

Python Molecular Viewer

Written by Ruth Huey, Sargis Dallakyan and Michel Sanner

*The Scripps Research Institute
Molecular Graphics Laboratory
10550 N. Torrey Pines Rd.
La Jolla, California 92037-1000
USA
13 July 2007*

Contents

| | |
|--|-----------|
| Contents | 2 |
| Introduction | 4 |
| Before We Start... .. | 4 |
| FAQ - Frequently Asked Questions | 5 |
| Exercise One: Getting Started: PMV Basics | 8 |
| Procedure: | 9 |
| Summary: what have we learned? | 16 |
| Add-On: Binding Commands to Keys | 16 |
| Hemolysin: Secondary Structure colored by Chain..... | 17 |
| Procedure: | 17 |
| Summary: what have we learned? | 20 |
| Add-On: Color by Secondary Structure | 21 |
| Add-On: Measure hemolysin beta barrel..... | 21 |
| HIV Protease: Active Site Residues and Inhibitor | 22 |
| Procedure: | 23 |
| Summary: what have we learned? | 25 |
| Add-On: Using the DejaVu GUI | 25 |
| Add-On: Creating an mpeg video file | 26 |
| HIV Protease: Binding Site Surface Colored by Property | 28 |
| Procedure: | 28 |
| Summary: what have we learned? | 32 |
| Add-On: Intermolecular Hydrogen bonds | 33 |
| Add-On: Displaying Interactions | 34 |
| HMG-I(Y)-DNA: Secondary Structure | 36 |
| Procedure: | 36 |
| Add-On: Input files for 3D Models | 37 |
| Postscript: Calculating electrostatic potential using Adaptive Poisson Boltzmann Solver (APBS). | 39 |
| Procedure: | 39 |
| Procedure: | 41 |
| Appendix 1: Advanced Display Control via the DejaVuGUI. | 46 |
| Procedure: | 47 |


| | |
|---|-----------|
| Summary: what have we learned? | 54 |
| Appendix 2: Selecting a subset..... | 55 |
| Procedure: | 55 |
| Summary: what have we learned? | 59 |
| Appendix 3: Working with MolKit in PMV's Python shell | 61 |
| Appendix 4: PMV StartUp Options | 64 |
| Appendix 5: PMV Modules..... | 66 |
| Appendix 6: Mouse Modifiers and Keystrokes recognized by the DejaVu Viewer | 71 |
| Appendix 7: Customization Options and Default Values for User Preferences..... | 72 |
| Customization Options for PMV | 72 |
| user_prefs defaults: | 72 |
| Appendix 8: Definitions of Named Residue and Atom Sets recognized by PMV | 74 |


Introduction

This tutorial will introduce you to **PMV**, which is short for Python-based Molecular Viewer. It is a general purpose, OpenGL-based viewer which supports interactively viewing molecules. It provides an extensible set of tools for displaying and editing molecules.

Before We Start...

Since PMV is distributed as a part of MGLTools, we need to download and install MGLTools first. Go to our website: <http://mgltools.scripps.edu>, click on “Downloads”, and download current release (1.4.5) of MGLTools corresponding to your operating system.

| | | |
|---|--|------------------------------|
|  | <ul style="list-style-type: none">■ Unix/Linux (21MB) | Instructions |
|  | <ul style="list-style-type: none">■ Mac OS X 10.4 - Intel (25MB)■ Mac OS X 10.4 - PowerPC (27MB)■ Mac OS X 10.3 - PowerPC (26MB) | Instructions |
|  | <ul style="list-style-type: none">■ Windows (13MB) | Instructions |
|  | <ul style="list-style-type: none">■ Source: All Platforms | Instructions |

Please read “Instructions” for information on how to install MGLTools. Once MGLTools is installed, you can start PMV either by clicking on the PMV icon -  - or by running the “pmv” script from the command line*.

* People attending the tutorial given at TSRI can also use PMV installed on TSRI network:
/mgl/prog/share/bin/pmv14

FAQ – Frequently Asked Questions

1. *Where can I find the download site for **PMV**?*

<http://mgltools.scripps.edu> under the Downloads link...

2. *Where should I start **PMV**?*

If you are working on a computer on the network at TSRI, you can start PMV in any directory. See instructions of page 4. If you are using a local installation of PMV, start it according to the instructions which came with it.

3. *How do I customize PMV according to my personal preferences?*

PMV will source a preference file, “_pmvrc” in the .mgltools/###/Pmv directory located in your home directory where ### is the number of the version you are running. The preference file can contain a list of PMV commands to be executed when the Viewer starts. For example, it is possible to set the font type and size in the _pmvrc. If you prefer to work with a ‘docked’ camera, you can set this in the _pmvrc. You can further customize PMV using a second _pmvrc file located in the directory where you start PMV. If another _pmvrc exists in the directory where you start PMV, it will be sourced after the one located in your home directory/.mgltools/###/Pmv .

Also, it is possible to customize PMV on the fly. PMV uses a dictionary of user preferences which can be set interactively. It is accessed from PMV via:

File → **Preferences** → **Modify Defaults**

4. *What documentation is available?*

<http://mgltools.scripps.edu> under the Documentation link has links to various forms of information such as Latest FAQs, Latest How-tos, Latest Tutorials, Latest Glossary Definitions among others. Documentation in the code itself is available in PMV under the **Help** button. For citations, you can use the citation command in the **Help** menu which gives this reference for PMV: Michel F. Sanner. Python: A Programming Language for Software Integration and Development, *J. Mol. Graphics Mod.* **17**, 57-61 (1999).

5. *Where can I get help finding specific commands?*

Descriptions of specific modules and the commands they contain can be accessed in PMV itself via Help->Commands Documentation.

Also, you can search for commands in PMV via Help->Search For Commands. This lets you find commands from strings you enter. For instance, searching on “closecontacts” reveals that a command “checkForCloseContacts” exists in the repairCommands module.

6. *How do I report bugs?*

Bugs can be reported directly from PMV via Help->BugReport. A widget opens which lets you immediately report any problems you encounter with PMV. Bugs can also be reported via **MGLBuzilla**, our bugs and issues tracking system (mgldev.scripps.edu/bugs). Before entering a new bug, search MGLBugzilla to see if the bug you have encountered has already reported. In some cases the bugs have also been fixed. If so, it may be possible to obtain a corrected module from the **cvs** repository. (cvs stands for Concurrent Versions System)

7. *How can I access the cvs repository?*

This is covered in the following How-to: <http://mgltools.scripps.edu/documentation/how-to/anonymous-access-to-our-cvs-repository>

8. *Whom do I contact with ideas/suggestions for added functionality?*

This can be done using MGLBuzilla or by sending email to mgltools@scripps.edu.

9. *Is there a mailing list for Pmv?*

Yes. Send messages to pmv@scripps.edu. To subscribe to this list go to mgldev.scripps.edu/mailman/listinfo/pmv. Type in your email address and name (optional) and choose a password, then click “Subscribe”.

10. *Can I view molecules in stereo in PMV? How can I change the stereo separation distance?*

Yes. You can switch from mono to stereo by clicking on the **mono/stereo** checkbox on the PMV GUI. It is also possible to adjust the stereo separation distance using a visual programming environment (**Vision**) network. To do

so, start a Vision network editor by clicking on the checkbutton labeled “Vision” on the right edge of the PMV GUI. In the network editor which opens, load the **3D Visualization** library. Find **StereoSep** in the macros panel and drag it onto canvas. Right click on this StereoSep node to display a list of possible actions and choose **Expand Node**. In the expanded representation of this node, you can directly access a **Dial** which lets you adjust the stereo separation distance interactively. Click on the Dial and drag to adjust the separation distance.

Exercise One: Getting Started: PMV Basics

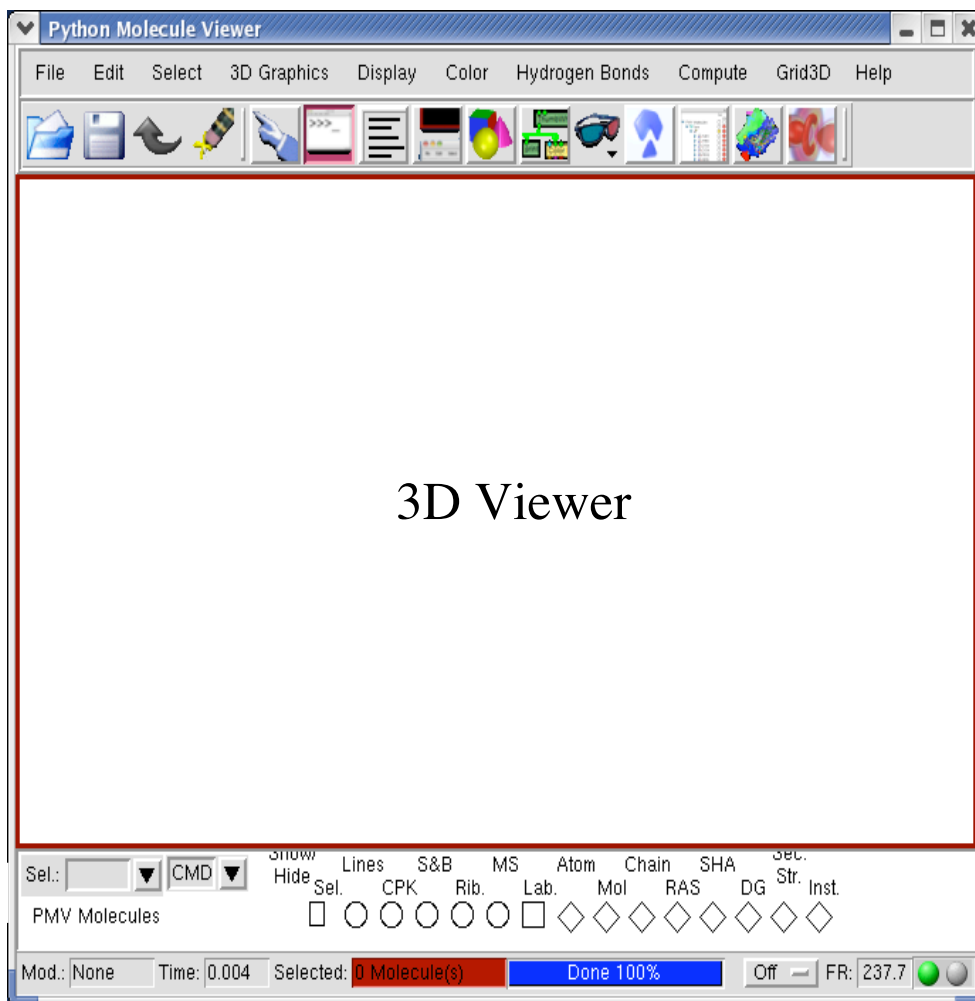
This exercise covers five basics of using **PMV**:

- loading and interacting with molecules
- graphical representations of molecules
- executing commands
- picking objects in the Viewer
- starting pmv with options

When you start PMV, a docked camera and a control panel open. Place the cursor over icons in Tool bar for more info.

Menu bar

Tool bar




Dashboard widget

Info bar

Note: You can submit bug reports directly from the **Help** menu in PMV. If you encounter any bugs today, please submit them! If you have any suggestions or feature requests, we'd like to hear about them, also...

Various menubuttons, such as **File**, **Edit**, **Display** and **Help**, are located on the **Menu bar**. Icons on the **Tool bar** provide quick access to most commonly used functions such as:

| | | | |
|---|------------------|---|--------------------|
|  | Read Session |  | Float Camera |
|  | Save Session |  | DejaVu GUI |
|  | Undo |  | Vision |
|  | Clear Selection |  | Toggle Mono/Stereo |
|  | Picking Commands |  | AutoDockTools |
|  | Python Shell |  | Cartoon Rendering |
|  | Message Box |  | Volume Rendering |

Note: Some keys modify the picking event. That is, if you hold down one of these **modifier** keys while you click on the left mouse button, a different type of picking event occurs which can be linked to a different command. The keys recognized by the Viewer are listed in Appendix 6.

The **PCOM bar**, which is displayed and activated by clicking on the Picking Commands icon, lets you bind commands to picking events. A picking event occurs when you click on the left mouse button while the cursor is over a geometry in the Viewer. The **Dashboard** widget lists all molecules currently loaded in PMV and provides quick access to various PMV commands. The **Message Box**, which is displayed below the Viewer when the Message Box icon is clicked, displays the log of recently executed commands. **The Info bar** contains information about the current keyboard modifier to picking, the time used for the last command, the number of entities in the current selection, a menu used to set automatic movement of the objects in the Viewer, information about frame rate and the progress of the current operation.

Note: Commands available via the Dashboard can also be invoked using the menus at the top of the PMV GUI. Here you could use this menu sequence to open the **file browser**:

File -> **Read Molecule**

Note: Instead of using the mouse to click on the **Open** button in the GUI, you could also press the <Enter> key on the keyboard while the cursor is still in the entry. This is true for many parts of the GUI in PMV, but not for all of them....

Procedure:

1. Load a Molecule:

A. Position the cursor over **PMV Molecules** in the Dashboard and click with the right mouse button.

This will open a file browser listing all the files in the current directory which have extensions in the list of input types we support at present. These types include 'cif', 'mol2', 'pdb', 'pqr', 'pdbq' among others.

B. Select **hsg1.pdbqs** and click on **Open**. This results in loading the molecule **hsg1** into PMV and displaying covalent bonds between its atoms as lines. By default, bonds are built based on the distances between atoms and are represented as lines while non-bonded atoms, such as

Note: You can change, add or remove commands which are applied to new molecules via:

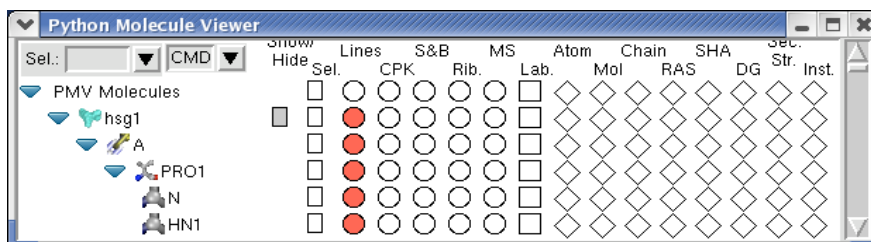
File → **Preferences** → **Set Commands to be Applied on Objects** Click on the button for **colorByAtomType** to display new molecules colored by atom type....

metal ions and oxygen atoms of water molecules, are shown as small 3D crosses.

2. A tour of the Dashboard Widget:

Select entry + Command Buttons →

Tree Widget



Note: Clicking on a shape - rectangle, circle, square or diamond - under a command causes the command linked to the shape to be applied to each node in the corresponding row. If the shape is off (colored white), the command will be applied to nodes and the shape will be colored red. If the shape is on (colored red), clicking on the command button will undo the command and the shape will be colored white. Circles are used for **display** commands, squares for **label** commands and diamonds for **color** commands. Coloring can be replaced by a different coloring scheme but cannot be undone. The gray rectangle is used for **show/hide** and the white rectangle for **select**.

The **Tree Widget** on the left lists all molecules currently loaded in PMV. Click on the arrows to navigate between molecules, chains, residues and atoms. Clicking on a **shape** in one of the columns in the right section executes the PMV command corresponding to the label at the top of the column on the group of nodes corresponding to the row. There 16 different commands that can be executed this way - gray rectangle(Show/Hide), select/unselect (**Sel.**), display lines(**Lines**), display CPK (**CPK**), display sticks and balls (**S&B**), display secondary structure (**Rib.**), display molecular surface (**MS**), display labels (**Lab.**), color by atom type (**Atom**), color by molecule (**Mol**), color by chain (**Chain**), color by residue according Rasmol (**RAS**), color by residue according Shapely (**SHA**), color according to David Goodsell colors (**DG**), color by secondary structure element type (**Sec.Str.**) and color by instance (**Inst.**).

To help users see the connection between molecular fragments and PMV commands, a crosshair is drawn when cursor is inside the Dashboard widget.

Right-clicking on a shape displays an input parameter panel for the command and allows the user to customize specific input parameters for the command.

Note: A **selection in the Tree** is used to build a group of nodes to be the target for commands linked to shapes. It is not the same as the current selection in the Viewer. It can be selected using the appropriate rectangles....

The **Sel:** entry in the top left corner of the Dashboard can be used to select entries in the Tree using a Pmv compound selector. Nodes matching the specified string will be selected. Selected nodes are outlined with a yellow selection box. When a shape is clicked for a selected node, the corresponding command is applied to all currently

selected nodes. Hovering over this entry shows samples of the required syntax.

