CAS and PubChem Chemical Descriptors (Draft)

David Hall and Hui Peng

September 18, 2020

Howdy!

In this brief exercise we'll be familiarizing (or introducing) ourselves to the world of cheminformatics. We'll discuss this in more detail as we work through this exercise, but for now it simply means we'll make use of *in silico* (i.e. computers) techniques to tackle problems in environmental chemistry.

Objectives

At the start of this exercise you'll be randomly assigned an environementally relevant chemical pollutant. **Note**, you'll be working with your assigned chemical throughout the remaining environmental chemistry section of the course. You'll then compare your unique chemical against a short list of 6 perfluoroalkyl substances (PFAS).

At the end of the exercise you should have the following:

1. A saved .csv file with the relevant chemical properties of your unique chemical and the short-list of PFAS compounds
2. A plot that explores the relationship between the chemical properties of your unique chemical and the short-list of PFAS compounds
3. A brief written analysis where you hypothesis/rationalize the results of your plot. Don't worry to much about being technically correct (boring), we're looking more for curiosity and neat ideas (what the world actually needs).
4. All of the above written in an R markdown document and submitted as a PDF (like this one!).

Obtaining your assigned chemicals

Before starting the R section of this lab, please complete the "Chemical Assignement Quiz" on Quercus. It's called a "quiz" because of the way QWuercus works, but really it simply assignes you a random chemical from a list and keeps track of it for us (the isntructors) to reference later on.

After the quiz you should have an assigned chemical (ex. *aspirin*), it's abbreviated name (ex. *ASA*) and it's CAS number (ex. *50-78-2*). Please keep track of these three values to prevent numerous headaches for all us down the line.

Packged needed for this lab

We'll be using the `webchem` and `tidyverse` packages from the CRAN repository. If you haven't already installed them you can run the following code in the console below or install them using the "Install" button under the "Packages" tab in the bottom right window of RStudio.

```
install.packages("tidyverse") # contains several packages such as dplyr and ggplot2.
install.packages("webchem")
```

After you’ve installed the packages, load them into your current R session by running the code below.

```
library(webchem)
library(tidyverse)
```

The `tidyverse` is a collection of R packages writen by H. Wickham et al. They work synergisticaly to streamline data analysis in R. `ggplot2` is an example of a tidyverse package, but there’s so much more to the tidyverse then that. Read more at tidyverse.org and please look into the *R for Data Science* book by Wickham and Grolemund (2017) to learn more. The book can be found online at r4ds.had.co.nz/.

The `webchem` package was written by E. Szocs et al. and allows the importation of chemical information from a variety of online databases. Example vignettes using `webchem` and the package reference manual are hosted on CRAN.

Importing your data into R

Obtaining pre-calculated chemical descriptors from PubChem

This needs to be written but I’ll probably talk about:

- What are CAS numbers
- What is PubChem & pubchem CIDs
- Point by Point description of each molecular desciptor

Depending on the how we assign their chemicals, I was planning on having them construct a data.frame from scratch and build it up column by column. What I did below was done on a list of all 24 chemicals, and we can use it for later reference.

Note: Can’t use `cir_query` because some CASRNs aren’t in the *Chemical Identifier Resolver*, the data base used by the code you sent me earlier.

```
# This file is a list of all 24 compounds mentionned across the 4 projects.
data <- read.csv(’data/CompoundsList_Env316.csv’, header = TRUE, fileEncoding="UTF-8-BOM")

CASRN <- as.vector(data$CAS)

cids <- get_cid(CASRN) # get’s the pubchem Compound IDs (cids)

# Get properties listed on pubchem
x <- pc_prop(cids$cid,
             properties = c("MolecularFormula",
                            "MolecularWeight",
                            "CanonicalSMILES",
                            "IUPACName",
                            "XLogP",
                            "MonoisotopicMass",
                            "TPSA", # total polar surface area
                            "Charge",
                            "Volume3D"))

# Combining everything into one file for subsequent lectures
dataPubChem <- cbind(data,x)
write.csv(dataPubChem,
          file = "data/CmpdListPubChem_Env316.csv",
          row.names = FALSE)
```

Table 1: The first 6 columns of our compound chemical descriptors data frame.

