JChemPaint viewer and editor applets: Interactive 2D molecular diagrams

Stefan Kuhn, Tobias Helmus, Egon Willighagen and Christoph Steinbeck
Cologne University Bioinformatics Center, Germany

Introduction

JChemPaint (http://jchempaint.sf.net/) is an open-source 2D structure editor for which recently two applet versions have been developed [1]. One is a smaller viewer-only applet; the other is a larger editor applet. JChemPaint has been under development for some years, going back to 1998 [2]. Over this period it has gone through many changes, e.g. swapping the underlying library to CDK [3] or the full integration into the CDK development. One goal had always been to develop a Java-Applet based on JChemPaint, which should fullfill the following requirements:

- small size
- fast operation
- compatible with a wide range of Java VMs, browsers and operating systems
- provide all functions needed in online-chemistry applications
- drawing of molecules and organic reactions
- several rendering modes

Based on the Chemistry Development Kit (CDK, http://cdk.sf.net/), the applet is quite rich in features, including 2D rendering of molecules and reactions, popup labels on atoms, structure diagram generation (diagram cleanup), optional use of templates, input/output capability including Chemical Markup Language (CML), SMILES, MDL molfile and many others, undo/redo capability, and much more.

The Applet Jars

The latest release is JChemPaint 2.2.1. Its source code uses a stable branch forked from the main development in December 2005. A binary download is available from the CDK page onhttp://sourceforge.net/. This download contains the two applet classes: an Editor for drawing two-dimensional structures and a Viewer for only viewing them.

The download consists of a zip file, which contains 35 jar files plus HTML example code. The jar files form the binaries for the applet. Their size is 6.2 MB, but they will typically not all be needed with additional code fetched on request. The Editor as well as the Viewer can start with jars of less than 1 MB in size. Over a fixed link internet connection this means that the applet starts in around three to fifteen seconds, depending on conditions. For comparison: Marvin (http://www.chemaxon.com/marvin/) is similar to JChemPaint in size, whereas JME (http://www.molinspiration.com/jme/) is only 37 KB large. Once loaded, the speed is no longer influenced by file size, Java VMs also cache the code. In operation, JChemPaint generally proves to be quite fast.

The current release is not signed, which implies it can not perform file system or clipboard operations. For workarounds see the section "Usage of the JCP applet"-section. Interested groups could also get the jars signed by some trusted organisation. The applet is compiled for the Java 1.3 platform, therefore it should run on all Java 1.3 or later (including 1.5) releases. It will not run on earlier Java VMs, noteably not on the Microsoft VM built into older Internet Explorer versions. We tested the applet with a range of current browsers (Mozilla, Firefox, IE, Konqueror) and found no problems.

Embedding in HTML

The prerequisite for linking the JCP applet into a web page is to have all the jars from the binary distribution in a web server directory. To embed the viewer the following HTML-code can be used:

```
<applet name="Viewer"

code="org.openscience.cdk.applications.jchempaint.applet.JChemPaintViewer
OnlyApplet"
  archive="jchempaint-applet-core.jar"
  width="600" height="500">
  <param name="load" value="mdl/big.mol" />
  <param name="atomNumbersVisible" value="true" />
  <param name="background" value="12632256" />
  <param name="detachable" value="true" />
</applet>
```

The archive-attribute must point to jchempaint-applet-core.jar relative to the directory containing the HTML page. All jars should be placed into this directory, even if not mentioned in the HTML code. The param load tells the applet which structure file it should load for diplay. The path to the file must either be relative to the html directory or a complete URL (the file must be publicly available). When detachable is set to true (default: false), the user can detach the applet by double-clicking on it and then resize it. The other parameters are self-explaining.

