
The Python based Molecular Viewer.
(PMV)

Tutorial presented by Sophie Coon, MGL.
(sophieco@scripps.edu)

PMV Tutorial

The Python-based Molecular Viewing Environment (PMV) is a general purpose viewer that can be integrated into any computational chemistry package available in Python. It relies on DejaVu for the 3-Dimensional visualization, ViewerFramework for the definition of individual commands mglutil for the GUI and MolKit for the representation of molecules. These components are independent and reusable in another context.

After getting a first contact with Pmv the user will be introduced in the second part of this tutorial to the fundamental concepts of the software. The third section will then present more advanced manipulations available in Pmv, and finally the last section will briefly present the customizable aspect of Pmv. This tutorial is intended to get the user comfortable using Pmv but also to demonstrate some of the features that set Pmv aside from other typical molecular viewers programs.

Install and start pmv:

If a working Pmv is already available on the user's machine start it and the window of the Fig1 should appear on your screen.

From a unix machine on the TSRI network:

cshell users:

```
% source /tsri/python/share/bin/initpmvcsh
```

bash shell users:

```
% source /tsri/python/share/bin/initpmvbash
```

Then start PMV by typing in your terminal window the following:

```
% pmv
```

The following window should then appear on your screen. If not please contact Sophie Coon (sophieco@scripps.edu).

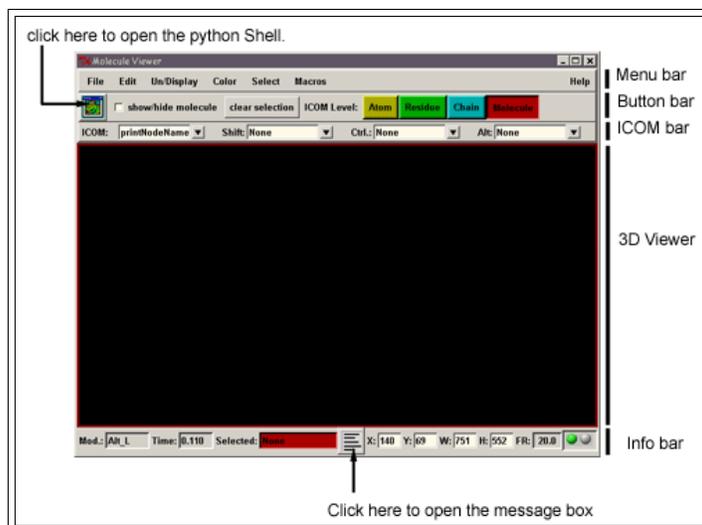


Fig1: PMV at start up.

From either a windows machine, a MacOSX or a unix/linux machine not on the TSRI network:

The user will need to download and install the complete release of the MGLTools from our download site (<http://www.scripps.edu/~sanner/software/packager.html>) for the chosen platform. Please contact Sophie Coon (sophieco@scripps.edu) for any installation problems.

Basic commands:

When PMV is started, a number of menus appears on the menu bar which indicates that commands have automatically been loaded at start up:

- File Commands under the File menu
- Color Commands under the Color menu
- Display Commands under the Display menu
- etc ...

The user can choose which modules and commands will be loaded at start up, this will be presented in the section "Extending PMV".

Reading the first molecule:

PMV has an extensible set of molecular data files parsers. It currently supports reading:

- PDB from the Protein Data Bank.
- PDBQ and PDBQS AutoDock formats.
- PQR Mead format.
- Mol2 Tripos format.

Step 1: Read the "protease.pdb" file describing the HIV protease.

- | |
|--|
| <ul style="list-style-type: none">• File -> Read Molecule• select protease.pdb in the file browser• click on the Open button to read the file. |
|--|

The red LED at the bottom right corner of the application turns red while PMV is reading the file. This LED turns red when the program is busy, after it completes a task the LED turns back to green. The time it took to PMV to execute the command is printed in the entry box labeled "Time" on the info bar.

The HIV protease is now loaded in PMV and displayed as lines (Fig 2.).

