://github.com/ropensci/webchem.

Grabner, M., K. Varmuza, and M. Dehmer (2012). "RMol: A Toolset for Transforming SD/Molfile Structure Information into R Objects". *Source Code for Biology and Medicine* 7, 12.

Guha, R. (2007). "Chemical Informatics Functionality in R". *Journal of Statistical Software* 18 (5), 1–16.

Guha, R. (2014). *rpubchem: Interface to the PubChem Collection*. R package version 1.5.0.2. URL: https://CRAN.R-project.org/package=rpubchem.

Heller, S. R., A. McNaught, I. Pletnev, S. Stein, and D. Tchekhovskoi (2015). "InChI, the IUPAC International Chemical Identifier". *Journal of Cheminformatics* 7 (1).

Howard, P. H. and W. Meylan (2016). *Physical / Chemical Property Database (PHYSPROP)*. URL: http://www.srcinc.com/what-we-do/environmental/scientific-databases.html.

Keyes, O. and C. Graul (2016). *WikidataR: API Client Library for Wikidata*. R package version 1.0.1. URL: https://CRAN.R-project.org/package=WikidataR.

Kim, S., P. A. Thiessen, E. E. Bolton, and S. H. Bryant (2015). "PUG-SOAP and PUG-REST: Web Services for Programmatic Access to Chemical Information in PubChem". *Nucleic Acids Research* 43 (W1), W605–W611.

Kim, S., P. A. Thiessen, E. E. Bolton, J. Chen, G. Fu, A. Gindulyte, L. Han, J. He, S. He, B. A. Shoemaker, and et al. (2016). "PubChem Substance and Compound Databases". *Nucleic Acids Research* 44 (D1), D1202–D1213.

Lang, D. T. and t. C. Team (2016). *RCurl: General Network (HTTP/FTP/...) Client Interface for R*. R package version 1.95-4.8. URL: http://CRAN.R-project.org/package=RCurl.

Lowe, D. M., P. T. Corbett, P. Murray-Rust, and R. C. Glen (2011). "Chemical Name to Structure: OPSIN, an Open Source Solution". *Journal of Chemical Information and Modeling* 51 (3), 739–753.

LUBW - Landesanstalt für Umwelt, M. u. N. B.-W. (2016). *Jahresdatenkatalog Fließgewässer 2013*. URL: http://jdkfg.lubw.baden-wuerttemberg.de/servlet/is/300/.

Marwick, B. (2016). "Computational Reproducibility in Archaeological Research: Basic Principles and a Case Study of Their Implementation". *Journal of Archaeological Method and Theory*, 1–27.

Münch, D. and C. G. Galizia (2016). "DoOR 2.0 - Comprehensive Mapping of Drosophila Melanogaster Odorant Responses". *Scientific Reports* 6, 21841.

NIH (2016). *NIH Chemical Identifier Resolver*. URL: http://cactus.nci.nih.gov/chemical/structure.

O'Boyle, N. M., R. Guha, E. L. Willighagen, S. E. Adams, J. Alvarsson, J.-C. Bradley, I. V. Filippov, R. M. Hanson, M. D. Hanwell, G. R. Hutchison, and et al. (2011). "Open Data, Open Source and Open Standards in Chemistry: The Blue Obelisk Five Years On." *Journal of Cheminformatics* 3, 37.

Ooms, J. (2014). "The jsonlite Package: A Practical and Consistent Mapping Between JSON Data and R Objects". *arXiv preprint*. URL: http://arxiv.org/abs/1403.2805.

PAN (2016). *Pesticide Action Network(PAN) Pesticide Database*. URL: http://www.pesticideinfo.org/.

Pence, H. E. and A. Williams (2010). "ChemSpider: An Online Chemical Information Resource". *Journal of Chemical Education* 87 (11), 1123–1124.

Peng, R. D. (2009). "Reproducible Research and Biostatistics". *Biostatistics* 10 (3), 405–408.

Poisot, T. (2015). "Best Publishing Practices to Improve User Confidence in Scientific Software". *Ideas in Ecology and Evolution* 8.

