chem-bla-ics

Oscar4 command line utilities

Egon Willighagen

Published November 18, 2010

Citation
Willighagen, E. (2010, November 18). Oscar4 command line utilities. *Chem-bla-ics*. https://doi.org/10.59350/jsfck-t351

Keywords
Oscar, Textmining, Beilstein

Abstract
One goal of my three month project is to take Oscar4 to the community. We want to get it used more, and we need a larger development community. Oscar4 and the related technologies do a good, sometimes excellent, job, but have to be maintained, just like any other piece of code. To make using it easier, we are developing new APIs, as well as two user-oriented applications: a Taverna 2 plugin , and command line utilities.

One goal of my three month project is to take Oscar4 to the community. We want to get it used more, and we need a larger development community. Oscar4 and the related technologies do a good, sometimes excellent, job, but have to be maintained, just like any other piece of code. To make using it easier, we are developing new APIs, as well as two user-oriented applications: a Taverna 2 plugin , and command line utilities. The Oscar4 Java API has slightly evolved in the last three weeks, removing some complexity. In this post, I will introduce the command line utilities.

Oscar4

Most people will be mostly interested into the full Oscar4 program, to extract chemical entities. Oscar3 was also capable of extracting data (like NMR spectra), but that is not yet being ported. The OscarCLI program takes input, extracts chemicals, and where possible resolves them into connection tables (viz. InChI).

To extract chemicals from a line of text (e.g. *"This is propane."*, you do:

```
$ java -cp oscar4-cli-4.0-SNAPSHOT.jar \
  uk.ac.cam.ch.wwmm.oscar.oscarcli.OscarCLI \
  This is propane.
propane: InChI=1/C3H8/c1-3-2/h3H2,1-2H3
```

For larger chunks of texts it is easier to route it via stdin, for which we can use the `-stdin` option:

```
$ echo "This is propane." | \
  java -cp oscar4-cli-4.0-SNAPSHOT.jar \
  uk.ac.cam.ch.wwmm.oscar.oscarcli.OscarCLI \
  -stdin
propane: InChI=1/C3H8/c1-3-2/h3H2,1-2H3
```

That way, we can easily process large plain text files (output omitted):

```
$ cat largeFile.txt | \
  java -cp oscar4-cli-4.0-SNAPSHOT.jar \
  uk.ac.cam.ch.wwmm.oscar.oscarcli.OscarCLI \
  -stdin
```

If you prefer RDF output, for further integration, use the `-output text/turtle`:

```
$ cat largeFile.txt | \
  java -cp oscar4-cli-4.0-SNAPSHOT.jar \
  uk.ac.cam.ch.wwmm.oscar.oscarcli.OscarCLI \
  -stdin -output text/turtle
```

This returns RDF using the CHEMINF ontology like:

```
@prefix dc:  .
@prefix rdfs:  .
@prefix ex:  .
@prefix cheminf:  .
@prefix sio: .

ex:entity0
  rdfs:subClassOf cheminf:CHEMINF_000000 ;
  dc:label "propane" ;
  cheminf:CHEMINF_000200 [
    a cheminf:CHEMINF_000113 ;
    sio:SIO_000300 "InChI=1/C3H8/c1-3-2/h3H2,1-2H3" .
  ] .
```

We can, however, also use Jericho to extract text from HTML pages, made available with the `-html` option, and pulling in a Beilstein Journal of Organic Chemistry paper with wget:

```
$ wget -qO- https://doi.org/10.3762/bjoc.6.122 | \
  java -cp oscar4-cli-4.0-SNAPSHOT.jar \
  uk.ac.cam.ch.wwmm.oscar.oscarcli.OscarCLI \
  -stdin -html
```

This will return 271 chemical entities recognized in the text, matching 48 unique chemical structures.