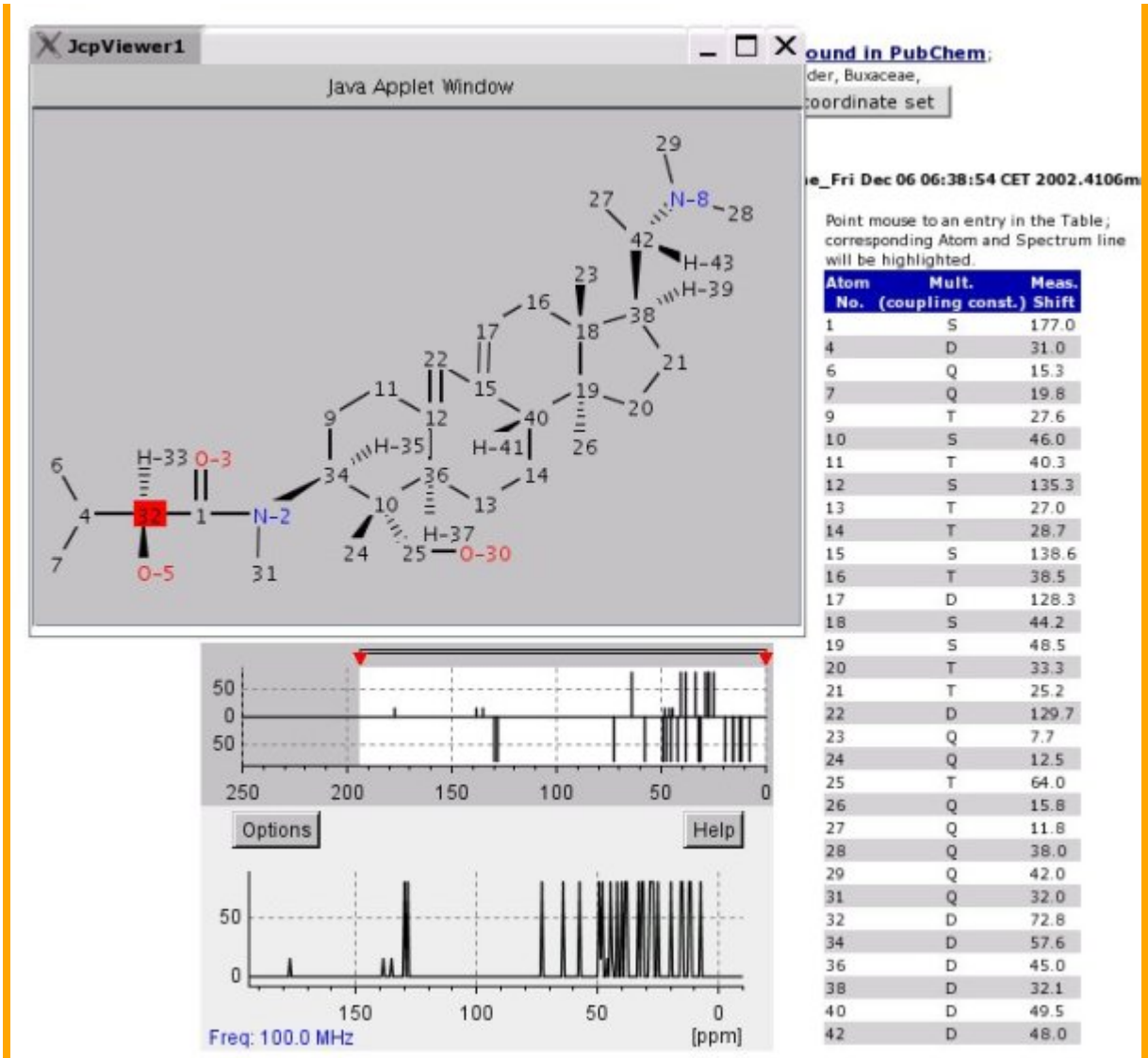

Figure: Screenshot showing the detached JChemPaint Viewer.

JavaScript

Javascript can also be used to dynamically interact with the applet. In the figure below you can see one atom marked in red. This is done dynamically when the user hovers over the table on the right. The code for this would be document.Viewer.selectAtom(6). Viewer is the name paramter in the <applet> tag, 6 the number of the atom to mark (naturally, atom numbers start with 0).