The option menu on the top allows the user to specify whether commands should be applied to the backbone atoms only (**BB**), the side chain atoms only (**SC**), the sidechain atoms and **CA** atoms (**SC+CA**) or the full molecular fragment (**ALL**). This setting can be overridden for each column (**CMD**).

Click on the gray rectangle under **Show/Hide**. Notice that the molecule in the viewer disappears. Click on the same rectangle again to redisplay it. Click on the rectangle under the **Sel** level to select or deselect all. Experiment by clicking on each of the other buttons. These are short cuts to a basic set of commands for displaying and coloring various molecular representations which we will cover in step 5.

Note: you can customize the mouse bindings (and other user preferences) in a `_pmvrc` preference file which will be sourced whenever you start PMV.

3. Mouse Bindings:

You can directly interact with the molecules in the PMV Viewer with your computer's mouse or touchpad. By default, PMV is configured to work with a three-button mouse. The mouse buttons can be used alone or with a modifier key to perform different operations. To zoom the molecule (make the molecule look bigger or smaller) in the Viewer window, press and hold down the <Shift> key and then click and drag with the middle mouse button. To rotate the molecule, just click and drag with the middle mouse button. To summarize what the mouse buttons do:

On Apple Computers:

option: Rotate

command: Translate left/right

shift+**option**: Scale or Zoom

shift+**command**: Translate in/out

Note: as you translate a molecule out in the Z dimension, it will disappear into the **fog** which is used for depth-cueing.

| Button Mod | Left | Middle | Right |
|---------------|----------------|-----------------------|--|
| None | <u>P</u> ick | <u>R</u> otate | <u>T</u> ranslate left/right (X) and up/down (Y) |
| Shift | <u>S</u> elect | <u>S</u> cale or Zoom | <u>T</u> ranslate in/out (Z) |

You can also press the following keys in the Viewer window to change the view of the molecule:

| Key | Action |
|----------|---|
| R | <i><u>Reset</u> view</i> |
| N | <i><u>Normalize</u> – scale so all visible molecules fit in the Viewer</i> |
| C | <i><u>Center</u> on the center of gravity of all the molecules</i> |
| D | <i>Toggle on/off <u>Depth-cueing</u> (blends molecule into background farther away)</i> |
| T | <i>Toggle between transform root (ie scene) and transform the Viewer's current object</i> |

Note: By default, the Viewer's current object is **root** so you will not see any changes here if you toggle between *transform root* and *transform current object*. The Viewer GUI lets you change the current object. This is covered in Appendix 1.

Note: here the menu equivalent is

Color → **by Atom Type**

Click on **All Geometries** and then click **OK**.

Most PMV menu commands can be undone. **ColorByAtomType** is one of them. Try it out by clicking on the **Undo** button (curved arrow icon) or this sequence:

Edit->UndoColorByAtom

Before going on, redo **colorByAtom** type.

4. Color molecular representations (i.e. geometries):

Each row in the Dashboard is linked to a single entity. The entity is either **PMV Molecules** which is the group comprised of all the Molecules in the Viewer, a single **Molecule**, a single **Chain**, a single **Residue** or a single **Atom**. The diamonds on the right side of the Dashboard are used to color the displayed geometries for that row's entity by different coloring schemes.

A. In the **PMV Molecules** row, click on the **diamond** under **Atom** to color the lines by Atom Type. In this coloring scheme, all of the displayed objects will be colored according to the chemical element, as follows:

- Carbons that are aliphatic (C) - white,
- Nitrogens (N) - blue,
- Oxygens (O) - red,
- Sulfurs (S) - yellow,
- Hydrogens (H) - cyan.

In the color scheme developed by David Goodsell (mgl.scripps.edu/people/goodsell), acidic residues such as ASP and GLU are colored red while basic residues such as LYS and ARG are colored blue. The more acidic or more basic the residue the more saturated the color used to display it. Sulfur atoms are colored yellow. Neutral atoms such as carbons are white.

Note: here the menu equivalents are:

Display → **CPK**

Display → **Sticks And Balls**

Display → **Secondary Structure**

Display → **Molecular Surface**

The **MSMS Parameters Panel** lets you set a **Surface Name** for the surface, whether to calculate the surface **Per Molecule**, the **probe radius** which determines the bumpiness of the surface and the **density** of the triangles used in drawing the surface where the more triangles the smoother the surface. Use the default name **MSMS-MOL** and default probe radius, **1.5**. For a smoother surface set the density to **2.0**. Then click on **OK**.

Note: here the menu equivalents are:

Color → **by Chain**

Display → **CPK**

click on **undisplay**.

B. Click on the other diamonds in the row to try coloring the lines by the other available coloring schemes which include by **Molecule**, by **Chain**, using **Rasmol** residue colors, using **SHApely** residue colors, using **David Goodsell** atom colors, by **Secondary Structure** element type and by **Instance**.

5. Display options for molecules:

The circles on the left side of a row in the Dashboard are used to display different geometries for the entity corresponding to the row.

A. In the **PMV Molecules** row, click on the **circle** under **CPK** to display hsg1 with a sphere for each atom whose radius depends on the element of the atom.

B. Undisplay CPK by clicking on the circle again. Positioning the cursor over a circle and clicking with the right mouse button opens an option panel for that display command. For CPK, you can adjust the radii, scaling each by a constant factor and/or adding a constant to each.

C. Try displaying hsg1 with **S&B** (Sticks and Balls), **Rib.** (secondary structure), and **MS** (molecular surface).

D. Right click on the MS circle to open the MS options widget. In the widget which opens, click on the square labeled **Per Molecule**. Click on the MS circle again. With **Per Molecule** turned off, the molecular surface is calculated only for the chain so the result is a closed surface. When calculated **Per Molecule**, calculating a molecular surface on a Chain results in a surface fragment.

6. Color a single representation:

A. Click on the other circles in the row to undisplay all geometries except CPK.

B. Color it by chain by clicking on the diamond under **Chain**. Note that each chain in hsg1 is colored a different color. We chose to color only the CPK geometry by **Chain**; the lines are still colored by **Atom**. To see this for yourself, use the circles to hide the CPK geometry and to display lines. Note that the lines are still colored by **Atom**.

You can see all the files you have opened with **File->Recent Files**,

Note: the menu equivalent is:

Display → **Show/Hide Molecule**

Click on the red circle at the top left of this widget (or **X** at the top right) to hide this widget.

Note: When you click on the left mouse button with the cursor over the Viewer, there may be multiple objects in the tiny picking 'rectangle' next to the cursor. If there is more than one object, the closest object in the z dimension is 'picked'.

7. Load a second molecule:

A. Position the cursor over **PMV Molecules** in the Dashboard and click with the right mouse button. This opens the file browser again.

B. Select **indinavir.pdbq** and click on **Open**. Now a second **Molecule** named '**indinavir**' has been loaded into PMV.

C. Using the circles in their respective rows, try displaying indinavir only, switch to show only hsg1, hide both molecules, redisplay both.

8. Picking commands:

PCOM stands for "picking command" (earlier versions used ICOM for "interactive command"— they are the same thing). A **pick** happens when you position the cursor over an object in PMV and left-click with the mouse (possibly while holding down a keyboard **modifier** key, such as <Shift>, <Ctrl> or <Alt>). The result of the pick is that the picking command is applied to the picked object. The type of the picked object is determined by the PCOM level. For instance, if the PCOM level is **Atom**, the pick will return an atom. Alternatively, picking can be done by dragging the mouse while holding down the left mouse button. In this case, all the objects in the dragged rectangle are picked. When the PCOM is **printNodeNames** and the PCOM level is **Atom**, the names of the picked atoms are printed in the Message box. By default, the PCOM is set to **selecting** at the **Atom** level.

A. Click on the **Pointing Finger** button to display the picking command controls:

PCOM level: is a drop-down menu which lets you set the level of object(s) to be picked. By default the level is set to **Atom**. Consequently picked objects are Atoms.

B. Click on **Atom =** to display other levels.

With 4 different modifiers, **None**, **Shift**, **Ctrl** and **Alt**, up to 4 different picking commands can be used at the same time. The commands are bound to these four via the 4 drop-down menus here:

printNodeNames = bound to picking with no modifier

Shift: **None=** no command bound to picking + Shift key

Ctrl.: **None=** no command bound to picking + Control.

Alt: **None=** no command bound to picking + Alt.

Note: PMV uses a **tree** data structure composed of these four levels. The nodes within a level are **siblings**. Each node, except for those of the lowest level-Atom, has **children** which is a set of nodes in the next level down. Each node, except for those of the highest level-Molecule, has both a **parent** and a **top**. For an atom, its parent is a residue while its top is the molecule to which it belongs. Its siblings are the other atoms in the same residue. See **Appendix 3** for more details.

PMV uses a hierarchical data structure to represent molecules. There are four levels in this hierarchy, and each level can include many instances of the structures at the next lowest level. In order from highest to lowest, these levels (and their PMV color code) are:

- Molecule - red
- Chain - cyan
- Residue - green
- Atom - yellow

In this section, we will use the mouse to pick on the objects in the Viewer to find their names.

C. Using the mouse, position the cursor over a line and click the left mouse button. Change the PCOM command to **PrintNodeNames** by clicking on the **down** arrow to the right of **select** to display the list of possible picking commands. Scroll down by dragging the rectangular box on the top right side of the list until you can see **printNodeNames**. Click on it to set the PCOM.

D. Open the Message Box by clicking on the Message Box button and look at the text which appears below the Viewer.

E. Change the PCOM level from **Atom** to **Residue** and click again. Try clicking somewhere else. You can pick on a region by dragging, that is: hold the left-mouse button down while you move the mouse. Try it on a small region. Look at the message which appears when you pick on **hsg1** with the PCOM level set to **Chain**.

Note: See **Appendix 4** for more details on available startup options.

9. Starting PMV using command line arguments

PMV will automatically load a molecule for each filename with a recognized extension listed after 'pmv'. Thus we could have used this shortcut instead of steps 1 and 5.

For **TSRI** tutorials **only**:

On linux machines:

/home/user##/MGLTools/share/bin/pmv

On mac os X:

/Library/MGLTools/share/bin/pmv

Preface pmv with the path to your local installation of
MGLTOOLS:

\$MGLTOOLS/pmv hsg1.pdbqs indinavir.pdbq

Summary: what have we learned?

1. PMV can load multiple molecules from a variety of file types. These molecules can be displayed and colored with different coloring schemes independently.
2. Any geometry can be colored by properties of the underlying entities.
3. Picking in the Viewer can be bound to commands acting on Atoms, Residues, Chains or Molecules using optional keyboard modifiers.
4. PMV startup options allow you to specify molecules to load and how they are to be displayed.

Add-On: Binding Commands to Keys

PMV allows you to bind executing a command on a particular set of entities to a keystroke. This provides a useful mechanism for repeated actions. To demonstrate this, we will bind displaying the first 10 atoms as CPK to the **F1** key.

Open the Python Shell by clicking on the Python-Shell button which is sixth button on the **Button** bar and type this:

```
mv.bindCmdToKey("F1", "None", mv.displayCPK, \
                (mv.allAtoms[:10],))

mv.bindCmdToKey("F2", "None", mv.displayCPK, \
                (mv.allAtoms[:10],), {'negate':True})
```

Now click the **F1** key. You can undisplay the CPK representation of these atoms using the **F2** key. Close the Python Shell by clicking on the Python-Shell button.

Before going on, clean up!

Delete all the molecules in the Viewer.
At this point we will delete all the molecules.

Edit → **Delete** → **Delete All Molecules**

You will be asked if you wish to continue because deleting cannot be undone. Click on **Continue**.

Note: Instead of typing these lines you could use

File → **Read** → **Python**
or Session scripts In the file browser which opens select **bindCmdToKeyExample.py** and click on **Open**.

Note: You could delete one or more molecules using **Edit** → **Delete** → **Delete Molecule**. In the widget which opens you can select a single molecule to delete or you can use the control key to select more than one molecule to delete. You will still be asked whether you wish to **Continue** or not.

Hemolysin: Secondary Structure colored by Chain.

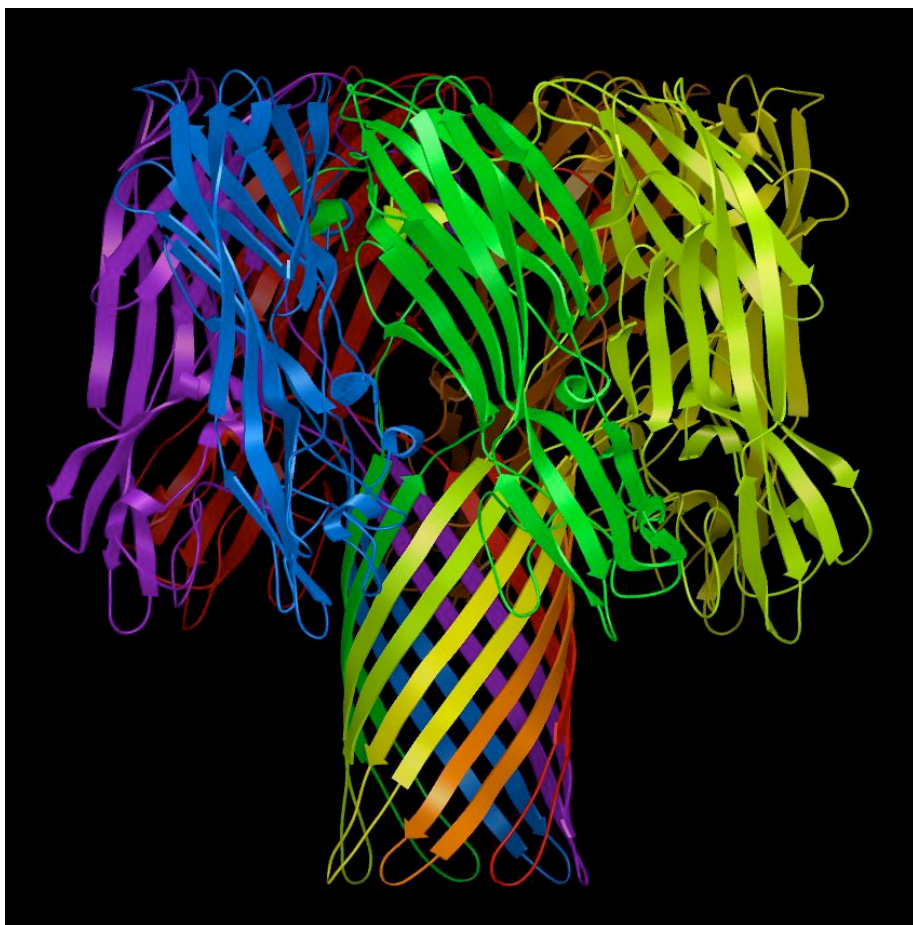


image courtesy of David Goodsell, TSRI

Procedure:

1. Load hemolysin:

Note: You can also use File → Read → PDB from RCSB.org menu to load hemolysin. Type 7ahl for the PDB ID and press OK to fetch this molecule from the web.

A. Position the cursor over **PMV Molecules** in the Dashboard and click with the right mouse button.

B. In the file browser which opens, select **7ahl.pdb** and click on **Open**.

Note: Representations of more sophisticated molecular properties are possible using the PMV module `secondarystructureCommands`,

Secondary structure representations of a molecule show sections of alpha helix as rectangular strips [extruded circles], beta sheets as arrow-capped strips [extruded arrows] and turns and random coils as thin tubes [extruded circles].

2. Construct a representation of the secondary structure of **7ahl** and display it:

A. Click on the circle under **Rib.** in the **7ahl** row of the Dashboard.

The displayed secondary structure, an example of a beta barrel, is composed of 46 sections of alpha helix, beta sheet, random coils and turns. 7 Coils, 9 Strands, 1 Helix and 6 Turns for each Chain.

3. Color the secondary structure to distinguish the chains:

A. Left-click on the diamond under **Chain** in the **7ahl** row of the dashboard to open the **colorByChains** widget.

B. Click on **secondarystructure** and **OK**.

Notice that each of the 7 chains in **7ahl** is colored a different color. We chose to color only the `secondarystructure` geometry by Chain; the color of lines is unchanged.

4. Clean up by removing the lines and adjust the view.

A. Click on the red circle under **Lines** in the **7ahl** row of the dashboard to undisplay the lines.

B. Use the middle mouse button to position **7ahl** so that you are looking down the cylinder at the center of the molecule.

In the final steps we will customize the colors of chains **B**, **D**, **F** and **G** to achieve the rainbow effect.