Project	Compound.Name	Abbreviation	CAS	CID	MolecularFormula
1	Perfluorodecanoate	PFDA	73829-36-4	21895380	C10F19O2-
1	Perfluoroundecanoate	PFUnDA	196859-54-8	23533165	C11F21O2-
1	Perfluorododecanoate	PFDoDA	171978-95-3	22174013	C12F23O2-
1	Perfluorotridecanoate	PFTriDA	862374-87-6	23084971	C13F25O2-
1	Perfluorooctane sulfonate	PFOS	1763-23-1	74483	C8HF17O3S

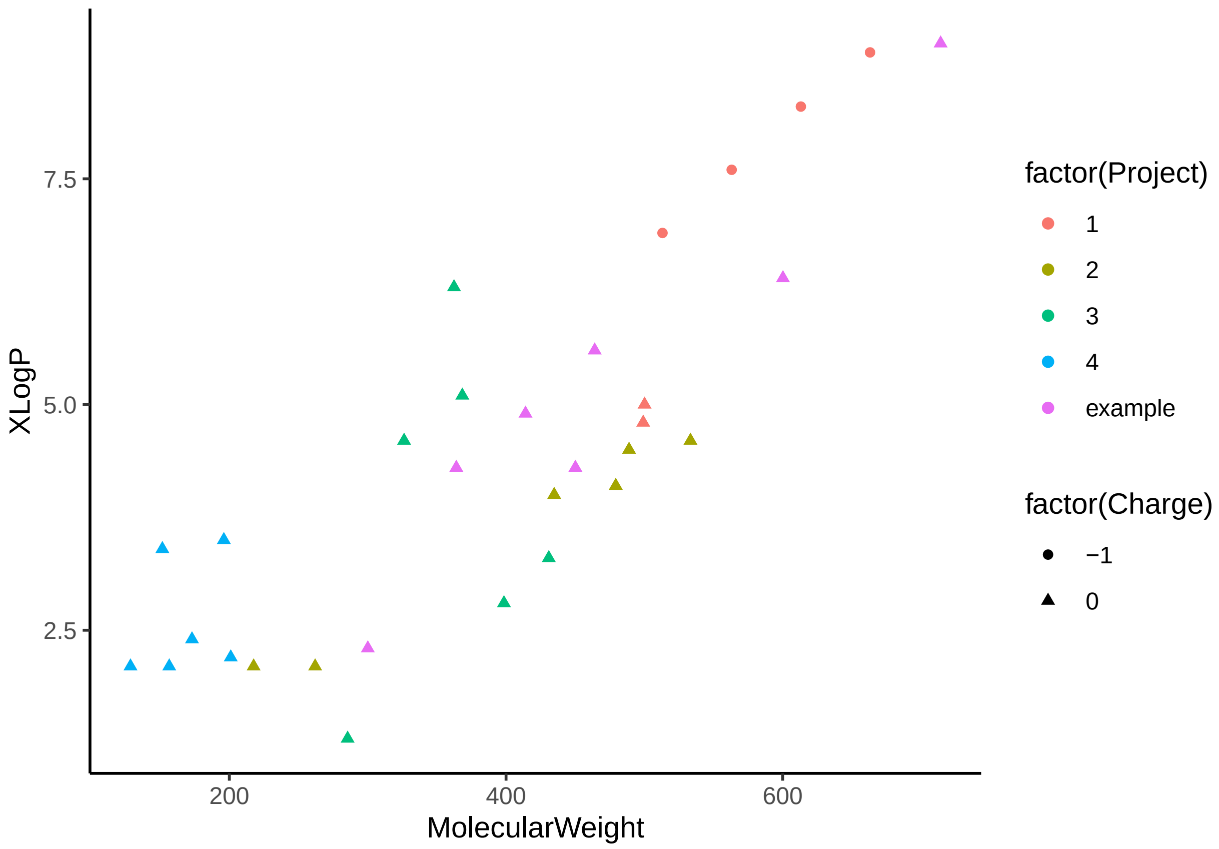
The calculated values scrapped from PubChem are: `CAS`, `CID`, `MolecularFormula`, `MolecularWeight`, `CanonicalSMILES`, `IUPACName`, `XLogP`, `MonoisotopicMass`, `TPSA`, `Charge`, `Volume3D`.

Quick data exploration of our compounds

Need to expand, but essentially quick hypothesis to test is if polarity increases with molecular weight. The theory being that larger molecules are more complex, and hence have a lower chance of cancelling out dipole moments. We'll use XLogP as a stand-in for polatiry and molecular weight as a standing for complexity. The higher the XLogP the more polar, the higher the molecular weight the greater the chemical complexity. **I know this needs a bit more justfication, but i'm just trying to show them how to make plots**.

```
dataPubChem <- read.csv("data/CmpdListPubChem_Env316.csv", header = TRUE)


ggplot(dataPubChem, aes(x = MolecularWeight,
                        y = XLogP,
                        colour = factor(Project),
                        shape = factor(Charge))) +
  geom_point() +
  theme_classic()
```



From the plot in Figure 1, it appears their is a positive correlation between MW and LogP. You can see the impact of formal charges on LogP in the top right.