The Editor applet can be included with code like this:

```
<applet

code="org.openscience.cdk.applications.jchempaint.applet.JChemPaintEditorA
pplet"
  archive="jchempaint-applet-core.jar"
  name="Editor"
  width="600" height="500">
  <param name="load" value="a-pinene.mol">
</applet>
```

The code is similar to that starting the Viewer, except for the classname. The Editor applet offers more possibilities for runtime manipulation. Firstly, you can set the structure via JavaScript as a molfile. This can be achieved by:document.Editor.setMolFile(thisBox.value). Edit or is the name of the applet, thisBox could be for example a text box. Another possibility is to upload a file to the server as part of a HTTP POST request, read this file on the server and display it. Both ways, you avoid the restrictions of the sandbox. Another restriction stemming from the sandbox architecture of the VM is the restriction not to use copy/paste (clipboard) functions on the client side. Thus, the user can display a molecule SMILES via Report->Generate SMILES but he will be unable to copy it into the buffer (it seems this works on some VMs). This might lead to the situation that the user draws a (complicated) structure, but cannot get it out of the applet, since he can neither save nor copy it. Instead, one can use the following code also be included:

```
function showmol(mol) {
  newwin = window.open("nmrshiftdbhtml/showmol.htm",
    "Fenster1", "width=310,height=400,left=0,top=0"
  );
  newwin.document.write("<html><body><pre>" +
    mol + "</pre></body></html>"
  );
}

<a href="javascript:showmol(document.Editor.getMolFile())">Show
editor content as mol file</a>
```

