The molecular QSAR descriptors in the CDK

Egon Willighagen ⬤

Published December 20, 2007

Citation

Willighagen, E. (2007). The molecular QSAR descriptors in the CDK. In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/47rre-wwc20

Keywords

Cdk, Qsar, Bioclipse

Abstract

Pending the release of Bioclipse 1.2.0, Ola asked me to do some additional feature implementation for the QSAR feature, such as having the filenames as labels in the descriptor matrix.

Copyright

chem-bla-ics

Pending the release of Bioclipse 1.2.0, Ola asked me to do some additional feature implementation for the QSAR feature, such as having the filenames as labels in the descriptor matrix. See also these earlier items:

- More QSAR in Bioclipse: the JOELib extension
- Further Bioclipse QSAR functionality development
- QSAR plugin for Bioclipse getting in shape
- Bioclipse now allows QSAR descriptor selection
- Bioclipse Workshop: short but productive

(How more open notebook science can you get?)

But I ran into some trouble when both JOElib and CDK descriptors were selected, or Ola really. Now, nothing much I plan to do on the JOElib code, but at least I code investigate the CDK code.

The QSAR descriptor framework has been published in the *Recent developments of the chemistry development kit (CDK) - an open-source java library for chemo- and bioinformatics* paper (DOI:10.2174/138161206777585274). However, while most molecular descriptors had JUnit tests for at least the `calculate()` method, a full and proper module testing was not set up. This involves a rough coverage testing and test methods for all methods in the classes.

So, I set up a new CDK module called `qsarmolecular`, and added the coverage test class QsarmolecularCoverageTest. This class is really short and basically only requires a module to be set up, as reflected by the line:

```
private final static String CLASS_LIST = "qsarmolecular.javafiles";
```

The actual functionality is inherited from the CoverageTest. The coverage testing requires, unlike tools like Emma for which reports are generated by Nightly, a certain naming scheme (explained in *Development Tools. 1. Unit testing* in CDK News 2.2).

Now, testing for a lot of the methods in the IMolecularDescriptor and IDescriptor interfaces are actually identical for all descriptors. Therefore, I wrote a MolecularDescriptorTest and made all JUnit test classes for the molecular descriptors extend this new class. This means that by writing only 10 new tests, with 29 assert statements, for the 45 molecular descriptor classes, 450 new unit tests are run without special effort, making to total sum of unit tests run each night by Nightly for trunk/ pass the 4500 unit tests.

Now, this turned out to be necessary. I count 52 new failing tests, which should hit Nightly in the next 24 hours.