5. First label the chains to help keep track in this process.

A. **Display** → **Label** → **by Properties**

B. In the widget which opens, change the **PCOM** level to **Chain** and choose **name** from the list of properties.

C. Click on **Label settings** to open the **ChainLabels** widget.

D. Set **global scale** to **5** and close this **ChainLabels** widget by clicking on the top left red button or top right x.

E. Select **First** for Label Location.

F. Finally click on **OK** in the **LabelAtom by properties** widget.

Using the labels, notice the chains are ordered clockwise from **A** to **G**. Here are the default colorings of the 4 chains we'll

change and the new custom colors we'll apply in the following steps 6-10:

Chain **A** is Blue
Chain **B** is Green → Magenta
Chain **C** is Red
Chain **D** is Cyan → Orange
Chain **E** is Yellow
Chain **F** is Magenta → Yellow-Green
Chain **G** is Cyan → Green

Note: In PMV the **current selection** is a particular homogeneous subset of the Atoms, Residues, Chains or Molecules currently loaded in the Viewer. It cannot contain entities of different levels such as Residues and Atoms. The current selection plays a pivotal role in PMV because most commands, such as display, color label etc, operate on the current selection. The current selection is dynamic: it

Note: you can also use Direct Select to unselect chain B. In PMV there are often several ways to do things. For sanity reasons, we will not try to cover them all.

6. Color chain **B** magenta:

A. Select chain **B**:

Click on the gray rectangle next to **B** in the Dashboard.

B. Color its secondary structure magenta.

Color → **Choose Color**

Click on **secondarystructure** in the **color** widget which opens and then click **OK**.

In the **Choose Color** widget which opens next, click on the magenta button and on **DISMISS**.

C. Unselect chain **B** by clicking on its gray rectangle.

7. Color chain **D** orange

A. Select chain **D** by clicking on its gray rectangle.

B. Color its secondary structure orange.

Color → **Choose Color**

Click on **secondarystructure** in the color widget and then click **OK**

In the color chooser widget, click on the **red** button and **Edit** → **Edit Selected Color** to open a **color wheel** for customizing the color.

In the color wheel, move the tiny box into the orange region by clicking on it with the left mouse button and holding the mouse button down while you drag it onto the orange edge of the color wheel. Click on **DISMISS**

C. Unselect chain **D** by clicking on its gray rectangle.

8. Color chain **F** yellow-green.

A. Select chain **F** by clicking on its gray rectangle .

B. Color its secondary structure yellow-green.

Color → **Choose Color**

Click on **secondarystructure** and then click **OK**

In the color chooser widget, click on the yellow button and

Edit → **Edit Selected Color**

In the color wheel, move the tiny box into the yellow-green region and click on **DISMISS**

C. Unselect chain **F** by clicking on its rectangle.

9. Color chain **G** green.

A. Select chain **G** by clicking on its gray rectangle.

B. Color its secondary structure green.

Color → **Choose Color**

Click on **secondarystructure** and then click **OK**

In the color chooser widget, click on the green button and

DISMISS

In the color wheel, move the tiny box into the yellow-green region and click on **DISMISS**

C. Unselect chain **G** by clicking on its rectangle.

10. Clean up by removing the chain labels.

A. Click on each red Lab. square in Dashboard.

Summary: what have we learned?

1. Pmv can display secondary structural properties of proteins using different geometries for alpha helix, beta sheet, turns and random coils.
2. In Pmn, commands are applied to the **current selection**. This is a homogeneous set of entities which are atoms, residues, chains or molecules.
3. Individual entities can be directly selected or deselected.
4. Geometries can be **labeled** according to properties of the underlying entities.
5. Specific geometries can be colored using a **custom** color.

Note: You may notice that the blue and the red chains stand out ('leap out') from the other colors. You can modify this by changing the **saturation** of these two colors using the color wheel. Saturation is the amount of white mixed into the color. Saturated colors have no white. Saturation varies along the radius of the wheel. Decrease the saturation of a chosen color by dragging the tiny box toward the center of the color wheel circle.

Note: Alternatively you could select all the chains by clicking on the first one and holding down the shift key while you click on the last. Then click on a single red Lab. Square to turn them all off at once.

Note: PMV colors secondary structure using the color appropriate for the alpha carbon of the corresponding residue for each section of secondary structure. Thus secondary structure should not be colored by per atom schemes. ColorByAtom would result in a white geometry because that is the CA color.

Add-On: Color by Secondary Structure

Position the cursor over the diamond under **Sec. Str.** and click with the left mouse button. This view of Hemolysin shows that the secondary structure of each of the seven identical chains is composed of beta strands which are yellow, an alpha helix which is pink, random coils which are white (turns would be colored blue).

Add-On: Measure hemolysin beta barrel

Set the picking command (PCOM) to measureDistance by clicking on the Picking COMmands ICON to display the PCOM Bar and then clicking on the down arrow next to the first entry to scroll down to measureDistanceGC. Click on measureDistanceGC.

Now left-mouse clicks in the viewer are bound to measuring distances. Left-click somewhere on the top edge of the molecule. In PMV all geometries are mapped to underlying atoms. A yellow sphere appears at coordinates of underlying atom. Left-click somewhere on the bottom edge of the molecule. Now a dotted line appears connecting the two picked atoms labeled by the distance. (You may need to undisplay the molecule using **show/hide molecule** to see this line.) Measure Distance displays up to 4 distances in a row. You can change this **measureDistanceSL** via

File → **Preferences** → **Modify Defaults**

To clear the displayed distances, set the PCOM to something else and then to measureDistanceGC again. Alternatively use:

Edit → **Measure** → **Distance**

You may want to redisplay the lines to make it easier to pick on specific atoms.

To stop measuring distances, use down arrow to the right of the PCOM entry to set the PCOM back to **printNodeNames**.

Before going on, clean up!

At this point we will delete all the molecules.

Edit → **Delete** → **Delete All Molecules**

Note: You may want to redisplay the lines to make it easier to pick on specific atoms.

Note: There is a list of all the user preferences and their defaults in **Appendix 7**. The newest one is "Startup Directory" which lets you specify where to start PMV.

Note: you can measure angles formed by 3 atoms using **Edit** → **Measure** → **Angle** and torsion angles formed by 4 atoms using **Edit** → **Measure** → **Torsion**.

HIV Protease: Active Site Residues and Inhibitor

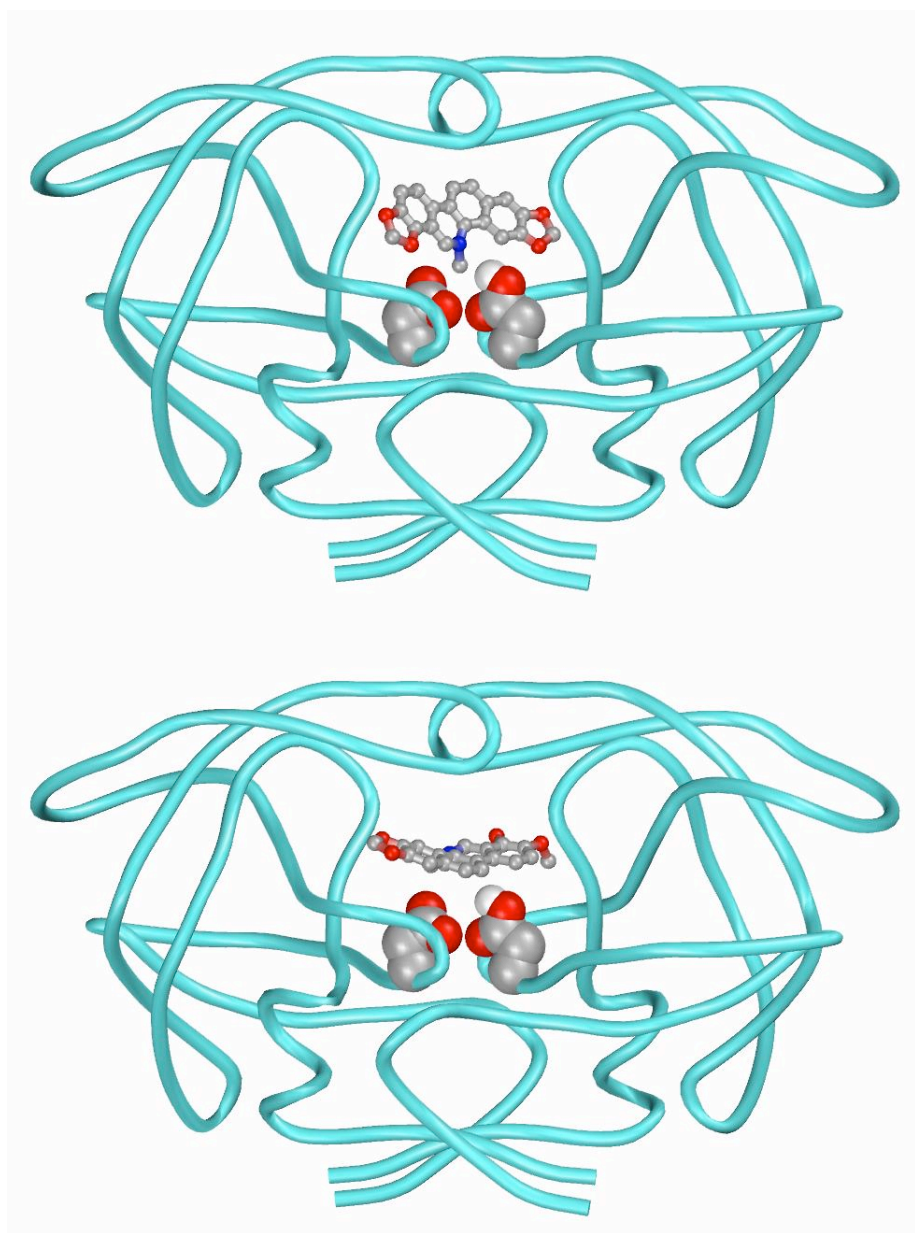


image courtesy of David Goodsell, TSRI

Note: you load what you need when you need it. Observe the effects of loading the traceCommands module: A new menubuttons

Compute → **Trace** appears in the GUI. Also, in the Browse Commands widget you can view the documentation strings of individual commands in modules listed by clicking on the Show Documentation checkbox and selecting a command. If the selected command has inline documentation, it will be displayed below the Show documentation

Note: the Browse Commands widget supports a shortcut device called **quick keys**. What this means is that if you first click in the list of modules, the next keystroke will be matched to the entries in this list. For example, if you first click in the list of modules then type the letter **m**, the first entry which starts with m will be highlighted and can be loaded using Load Module. You can match the first 2 characters if you can type fast enough.

Note: **splines** are smooth curves computed based on user-defined control points which result in a 3D representation of a molecule as a thin tube. They can be either **interpolating** or **approximating** splines, depending on whether the curve goes through the control points or approximates them. PMV **traceCommands** use approximating splines while PMV **splineCommands** use interpolated splines. Here we are using the smoother, Calpha-trace geometry.

By default, for a protein the control points are the CA atoms of the residues and the resulting tube represents the backbone of the protein.

Procedure:

1. Load a new command we will need for this exercise:

- Click on **File** → **Browse Commands**
- Select **Pmv** package in the list of Packages.
- Highlight **traceCommands** in the list of available PMV modules.
- Click on **Load Module** to load it.
- Click on **DISMISS** to close the widget.

2. Display a Calpha-trace representation of hiv protease.

- Load **hsg1** by positioning the cursor over **PMV Molecules** in the Dashboard and click with the right mouse button. In the file browser which opens, select **hsg1.pdbqs** and click on **Open**.
- Construct a CATrace representation of the secondary structure of **hsg1** and display it:

Compute → **Trace** → **ComputeExtrude Trace**

A CATrace is a spline, a smooth 3D curve calculated from the coordinates of the calpha atoms. It represents the backbone of the protein. The default extrudes a spline using a circle of radius 0.1. You could use compute and extrude as separate steps if you wish to change the defaults.

C. Color the CATrace geometry cyan.

Color → **Choose Color**

Click on **CATrace** and then click **OK**

In the color chooser widget, click on the cyan button and **DISMISS**

D. Clean up by removing the lines.

Click on the red circle under **Lines** in the **hsg1** row of the Dashboard to undisplay the lines.

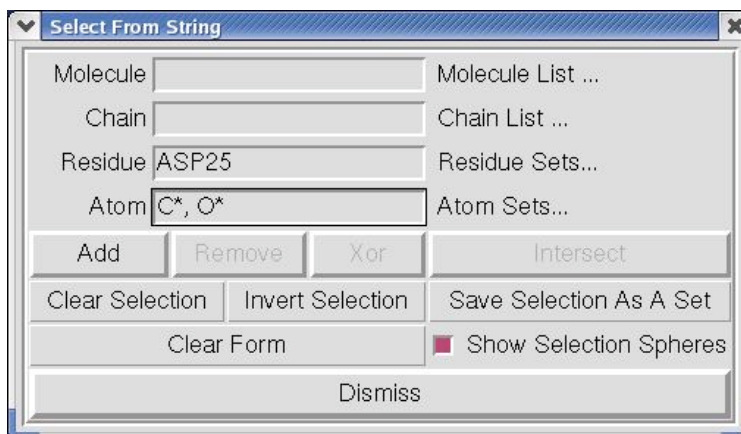
3. Display the side chains of the active site residues as CPK.

- Select the carbon atoms and the oxygen atoms in the two **ASP25** residues at the active site. Clicking on

Select → **Select From String** opens this GUI:

Note:

1. The 4 entries specify strings which are matched at the molecule, chain, residue and atom levels resulting in a new group of entities.
2. The menus which drop down from the Molecule List..., Chain List..., Residue Sets..., Atom Sets... **buttons** let you add or remove strings by clicking on checkboxes.
3. The buttons **Add**, **Remove**, **Xor** and **Intersect** define how the specified subset will be combined with the current selection:
currentselection **op** newgroup
4. When the current selection is empty, the only allowable operation is Add. Therefore, the buttons Remove, Xor and Intersect are disabled when nothing is selected.



Select From String lets you build a selection based on strings you enter for the Molecule, Chain, Residue and/or Atom level. These strings can be names, numbers, ranges of numbers, or lambda expressions which are evaluated to build a set. A comma-separated list of items can be entered in any entry. The strings can contain regular expressions including wild cards such as * which match anything. This command also matches residue sequences in single letter format. In addition, it supports selecting user-defined sets as well as predefined residue and atom sets. (See Appendix 2 for more details)

Note: C* matches all the atoms in the two **ASP25** residues whose names begin with C. O* matches all the atoms in these two residues whose names begin with O. SelectFromString uses **regular expressions** to build sets.

A. Type: (!Caution!: there are no spaces in C*,O*)

ASP25 in the Residue entry

C*, O* in the Atom entry

B. Click on **Add** You will be asked if you want to set the selection level to MolKit.molecule.Atom. Click **YES** and

Dismiss

C. Display these selected atoms using a CPK representation

Display → **CPK**

Using the **thumbwheels**, set **Scale Factor** to 0.7, make sure **display** is on and then click on **OK**

D. Color the CPK representation of these selected atoms by Atom

Color → **By Atom Type**

In the colorByAtomType widget, select the **CPK** geometry and click on **OK**

To set a value using a thumbwheel:

Position the cursor over the thumbwheel and click the left-mouse button. While continuing to hold down the left-mouse button, drag the mouse to the right to increase the value and to the left to decrease the value. Alternatively, you can right-click on the thumbwheel to open a widget which lets you type in the desired value. With the latest version, you can position the cursor over the thumbwheel and type the new value followed by clicking on **enter**.

- E. **clear selection** before going on.
3. Display a **sticks and balls** representation of **indinavir**, an hiv protease inhibitor.
- A. Load **indinavir**:
- Position the cursor over **PMV Molecules** in the Dashboard and click with the right mouse button.
- In the file browser which opens, select **indinavir.pdbq** and click on **Open**.
- B. Display indinavir as Sticks and Balls:
- Position the cursor over the circle under S&B in the hsg1 row in the Dashboard and click with the left mouse button.
- C. Color indinavir according to atom type:
- Position the cursor over the diamond under Atom in the hsg1 row in the Dashboard and click with the left mouse button.

Summary: what have we learned?

1. Modules can be loaded when they are needed.
2. Splines approximating the coordinates of the Calpha atoms can be used to represent a molecule.
3. The current selection can be set using strings to specify particular subsets of atoms, residues, chains or molecules.
4. Subset of atoms can be displayed using a different representation and coloring.

Add-On: Using the DejaVu GUI

PMV provides the user with direct access to its graphics engine **DejaVu**, which is an OpenGL-based 3D Viewer. The DejaVuGUI is a graphical user interface to the Viewer.