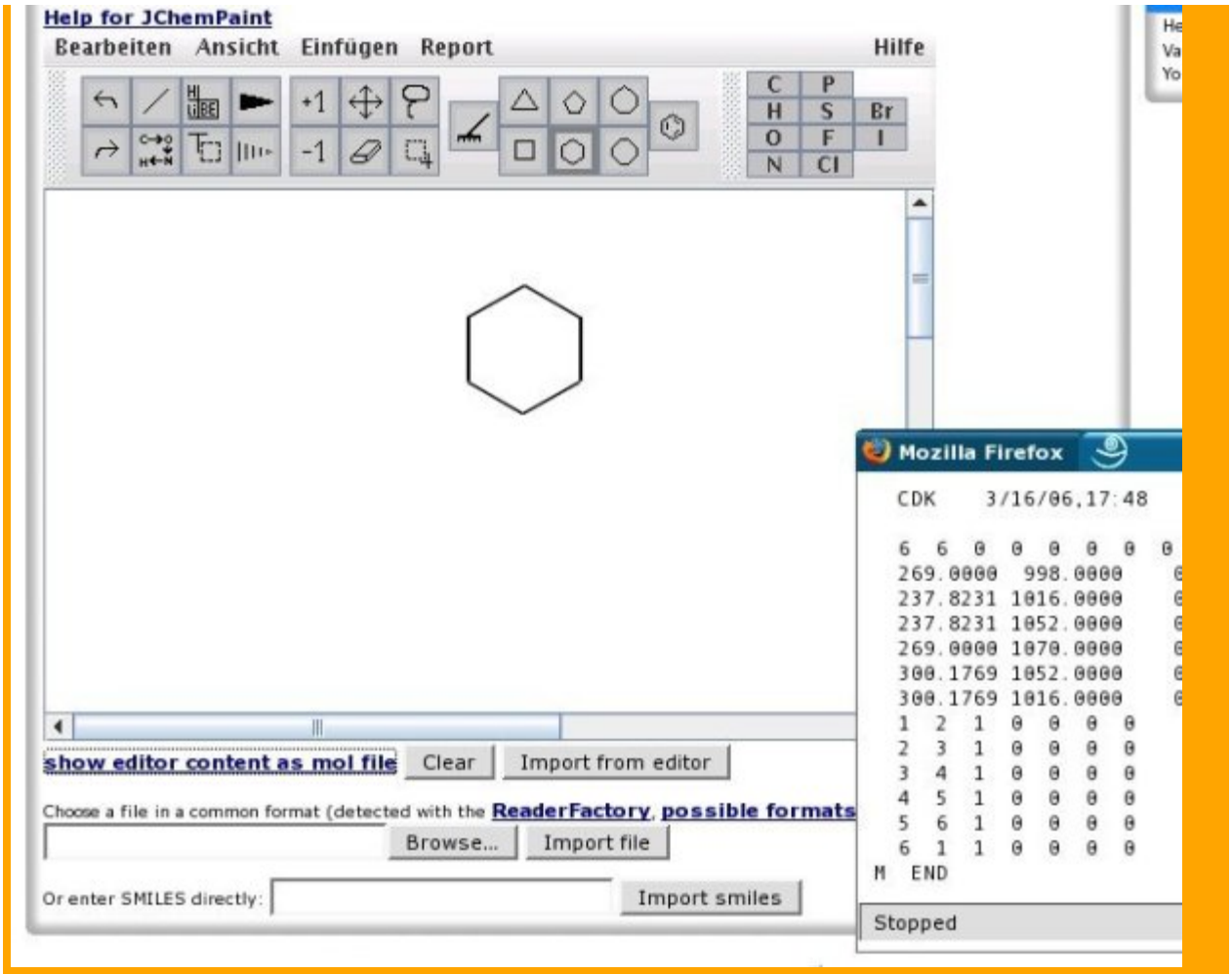

Figure: The MDL molfile content of the applet in a popup window.

Examples

This section will show three applications where the Viewer and Editor applets of JChemPaint are put to use. The first example shows a module from a drug design tutorial one of the author participated in (EW). In this example the user designs a new ligand for a given binding site, with much prior knowledge.

Drug Design Module

This example shows a module from a drug design tutorial one of the author participated in (EW). In this example the user designs a new ligand for a given binding site, with much prior knowledge. It resembles to situation where a new molecule is designed and synthesized and the binding affinity is experimentally determined, though that step is replaced, in the example, by a virtual screening calculation. Calculating of the affinity is done on the server side and the molecular structure is extracted from the Editor applet.

