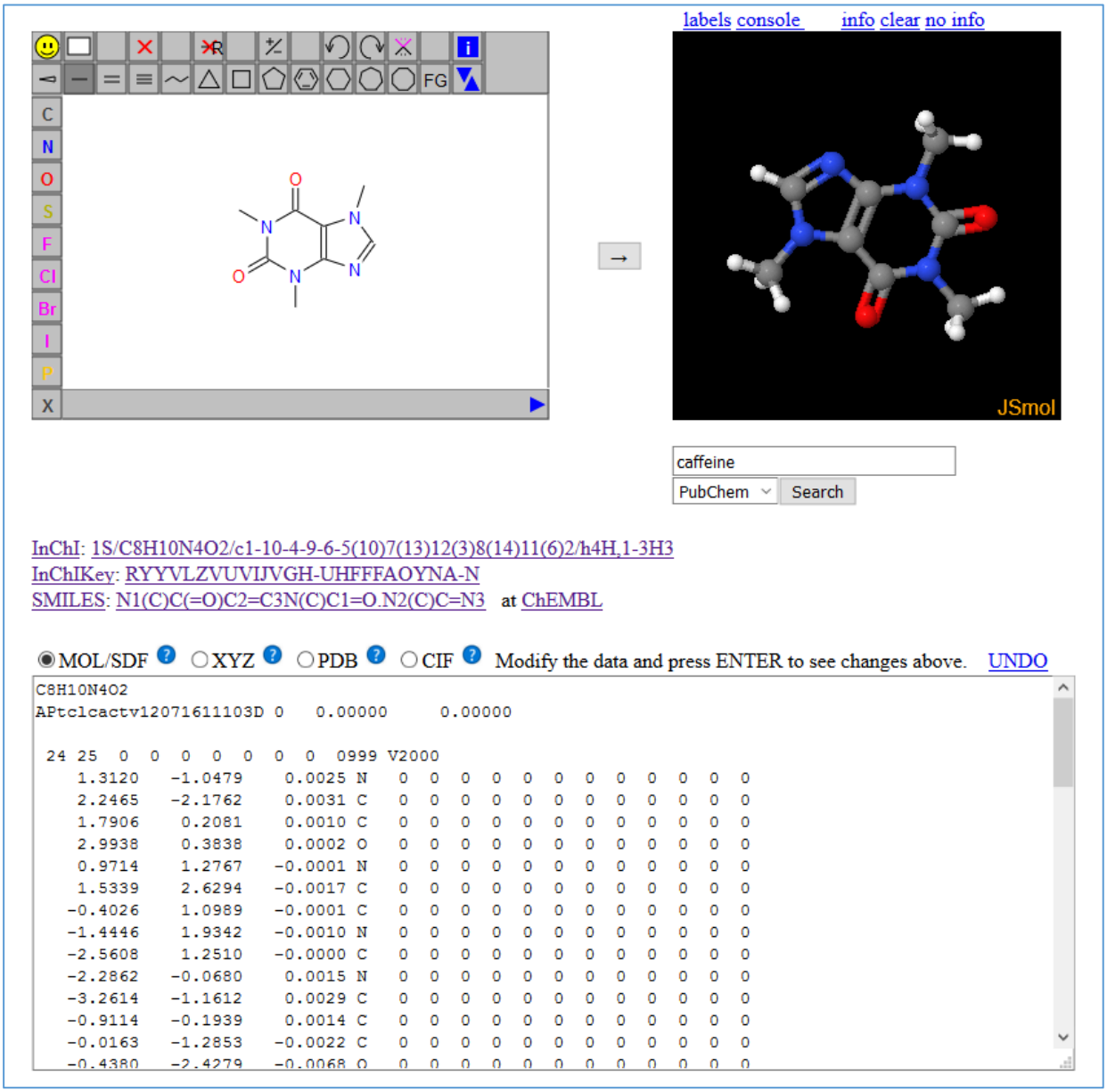
**Hack-a-Mol -- How It Works**

Hack-a-Mol (https://chemapps.stolaf.edu/jmol/jsmol/hackamol.htm) was written by Bob Hanson on a cold December evening in 2016 when he had nothing better to do. The HTML5 JavaScript application utilizes two modules – a 2D drawing application (JSME, http://peter-ertl.com/jsme) and a 3D application (JSmol, https://sourceforge.net/projects/jsmol/). Most of the action on the page involves JSmol.




