chem-bla-ics

Validating MDL SD files and Symyx molfiles with the CDK

Egon Willighagen

Published January 30, 2010

Citation

Willighagen, E. (2010). Validating MDL SD files and Symyx molfiles with the CDK. In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/zwv8r-yey62

Keywords

Cdk, Chemistry

Bioclipse 2.0 introduced a new, powerful molecular table support, and we have been eager to test that on large SD files. A recent ChEBI SD file failed to open, and eyes were immediately at the CDK, which is the cheminformatics library used in Bioclipse.

After careful investigations, it turned out that the ChEBI file contained a few entries which were not MDL molfiles, but queries for the ISISBase system. Those cannot be read by the CDK MDLV2000Reader. However, it crashed on it, instead of failing more savely. That's not nice, and fixed. But, the problem is rather recurrent, and the reason why I like CML so much: invalid input. CML, based on XML, has several general validation approaches that give in-depth error messages of what is wrong with the file.

So, I asked on the BOx what the Open Source cheminformatics community had to offer for this. Turns out that several tools find problems in the files, but none could report where the error occurred.

Validation

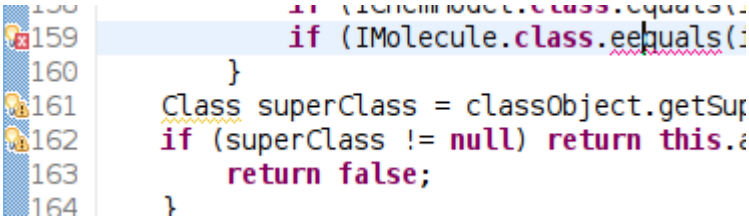
Now, some time ago, I played with two reading modes, RELAXED and STRICT, as faulty files is core cheminformatics material, and the software is blamed if the QSAR model resulting from it is not good (seriously). Anyways, a small API change in the CDK would make a validating MDLV2000Reader quite a step closer, but I had not followed up on it until last Friday where I patch I was reviewing caused 6 new unit test fails. The new fails were caused by a assumption which turned out the be false in the test files used in those 6 unit tests.

The MDL (or Symyx) molfile specifications (not an Open Specification) defines an atom block line as:

```
xxxxx.xxxxyyyyy.yyyyzzzzz.zzzz aaaddcccssshhhbbbvvvHHHrrriiimmmnnneee
```

but does not specify which fields are optional. And indeed, many tools around save MDL molfiles with one or more fields missing, leading to shorter than expected line lengths. And, as you might have expected, the failing unit tests had files with lines missing the field introduced by the patch, causing Exceptions being thrown around. *I have yet to make up my mind of the lack of those fields is a problem in the file, or allowed by the format.* In either case, the information from that field is not available, and the reader could safely ignore the missing information. Per user demand.

Now, personally, I rather send the file back to the user with a proper error report and show them what is wrong with the file. Or better, provide them with a MDL V2000 text editor (e.g. in Bioclipse) which would graphically highlight errors, as many of us are used to with Eclipse:



CDK Patch

So, I am hacking up a patch for CDK master to allow error reporting by IChemObjectReaders. The initial version of the API update and use in the MDLV2000Reader are available as Gist 290659. They are not final yet, as I realized when making the above screenshot, that merely int col is not enough, and that I actually need the *startCol* and *endCol* positions instead. Also, there are only an error level at this moment, and no warning level as in the screenshot.

That said, I created a jar (*ant dist-large*) and saved it as mdlCheck.jar, and wrote a bit of Groovy:

which defines a class implementing the new *IChemObjectReaderErrorHandler* and then reads a MDL molfile. And the output looks like it fulfills my needs (test6.sf):

```
$ CLASSPATH=mdlCheck.jar groovy mdlCheck.groovy src/test/data/mdl/test6.sdf
location: 5, 35: Could not parse mass difference field.
  -> For input string: ""
location: 6, 35: Could not parse mass difference field.
  -> For input string: ""
```

Note to myself, that atom block does not like like a MDL molfile atom block at all! Every second line outputs the Exception passed to the error handler. I have to say, those messages are rather cryptic, but resulting from a NumberFormatException, if not mistaken.

Or, another common found issue (using D and T as element symbols; hisotopes.mol):

```
$ CLASSPATH=mdlCheck.jar groovy mdlCheck.groovy src/test/data/mdl/hisotopes.mol
location: 6, 32: Invalid element type. Must be an existing element, or one in: A, Q, L, L
location: 7, 32: Invalid element type. Must be an existing element, or one in: A, Q, L, L
```

Enough for now... dinner time.