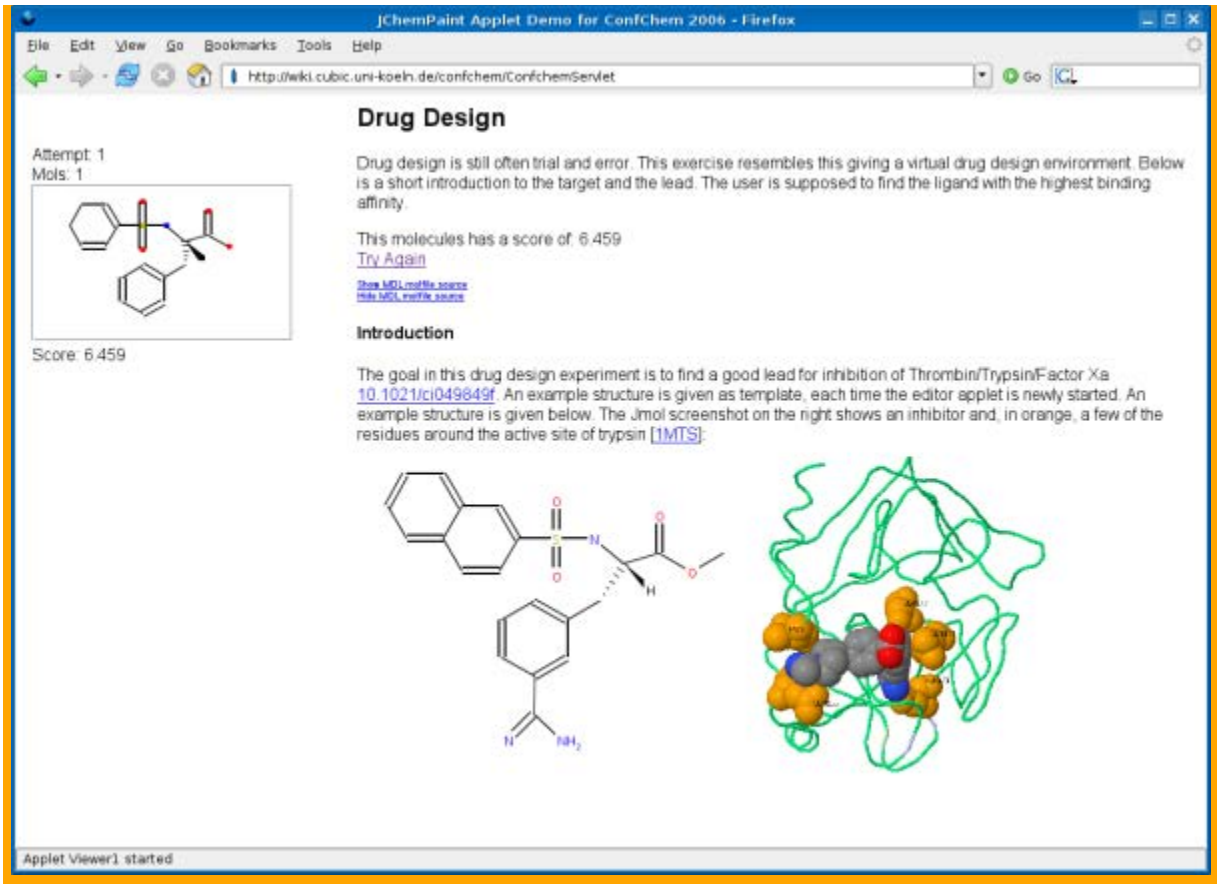

Figure: Screenshot showing the drug design module with JChemPaint in reduced rendering mode on the left [Online module].

NMRShiftDB

NMRShiftDB is a transparent structure-property database for the storage and retrieval of small organic molecules and their NMR chemical shift data [4]. The software has reached a stable state and has successfully been operating over the last few months (http://www.nmrshiftdb.org). It is intended to grow into a general spectroscopic information system by extending it to store other types of spectroscopic data, like mass and infrared spectra.

The database provides the following major functionality:

- Spectra and subspectra similarity searches
- Structure-, substructure- and structural similarity searches
- Prediction of NMR spectra using machine learning approaches
- Interface for peer-reviewing submitted data for quality assurance

And it also offers various other, non-spectrum related search facilities, on e.g. the chemical name, formula, molecular mass etc.

The first stable release of NMRShiftDB was released in November 2003 on http://www.nmrshiftdb.org. At the time of writing, the system's extent is characterized by 17921 structures, 20230 spectra (most of them 13C spectra, some 1H and a few 31P) and almost 700 registered contributors. A standalone client is being developed to aid in collecting data locally and contributing them to NMRShiftDB later. This tool will allow for easier contribution and a more convenient assembly of private, specialized or inhouse collections.