Click on the **Sphere-Cube-Cone** button, the ninth button in the row of buttons to open the DejaVuGUI. At any time, the DejaVuGUI lets you set properties of the **current** object, camera, clipping plane or light. The properties control panel manages which set of options is displayed at any time. For this

section, we will set properties of the Camera to improve the image. [see Appendix 1 for more details]

Click on **Camera** in the **Properties Control** panel to display the Camera Properties Panel.

1. Increase antialiasing. [see Appendix 1 for details]

Click on **Scene Antialiasing** and **15**

2. Set the background color to white.

Click on **Background Color** and set **Value** slider near **1.0** by left clicking on its small triangle and dragging it to the right.

3. Save the image

Save → **Save Image As**

In the widget which open type in "hsg1_activesite.tif" and click on **OK**.

Note: You can also set the camera's background color using:
3DGraphics->SetBackgroundColor
and adjust antialiasing using:
3DGraphics->SetAntialiasing

Note: in general you should save images as 'tif' files. When you save an image as 'jpg' with PMV it automatically compresses the file so you lose resolution with that format.

Note: it is possible to save the image with a transparent background. You would need a transparent background if you wanted to use the image in a complicated composite. Transparent background images can only be saved in **png** format.

Add-On: Creating an mpeg video file

DejaVu GUI also provides access to video recorder powered by PyMedia - <http://www.pymedia.org>. Click on Camera under "Show properties panel for:" group. Next, click on the "Video Recorder" button to open panel shown below. Click on start/stop recording button (●) to start the recording. This button changes its color to red to indicate that all changes in the Viewer are recorded in out.mpg file. For demonstration purposes, rotate molecule and press stop recording button (■). This finishes up recording by closing out.mpg in the current working directory. Save As button can be used to save mpg file at a different location.



Before going on, clean up!

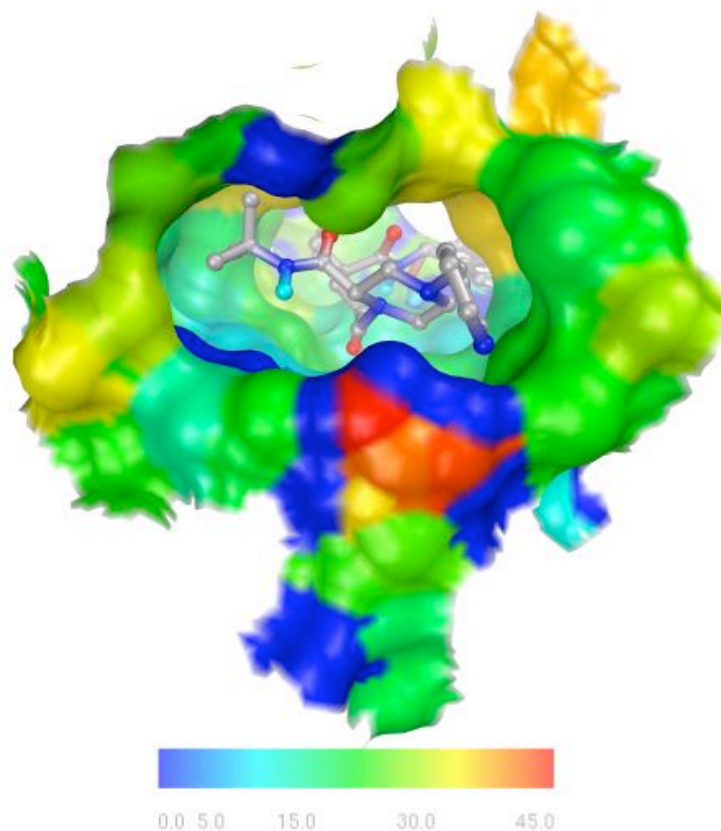
Delete all the molecules in the Viewer.

Edit → **Delete** → **Delete All Molecules**

You will be asked if you wish to continue because deleting cannot be undone. Click on **Continue**.

If you have changed the Camera background color, restore it to black before going on. Use the DejaVuGUI for this. Close the DejaVuGUI before going on. Set Scene Antialiasing back to 0 if you have changed it.

HIV Protease: Binding Site Surface Colored by Property



MSMS surface for residues within 4Angstrom of indinavir, colored by Temperature Factor

Procedure:

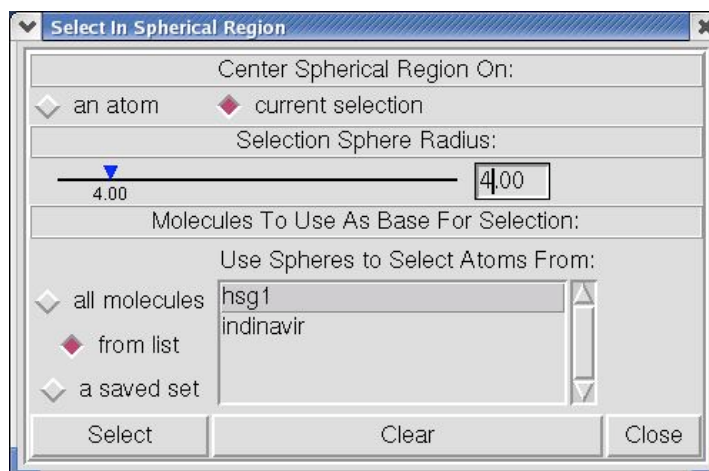
1. Display a molecular surface for hiv protease.

A. Load **hsg1**:

Position the cursor over **PMV Molecules** in the Dashboard and click with the right mouse button.

In the file browser which opens, select **hsg1.pdbqs** and click on **Open**.

- B. Compute the molecular surface for **hsg1** and display it:
Position the cursor over the circle under **MS** in the **hsg1** row in the Dashboard and click with the left mouse button.
2. Load indinavir and select it.
- A. Position the cursor over **PMV Molecules** in the Dashboard and click with the right mouse button.
- B. In the file browser which opens, select **indinavir.pdbq** and click on **Open**.
- C. Select indinavir by clicking on the grey rectangle in the indinavir row.
- D. Display it as Sticks and Balls colored by Atom Type by clicking on the circle under S&B and the diamond under Atom in this row.
3. Select all the atoms in all residues in hsg1 which have any atom within 4 Angstrom of an atom in indinavir:
- A. Click on **Select** → **SphericalRegion**
- B. In the widget which opens:
1. Click on **current selection** to use the atoms currently selected, as the centers for the ‘spherical region.’
 2. Type **4.00** in the entry for **Selection Sphere Radius**.
 3. To select atoms in hsg1 only, click on **from list** and **hsg1**.
 4. Finally click on **Select**.



Close the widget by clicking on **Close**.

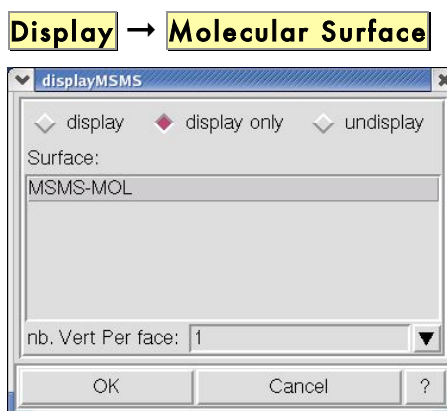
This results in selecting **72 atoms**.

C. Select all the Atoms in all the Residues with any atom in this group of 72:

1. Use **Select** → **Set Selection Level** to change the selection level to **Residue**. Now the selection consists of **26 residues** in hsg1 which have an atom within 4 Angstrom of an atom in indinavir.
2. Use **Select** → **Set Selection Level** to change the selection level to Atom. Now the selection consists of **214 atoms**.

Save these atoms as a set for use later on. **Select** → **Save current selection as a set**. In the **Set Name** widget which opens type in **ats** and click on **OK**.

4. Show the MSMS surface only for these atoms:



Click on **display only**, **MSMS-MOL** and **OK**.

5. Color MSMS surface according to the **Temperature Factor** property of the underlying atoms:

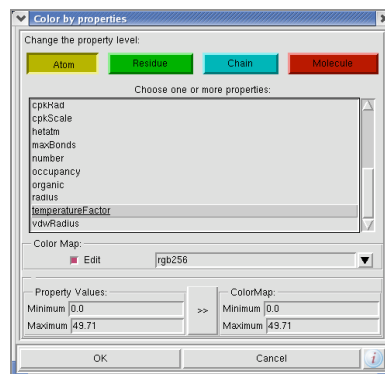
- A. Click on **Color** → **by Properties**

In the **colorByProperty** widget which opens, select **MSMS-MOL** and click on **OK**.

In the **Color by properties** widget which opens:

- Change the property level: to **Atom**.
- Scroll down to **temperatureFactor** and click on it.
- Click on **Edit** under **Color Map**:
- Click on **>>** to accept Property Values range 0-49.71

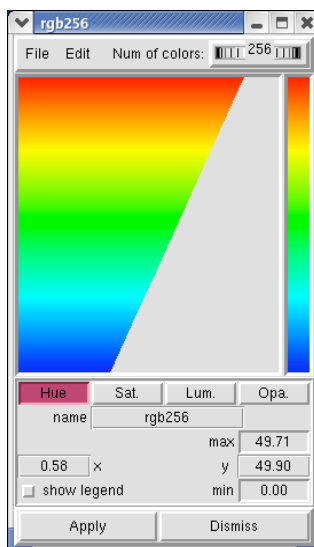
- Finally click on **OK**.



This results in coloring the MSMS surface according to the temperatureFactor of the underlying atoms. It also opens a widget which lets you edit the colorMap and permits you to add a colorMap Legend to the VIEWER.

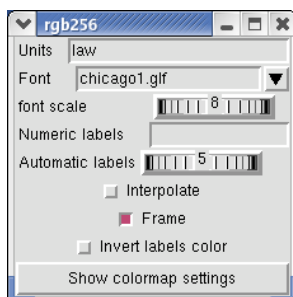
6. ColorMaps map a range of values to a range of colors. The red-blue-green colorMap **rgb256** maps the lowest values to blue and the greatest values to red using 256 different colors. Here we are using the range of temperatureFactors in the atoms in residues near indinavir, 0-49.71, for the range of values. Atoms with the lowest temperature factor will be colored blue and those with the highest red. Each polygon used to display the msms surface is colored according to the temperatureFactor of its underlying atom.

You can edit the rgb256 colorMap using its graphical-user-interface:



This widget lets you customize the Hue, Saturation, Value or Opacity range of the colorMap by clicking on the buttons in the middle of the widget. Currently, **Hue** is displayed and can be changed using the mouse. Click on any portion of the center rectangle to change the color for the corresponding section of the map. Clicking on **Apply** updates the objects colored using this colorMap in PMV. You can use Edit->Reset to restore the previous values.

You can add a color map legend to PMV by clicking on **show legend**. The rgb256 widget's **Edit** -> **Edit Legend** menu lets you enter a string for the Units at the top of the legend, set the text font, scale the text font, enter Numeric custom labels or the number of automatic labels to use,, whether to display the Frame around the ligand, whether to invert the color of the labels from white to black. Clicking on the legend (especially the white square at the top when it is displayed) lets you drag it from vertical to horizontal and/or reposition it. If you have dragged it, click on **show legend** in the color map widget to refresh its colors.



Note: To view the active site msms surface from the side, you need to use the DejaVuGUI to set culling for the msms surface to None. By default, the back side of all geometries is culled and the geometry is 'invisible.'. To change the culling of a geometry, use the DejaVuGUI to make the geometry the current object in the Viewer and then set culling in the Object Panel. See Appendix 1 for more details.

Summary: what have we learned?

1. You can select atoms in specified molecules which are within spherical regions centered on one or more atoms.
2. Geometries can be colored according to specific properties of the underlying atoms, using color maps which assign a unique color to each value in the range of the property in the atoms to be colored.
3. The color maps can be edited interactively and saved for future use.

Add-On: Intermolecular Hydrogen bonds

The **hbondCommands** module contains commands which allow you to build hydrogen bonds between specified atoms using adjustable parameters for distances and angles. This module also contains commands which allow you to display the resulting hydrogen bonds using various geometries.

Here we will use the default parameters to build **intermolecular** hydrogen bonds between hsg1 and indinavir and display them as lines.

Hydrogen Bonds → **Build** → **Set Parms + Build**

In the widget which opens, you opt to either to use all atoms or to specify two sets. Also, you can adjust the default **criteria** and decide whether to remove previous hydrogen bonds or not. This command builds hydrogen bonds based pairs of interacting atoms it detects using a dictionary of angle and distance cutoffs for pairs of sp2 or sp3 atoms. You can change any of the parameters. For hsg1, if you were to use all atoms vs. all atoms, >120 hydrogen bonds would be formed. Instead we want to build hydrogen bonds between hsg1 and indinavir only.

Note The **criteria** used for building hydrogen bonds includes distance and angle values for specific pairs of atom types. Atom types in this context means **sp2 donors**, **sp3 donors**, **sp2 acceptors** and **sp3 acceptors**.

A. Build hydrogen bonds:

Hydrogen Bonds → **Build** → **Set Parms + Build**

In the widget which opens:

- Click on **Specify two sets**.
- Click on the top **Molecule List** button and select **hsg1** in the list which drops down.
- Click on the lower **Molecule List** button and select **indinavir** in the list which drops down.
- Click on **Ok** to build hydrogen bonds.

A widget should open informing you that 4 hydrogen bonds have been formed. The hydrogen bonds are not displayed until you specifically display them.

B. Display the hydrogen bonds:

You can display hydrogen bonds as lines, extruded shapes, cylinders or spheres.

Hydrogen Bonds → **Display** → **As Lines**

Note: You may want to redisplay the hsg1 lines to see specific atoms in hsg1 which are involved in the hydrogen bonds. To do this, use **Un/Display-> lines**

Note: hydrogen bonds are displayed while the DisplayHbondsAsXXX widgets are open and are undisplayed when the widgets are closed. [this is not a popular feature]

The hydrogen bonds are shown as green lines between atoms. By default these lines are labeled by the distance between the atoms. You can use check buttons in the widget to undisplay distance measurements. It is also possible to display the angles between the atoms involved and an energy value for the hydrogen bonds.

Before the Add On, clean up the display to show only lines for each molecule:

- Click on the circle under MS to display MS for the entire molecule. Click on it again to hide the MS.
- Click on the circle under S&B to undisplay the StickAndBall geometry for indinavir.
- Click on the circles to show both molecules as lines.
- Close the hydrogen bond widget to undisplay the hydrogen bonds.

Add-On: Displaying Interactions

Interactions between two molecules can be visualized in PMV using a new command which was initially developed to show interactions between ligands and receptors. Here is a quick look at it:

1. Click on **Display** → **Interactions** to open a widget which lets you **Specify two groups of nodes for interaction display:**
2. Select **indinavir** in the list which drops down when you click on the top **Molecule List**.
3. Select **hsg1** in the list which drops down when you click on the lower **Molecule List**.
4. Click on **Ok**

As a result, the ligand is displayed with a molecular surface.

In this surface, the ligand atoms which do not interact with receptor atoms are colored dark gray while the ligand atoms which are interacting with receptor atoms are colored by atom type. Atoms in the receptor which are interacting with atoms in the ligand are displayed as spheres. Secondary structure is displayed for sequences of at least 3 residues in row in the receptor which have atoms interacting with the ligand. These interactions are either close-contacts or hydrogen bonds.

Note: In principle, this command can be used to illustrate interactions between any two groups of nodes in PMV. For example, you could choose hsg1:A for the first group and hsg1:B for the second to look at the intramolecular interface between these two chains. Please be aware, however, that this new command is under development.....

The **Interactions Display Options** widget lets you customize this display by toggling on or off different representations.

It is also possible to calculate and display pi-pi (and t-shaped) interactions, if they exist and/or pi-cation interactions. This calculation is slow...

Clicking on the down arrow attached to the entry for output list of results in writing the list of interacting residues and atoms in the python shell, which you open by clicking on its ICON on the ICON Bar.

You can use the Save Image button to write the image to a file. In principle, you can undo the display created by this command via the Revert button which also Closes the Interaction Display Options widget.

