Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. doi: 10.18637/jss.v000.i00

webchem: An R Package to retrieve Chemical Information from the Web

Eduard Szöcs
Universität Koblenz-Landau

Ralf B. Schäfer
Universität Koblenz-Landau

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.

Keywords: ecotoxicology, chemistry, data cleaning, web scraping, ropensci.

1. 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 datasets (Chapman 2005). In practice, the data cleaning step is often more time consuming than the subsequent statistical analysis, particularly, when the analysis relies on multiple data sources that need to be joined.

When dealing with chemical datasets (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 2015; Wehrens 2011). R is one of the most widely used software 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**.

2. 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 (`https://github.com/ropensci/webchem`) and a stable version is released on the official R repository (CRAN, `https://goo.gl/OKbZaF`). **webchem** is part of the rOpenSci project (`https://ropensci.org/`), which aims at fully reproducible data analysis.

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

webchem builds on top of the following R packages: **RCurl** (Lang and team 2015) and **httr** (Wickham 2015b) for data transfer, **stringr** (Wickham 2015d) for string handling, **xml2** (Wickham 2015e) and **rvest** (Wickham 2015c) 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 from 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 this is provided. 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 timeout of 0.3 to 2 seconds depending on the datasource. Therefore, processing larger datasets 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.

3. 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 legal information (Figure 1, a detailed overview of all sources is included as supplement):

NIH Chemical Identifier Resolver (CIR) (NIH 2016) A web service that converts from and to various chemical identifiers.

Chemical Translation Service (CTS) (Wohlgemuth *et al.* 2010) A web service that converts from and to various chemical identifiers.

ETOX (UBA 2016) 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.

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

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

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

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

Compendium of Pesticide Common Names (Wood 2016) The compendium provides information on pesticide common names, identifiers and classification.

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

ChemSpider (Pence and Williams 2010) 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.

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

Though the data sources exhibit some overlap in the provided information, each has been selected because it provides unique information and we encourage the interested reader to