R Core Team (2016). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Reichman, O. J., M. B. Jones, and M. P. Schildhauer (2011). "Challenges and Opportunities of Open Data in Ecology". *Science* 331 (6018), 703–5.

Stieger, G., M. Scheringer, C. A. Ng, and K. Hungerbühler (2014). "Assessing the Persistence, Bioaccumulation Potential and Toxicity of Brominated Flame Retardants: Data Availability and Quality for 36 Alternative Brominated Flame Retardants". *Chemosphere* 116, 118–123.

Swain, M. (2015a). *ChemSpiPy*. URL: https://github.com/mcs07/ChemSpiPy.

Swain, M. (2015b). *PubChemPy*. URL: https://github.com/mcs07/PubChemPy.

Swain, M. (2016). *CIRpy*. URL: https://github.com/mcs07/CIRpy.

Tomasulo, P. (2002). "ChemIDplus - Super Source for Chemical and Drug Information". *Medical Reference Services Quarterly* 21 (1), 53–59.

Travis-CI (2016). URL: https://travis-ci.org/.

UBA (2016). *ETOX: Information System Ecotoxicology and Environmental Quality Targets*. URL: https://webetox.uba.de/webETOX/index.do.

U.S. EPA (2016). *ECOTOX database*. URL: http://cfpub.epa.gov/ecotox/.

Wehrens, R. (2011). *Chemometrics with R: Multivariate Data Analysis in the Natural Sciences and Life Sciences*. Springer.

Weininger, D. (1990). "SMILES. 3. DEPICT. Graphical Depiction of Chemical Structures". *Journal of Chemical Information and Computer Sciences* 30 (3), 237–243.

EU-WFD (2000). "Directive 2000/60/EC of the European Parliament and of the Council Establishing a Framework for the Community Action in the Field of Water Policy". *The European Parliament and Council* (L327/1).

Wickham, H. (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag. URL: http://had.co.nz/ggplot2/book.

Wickham, H. (2011). "testthat: Get Started with Testing". *The R Journal* 3, 5–10.

Wickham, H. (2015a). *Advanced R*. The R Series. CRC Press.

Wickham, H. (2015b). *rvest: Easily Harvest (Scrape) Web Pages*. R package version 0.3.1. URL: https://CRAN.R-project.org/package=rvest.

Wickham, H. (2015c). *stringr: Simple, Consistent Wrappers for Common String Operations*. R package version 1.0.0. URL: http://CRAN.R-project.org/package=stringr.

Wickham, H. (2015d). *xml2: Parse XML*. R package version 0.1.2. URL: https://CRAN.R-project.org/package=xml2.

Wickham, H. (2016). *httr: Tools for Working with URLs and HTTP*. R package version 1.1.0. URL: https://CRAN.R-project.org/package=httr.

Wickham, H., P. Danenberg, and M. Eugster (2015). *roxygen2: In-Source Documentation for R*. R package version 5.0.1. URL: http://CRAN.R-project.org/package=roxygen2.

Wikipedia (2016). *WikiProject Chemistry*. URL: https://www.wikidata.org/wiki/Wikidata:WikiProject_Chemistry.

Wilson, G., D. A. Aruliah, C. T. Brown, N. P. Chue Hong, M. Davis, R. T. Guy, S. H. D. Haddock, K. D. Huff, I. M. Mitchell, M. D. Plumbley, B. Waugh, E. P. White, and P. Wilson (2014). "Best Practices for Scientific Computing". *PLoS Biology* 12 (1), e1001745.

Wohlgemuth, G., P. K. Haldiya, E. Willighagen, T. Kind, and O. Fiehn (2010). "The Chemical Translation Service – a Web-Based Tool to Improve Standardization of Metabolomic Reports". *Bioinformatics* 26 (20), 2647–2648.

Wood, A. (2016). *Compendium of Pesticide Common Names*. URL: http://www.alanwood.net/pesticides/index.

Zenodo (2016). *webchem: Retrieve Chemical Information from the Web*. URL: http://dx.doi.org/10.5281/zenodo.33823.