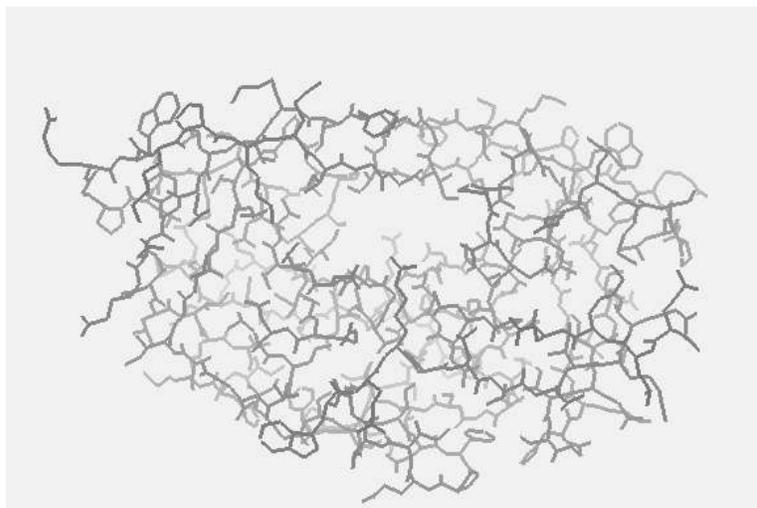


Fig 2: HIV protease (protease.pdb) displayed as lines in PMV.

Basic interactions with the viewer:

Once a molecule is loaded in the application, the user can interact with it using the mouse buttons. The table below lists the default mouse bindings for a 3-buttons mouse. On a laptop with a 2-buttons mouse pad, clicking on the two buttons simultaneously usually emulates the middle button of the 3-buttons mouse.

Action	Mouse Button	Modifier
Rotation	Middle / Wheel	None
Scaling	Middle / Wheel	SHIFT
XY Translation	Right	None
Z Translation	Right	SHIFT
Picking	Left	None

Those bindings can be modified by using the “bind action to mouse button” command located under the “File -> Customize” menu.

If the molecule moves out of the viewport, the user can always do the following to get it back:

- Use the **R** or **r** key to reset all the transformation performed so far on the molecule.
- Use the **N** or **n** key of your keyboard to normalize the scene.
- Use the **C** or **c** key of your keyboard to set the center of rotation at the center of the molecule. to the center of your molecule.
- The **D** or **d** key of your keyboard allows you to turn on or off the depth cueing.

Basic graphical representations of a molecule in PMV:

Color commands:

One way to alter the representation of a molecule is to change its color. PMV provides several coloring schemes available in the "colorCommands" module. Some of these commands have been loaded by default in PMV and are available under the "Color" menu.

Color the protease by atom type

This coloring scheme gives information on the protein atomic composition.

- Color -> By Atom type
- Select the check box labeled **lines** to color the lines geometry representing the molecule. This is the only choice because right now the protease is only represented by lines
- Click on the OK button to carry out the color command.

Now all the Carbon atoms of the protease are colored in gray, the Oxygen atoms in red, the Nitrogen atoms in blue and the Sulfur atoms in yellow.

Color the protease by residue type

Coloring the protease by residue type using either the **Rasmol** (Fig 3A) or the **Shapely** (Fig 3B) coloring schemes will give the user some information on the residue composition of the protein. In both schemes the polar residues, which can be found on the surface of the protein, have a bright color whereas the non-polar residues have a darker color.

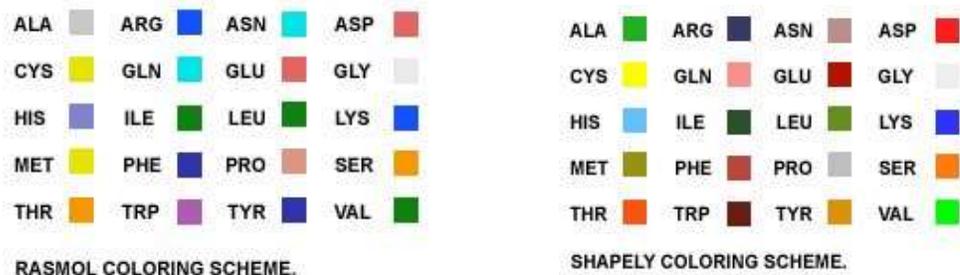


Fig 3: (A) Rasmol Coloring scheme, (B) Shapely Coloring scheme

Color the protease by Chain or by Molecule

Coloring a molecule by chain allows the user to distinguish the various parts of a multimeric structure whereas coloring by molecules will distinguish the different molecules loaded so far in the application.

Both the **By Chain** and **By Molecule** coloring schemes use a predefined palette of 20 different colors.

Display Commands:

The user can also alter the representation of a molecule by changing the geometry used to represent it. PMV provides a set of display commands available in the displayCommands module. A subset of these commands has been loaded at start up and is located under the **Un/Display** menu. In PMV, a molecule can be represented by several geometries at the same time and each

geometry can be colored using a different color scheme.