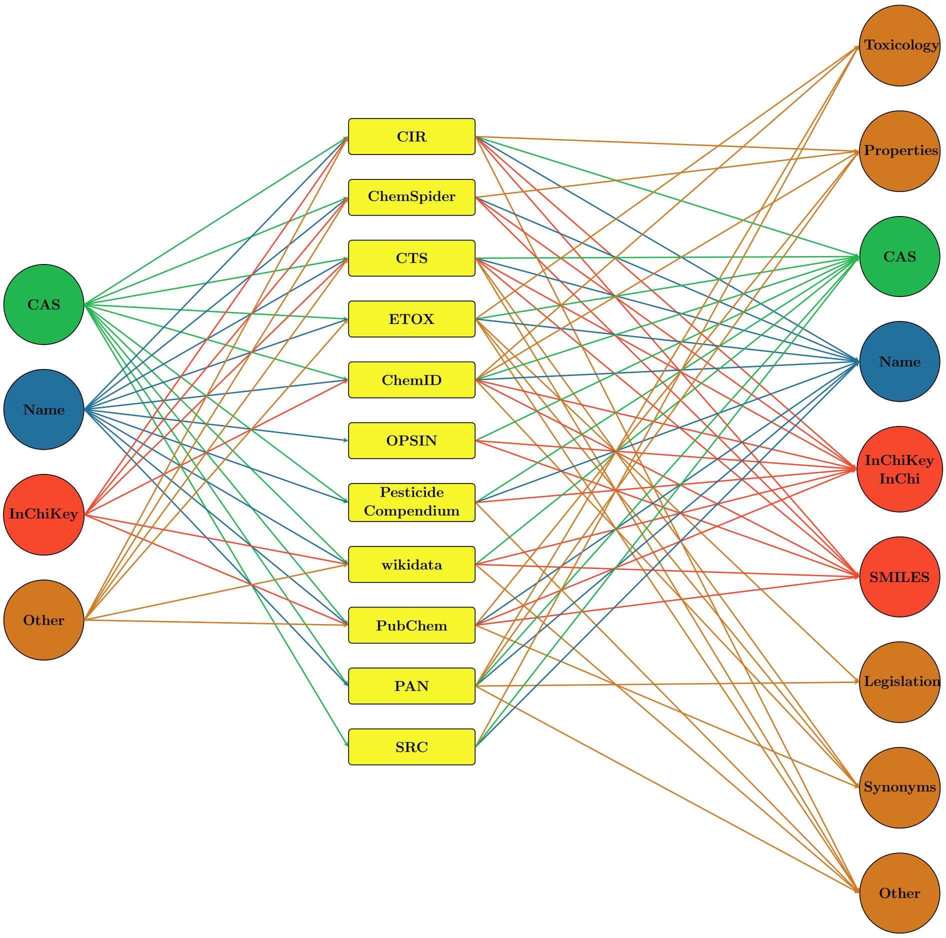

Figure 1: Overview of current data sources. Input and output possibilities currently implemented in the package.

consult the related source for details. However, we provide a brief overview in the Supporting Information.

4. Use Cases

4.1. 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 http://dx.doi.org/10.5281/zenodo.46930. This document has been created using **webchem** version 0.1.

4.2. Sample datasets

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

(i) `jagst`: This dataset comprises environmental monitoring data of organic substances in the river Jagst, Germany, collected 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.

```
R> # load jagst dataset
R> data(jagst)
R> # print first 6 lines of data
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 dataset 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 dataset identifies the substances only by CAS numbers.

4.3. Query Identifiers

The `jagst` dataset 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 were 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:

```
R> # unique substance names
R> subs <- unique(jagst$substance)
R> # search ETOX IDs
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 output / 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 is 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 other source using CAS numbers (Figure 1):

```
R> # query SMILES from PubChem
R> cids <- get_cid(etox_cas)
R> pc_data <- pc_prop(cids, properties = c('CanonicalSMILES'))
R> pc_smiles <- smiles(pc_data)
R>
R> # query InChiKey from ChemSpider
R> # token <- '<yourtokenhere>'
R> csids <- get_csid(etox_cas, token = token)
R> cs_data <- cs_compinfo(csids, token = token)
R> cs_inchikey <- inchikey(cs_data)
```

```
R> # combine in 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 webpage. We end up with a compound table containing many different identifiers (Table 1), allowing easy identification and merging with other datasets, e.g. the `lc50` dataset based on CAS.

Name	CAS	SMILES[a]	CID	InChIKey[a]	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: Identifiers for the jagst datasets as queried with **webchem**. Only the first 6 entries are shown. - = not found. [a] Only the first 7 characters are shown.

4.4. 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` dataset. The Compendium of Pesticide Common Names (Wood 2016) contains such information and can be easily queried with **webchem**.

```
R> # query the compendium using CAS numbers
R> aw_data <- aw_query(lc50$cas, type = 'cas')
```

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

```
R> # shows internal structure of the data object
R> # str(aw_data[[1]])
R> # extract chemical group
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 2 displays the result after some more 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).