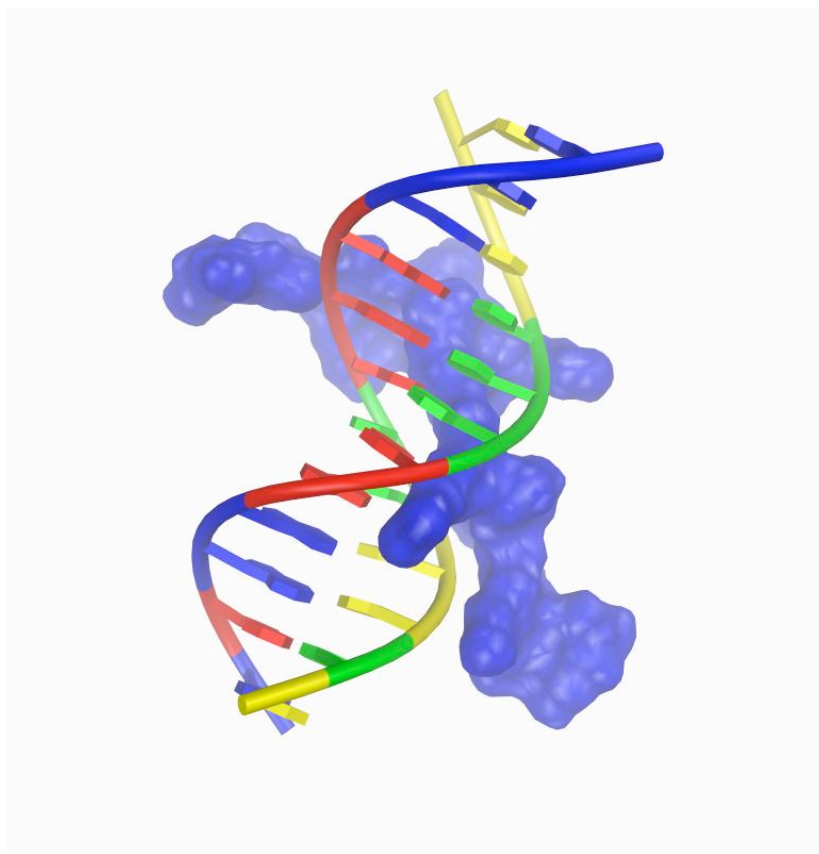
Before going on, clean up!

Delete all the molecules in the Viewer.

Edit → **Delete** → **Delete All Molecules**

In the widget which opens, click on **Continue**.

HMG-I(Y)-DNA: Secondary Structure



MSMS surface for second dna binding domain of human HMG-I(Y) and ribbon representation of DNA dodecamer

Procedure:

1. Load the molecule:
 - A. Position the cursor over **PMV Molecules** in the Dashboard and click with the right mouse button.
 - B. In the file browser which opens, select **2ezd.pdb** and click on **Open**. This results in loading the molecule **2ezd** into PMV and displaying covalent bonds between its atoms as lines.
2. Compute a molecular surface for chain **A** and display it:

- A. Position the cursor over the Circle under **MS** in the **A** row in the Dashboard and click with the right mouse button to open the **MSMS Parameters Panel** widget.
 - B. Click on the red square next to **Per Molecule** to calculate a surface for chain A only and then click on **OK**.
3. Color the Molecular Surface for **2ezd:A** blue:
 - A. Position the cursor over the diamond under Chain in the **A** row in the Dashboard and click with the left mouse button.
4. Display secondary structure representation for **2ezd:B** and **2ezd:C**:
 - A. Position the cursor over the circle under **Rib.** in the **B** row and click with the left mouse button.
 - B. Click **OK** in the **Set Nucleic Acids colors and scaling factor** widget which opens.
 - C. Repeat this process for chain **C**.
 - D. Clean up by undisplaying the lines by positioning the cursor over the red circle under Lines in the the **2ezd** row and clicking with the left mouse button.

Add-On: Input files for 3D Models

Printing 3D models is a new service offered at TSRI by the Olson lab in conjunction with Biomolecular Graphics. PMV can be used to produce the input files required by this service. The following steps result in a file which would produce a model of this representation of 2ezd.

1. Load fileCommands.

File → **Browse Commands**

In the **Load Modules and Commands** widget which opens click on **Pmv** in the list of packages. Scroll down the list under **Select a module** until you see **fileCommands**. Click on **fileCommands**, **Load Module** and **DISMISS**

2. Save a VRML 2.0 file:

Save → **Write VRML 2.0**

Type in **indinavir.wrl** and click on **OK**.

Before going on, clean up!

Delete all the molecules in the Viewer.

Edit → **Delete** → **Delete All Molecules**

In the widget which opens, click on **Continue**.

Postscript: Calculating electrostatic potential using Adaptive Poisson Boltzmann Solver (APBS).

So far we have introduced two main uses of PMV: as a toolset for looking at molecules and for preparing publication quality images. In closing we point to a third use of PMV: as a platform for the development of applications and we take a look at the APBS commands.

Note: We have a separate tutorial about ADT called *Using AutoDock with AutoDockTools*.

AutoDockTools (ADT), the graphical-user-interface to AutoDock, is one example of an application built on top of PMV. It extends PMV with 5 specialized modules for the preparation of AutoDock input files and for the analysis AutoDock results. Someone using ADT has access to all of the PMV functionality.

Another example is the APBSCommands module in PMV which provides a GUI for APBS (for more information on APBS, see <http://apbs.sourceforge.net>). APBS is a versatile program that can calculate electrostatic interactions between molecular solutes in aqueous media as well as calculate solvation and binding energies. The APBSCommands module implements setting up APBS input files, running the APBS program and analyzing the results.

Here we present a brief look at the APBS module.

Procedure:

1. Load histidine:

File → **Read Molecule**

Select **HIS.pdb** and click on **Open**.

2. Run APBS to compute the electrostatic potential using the Linearized Poisson-Boltzmann (PB) equation with default parameters:



Compute APBS → **Electrostatics** → **Compute Potential using**

Note: APBS needs partial charges (q) and radii (r) for each atom in pqr format. Since the molecule we read is in pdb format, we use pdb2pqr package to convert pdb to pqr before we pass it to APBS.

Wait until the calculation is complete and you can see this message: “Thanks for using APBS” in the **stdout** widget which opens. Click on **OK**

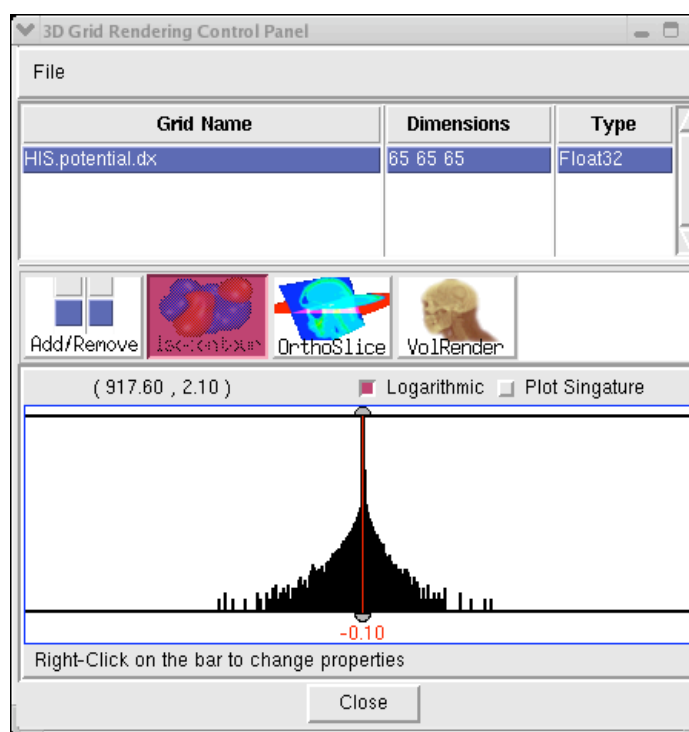
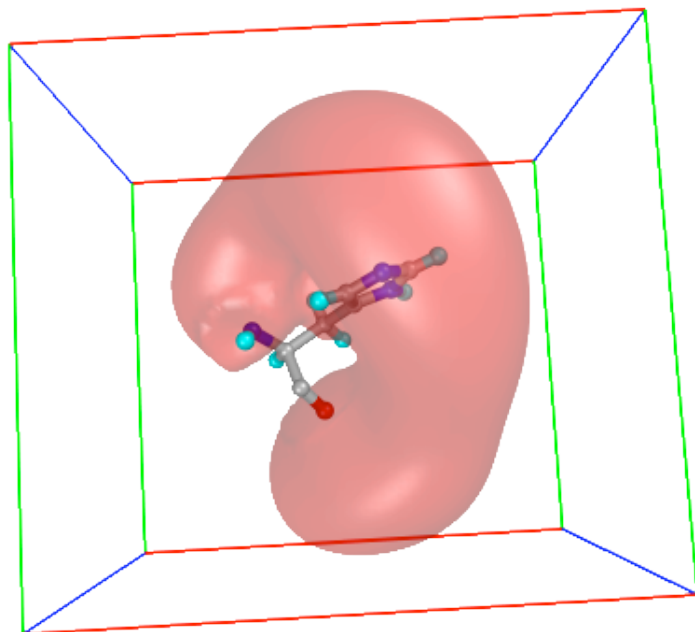
Note: *HIS.potential.dx* is the output electrostatic potential in **OpenDX** format for a 3D grid with 65 grid points in each dimension. The coordinates of the grid's back lower left corner (origin) are -27.853, -40.546, -9.7595 and the step sizes are .2536 in x, .22746 in y and .21292 in z. The data values $u(nx,ny,nz)$ are stored with increasing z-index, followed by increasing y-index and then by increasing x-index. (See the APBS User's Guide for more details: apbs.sourceforge.net/doc/user-guide/index.html)

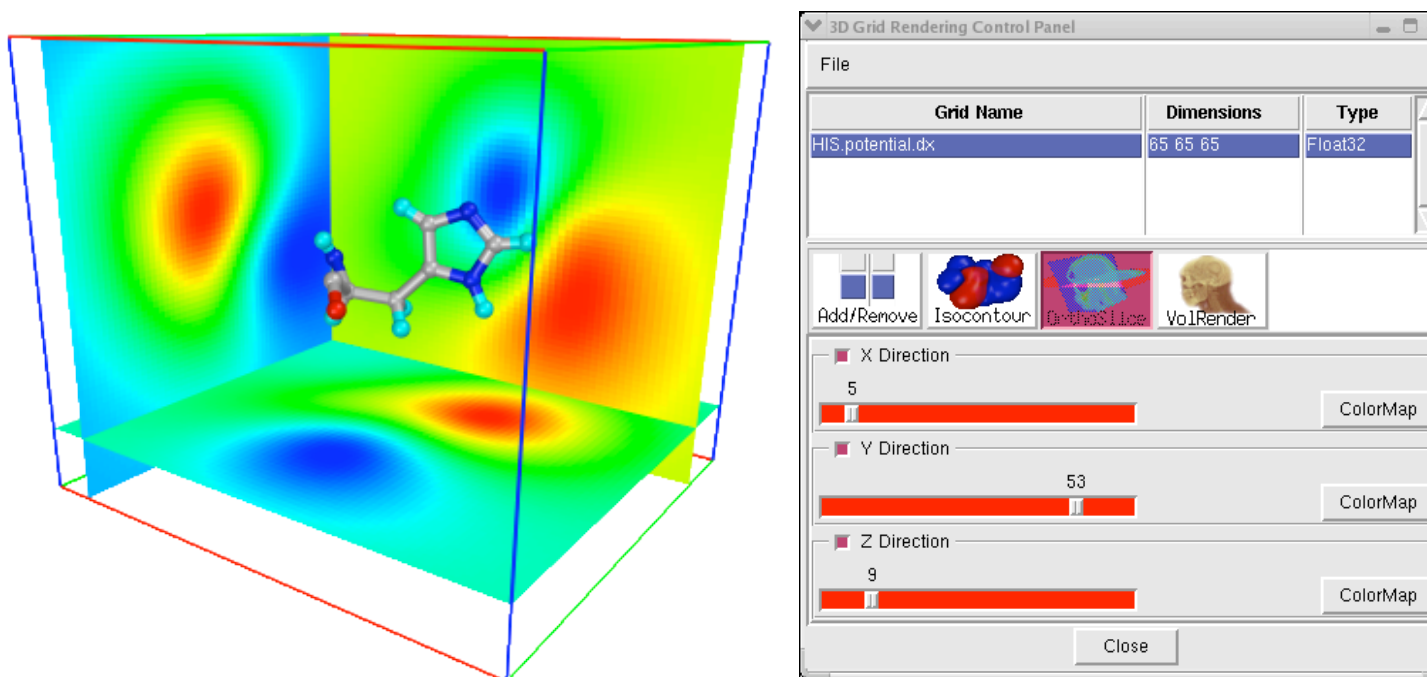
At this point we should have created an **apbs-HIS** subdirectory in the current working directory. This directory contains the input files for APBS as well as a file named **HIS.potential.dx** produced by APBS.

3. Click on the **3D Grid/Volume Rendering** icon  on the Tool bar to open the 3D Grid Rendering Control Panel widget. This widget contains a table of the 3D grids currently loaded in PMV, together with Add/Remove, Isocontour, VolRender commands. Click on the **Add** icon  in the 3D Grid Rendering Control Panel widget, navigate to apbs-HIS directory and select HIS.potential.dx.

Note: **Isocontours** are shown in PMV as a surface or as a set of lines through points of a constant value in 3D data. The constant value is called the **isovalue**

4. Click on the **Isocontour** icon to show the histogram of electrostatic potential. Shift-Click inside the widget to add an **isocontour**. This should create a bar inside the histogram and corresponding isocontour geometry in the PMV viewer. Move this bar to change the **isovalue**.





4. Click on OrthoSlice icon to see options available for displaying orthogonal slices along X, Y and Z directions. Checkbuttons toggle visibility of orthogonal slices and the sliders lets you move slices along corresponding directions. ColorMap buttons open ColorMap GUI where user can change the color map used to display orthogonal slices.

Add-On: How-to modify defaults and use APBS Setup: case study for superoxide dismutase.

Before we finish let's have a look at different parameters needed to Setup APBS.

Procedure:

1. Delete all the molecules in the Viewer.

Edit → **Delete** → **Delete All Molecules**

In the widget which opens, click on **Continue**.

2. Load **2sod_O.pqr**:

File → **Read Molecule**

This will open a file browser, showing all the pdb files in the current directory. Click on ‘-‘ in the file browser to display other file extensions. Select “.pqr” to list the pqr files. Select **2sod_O.pqr** (or just type **2sod_O.pqr** in the entry) and click on **Open**.

3. Run APBS:

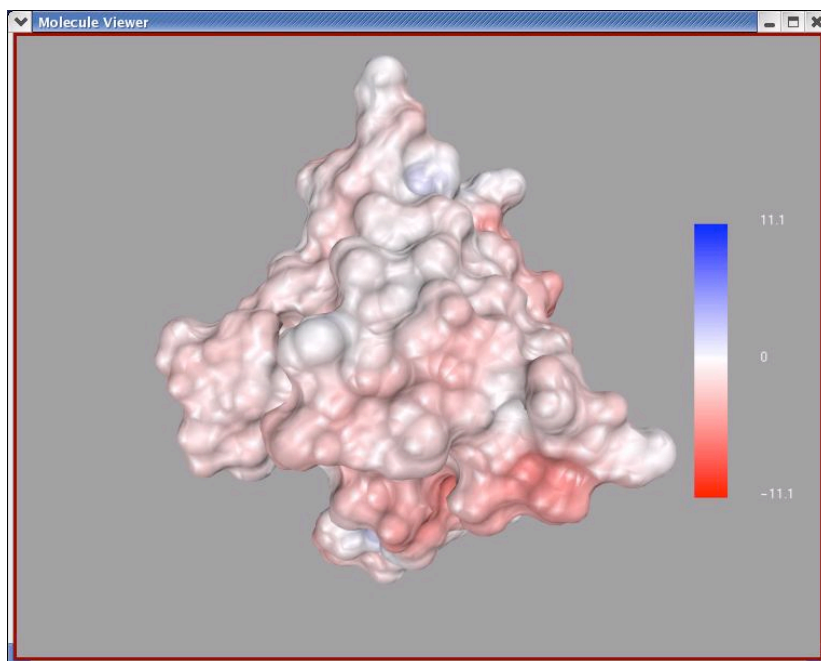
Compute → **Electrostatics** → **Compute Potential using APBS**

Wait until the calculation is complete and you will see this message:

“Thanks for using APBS”. Click on **OK**

4. Color the MSMS surface by APBS Potential.

Compute → **Electrostatics** → **Map Potential to Surface**



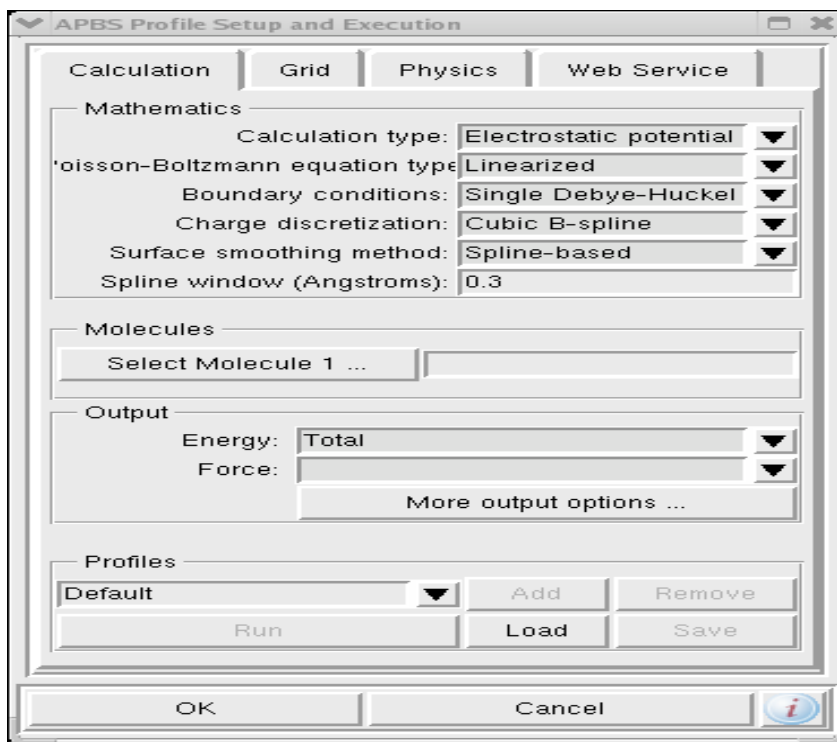
5. Open APBS Setup widget

Compute → **Electrostatics** → **Setup**

There are the following four tabs in this widget:

- Calculation – used for setting up Calculation type, Boundary conditions, selecting molecules, specifying the output, etc..
- Grid – used for setting up the grid dimensions.
- Physics - contains protein and solvent dielectric constant, etc.
- Web Services - used for running APBS remotely.