Display the protease by CPK

- Un/Display -> CPK
- In the display panel: select "display", set the scale factor to 1, and the quality to 10.
- Click on the OK button to carry out the command with the chosen parameters.

Each atom is, now represented by a sphere. This representation is also known as "Space filling" representation.

Color this new geometry using the "Shapely" coloring scheme

- Color -> By Residues -> Shapely
- In the Choose Geometry panel select the "CPK" geometry
- Click on the OK button to carry out the coloring command.

Display the protease by sticks and balls

When representing a molecule by "sticks and balls" each bonds is displayed as a cylinder and each atom as a sphere. The following parameters can be set by the user:

- the size of the spheres and the quality of the balls
- the size of the cylinder and the quality of the sticks
- whether or not to display the balls

Undisplay CPK and sticks and balls

PMV fundamentals

The selection in PMV

In PMV, commands are always applied to the **current selection**. However for convenience when nothing is selected, commands are applied to all the molecules in the viewer. This behavior can be modified through the setUserPreference command located under the File->Preferences menu. In this section the user will learn how to identify what the current selection is and how to create one.

In PMV, the molecules are represented by a 4 level hierarchical tree, which mirrors the inner hierarchy of a protein. These four levels are Molecule, Chain, Residue and Atom.

On the right hand side of the button bar there are 4 buttons labeled: "Molecule", "Chain", "Residue" and "Atom". These buttons allow the user to choose at which level of the molecule a command must be carried out.

A set of selection commands has been loaded by default at start-up and is available under the Select menu.

Select the chain B of the protease using the Select From String commands

- Select -> From String
- In the "Select From String" panel (Fig 4)
- click on **Chain List...** in the dropdown menu select the '**A**' checkbox. The type-

in entry labeled "Molecule" now contains "protease" string and the type-in entry labeled "Chain" the word "B".

- Click on the **Select** button to perform the selection
- In the panel that appears click **OK** to change the level of the selection from Molecule to Chain.

Several things happened:

The Chain Button level is now selected

The background color of the entry labeled selection is now Cyan and the selection holds 1 Chain.

Yellow crosses are now displayed on each atom of the current selection.

Display the chain B by CPK and color by Residues using the Shapely coloring scheme

- Display -> CPK
- Color -> By Residues -> Shapely

As expected only the chain B of the protease is now displayed by CPK and the CPK geometry is colored by residue type.

The current selection stays active until the user clears it using the clearSelection command that can be found on the button bar.

Clear the selection.

- Button bar -> Clear Selection

The "SelectFromString" command is a powerful command and as mentioned earlier the user can type a string in each entry that will be matched against the name of the nodes (Fig 3). The '*' character replaces any character, ',' allows the user to specify a list of nodes and '-' a range of nodes.

Select all the carbons all the oxygen atoms of all the isoleucine residues of the chain A of the protease.

- Select -> Select From String
- Click on the Clear Form and Clear Selection.
- In the type-in entry labeled Atom type: O*,C*
- In the type-in entry labeled Residue : ILE*
- In the type-in entry labeled Chain : A
- In the type-in entry labeled Molecule: protease
- Click on Select to perform the selection.
- Click on the OK button to change the level of the selection from Chain to Atom.

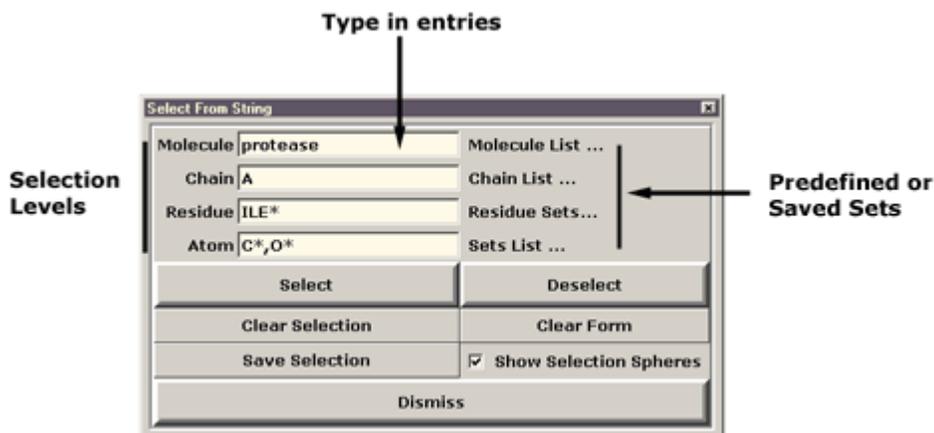


Fig 4: Select From String GUI.

