chem-bla-ics

Bioclipse for CDK Developers #1

Egon Willighagen

Published February 15, 2009

Citation
Willighagen, E. (n.d.). In *chem-bla-ics*. chem-bla-ics. https://doi.org/10.59350/zv2f4-ac581

Keywords
Bioclipse, Cdk

Copyright
Copyright © Egon Willighagen 2009. Distributed under the terms of the Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Ola has released the second beta for Bioclipse 2.0. Things are getting along, and I will not go into details on the molecules table Arvid is working on, the 1GB+ SD file support, the validating CML editor , the support for XMPP services , or the brand new welcome page which will guide new users around in what Bioclipse has to offer.

This blog will focus on what Bioclipse has to offer CDK developers.

While Bioclipse 1.x (doi:10.1186/1471-2105-8-59) was a prototype that showed the power if integrating different bio- and cheminformatics tools, Bioclipse2 was designed from scratch, taking advantage of the latest Eclipse RCP technologies. More importantly, the team in Uppsala decided to have all functionality work via managers, allowing all actions to be recorded. *And*, scripting of Bioclipse. I blogged earlier about scripting JChemPaint , and creating UFF optimized 3D structures from SMILES . Example scripts can be found on GitHub (this is their coverage), and are indexed on Delicious.

R for cheminformatics

The fact that we can script everything makes Bioclipse an ideal platform for doing cheminformatics: we have access to a variety of cheminformatics libraries, *and* the means to visualize results via JChemPaint and Jmol. It is like R for cheminformatics: Bioclipse being the R command line, Bioclipse plugins the R packages. Eclipse provides an mechanism called *Update Sites*, which makes something like CRAN redundant. Back to the Chemistry Development Kit.

Over the next weeks, I will blog about scripts aimed at CDK developers and people who want to learn more on how the CDK internals work. This series assumes Bioclipse 2.0 beta2 (or better) and the CDK Feature installed. I'll be using the Gist widget to embed scripts in this blog, but you can always download the Gist directly into Bioclipse, with the GUI as described here .

Bioclipse uses JavaScript (maybe other scripting languages in the future. File a wishlist report if you like to see Jython, BeanShell or other support in the Bioclipse bug track system.) Bioclipse managers are visible using special variables, such as:

Bioclipse Feature	ui	Bioclipse UI interaction
Cheminformatics Feature	cdk	CDK functionality
	jmol	Jmol functionality
CDK Feature	cdx	CDK Developer functionality

Bioclipse scripting has TAB completion support, so you can type cdk. (notice the dot at the end) to which methods the cdk manager provides.

Debugging CDK's Atom Type

As I wrote last week with the email on the first CDK 1.2 release candidate , the new CDK atom typer is a core component of the new CDK. The new implementation covers all atom types used

in CDK 1.0, and many more. In particular, Miguel boosted support for charged and radical atom types.

However, the atom types in your data set may not be covered, or perception fails otherwise. That happens. Bioclipse2 makes debugging of this important step in cheminformatics quite insightful. The following script reads a molecule from SMILES, visualizes 2D diagram in JChemPaint, and perceives atom types: The atom type perception results are return to the JavaScript console, and if there are *nulls* given, then the CDK algorithm did not find a matching atom type for that atom. If you are sure your cheminformatics representation is in order, I welcome a bug report here.

CDK developers can take advantage of this functionality, to eliminate possible causes why a certain algorithm fails. CDK atom typing is used for a variate of algorithms, including counting implicit hydrogens, which many other algorithms need to know.

How does the CDK read a SMILES

A use case for people who want to know if a particular SMILES feature is read or to make sure it is read correctly: This script uses the *diff* functionality introduced in CDK 1.2, and shows two aspects of the SMILES specification: 1. it picked up the isotope information given in the second SMILES; 2. the second SMILES does not include the implicit hydrogen count, which the SMILES specification then defaults as zero.

Summary

The CDK managers in Bioclipse (*cdk* and *cdx*) expose functionality of the CDK, and allows using it in Bioclipse' rich visual workbench environment.