The widget opens with Calculation tab selected where one of the most important options to select is the Calculation type. By default, Calculation type is set to Electrostatic potential, which tells APBS to calculate electrostatic potential for a selected molecule. Two other options for Calculation type are Solvation energy and Binding energy.



When Calculate type is set to Solvation energy, APBS calculates total energy of a molecule twice, once in a solvent (E_{solvent}) and once in a vacuum (E_{vacuum}). The Solvation energy is outputted as $E_{\text{solvent}} - E_{\text{vacuum}}$.

User can also choose to calculate binding energy between two molecules by setting Calculation type to Binding energy, in which case s/he should provide Molecule1, Molecule2 and the complex between these two molecules. APBS then calculates the energy for Molecule1 (E_1), Molecule2 (E_2) and Complex (E_3), and outputs the binding energy as $E_3 - E_2 - E_1$.

Before finishing, let's have a look at the parameters in the Physics tab that can be changed as needed.

| | | | |
|------------------------------|------|-------------------------------------|-------------|
| Calculation | Grid | Physics | Web Service |
| Parameters | | | |
| Protein dielectric: | | <input type="text" value="2.0"/> | |
| Solvent dielectric: | | <input type="text" value="78.54"/> | |
| Solvent radius (Angstroms): | | <input type="text" value="1.4"/> | |
| System temperature (Kelvin): | | <input type="text" value="298.15"/> | |

Appendix 1: Advanced Display Control via the DejaVuGUI.

In these exercises, we introduced some tools useful for a primary function of PMV which is looking at molecules. PMV is also useful in preparing publication quality representations of molecules. PMV is well-suited to this purpose in that it provides the user with direct access to its graphics engine **DejaVu**, which is an OpenGL-based 3D Viewer.

DejaVu GUI, the graphical user interface to DejaVu, lets you alter any displayed geometry (i.e. graphical representation) interactively.

Note: An early DejaVu tutorial exists:
www.scripps.edu/~sanner/software/documentation/index.html

Note: whereas PMV uses the concept of the **current selection** to which most commands are applied, DejaVu uses the concept of the **CurrentObject** to which specified actions are applied. The CurrentObject at any time is highlighted with a yellow stripe. Initially, the CurrentObject is **root**, which is the parent of all other objects in the Viewer. By clicking on the + to the left of a node, you display its children. Expanding the **root** node shows children of root, which include all 'toplevel' objects present in the Viewer. You set the CurrentObject to any geometry at any level in the tree by clicking on its name.

Object Property Panel

Note: The tiny tab above the left edge of the yellow stripe is an Easter Egg. If you click on it and drag it to the right, it enables and opens a History panel which provides shortcuts for accessing previous CurrentObject geometries. Double click on a CurrentObject to add it to this list. Click in the list to set a CurrentObject.

DejaVu GUI:

Transformation Target

Object Tree

Scene Control

Properties Panel Selector

CurrentObject menu

Inheritance control

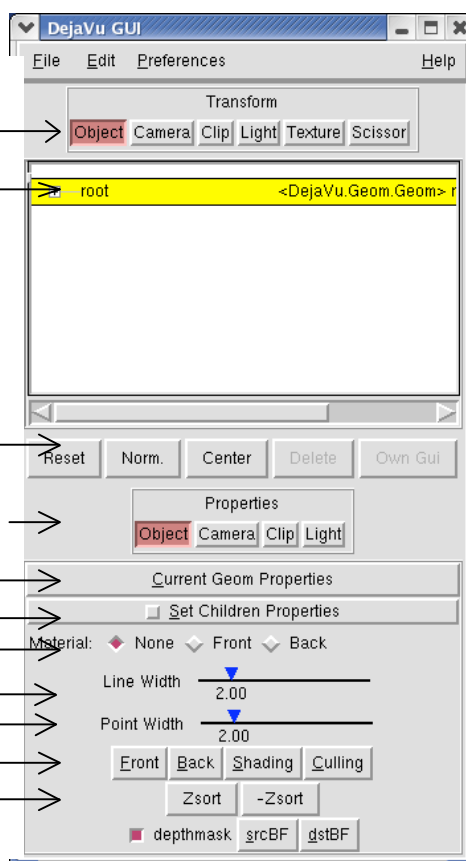
Material menu

Line width

Point width

Rendering Buttons

Zsort



Note: The current Transformation Target is transformed by mouse even Here the target is **Object**. a the CurrentObject is **root**. If if you hold down the right button and drag the mouse, hence the whole scene, will translated.

Note: The **Properties Panel Selector** buttons control which Property Panel is displayed below them. Here, the currently active Property Panel Selector button is **Object**, and the Object Property Panel is displayed. The other three panels - Camera Property Panel, Clip Property Panel and Light Property Panel - are shown in the following pages.

In this exercise we introduce six 'advanced' uses of the DejaVuGUI: We will use it to demonstrate moving geometries independently, to access the **Material Editor** for customizing materials, to smooth the edges of the lines and polygons via scene-antialiasing, to change the background color, to set

clipping planes and to set light direction and color. Finally, we will save the resulting image to a file. [Read in hsg1 and display it as SticksandBalls, cpk and compute an msms surface for it if necessary. Also read in indinavir if necessary]

Procedure:

1. Open the DejaVu GUI by clicking on the **Sphere-Cube-Cone** Button on the Tool bar.
2. Move one object independently of the rest:

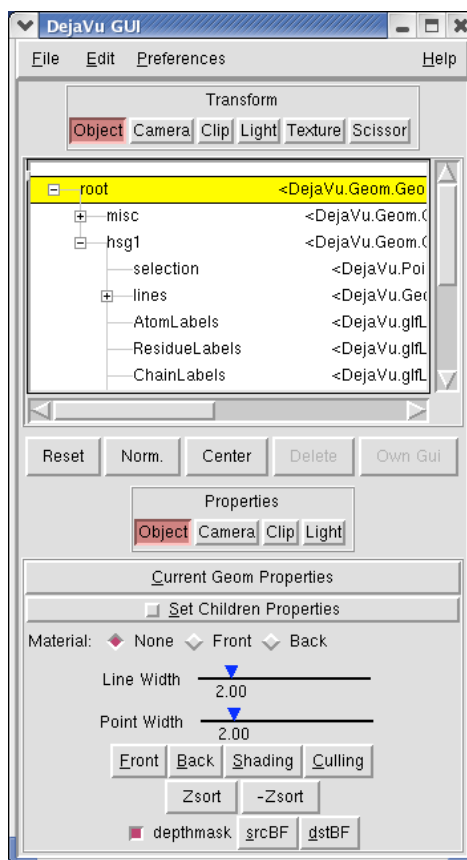
Display the object tree by clicking on the + to the right of **root** in the **Object Tree**.

Note: Reset, normalize and center operations are bound to the current object. By default this is the root, so everything is reset, normalized or centered with 'r', 'n' or 'c'. If you are transforming a current object other than root, you can reset it. If you move one object, then switch back to transform root only and transform the scene, you may not be able to reset the object.

Note: DejaVu Quick Keys provide a convenient way to switch between transforming a geometry and transforming the scene. Here is how to use them:

1. **Preferences-> Display Quick Key Panel**
2. **Add Quick Key**
#this adds button **Xform** root to panel
3. **Preferences-> Trans. Root only** #off
4. Make **cpk** geometry CurrentObject
5. **Add Quick Key**
this adds button **Xform root/hsg1/cpk**

Use the buttons to alternate between moving only the cpk geometry and moving the whole scene.



Find the object you want to move in the object tree (you may need to open its parent) and click on it. This makes it the CurrentObject in the Viewer.

In the **Preferences** menu at the top of DejaVu GUI turn off **Tranf.Root Only**. [You can also just type 'T' with the cursor in the camera]

Use the mouse to position the CurrentObject as you wish. Be sure to restore **Transf.Root Only** when done.

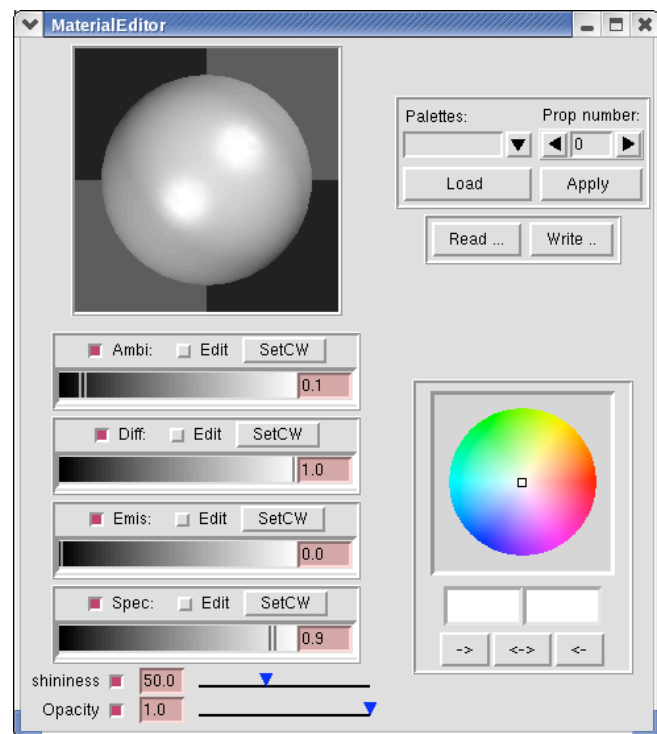
3. Set **Material** properties of a Geometry:

Click on **Front** button in the **Material** menu to open the **Material Editor**. To edit the material of an object, it must be the DejaVu Viewer's CurrentObject. Also it must not inheritMaterial. InheritMaterial is controlled by a checkbox in the menu displayed under **Current Geom Properties**.

Note: The large **sphere** in the top left corner illustrates the current light settings for the 4 different kinds of light: **Specular**, **Emissive**, **Diffuse** and **Ambient**.

- Specular refers to light from a light source which is reflected by an object. The sphere currently shows two round, specular reflects, one from each light source. The size of the specular reflect decreases as the shininess of the object increases.
- Emissive light is light which is emitted by a glowing object.
- Diffuse light is light from a light source which is not reflected back. The diffuse light is shown as the lighter-sector surrounding the two specular reflects.
- Ambient is the light from the rest of the scene. In the sphere the darker sectors at the top and bottom of the sphere illustrate the ambient light.
- Changes to the color, shininess and/or opacity are applied to the types of light for which the **Edit** button is checked when the change is made.
- Ambient and Diffuse light should be set together.
- If the current object is colored by part, ie an array of colors, these menus will be disabled.**

There are some preset 'light' palettes which can be applied to the current object. They are accessed and used via the buttons in the top right corner.



Experiment with setting the color (**HSV**) and the opacity of the msms surface. Opacity is controlled by a slider. The buttons directly below the color wheel let you save and restore a color. The left box displays the current color; the right box displays the previous color. The buttons underneath the boxes switch the two colors as indicated by their arrows. **->** saves the current value; **<-** restores the saved value; **<->** switches them.

4. Change the color of the background using a color wheel:

Note: Now you can also set the **background** color from the PMV Menu bar using:

3D Graphics->SetBackGroundColor

and use **Edit->Edit Selected Color** to customize it

Display Camera Property Panel by clicking on **Camera** in the Properties Panel Selector.

Click on **Background color** checkbox to display a color wheel and a Value slider.

Set the slider to 0.6 to set value

Click any where else in the color wheel to change the background hue.

Click near the center for a light colored background

Note: The color wheel uses a **Hue, Saturation, Value** scheme. Hue is the color and changes along the edge of the wheel. Saturation is the amount of white mixed into the color. Saturated colors have no white. Saturation varies along the radius of the wheel. Value is the amount of black added to the color. It is set using the slider under the wheel. Grey is obtained with 100% white and Value <1.0.

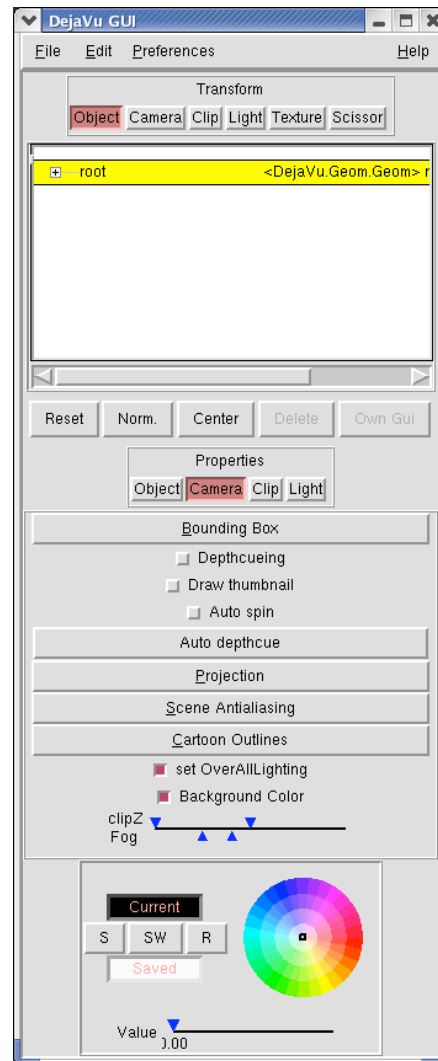
Note: The **Current** box displays the current color. The **Saved** box displays the color in memory.

S: copy current color into memory.

R: copy memory color into current.

SW: swap current color and memory color.

Note: The **fog** used for **depthcueing** changes along with the background color.



Note: Scene antialiasing is done by a **cycle** of drawing the image, accumulating data about the image in a buffer, jittering the scene by moving the image to a slightly different position which is repeated '**n**' times. The final displayed image is the average of the **n** different images. The higher the value of **n**, the smoother the lines in the images but the longer it

5. Smooth edges by increasing **scene antialiasing**:

Click on **Scene Antialiasing** to drop down menu of antialiasing choices.