- **type-in entries:** corresponding to the 4 levels of selection to allow the user to enter a string that will be matched against the name of the corresponding node.

Molecule : name of the file without the extension

Chain : chain ID of the pdb file

Residue : 3 letter AA name concatenated to the residue number and the residue seq ID

Atom : element type and the letter code and number for its position in the residue (CA for carbon alpha, O1).

- List of sets for each level is available by clicking on the label on the right hand side of the type in entry.

- **Select** button :to add to the current selection

- **Deselect:** to remove from the current selection,

- **Clear Selection:** to clear the current selection

- **Save Selection:** to save the current selection as a set

- **Clear Form:** to clear the form

- **Show Selection Spheres:** toggle on or off the selection spheres which are the yellow crosses.

- **Dismiss** button to withdraw the form.

Now the selection level is the "Atom" level and the current selection holds 26 atoms. If the user hadn't cleared the previous selection the new selected atoms would have been added to the previous selection.

In PMV the current selection is **homogeneous** which means that two residues and three atoms cannot be selected at the same time. If the user changes the selection level from Atom to Residue, the current selection will be expanded to all the residues having at least one atom in the current selection. When going back to the Atom level, all the atoms of the selected residues will now be part of the current selection.

Undo capability:

Most of the PMV commands are undoable using the undoCommands available under the "Edit" menu.

To undo the two "setlcom Level" commands

- Edit -> Undo Setlcom Level (to undo the command that changed the selection level from Residue to Atom)
- Edit -> Undo Setlcom Level (to undo the command that changed the selection

level from Atom to Residue.)

PMV keeps a stack of the last undoable commands that have been executed. By default the stack can hold up to 100 commands. This number can be modified through the `setUserPreference` command. When a molecule is deleted the stack will be reset.

Loading new commands and modules in PMV:

There are many more commands available in PMV than the subset used so far. New commands can be loaded dynamically by the user in the application at any time. Commands are grouped into modules, which are in turn grouped into packages.

Either a whole module can be loaded at once in the application or individual commands using the `browseCommands` command.

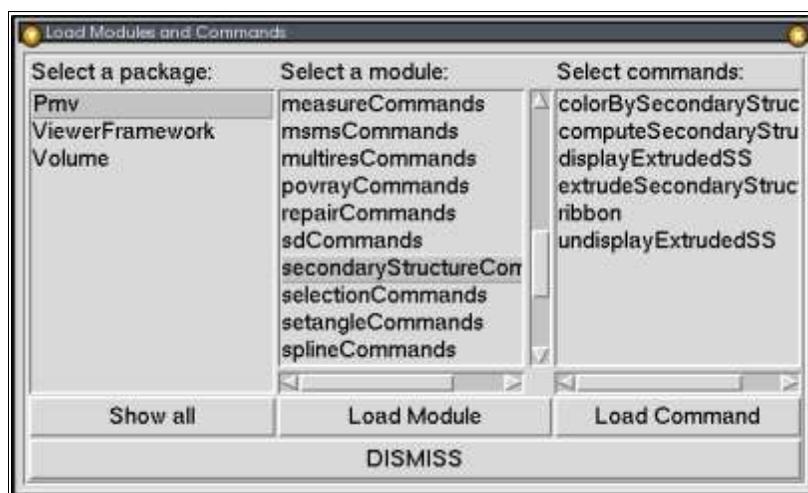


Fig : BrowseCommands GUI.

Load the secondaryStructureCommands module into PMV:

- File -> browseCommands
- Select the "PMV" entry in the "Select a package" listbox
- Select the "secondaryStructureCommands" in the "Select a module"
- Click on the "Load Module" button to load the selected module.
- Click on the "Dismiss" button to withdraw the panel

A new menu "Compute" appears on the menu bar with a cascade called "secondary structure". The menus "Color" and "Un/display" have also two new entries "By Secondary Structure Type" and "Secondary structure".

Load the computeMSM S and displayMSMS, undisplayMSMS commands in PMV:

- File -> browseCommands
- Select the "PMV" entry in the "Select a package" list, msmsModule being a PMV module.

