

## What do molecular modeling programs do?

Molecular modeling programs visually render molecules in 3D from an array of spatial coordinates stored in a standard file type (ex: .pdb). These files are created when structures of macromolecules have been resolved via crystallography or NMR. These coordinate files are then submitted to a database (e.g. Protein Data Bank) or further processed by advanced applications.

As you become familiar with PDB files (and other coordinate file types), it is important to note that *very little* structural information is contained within these files. The atoms are labeled and x,y,z coordinates are supplied, but it is usually up to the modeling program (and the user) to decide how to render the bonds, surfaces, and other stylized representations of the molecules.

There are many different molecular viewing programs, and what differentiates them is the way in which they interpret, process and display the features of a macromolecule. Some programs also offer tool sets to manipulate and save modified structures. Still others are used to animate or perform advanced computations involved in single-molecule kinetics.

Many of these programs run as “standalone” applications that are downloaded and installed on your computer. However, this means that the molecular coordinate file must also be present locally. This is inconvenient for those wanting to share a large repository of information or for those who wish to use molecular viewing for educational purposes. This is Jmol's specialty.

## What is Jmol?

Jmol is a free, open source molecule viewer for students, educators, and researchers in chemistry and biochemistry. Jmol is a successor to such legacy products as XMol, Rasmol, and the Chime plugin.

Because Jmol's core rendering engine was written in Java, it is cross-platform, running on Windows, Mac OS X, and Linux/Unix systems. However, v1.4.2 (or higher) of Sun Java must be installed. See below for installation instructions.

The class will use the JmolApplet implementation for structure-based homework. The applet is generally stored on a remote web server (e.g. Protein Data Bank) and downloaded whenever a user visits a web page that integrates Jmol. At this point in time, the applet is unsigned or “untrusted”, meaning that it runs in a protected environment and cannot affect your machine.

The Jmol *application* is an optional download from the Jmol development web site. It is a Java-based “wrapper” around the Jmol rendering engine that can be run locally from the desktop.

## How do I install Jmol?:

**Step 1:** Make sure that the latest version of Sun Java is installed on your operating system.

The easiest way to check is to visit <http://www.java.com>, click on the link to “download the latest Java software”, and follow the instructions. If you already have the latest version, the installation will terminate. If not, make sure that you install Java, including extensions for the browser(s) you plan on using to view molecules.

**Step 2:** Visit <http://www.jmol.org> to test your Java installation and the Jmol applet.

If the main graphic on the front page is displayed, installation was successful. That's it! Remember, Jmol doesn't actually exist on your machine as an application, so there is nothing else to install or configure.

## How do I use Jmol?:

When you visit a website that contains the JmolApplet, the molecule will load within a square viewing area. Clicking and dragging the mouse will rotate the molecule, while Shift + Drag will zoom in and out.

Jmol has several simple commands coded into right-click context menus. While fine for generic viewing and browsing of molecules, these commands are limited in scope. To use select and script efficiently, you will need to revert to the Jmol Console; it is often much easier to use the console exclusively.

Jmol does not alter coordinate files in any way. However, depending on what commands or scripts are run, you can alter the way in which Jmol interprets, displays, and labels various components of the macromolecule. To this end, Jmol provides a set of commands that are similar to Rasmol. These commands may be issued individually or in groups and are entered through the Jmol Console (see below).

The documentation for these commands can be found at Jmol Interactive Scripting web site:

<http://www.stolaf.edu/people/hansonr/jmol/docs/>

Although some of the information is rather technical, most definitions have links to example web pages designed to show you their effects. Commands can be grouped into 4 main functions:

- |                                    |  |
|------------------------------------|--|
| <b>Selecting Entities</b>          | <b><i>select</i></b> is perhaps the most important command to master when using Jmol. An entity such as an atom, chain, helix, etc. must be selected before any changes can be made to its display. The <b><i>select</i></b> command has an extensive format with many different variables or flags. These can be used to tell Jmol exactly which item or sets of items you want to manipulate. Learning how to use this command involves understanding how Jmol groups atoms. |
| <b>Altering Displays</b>           | Once you have selected a group of atoms, these commands alter their rendering within Jmol; changes can be made to atom radii, bond representations, colors, labels, etc. Most of these commands have a range of acceptable input values, so displaying a certain characteristic is usually not limited to on/off. This type of command makes up the bulk of Jmol's tool set and they are too numerous to list here; please refer to the online script doc.                     |
| <b>Altering Frame of Reference</b> | These commands are used to alter the position of the molecule "camera" through translation, rotation, and zooming. Jmol can also be instructed to animate changes in position, providing smooth motion for automated scripts.  |
| <b>Jmol Environment</b>            | These commands are used to input and extract information from the Jmol program. These include the <b><i>set</i></b> command for setting environment characteristics, advanced scripting commands (e.g. <b><i>loop</i></b> , <b><i>define</i></b> , <b><i>script</i></b> ), and <b><i>show</i></b> commands, which output information to the Jmol Console   |

### **Can I save my work?**

As mentioned before, Jmol does not modify the coordinate file, so there is no simple method to "save" your particular rendering. However, the commands used to manipulate the molecule may be placed into a flat text file, called a script. Scripts are lists of commands that can be pasted into Jmol console or read directly using the ***script*** command. Each line is processed sequentially until the end of the file is reached. Scripts are composed of commands universal to most Jmol versions and can be exchanged freely or linked to the applet through web page programming.

### **Links:**

Jmol Development – note the series of links to various resources on the top right of the page:

<http://www.jmol.org>

Jmol Interactive Scripting Doc (the bible):

<http://www.stolaf.edu/people/hansonr/jmol/docs/>

University of Arizona BIOC 462a lecture scripts:

<http://www.biochem.arizona.edu/classes/bioc462/462a/jmol/routines/routines.html>

Protein Data Bank (updated as of 1/1/2006):

<http://www.pdb.org>

For those interested in the Jmol.js library and scripting with the JmolApplet:

<http://jmol.sourceforge.net/jslibrary/>

<http://jmol.sourceforge.net/demo/>

For more information on the nature and source of 3D modeling data:

[http://pd-beta.rcsb.org/pdb/static.do?p=general\\_information/about\\_pdb/nature\\_of\\_3d\\_structural\\_data.html](http://pd-beta.rcsb.org/pdb/static.do?p=general_information/about_pdb/nature_of_3d_structural_data.html)