4.5. Querying partitioning coefficients

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

```
R> # query PHYSPROP DATABASE
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> # look  at internal structure
R> # str(pp_data[[1]])
R> lc50$logp <- sapply(pp_data, function(y){
+    if (length(y) == 1 && is.na(y))
+      return(NA)
+    y$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 gives users full

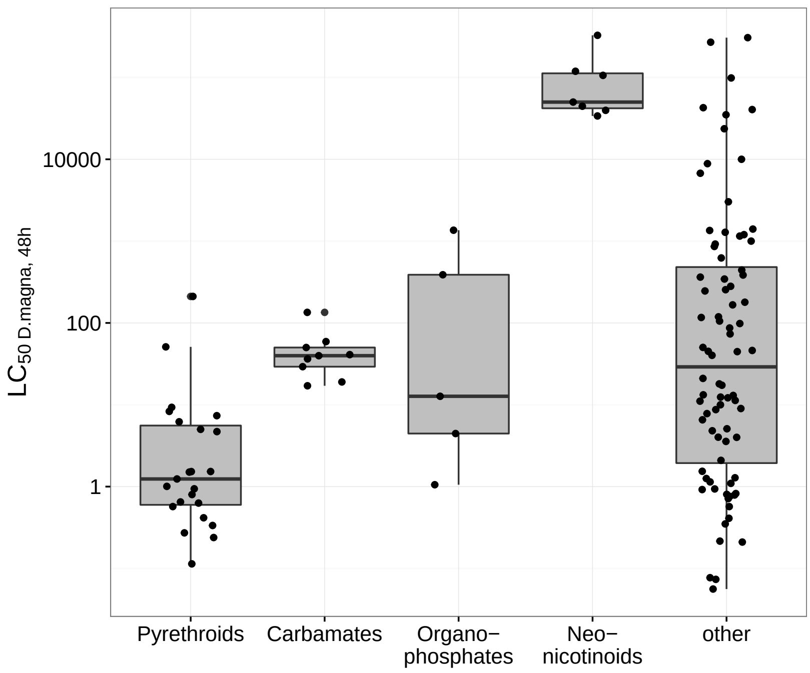

Figure 2: Toxicity of different pesticide groups. LC_{50} values have been retrieved from EPA ECOTOX database, chemical groups from the Compendium of Pesticide Common Names (Wood 2016)

control over how to organize their data. Nevertheless, we provide in the documentation of each function examples on how to extract more complicated parts of the data. The resulting data and model is displayed in Figure 3.

4.6. Utility functions

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

```
R> # simple formatting check
R> is.inchikey('BQJCRHHNABKAKU-KBQPJGBKS-AN')
```

```
Hyphens not at position 15 and 26.

[1] FALSE
```

```
R> # or using ChemSpider
R> is.inchikey('BQJCRHHNABKAKU-KBQPJGBKSA-5',
+         type = 'chemspider')
```

```
[1] FALSE
```

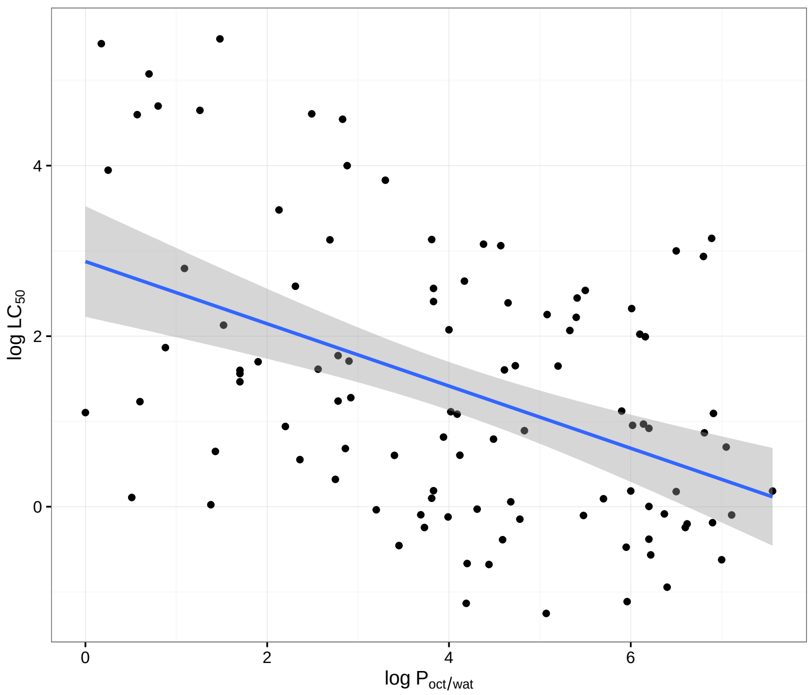

Figure 3: Simple QSAR for predicting log LC_{50} of pesticides by log P. Log P values have been retrieved from SRC Physprop database (97 experimental data, 9 estimated data and 18 substances without data). Blue line indicates the regression model ($\log LC_{50} = 2.88 - 0.37 \log P$, RMSE = 1.45).