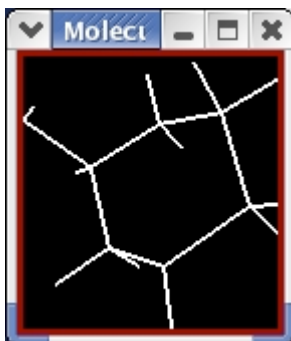
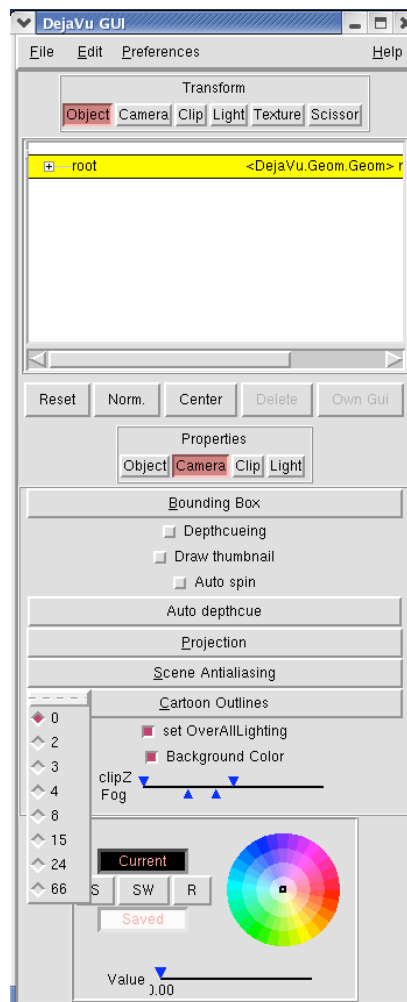
Choose 15. Notice the changes to the lines.

Note: Now you can also set scene antialiasing from the PMV Menu bar using:

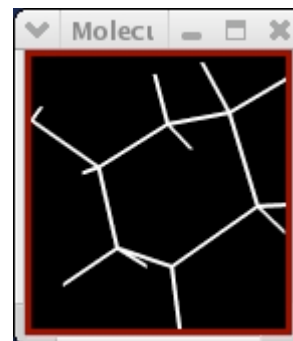
3D Graphics>SetAntialiasing

and clicking on the desired **Value** in the **SetAntialiasing** widget which opens.

Camera Property Panel



no anti-aliasing



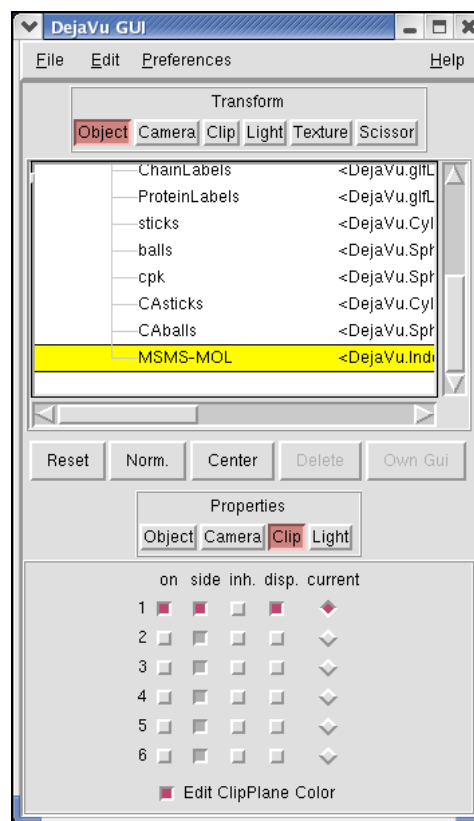
with anti-aliasing

Note: **Clipping planes** can be bound to any displayed geometrical object which they divide into visible or displayed and invisible or undisplayed sections. The position and orientation of the plane can be controlled by the user.

6. Activate and position a clipping plane

Display Clip Property Panel by clicking on **Clip** in the Properties Panel Selector.

Clip Property Panel



Note: If you rotate a clipped scene, you may notice that some geometries appear to disappear. This is because **Culling** is set to **back** by default. In the PropertyPanel, set it to **None** instead. You can control the representation of these back faces. Experiment by setting the **Back** to **Fill** or **Line**

Open the object tree by clicking on the **+** next to **root** in the object list.

Show the list of **hsg1** geometries by clicking on the **+** next to **hsg1** and select **MSMS-MOL**. Now the msms surface is the current object of the DejaVu Viewer.

Clicking the **'on'** checkbox enables a clipping plane. When you enable a clipping plane, it is bound to the Viewer's CurrentObject. You must set the object **before** enabling the clipping plane.

Enable the first clipping plane and display it by clicking on the checkbox under **'disp'**.

Note: Compute->Secondary structure->Ribbon if you have not already done so.

Toggle which side of the clipping plane is removed by clicking on checkbox under '**side**.'

Edit its color. Changing the clipping plane color is useful when you are working with more than one clipping plane.

Set the Transform target to **Clip**. Use the mouse to position the clipping plane.

Now select hsg1's **secondarystructure** in the Viewer's Object Tree making it the CurrentObject. Look at the Clip Property Panel. Note that **on** is disabled for the current clipping plane because its associated geometry is no longer the CurrentObject in the Viewer. Make the second clipping plane current by clicking on the radiobutton at the right of the second row.

Click 'on' in this row. Nothing happens because secondarystructure is a geometry container which does not display any geometry.

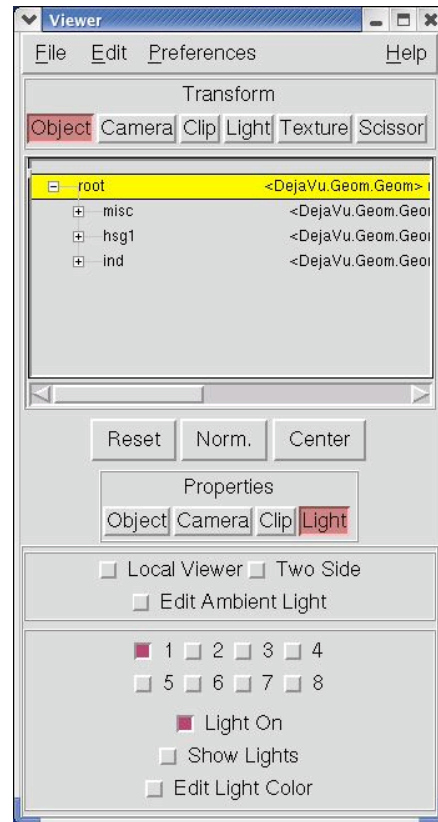
Click on the **inherit** checkbox labelled "**inh**". Now all the component secondary structure geometries are clipped.

Click on '**side**'. Note that secondarystructure is clipped on 1 side and MSMS-MOL on the other side.

7. Customize the lights.

Display the Object Property Panel using the Object button in the Property Panel Selector

Light Property Panel



Show the current light as a line by clicking on Show Lights. Set the Transformation Target to Light and use the mouse to position the light. Notice the changes to the image. Customize the color of the ambient, specular and diffuse light by clicking on the Edit Light Color button to open a color wheel. Add a second light by clicking on the checkbox labelled 2. Position and color it.

Note:
RenderLargeImage
exists in PMV

8. Save the image to a file.

File → **Save** → **Save image as**

In the SaveImage widget which opens, you can choose to have a Transparent Background but images with transparent backgrounds can only be written to **png** files. You can either enter a filename or click on **Choose** to open a filebrowser. The extension of the filename determines the filetype. PMV supports writing TIFF, JPEG, GIF, PNG, PPM, EPS, IM and PDF files.

Click on ‘-’ in file browser for a list of supported formats.

Type in **first_image.tif**.

Click on **OK** to write the file and **DISMISS** .

Use xv to look at the new image:
xv first_image.tif

Summary: what have we learned?

1. DeJaVuGUI allows you to interactively modify images.
2. The background color can be modified.
3. Properties of materials used to display geometries can be customized.
4. Any geometry in the Viewer can be clipped and clipping planes can be enabled for multiple geometries.
5. Geometries can be moved independently.
6. Light sources can be positioned and colored.
7. Images can be saved to files of a variety of types.

Appendix 2: Selecting a subset

Note: Atoms in the entities which are currently selected are marked with yellow 3D crosses. Info bar shows both the current level and the current number of selected entities. You can set the current selection level directly using: **Select->Set Selection Level**. In previous versions of PMV, setting the PCOM level also set the selection level. The PCOM level and the selection level are now separate.

Note: if there is no current selection, PMV expands the selection to include all atoms in the viewer. If the userpref, 'warnOnEmptySelection' is set to 1, PMV will ask you if it should "expand empty selection to all molecules." The default behavior is to not ask you if you want the empty selection to be expanded to include every molecule in the viewer.

In PMV the **current selection** is a particular homogeneous subset of the Atoms, Residues, Chains or Molecules currently loaded in the Viewer. It cannot contain entities of different levels such as Residues and Atoms. The current selection plays a pivotal role in PMV because most commands (display, undisplay, color, label, etc...) operate on the current selection. The current selection is dynamic: it can be modified, saved, cleared or restored.

This appendix covers the four principal ways of constructing the current selection in **PMV**:

- directly selecting molecules, chains or sets
- selecting by picking objects in the Viewer
- specifying strings to match at specific levels
- specifying center(s) and a radius for spherical regions of selection.

In this exercise we will experiment with building selections using these methods. We will build, clear, invert and save selections and examine the effects of changing the selection level. [Read in hsg1.pdb and indinavir if necessary]

Procedure:

1. Load all the selection commands:

File → **Browse Commands**

Select **Pmv** package. Highlight **selectionCommands** in the list of available PMV modules. Click on **Load Module** to load it. Click on **DISMISS** to close the widget.

Note: you load what you need when you need it. You can observe the effects of loading the selectionCommands module if you display (by clicking on it) and tear off the **Select** menu before you load selectionCommands module. **Also**, you can view the documentation strings of individual commands in modules listed by clicking on the Show Documentation checkbox and selecting a command. If the selected command has inline documentation, it will be displayed below the Show documentation checkbox.

2. Construct the current selection using **Direct Select**:

Note: you can use the Python shell to check the current selection at any time. After selecting both hsg1 and ind, open the Python shell and type:
`mv.getSelection().name`
The result is a list of names of the currently selected molecules:
`['hsg1','ind']`
See [Appendix 3](#) for more examples.

Select → **Direct Select**

This opens a widget which lets you select a **Molecule**, **Chain** or user-defined **Set** present in PMV by clicking on its checkbox. Clicking on an ‘active’ checkbox deselects. This is a very simple way to set the current selection.

Try these operations using this widget:

Select both **hsg1** and **indinavir**

Deselect **hsg1**

Select chain **hsg1:W**

You can always empty the current selection by clicking on

clear selection which is located on Button bar.

It is also possible to **invert** the selection which deselects everything in the current selection and selects everything else. This is done via **Select** → **invert selection**. The level for the inversion can be **all** the molecules in the Viewer or, if all the selected entities belong to a single molecule, a single **molecule**.

Try selecting one chain in hsg1, then invert the selection using ‘**all**’ as the level. Clear the selection and try it again using ‘**molecule**’.

At any time the current selection can be saved as a set. This lets you easily restore a previous selection.

Use the **Select** → **Save current selection as a set** to save chain **W** as “**waters**”. **clear selection** Restore it using **Select** → **Select a set** or using **Sets List...** in the Direct Select widget.

2. Construct a subset using picking:

In this section, we will use the mouse to select subsets and investigate the effects of changing the picking level.

Note: typing **Ctrl-s** with the cursor in the Viewer is a short cut to setting the PCOM to **select**. Typing **Ctrl-d** is a short cut for setting it to **deselect**.

Try these operations after setting **PCOM** to **select**:
Click on the down arrow to the right of the PCOM

window to display a list of possible PCOM commands.
Choose **select**.

PCOM → **select**

Set the PCOM level to **Atom**.

Pick on any vertex in **hsg1**.

Change the selection level to **Residue** using

Select → **Set Selection Level**

Notice that if any atom in a Residue was previously selected, now the selection is expanded to include all the atoms in that residue.

Try selecting two Residues in hsg1 by setting the PCOM level to Residue and holding down the left-button while you move the mouse. Save the selected residues as a set named **"two_Residues"**.

Restore 'two_Residues' using **Select** → **Select a set**.

Change the selection level to **Atom**. Note the entities in the current selection change to 13 Atom(s) but the selection does not expand.

Bind **Shift-PCOM** to **deselect**.

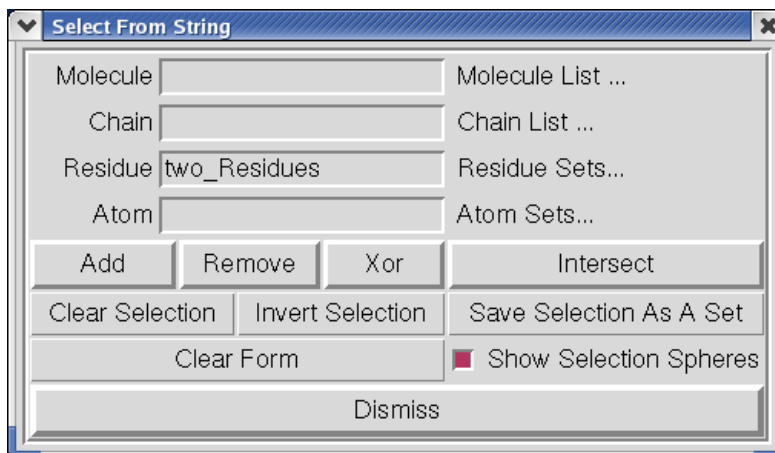
Try it out!

3. Use strings to specify a selection.

Note:

1. The 4 entries specify strings which are matched at the molecule, chain, residue and atom levels resulting in a new group of entities.
2. The menus which drop down from the Molecule List..., Chain List..., Residue Sets..., Atom Sets... **buttons** let you add or remove strings by clicking on checkboxes.
3. The buttons **Add**, **Remove**, **Xor** and **Intersect** define how the specified subset will be combined with the current selection:
currentselection **op** newgroup
4. When the current selection is empty, the only allowable operation is Add. Therefore, the buttons Remove, Xor and Intersect are disabled when nothing is selected.

Select → **Select From String**



Select From String lets you build a selection based on strings you enter for the Molecule, Chain, Residue

and/or Atom level. These strings can be names, numbers, ranges of numbers, or lambda expressions which are evaluated to build a set. A comma-separated list of items can be entered in any entry. The strings can contain regular expressions including wild cards such as * which match anything. This command also matches residue sequences in single letter format. In addition, it supports selecting user-defined sets as well as predefined residue and atom sets.

For example, to select all atoms (*) in residues named HOH* type **HOH*** in the Residue entry and type * in the Atom entry. Now click **Add**. You get a warning asking you if you want to “change selection level to Atom”: click **Yes**. [127 Atoms]

Try these, clear selection and clear form after each:

To select all alanine residues:

type “**ALA***” in the Residue Entry [6 residues]

To select residues 23 and 47 in ChainB:

type “**B**” for Chain, and “**23, 47**” for Residue.

[hsg1:B:LEU24, GLY48]

To select **aromatic** residues:

type “**aromatic**” for Residue. [12 residues]

To select all atoms whose name starts with **N**:

type “**N***” for Atom [67 atoms]

Clear the form but keep N* selection for next step:

To deselect all atoms in Chain **B**:

type “**B**” for Chain, “*” for Residue, “*” for

Atom. Now click **Remove**. [36 Atoms]

Note: the named residue sets are similar to the ones used by **Rasmol**. For details on these sets, please see Appendix 2

Integers are matched to position. 0-2 selects the first 3 atoms in the Viewer. #integers are matched to relative positions.

For example:

Typing 0 for Atom, selects the first atom in the first Molecule in the Viewer. [1 Atom]

Typing #0 for Residue, selects the first Residue in each Chain in the Viewer. [4 Residues]

To select a specific sequence, type in the Residue entry:

ITLW

This selects 8 residues because hsg1 has 2 identical Chains.

To select all the atoms with one bond, type in the Atom entry:

lambda x: len(x.bonds)==1

[517 atoms]

4. Select within spherical regions.

Select → **SphericalRegion**

Note: if you select using all the atoms in indinavir as centers for selection spheres vs **all** the molecules, ind will be selected. If instead you select using all the atoms in ind vs **hsg1**, the resulting selection will contain atoms in **hsg1** only. That is, in this case the selected atoms in ind will be deselected.

This command lets you select atoms in specified molecules which are within a given distance of another atom or atoms. The reference atom or atoms can either be the current selection or a picked atom. The radius of selection can be dynamically adjusted. You specify the basis set of molecules from which the selection is made: either all molecules in the Viewer OR molecules which you select in a list of all molecules in the Viewer OR a user-defined saved set.

To select all the residues in hsg1 which are within 4 angstrom of any atom in indinavir:

1. **Direct Select** 'indinavir'
2. **Select** → **Set Selection Level** Set the level to **Atom**.
3. **Select** → **SphericalRegion**

In the widget which opens:

Choose "**current selection**"

Set radius to **4**.

Choose "**from list**" and highlight '**hsg1**'

Click **Select** and **Close**.