- Select the “msmsCommands” in the Choose a module list, the commands to load are part of the msmsCommands module.
- Select the computeMSMS, displayMSMS and undisplayMSMS entries in the “Select Commands” list.
To make multiple selection either use the SHIFT modifier to select a range of entries or the CTRL modifier to select multiple entries.
- Click on the **loadCommand** button to load the commands in PMV and the **DISMISS** button to withdraw the panel.

A new cascade named “MSMS” under the “Compute” menu has been created and a new entry named “MSMS” under the “Un/display” menu has been created as well.

Picking commands:

So far all the commands were applied to the current selection, which was created using one of the select commands available in PMV.

Another way to carry out a command is to use the mouse. PMV maintains the relationship between the data structures representing a molecule and the various graphical representations of that molecule (called geometries). This relationship makes it possible to indicate molecular fragments by picking or dragging a box around their graphical representation. Such commands are called picking commands (PCOM). Any command that has only one required argument that describes a molecular fragment, can become a picking command. The printNodeName command is bound to the picking event by default. At the Molecule level when picking on the lines representing the molecule the name of the molecule will be printed in the message box. To open the message box click on the "message box" icon at the bottom of PMV(Fig1). Changing the PCOM level from Molecule to Atom will print the full names of the underlying atoms. Other command can be bound to the mouse event or to the mouse event and a modifier (SHIFT, CTRL, ALT, ...) using the PCOM combobox.

Bind the select command to the mouse:

- PCOM -> first combobox -> “select”

The PCOM level is Atom so when the user picks on the lines, the underlying atoms will be added to the current selection. In fact, the number of atoms in the current selection increases.

Picking commands can be also bound the picking event combined to keyboard modifiers such as SHIFT, CTRL, and ALT. Therefore you could have the select command bound to the picking event and the deselect command bound to the picking event combined with the SHIFT button.

Advanced manipulations

Secondary Structure representation:

In this section the user will represent the protease by a traditional ribbon diagram using the ribbon command from the secondaryStructureCommands module loaded previously.

Select the protease using the interactive command "select" as described above.

- Set the PCOM level to Molecule
- Pick on the lines representing the molecule to add it to the current selection.

Represent the current selection by a traditional ribbon diagram

- Compute -> secondary structure -> Ribbon
- Un/Display -> Lines -> undisplay

The protease is now represented by a traditional ribbon diagram where the:

- beta strands are represented by an extruded arrow
- helices by a extruded rectangle
- coils and Turns by a extruded circle.

In this representation, the two flaps of the HIV protease are really visible (Fig 4.)

The Secondary structure geometries, like any other geometry in PMV, can be colored using one of the available coloring commands.

Color the protease by secondary structure type

- Color -> Secondary Structure Type
- Select the secondary structure geometry

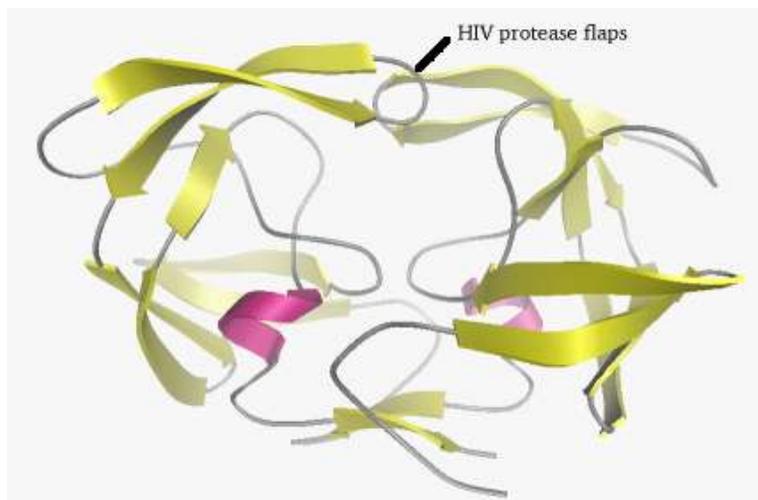


Fig 4: HIV protease represented as a traditional ribbon diagram colored by secondary structure. (beta strand: yellow, alpha helices: pink, coil: gray, turn: blue)

The "Ribbon" command is actually a shortcut to display the current selection using a traditional ribbon diagram. The "Ribbon" command executes first the "computeSecondaryStructure" command on the current selection then the "extrudeSecondaryStructure" command. The secondary structure can be represented by other geometries.