```
R> # formatting check
R> is.cas('64-17-6')

Checksum is not correct! 5 vs. 6

[1] FALSE
```

5. Discussion

5.1. Related software

Within the R ecosystem, there are only few similar projects: **rpubchem** (Guha 2014) and **ChemmineR** (Cao *et al.* 2008) both provide an interface to PubChem that is more extensive than the current one in **webchem**. **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 (a popular programming language) 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 legal data).

5.2. Open Science

An increasing number of scientific data is becoming publicly available (Gewin 2016; Reichman *et al.* 2011; O'Boyle *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.

webchem facilitates chemical data availability to researchers. However, data quality is also crucial for data analysis (Stieger *et al.* 2014). Ensuring good quality requires additional effort and methods to be developed.

5.3. 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.

6. 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.

Acknowledgements

We thank all resource maintainers for their work making their data open to the public. We thank Johannes Ranke and Daniel Münch for their contributions to the webchem package, as well as all users who provided feedback and feature requests.

References

Cao, Y, Charisi, A, Cheng, C L, Jiang, T, Girke, T (2008). "**ChemmineR**: a compound mining framework for R." *Bioinformatics*, **24**(15), 1733–1734.

Chamberlain SA, Szöcs E (2013). "**taxize**: Taxonomic Search and Retrieval in R." *F1000Research*, **2**(191). URL `http://f1000research.com/articles/2-191/v2`.

Chapman A (2005). *Principles and Methods of Data Cleaning*. Report for the Global Biodiversity Information Facility, Copenhagen. GBIF. ISBN 978-87-92020-04-8. URL http://www.gbif.org/orc/?doc_id=1262.

Ertl P, Patiny L, Sander T, Rufener C, Zasso M (2015). "Wikipedia Chemical Structure Explorer: Substructure and Similarity Searching of Molecules from Wikipedia." *Journal of Cheminformatics*, **7**(1). doi:10.1186/s13321-015-0061-y.

Gewin V (2016). "Data sharing: An open mind on Open Data." *Nature*, **529**(7584), 117–119. doi:10.1038/nj7584-117a.

Grabner M, Varmuza K, Dehmer M (2012). "**RMol**: a Toolset for transforming SD/Molfile Structure Information into R objects." *Source Code for Biology and Medicine*, **7**, 12. doi:10.1186/1751-0473-7-12.

Guha R (2007). "Chemical Informatics Functionality in R." *Journal of Statistical Software*, **18**(6).

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 SR, McNaught A, Pletnev I, Stein S, Tchekhovskoi D (2015). "InChI, the IUPAC International Chemical Identifier." *Journal of Cheminformatics*, **7**(1). doi:10.1186/s13321-015-0068-4.

Howard PH, Meylan W (2016). "Physical/chemical property database (PHYSPROP)." URL http://www.srcinc.com/what-we-do/environmental/scientific-databases.html#physprop.

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

Kim S, Thiessen PA, Bolton EE, Bryant SH (2015). "PUG-SOAP and PUG-REST: Web Services for Programmatic Access to Chemical Information in PubChem." *Nucleic Acids Research*, **43**(W1), W605–W611. doi:10.1093/nar/gkv396.

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

Lang DT, team tC (2015). ***RCurl**: General Network (HTTP/FTP/...) Client Interface for R*. R package version 1.95-4.7, URL http://CRAN.R-project.org/package=RCurl.

Lowe DM, Corbett PT, Murray-Rust P, Glen RC (2011). "Chemical Name to Structure: OPSIN, an Open Source Solution." *Journal of Chemical Information and Modeling*, **51**(3), 739–753. doi:10.1021/ci100384d. URL http://opsin.ch.cam.ac.uk/instructions.html.