This results in selecting 65 atoms in hsg1.

4. **Select** → **Set Selection Level** Set the level to **Residue**. The 36 residues in hsg1 with at least one atom within 4 Angstrom of any atom in indinavir are now selected.

Summary: what have we learned?

1. The current selection is must be homogeneous: it cannot be made up of Residues and Atoms. It is dynamic: it can be modified, saved, cleared or restored.

2. Sets of entities in the Viewer have stringRepr attributes which are shorthand representations of the set. See Appendix3 for more details.
3. When you make a selection by choosing some entities, you can combine them with the current selection using Boolean operations: add, remove, intersect and xor. When you clear the selection, all currently selected entities are removed from the selection.
4. Changing to a higher level expands the selection to include any non-selected siblings. Changing to a lower level does not expand the selection.
5. When you invert the selection, all the currently selected entities are removed from the current selection and all the currently unselected entities are added to the selection.

Appendix 3: Working with MolKit in PMV's Python shell

In the Python shell, PMV itself can be referred to as 'mv'. All molecules loaded into PMV are added to a molecule set stored in the 'Mols' attribute of the mv object.

```
>>> mv.Mols
```

```
<MoleculeSet instance> holding 2 Protein, "hsg1, indinavir"
```

Molecules have a hierarchical structure with 4 levels (Molecule, Chain, Residues, Atoms).

Below are some examples of navigating and inspecting this hierarchical structure:

```
# sets can be indexed, here we get a handle to the first molecule
>>> mol = mv.Mols[0]
```

```
# we print the name attribute of the molecule object
>>> print mol.name
```

```
# print the name attribute of each molecule in the set
>>> print mv.Mols.name
```

```
# find out about object attributes:
# print the name of all attributes of the 1st chain of the 1st molecule
>>> dir(mv.Mols[0].chains[0])
```

```
# navigate the tree
```

```
# print the names of all residues in all chains in mol
>>> print mol.chains.residues.name
```

```
# build a list of all atoms in all residues or all chains of mol,
# and get the subset of atoms 20 through 84
# and call the 'full_name' method on each of these 64 atoms
>>> print mol.chains.residues.atoms[20:85].full_name()
```

```

# ask a molecule to report all of its atoms (as an AtomSet)
>>> from MolKit.molecule import Atom
>>> allAtoms = mol. findType(Atom)

# select all atoms with the attribute temperatureFactor larger
than 20.
>>> set1 = allAtoms.get("lambda x: x.temperatureFactor
>20.0")

# new attributes can be created interactively.
# Here we compute the geometric center of all atoms
# and save it in the residues
>>> allResidues = allAtoms.parent.uniq()
>>> import Numeric
>>> for r in allResidues:
...     coords = r.atoms.coords
...     r.geomCenter = Numeric.sum(coords) / len(coords)

```

Selecting molecular fragments in PMV

Pmv's molecules are stored in a hierarchy of 4 levels. One can select molecular fragments by selecting subsets at each level of the hierarchy. However, a selection string must always start at the molecule level.

The general syntax of a selection string is:

Mol_selector:chain_selector:residue_selector:atom_selector

- can be used to specify ranges example
- # are positions
- * matches any number of any characters
- , separated lists of selectors
-

Multiple selection strings have to be separated by a ';' semi-colon character

The results of a selection can be added, subtracted, intersected, xor'ed using the +, -, &, ^ operators

Example:

An atom the nitrogens in hsg1 alanine residues plus all the carbons in indinavir

```
hsg1::ALA*:N/+/:indinavir::C*
```

The **s** operator allows select a subset from a previous selection

Get all the backbone atoms in the viewer

```
bb = mv.get('::backbone') # other set names are in Appendix  
2
```

```
bb = bb + mv.get("::CYS36:/+/:HIS*:/-/:H*")
```

Appendix 4: PMV StartUp Options

Note: the flags for display mode and color mode must precede the list of molecule filenames.

PMV will also automatically display molecules according to a display mode entered on the command line following **-d** flag and a color mode entered following **-c** flag. Type this at the shell prompt to display these molecules as cpk colored by residue:

```
pmv -d cpk -c cr hsg1.pdb indinavir.pdb
```

Here are the currently recognized **command line** arguments:

| | |
|---------------------|---|
| -h or -help | : print this message |
| -i | : use Unix shell as the Python shell |
| -a or -again | : play back last log file |
| --overwriteLog | : overwrite log file |
| --uniqueLog | : create a log file with a unique name |
| --noLog | : turn off logging |
| --die | : do not start GUI event loop |
| --customizer file | : run the user specified file |
| --lib packageName | : add a library of commands |
| -d or --dmode modes | : specify a display mode which can be any combination of display modes |
| | 'cpk' : cpk |
| | 'lines' : lines |
| | 'ss' : secondary structure ribbon |
| | 'sb' : sticks and balls |
| | 'ms' : molecular surface |
| | 'ca' : C-alpha trace |
| | 'bt' : backbone trace |
| | 'sp' : CA-spline |
| | 'sssb' : secondary structure for proteins, sticks and balls for other residues with |
| bonds | lines for other residues without bonds |
| -c or --cmode modes | : specify a display mode color scheme |
| | : |
| | 'ca' : color by atom |
| | 'cr' : color by residue (RASMOL scheme) |

‘cc’ : color by chain
‘cm’ : color by molecule
‘cdg’ : color using David Goodsell’s scheme
‘cs’ : color residues using Shapely scheme
‘css’ : color by secondary structure element

For example: to display protein as ribbon, non-protein as sticks and balls and color by secondary structure element:

pmv -d sssb -c css hsg1.pdb indinavir.pdb

Restart PMV restoring hsg1 and indinavir as lines colored by atom before going on to Exercise Two.

pmv -d lines -c ca hsg1.pdb indinavir.pdb

Appendix 5: PMV Modules

Note: Each module is listed by name and followed by a list of the commands in it. Commands in **bold** were used in this tutorial.

APBSCommands.py

APBSSetup
APBSRun
APBSMap_Potential_to_MSMS
APBSDisplay_Isocontours
APBSLoad_Profile
APBSSave_Profile
APBSWrite_APBSParameter_File
APBSPreferences

bondsCommands.py

BuildBondsByDistance
AddBondsCommand
RemoveBondsCommand

colorCommands.py

Color
ColorFromPalette
ColorByAtomType
ColorByDG
ColorByResidueType
ColorShapely
ColorFromRamp
ColorByChain
ColorByMolecule
ColorByInstance
ColorByProperties
ColorByExpression

colorPaletteCommands.py

EditColorPaletteByAtomType

dashboardCommands.py

FloatDashboard
DockDashboard
DashboardSuspendRedraw
ShowDashboard
AddDashboardCmd

deleteCommands.py

DeleteMolecule
DeleteAllMolecules
DeleteAtomSet
DeleteHydrogens

displayCommands.py

- DisplayLines**
- UndisplayLines**
- DisplayCPK**
- UndisplayCPK
- DisplaySticksAndBalls**
- DisplaySSSB**
- UndisplaySticksAndBalls
- DisplayBackboneTrace**
- UndisplayBackboneTrace
- ShowMolecules**
- BindGeomToMolecularFragmentBase
- BindGeomToMolecularFragment
- DisplayBoundGeom
- UndisplayBoundGeom
- DisplayComplexCommand

editCommands.py

- TypeAtomsCommand
- EditAtomTypeCommand
- TypeBondsCommand
- AddHydrogensCommand
- FixHydrogenAtomNamesCommand
- MergeNPHsCommand
- MergeLonePairsCommand
- ComputeGasteigerCommand
- AddKollmanChargesCommand
- AssignAtomsRadiiCommand

extrusionCommands.py

- ComputeSheet2DCommand
- DisplayPath3DCommand

fileCommands.py

- PDBReader**
- MMCIFReader
- GROReader
- PDBQReader**
- PDBQSReader**
- PDBQTReader
- PQRReader**
- MOL2Reader
- PDBWriter
- PQRWriter
- PDBQWriter
- PDBQSWriter
- PDBQTWriter
- SaveMMCIF
- STLWriter
- VRML2Writer
- readFromWeb**

genparserCommands.py

GenPDBReader
DefinePdbSpecifications

grid3DCommands.py

readAnyGrid
Grid3DCommands
AddRemove
IsocontourCommand
OrthoSliceCommand
VolRenCommand

gridCommands.py

GridReader
AutoGridReader
DXReader
AutoGridReader
DeleteAutoGrid
AutoGridIsoSurface
SetIsovalue
GetOrthoSlice

hbondCommands.py

GetHydrogenBondDonors
GetHydrogenBondAcceptors
GetHydrogenBondEnergies
ShowHBDonorsAcceptors
ShowHydrogenBonds
BuildHydrogenBonds
AddHBondHydrogens
LimitHydrogenBonds
DisplayHBonds
DisplayHBondsAsSpheres
DisplayHBondsAsCylinders
ExtrudeHydrogenBonds
WriteAssemblyHBonds
WriteIntermolHBonds
ReadIntermolHBonds
AddHBondComman
RemoveHBondCommand

helpCommands.py

mailingListsCommand
BugReportCommand

interactiveCommands.py

MVSetInteractiveCmd
MVIcomGUI
MVBindCmdToKey

labelCommands.py

LabelByProperties

LabelByExpression

measureCommands.py

MeasureAtomCommand

MeasureDistance

MeasureAngle

MeasureTorsion

msmsCommands.py

ReadMSMS

SaveMSMS

ComputeMSMS

DisplayMSMS

UndisplayMSMS

ComputeMSMSApprox

MsmsNPR

IdentifyBuriedVertices

DisplayBuriedTriangles

DisplayIntermolecularBuriedTriangles

AssignBuriedAreas

multiresCommands.py

GoCoarserFiner

goCoarserFiner

Multires

pdb2pqrCommands.py

SetupPdb2Pqr

povrayCommands.py

Povray

repairCommands.py

CheckForMissingAtomsGUICommand

CheckForMissingAtoms

CheckForCloseContactsGUICommand

CheckForCloseContacts

RepairMissingAtomsGUICommand

RepairMissingAtoms

EditHistHydrogensGUICommand

EditHistHydrogens

AddOXTGUICommand

AddOXT

ModifyCTerminus

ModifyNTerminus

ModifyTermini

secondaryStructureCommands.py

ComputeSecondaryStructureCommand

ExtrudeSecondaryStructureCommand
DisplayExtrudedSSCommand
ExtrudeSecondaryStructureCommand
DisplayExtrudedSSCommand
RibbonCommand
ColorBySSElementType

selectionCommands.py

MVClearSelection
MVSaveSetCommand
MVCreateSetIfNeeded
MVSelectSetCommand
MVSelectFromStringCommand
MVDirectSelectCommand
MVSelectSphericalRegion
MVInvertSelection

setangleCommands.py

SetRelativeTorsion
SetTranslation
SetQuaternion
SetTorsion

splineCommands.py

ComputeSplineCommand
ComputeSplineGUICommand
ExtrudeSplineCommand
DisplayExtrudedSplineCommand
DisplaySplineAsLineCommand
CustomSplineCommand
ComputeExtrudeSplineCommand

traceCommands.py

ComputeTraceCommand
ExtrudeTraceCommand
DisplayTraceCommand
ComputeExtrudeTraceCommand
ExtrudeSecondaryStructureCommand
DisplayExtrudedSSCommand
RibbonCommand
ColorBySSElementType

trajectoryCommands.py

OpenTrajectoryCommand
PlayTrajectoryCommand

updateCommands.py

Update

writeMsmsAsCommands.py

WriteSTL
WriteColoredM

Appendix 6: Mouse Modifiers and Keystrokes recognized by the DeJaVu Viewer

Mouse button modifiers:

- None
- Shift
- Control
- Alt
- Meta

Keystrokes:

- R/r
- N/n
- C/c
- D/d
- T/t
- KeyRelease
- KeyPress

Appendix 7: Customization Options and Default Values for User Preferences

Customization Options for PMV

_pmvrc
user_prefs

user_prefs defaults:

| | |
|-------------------------------|---|
| 'warnOnEmptySelection' | no /yes ask user before expanding to all |
| 'NumberOfUndo' | 100 # of cmds that can be undone |
| 'centerScene' | firstMoleculeOnly center scene on new molecule |
| 'useDepthCueing' | yes /no depthcueing on by default |
| 'showProgressBar' | show /hide progress bar |
| 'useBusyCursor' | 0 /1 change cursor shape while cmd running |
| 'changeCursor' | 0 /1 change cursor shape <window may flash!> |
| 'expandNodeLogString' | 1 /0 use fullname of nodes in log string or not |
| 'transformationLogging' | final /continuous/no when transformations are logged |
| 'showSelectionSpheres' | 0 /1 display yellow crosses on selected atoms |
| 'visualPickingFeedBack' | 0 /1 display sphere on picked vertex |
| 'fillSelectionBox' | 1 /0 draw box around drag select after delay |
| 'fillSelectionBoxDelay' | 200 msec w/out motion before drawing box |
| 'raiseExceptionForMissingKey' | 0 /1 raise exception if key not in dict |
| 'warningMsgFormat' | pop-up /printed format for warning messages |
| 'inputFormInitializationMode' | lastValue /defaultValue |
| 'measureDistanceSL' | 4 # of labelled distances displayed |
| simultaneously | |
| 'measureAngleSL' | 4 # of labelled angles displayed simultaneously |
| 'measureTorsionSL' | 4 # of labelled torsions displayed |
| simultaneously | |
| 'continuousUpdateDist' | yes /no update distance if move objects independently |
| 'continuousUpdateAngle' | yes /no update angles if transform objects independently |
| 'continuousUpdateTorsion' | yes /no update torsions if transform objs |
| independently | |
| 'selectStringMatchMode' | caseSensitive /caseInsensitive/CIWEC |
| 'Startup Directory' | current working directory #can be set in |
| _pmvrc | |

You can change the user_prefs values via:

File → **Preferences** → **Modify Defaults**

See also the following how-tos:

- How-to change default parameters using _pmvrc:
<http://mgltools.scripps.edu/documentation/how-to/changing-default-parameters>
- How-to change Camera size on start-up or on-the-fly:
<http://mgltools.scripps.edu/documentation/how-to/ChangingCameraSize>
- How-to change the default font:
<http://mgltools.scripps.edu/documentation/how-to/change-the-default-font>

Appendix 8: Definitions of Named Residue and Atom Sets recognized by PMV

Named Residue Sets:

'acidic' = asp, glu
'acyclic' = ala, arg, asn, asp, cys, glu, gln, gly, ile, leu, lys, met, ser, thr, val,
'aliphatic' = ala, gly, ile, leu, val
'aromatic' = his, phe, trp, tyr
'basic' = arg, his, lys
'buried' = ala, cys, ile, leu, met, phe, trp, val
'charged' = arg, asp, glu, his, lys
'cyclic' = his, phe, pro, trp, tyr
'hydrophobic' = ala, gly, ile, leu, met, phe, pro, trp, tyr, val
'large' = arg, glu, gln, his, ile, leu, lys, met, phe, trp, tyr
'medium' = asn, asp, cys, pro, thr, val
'negative' = asp, glu
'neutral' = ala, asn, cys, gln, gly, his, ile, leu, met, phe, pro, ser, thr, trp, tyr, val
'polar' = arg, asn, asp, cys, glu, gln, his, lys, ser, thr
'positive' = arg, his, lys
'small' = ala, gly, ser
'surface' = arg, asn, asp, glu, gln, gly, his, lys, pro, ser, thr, tyr

Named Atom Sets:

'backbone' = C, CA, N, O
'backbone+h' = C, CA, N, O, HN, HN1, HN2
'sidechain' = SG, SD, CB, CG, CD, CD1, CD2, CE, CE1, CE2, CE3, CG1, CG2, CZ, CZ2, CZ3, CH2, ND1, ND2, NE, NE1, NE2, NH1, NH2, NZ, OD1, OD2, OG, OG1, OE1, OE2, OH, HD1, HE1, HE2, HE, HE21, HE22, HH23, HH11, HH12, HG, HG1, HH21, HD22, HD21, HZ1, HZ2, HZ3, HH

One Character Residue Names:

ALA=A, ARG=R, ASN=N, ASP=D, ASX=B, CYS=C,
GLU=E, GLN=Q, GLX=Z, GLY=G, HIS=H, ILE=I, LEU=L,
LYS=K, MET=M, PHE=F, PRO=P, SER=S, THR=T, TRP=W
TYR=Y, VAL=V, XAA=X, SEL=U

#www.ensembl.org/Docs/Pdoc/bioperl/Bio/SeqUtils.html