**2D to 3D**

When the user draws a structure in the JSME application and presses the → button, JSME delivers its structure in the form of a SMILES string. This SMILES string is passed to JSmol using a LOAD command, shown here for acetone:

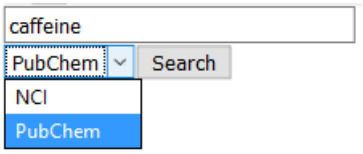
*LOAD $CC(O)C*

The "$" instructs JSmol to make a request to the NCI Chemical Information Resolver (CIR) in Frederick, Maryland, where the SMILES string is processed, and a 3D MOL file is returned. The call to the CIR looks like this (found in the JavaScript console accessed through the info link):

https://cactus.nci.nih.gov/chemical/structure/C(C)(=O)C/file?format=sdf&get3d=true

*Note: In some complicated cases, CIR cannot convert the model to a 3D file, and a 2D file is returned. This can result in an inappropriately flat structure being displayed in the JSmol window.*

**Chemical name to 3D**




The box under the JSmol window allows entry of a chemical identifier – a chemical name, a SMILES string, an InChI string, a CAS registry number, for instance. When the Search button is pressed, either the CIR or PubChem is sent this information, and a 3D mol file is returned. The call is the same as for 2D to 3D, above for the CIR; for PubChem it looks like this:

https://pubchem.ncbi.nlm.nih.gov/rest/pug/compound/name/acetone/SDF?record_type=3d

**3D to structure data**

JSmol has been initialized such that when a model is loaded, the page's JavaScript function *loadStructCallback()* is fired.

```
var loadStructCallback = function(app,filename) {
  if (!filename)
   return;
  setFormat();
  if (!from2D)
   to2D();
  showSmiles(true)
}
```

When this function runs, the function *setFormat()* runs. This function loads the textarea with the model structure, obtained from JSmol with a call to its SHOW("file") Jmol scripting method (if the file is the same type as selected using the radio buttons) or the WRITE(type) Jmol scripting method (if not).

**3D to 2D**

We need to maintain both a 3D and a 2D representation when a 3D model is loaded. In order to do that, we again tap into the CIR at NCI. This is done in the *to2D()* function, which simply runs a built-in function in JSmolApplet.js *Jmol.show2d(jmol, true)*. That function ultimately sends the following script command Jmol:

*select visible;show chemical "file?format=jme*"

which then sends the following command to the CIR, again using a SMILES string to communicate:

https://cactus.nci.nih.gov/chemical/structure/C(=O)C/file?format=jme

The return from the call is a 2D format that JSME can read and display.

**3D to SMILES, InChI, and InChIKey**

The third function executed from loadStructCallback(), *showSmiles()*, issues three Jmol scripting methods:

*{visible}.find('SMILES/noaromatic')*

*show('chemical stdinchi')*

*show('chemical inchikey')*

and deposits the results of those requests into the appropriate spots on the page. The first of these is handled within JSmol itself. Note that we turn off aromaticity, because the lower-case aromatic SMILES syntax can cause problems in communication, as different programs have different definitions of what "aromatic" means. (See *Jmol Smiles and Jmol Smarts: Specifications and Applications,* https://jcheminf.springeropen.com/articles/10.1186/s13321-016-0160-4.) The other two methods (again) involve calls to the CIR:

https://cactus.nci.nih.gov/chemical/structure/C(C)(=O)C/stdinchi
https://cactus.nci.nih.gov/chemical/structure/C(C)(=O)C/inchikey

**Structure Hacking**

When the user modifies the structure (or pastes into the textarea some sort of structure file data) and presses ENTER, the jQuery *keydown* event is caught and sent to this method:

```
$("#strucfile").keydown(
   function(e) {
     if (e.keyCode != 13)return;
     from2D = false;
     loadMol();
     return false;
   }
 );
```

The ENTER key event (keyCode 13) is trapped, and the *loadMol()* method is executed. This method checks to see if the data is 2D mol data (the characters "2D" starting at column 21 on the second line of the file) or 3D data (anything else), and then passes the data either to the appropriate module -- JSME, using *Jmol.jmeReadMolecule(jme,s)*, or JSmol, using the LOAD DATA scripting syntax:

*load data 'mydata'…end 'mydata'*

**Summary**

That's pretty much all there is to it. The key is that messages are being sent behind the scenes between the page and various servers – primarily the NCI Chemical Information Resolver – in order to convert various structure data types -- 2D mol, 3D mol, SMILES, InChI, etc. – into the needed data type. The code on the page is not complex. It simply taps into a set of functions built into JSmol to good effect.

Bob Hanson, 2016.12.08 https://stolaf.edu/people/hansonr