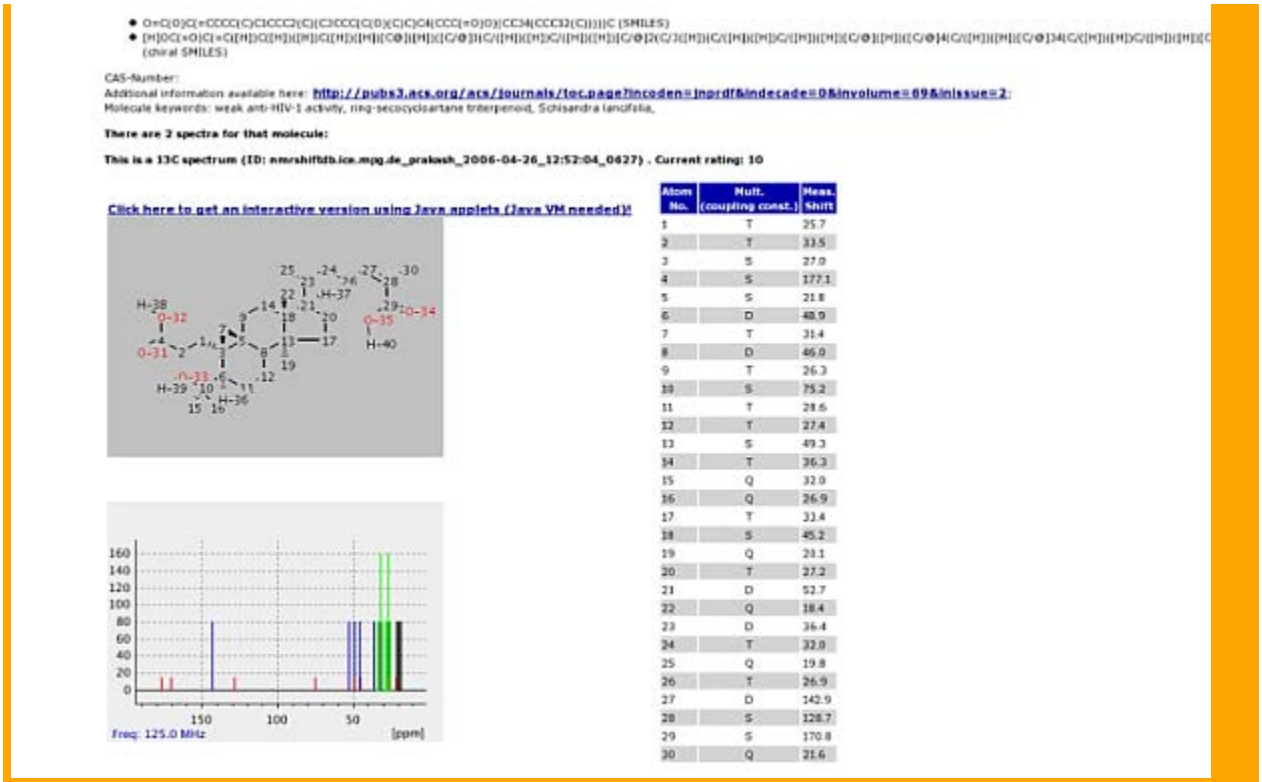

Figure: Screenshot showing NMRShiftDB's details page of the molecule lancifoic acid A in the JChempaint Viewer applet with atom numbering turned on.

www.chemistry-development-kit.org

This website has been set up to showcase some CDK functionality. Since CDK is a library, it, in general, does not provide full fledged applications. As a result it is sometimes difficult to demonstrate specific features to potential users. For providing exactly this http://www.chemistry-development-kit.org/was developed [5]. It also can be used as a structure workbench and for teaching purposes in the field of chemoinformatics. Currently the functions used are file I/O, structure analysis and 2D/3D rendering. The structure of the webpage allows the user to upload a molecule in a format by cdk and then is able to apply a variety of functions (e.g. the calculation of QSAR descriptors, general molecular properties...) to this molecule. The results will then be displayed on the webpage in a structured and consise manner.

