

# Retrieving CCS Values from PubChem

Emma SCHYMANSKI

05/07/2022

## Background and Setup

This is a brief RMarkdown document to describe how to retrieve experimental CCS values from PubChem. First, set up the packages etc:

```
library(rjson)
library(httr)
library(RChemMass)
library(webchem)
library(data.table)

# Get the extract annotations script
extractAnno_url <- "https://gitlab.lcsb.uni.lu/eci/pubchem/-/raw/master/
                    annotations/tps/extractAnnotations.R"
download.file(extractAnno_url, "extractAnnotations.R")
source("extractAnnotations.R")
# Get the CCS-specific function:
CCSanno_url <- "https://gitlab.lcsb.uni.lu/eci/pubchem/-/raw/master/
               annotations/CCS/getPcAnno_CCS.R"
download.file(CCSanno_url, "getPcAnno_CCS.R")
source("getPcAnno_CCS.R")
```

## Testing existing functions

Now, test that the existing annotations functions work on the CCS cases. First, find the total number of pages. The two data sources are [CCSbase](#) and [NORMAN Suspect List Exchange](#)

```
source_1 <- "NORMAN Suspect List Exchange"
source_2 <- "CCSbase"
heading <- "Collision Cross Section"
getPcAnno.TotalPages(source_1, heading, timeout = 100)
```

```
## [1] 2
```

```
getPcAnno.TotalPages(source_2, heading, timeout = 100)
```

```
## [1] 15
```

Now, retrieve CCS values for the first page for the [NORMAN Suspect List Exchange](#) (NORMAN-SLE) as a test:

```
source_1 <- "NORMAN Suspect List Exchange"
heading <- "Collision Cross Section"
getPcAnno.CCS(source_1, heading, file_name = "SLE_CCS_test_p1.csv",
               timeout = 100)
```

```
## [1] "SLE_CCS_test_p1.csv"
```

Now, retrieve all CCS values for the [NORMAN-SLE](#):

```
source_SLE <- "NORMAN Suspect List Exchange"
heading <- "Collision Cross Section"
getPcAnno.allCCS(source = source_SLE, heading = heading, base_file_name = "SLE_CCS")
```

Then, retrieve all CCS values for [CCSbase](#) (several pages):

```
source_CCSbase <- "CCSbase"
heading <- "Collision Cross Section"
getPcAnno.allCCS(source = source_CCSbase, heading = heading,
  base_file_name = "CCSbase_CCS")
```

Now ... try to merge them all together:

```
CCS_data <- read.csv("SLE_CCS_all.csv", stringsAsFactors = F)
CCS_data2 <- read.csv("CCSbase_CCS_all.csv", stringsAsFactors = F)
CCS_data_all <- merge(CCS_data, CCS_data2, all = TRUE)
write.csv(CCS_data_all, "All_CCS_in_PubChem.csv", row.names = F)
```

Remove some of the files no longer needed:

```
# clean up

# SLE files
intermediate_files <- list.files(path = ".", pattern = "SLE_CCS_page",
  recursive = F, include.dirs = F, full.names = T)
file.remove(intermediate_files)

# test file
test_file <- list.files(path = ".", pattern = "SLE_CCS_test",
  recursive = F, include.dirs = F, full.names = T)
file.remove(test_file)

# CCSbase files
intermediate_files <- list.files(path = ".", pattern = "CCSbase_CCS_page",
  recursive = F, include.dirs = F, full.names = T)
file.remove(intermediate_files)
```

## Value add the “All CCS in PubChem” Dataset

The code above retrieves the CCS values and accompanying annotations, along with the PubChem compound identifier (CID). Next, split out the CCS annotation (all merged in one field) into the CCS values, the adduct species and, where applicable, the comment (instrumentation details).

```
# CCS_data_all <-
# read.csv('All_CCS_in_PubChem.csv', stringsAsFactors = F)
CCS_data_all$CCS_A2 <- ""
CCS_data_all$Adduct <- ""
CCS_data_all$Comment <- ""

for (i in 1:length(CCS_data_all$data_CCS)) {
  CCS_entry <- CCS_data_all$data_CCS[i]
  CCS_entry_spl <- strsplit(CCS_entry, " ")[[1]]
  len_spl <- length(CCS_entry_spl)
  CCS_data_all$CCS_A2[i] <- CCS_entry_spl[1]
```

```

CCS_data_all$Adduct[i] <- CCS_entry_spl[3]
# if longer than 3, there is a comment
if (len_spl > 3) {
  comment_str <- paste(CCS_entry_spl[4:len_spl], collapse = " ")
  # comment_str <- sub('[', '', comment_str)
  # comment_str <- sub(']', '', comment_str)
  CCS_data_all$Comment[i] <- comment_str
}
}

```

Next, fill in the chemical information using webchem and datatable.

```

selected_properties <- c("Title", "ExactMass", "MolecularFormula",
  "XlogP", "InChI", "IsomericSMILES", "InChIKey", "IUPACName")
cids <- unique(CCS_data_all$linking_cid)
cids <- cids[-grep("NULL", cids)]

# retrieve info with webchem
CID_info_all <- as.data.table(webchem::pc_prop(cids, selected_properties))

# merge
CCS_CID_info_all <- merge(CCS_data_all, CID_info_all, by.x = "linking_cid",
  by.y = "CID")

# tidy col names
col_names <- colnames(CCS_CID_info_all)
col_names[1] <- "PubChem_CID"
colnames(CCS_CID_info_all) <- col_names

# output - note this removes NULL entries
write.csv(CCS_CID_info_all, "All_CCS_in_PubChem_wInfo.csv", row.names = F)

```

This dataset is available for download and use on Zenodo [DOI:10.5281/zenodo.6800138](https://doi.org/10.5281/zenodo.6800138) or [GitLab](#).

If you have any questions, please contact the [ECI NORMAN-SLE team](#) or [PubChem help mailing list](#) for more information.

**Enjoy!**