LUBW - Landesanstalt für Umwelt, Messungen und Naturschutz Baden-Württemberg (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*, p. 1–27.

Münch D, Galizia CG (2016). "DoOR 2.0 - Comprehensive Mapping of Drosophila melanogaster Odorant Responses." *Scientific Reports*, **6**, 21841. doi:10.1038/srep21841.

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

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

O'Boyle NM, Guha R, Willighagen EL, Adams SE, Alvarsson J, Bradley JC, Filippov IV, Hanson RM, Hanwell MD, Hutchison GR, et al (2011). "Open Data, Open Source and Open Standards in chemistry: The Blue Obelisk five years on." *Journal of Cheminformatics*, **3**, 37.

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

Pence HE, Williams A (2010). "ChemSpider: An Online Chemical Information Resource." *Journal of Chemical Education*, **87**(11), 1123–1124. doi:10.1021/ed100697w.

Peng RD (2009). "Reproducible Research and Biostatistics." *Biostatistics*, **10**(3), 405–408. doi:10.1093/biostatistics/kxp014.

Poisot T (2015). "Best Publishing Practices to Improve User Confidence in Scientific Software." *Ideas in Ecology and Evolution*, **8**. doi:10.4033/iee.2015.8.8.f.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Reichman OJ, Jones MB, Schildhauer MP (2011). "Challenges and Opportunities of Open Data in Ecology." *Science*, **331**(6018), 703–5.

Stieger G, Scheringer M, Ng CA, Hungerbühler K (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. doi:10.1016/j.chemosphere.2014.01.083.

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. doi:10.1300/J115v21n01_04.

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

US 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. ISBN 3-642-17840-5.

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

Wickham H (2009). ***ggplot2**: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-0-387-98140-6. URL http://had.co.nz/ggplot2/book.

Wickham H (2011). "**testthat**: Get Started with Testing." *The R Journal*, **3**, 5–10. URL http://journal.r-project.org/archive/2011-1/RJournal_2011-1_Wickham.pdf.

Wickham H (2015a). *Advanced R*. The R series. CRC Press. ISBN 978-1-4665-8696-3.

Wickham H (2015b). ***httr**: Tools for Working with URLs and HTTP*. R package version 1.0.0, URL https://CRAN.R-project.org/package=httr.

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

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

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

Wickham H, Danenberg P, Eugster M (2015). ***roxygen2**: In-Source Documentation for R*. R package version 5.0.0, URL http://CRAN.R-project.org/package=roxygen2.

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

Wilson G, Aruliah DA, Brown CT, Chue Hong NP, Davis M, Guy RT, Haddock SHD, Huff KD, Mitchell IM, Plumbley MD, Waugh B, White EP, Wilson P (2014). "Best Practices for Scientific Computing." *PLoS Biology*, **12**(1), e1001745. doi:10.1371/journal.pbio.1001745.

Wohlgemuth G, Haldiya PK, Willighagen E, Kind T, Fiehn O (2010). "The Chemical Translation Service – a web-based Tool to improve Standardization of Metabolomic Reports." *Bioinformatics*, **26**(20), 2647–2648. doi:10.1093/bioinformatics/btq476. URL http://cts.fiehnlab.ucdavis.edu/.

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

Affiliation:

Eduard Szöcs
Institute for Environmental Sciences
Universität Koblenz-Landau
Fortstraße 7
76829 Landau, Germany
E-mail: szoecs@uni-landau.de
URL: https://edild.github.io

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
MMMMMM YYYY, Volume VV, Issue II *Submitted:* yyyy-mm-dd
doi:10.18637/jss.v000.i00 *Accepted:* yyyy-mm-dd