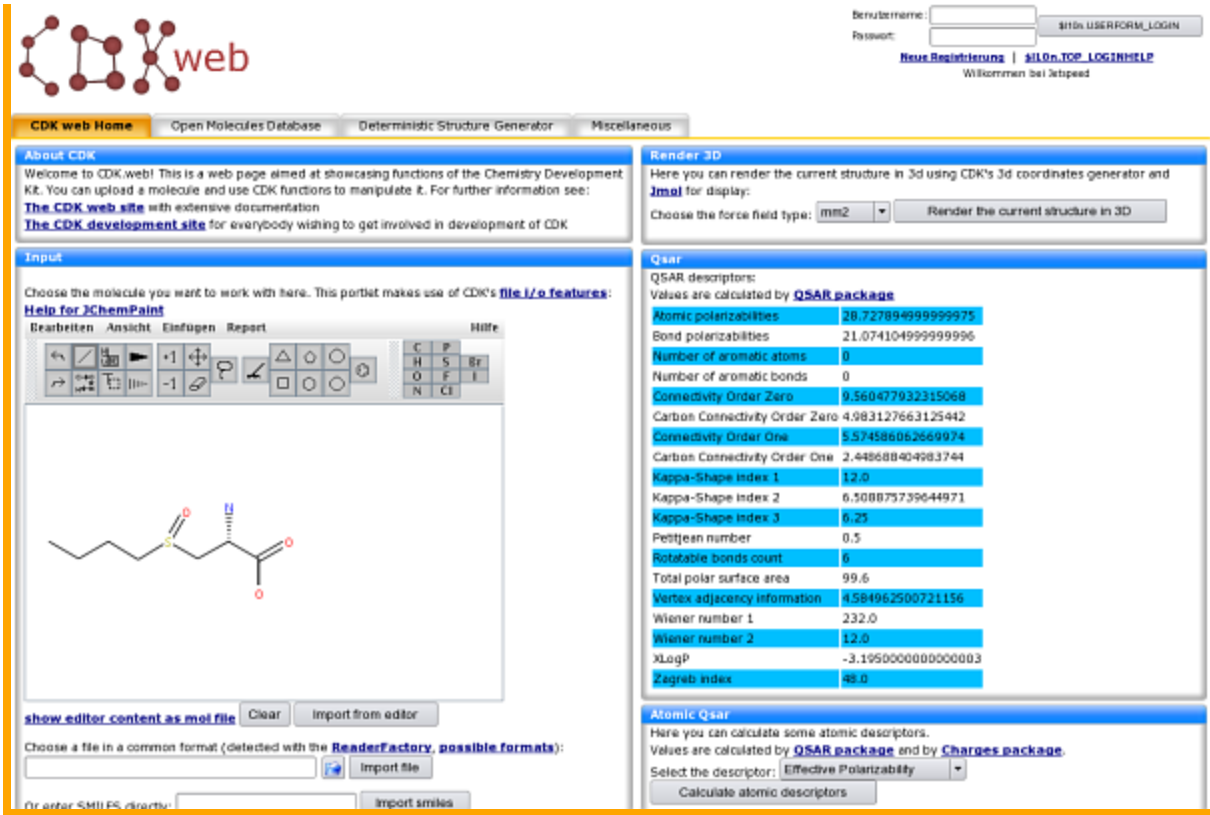

Figure: Screenshot of the chemistry-development-kit.org webpage showing the JChemPaint Editor applet and calculated QSAR descriptors.

Conclusion

This poster briefly introduces the JChemPaint applet. Development relies on contributions of scientists, educators, and programmers whos primary focus is on something other that software development. Consequently, there is room for improvement. The open source nature of JChemPaint and the CDK make this possible. The discussed examples show, that the applet is ready for mainstream applications.

References

1. S. Kuhn, An Applet Release of JChemPaint, **2006**, *CDK News*,3(1):21-23, http://almost.cubic.uni-koeln.de/cdk/cdk_top/cdk_news/

2. S. Krause, E.L. Willighagen and C. Steinbeck, JChemPaint - Using the Collaborative Forces of the Internet to Develop a Free Editor for 2D Chemical Structures, **2000**, *Molecules*, 5:93-98,http://www.mdpi.org/molecules/html/50100093.htm

3. C. Steinbeck, Y. Han, S. Kuhn, O. Horlacher, E. Luttmann, and E. Willighagen, The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics, **2003**,*Journal of Chemical Information and Computer Sciences*,43:493-500, DOI: 10.1021/ci025584y

4. C. Steinbeck, S. Kuhn, and S. Krause, NMRShiftDB - Constructing a Chemical Information System with Open Source Components. Journal of Chemical Information and Computer Sciences, **2003**, *Journal of Chemical Information and Computer Sciences*, 43(6):1733-1739,
5. S. Kuhn, Extending www.chemistry-development-kit.org, **2005**,*CDK News*, 2(1):14-16, http://almost.cubic.uni-koeln.de/cdk/cdk_top/cdk_news/