Represent the helices by an extruded ellipse:

- Select -> Select a set
- Select all the helices entries in the "Choose an item" using the Shift and Ctrl key to select several item at a time.
- Click on the OK button to carry out the selection.
- Compute -> Secondary Structure -> Extrude Secondary Structure

- Select ellipse in the "Choose a shape" listchooser
- keep or set the size of the demi-grand axis
- keep or set the size of the demi-small axis
- Add or not a front and /or a end cap
- Click on the **OK** button to extrude the chosen shape.

Now the helices of the protease are represented by extruded ellipses. Although, each secondary structure elements can use its own shape for the extrusion, only one shape is allowed by secondary structure element.

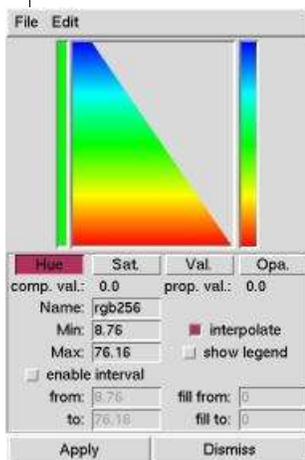
Representing the protease by a Molecular surface using MSMS

MSMS stands for Maximal Speed Molecular Surface.

To represent the protease using a molecular surface the user will need the Compute MSMS command loaded earlier.

Select the protease using the "Direct Select" command:

- Select -> Direct Select
- In the "Direct Select" panel that appears
- select **protease** in the Molecule List drop down check box



- click on the **Dismiss** button to withdraw the form.

Compute the solvent excluded surface of the molecule in the current selection

- Compute -> MSMS -> ComputeMSMS
- In the panel that appears set the following parameters.
 - The user can specify the name of the surface or use the default. If the surface name already exists the surface for that name will be recomputed with the new parameters.
 - If Per Mol is True a MSMS surface will be computed for each molecule in the current selection, else a MSMS surface will be computed for the current

- selection.
- 1.5 for the probe radius
- 3.0 for the density of triangles used to represent the surface.
- Click on the **OK** button to compute the surface of the molecule in the current selection.

The protease is now displayed using its MSMS surface. This new geometry can be colored using any coloring command available in PMV.

ColorByProperty and ColorByExpression commands

The introspection capability of Python allows to find out at runtime what attributes of molecular fragments can be used for coloring a given geometry.

ColorByProperty

The ColorByProperty command interactively builds a panel that lets the user choose the property to be used for coloring. The list of properties available depends on the selection level (Atoms have different attributes than Residues).

To color the msms surface by temperature factor

- load the colorByProperty and the colorByExpression commands
- Select the protease using the selection command of your choice.
- Color -> ColorByProperty : Choose 'msms' geometry.
In the "Color by properties" panel that appears:
- Select the Atom level to display the atoms properties in the "Property listchooser"
- Select temperatureFactor in that list
- Select the "Edit" radiobutton to edit the colormap
- Click on the OK button to carry out the color command.

The msms is now colored by temperature factor using a RGB ramp. The user can modify this color map using the colormap editor.

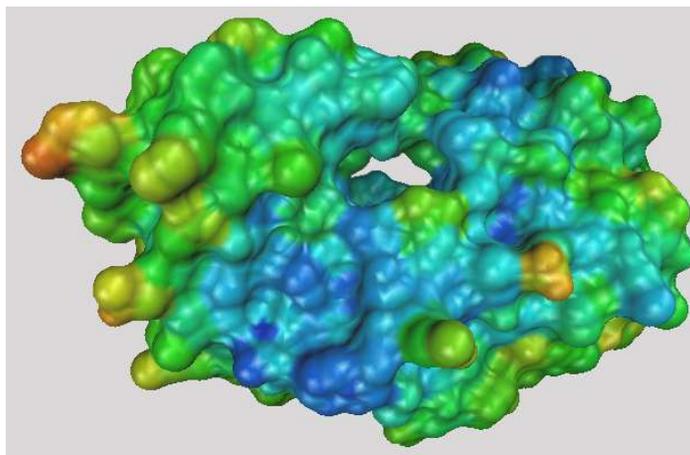


Fig5 : HIV protease (protease.pdb) represented by a msms surface colored by the atom property temperature factor using a RGB ramp.

Using this editor the user can for example decrease the opacity of the low values of the color map therefore the blue regions of the surface will then be more transparent.

Fig 6: HIV protease (protease.pdb) represented by a msms surface and secondary structure, colored by the atom property temperatureFactor using a RGB ramp where the low values are made more transparent.

Display and transform the colorMapLegend:

A colorMap legend is a DejaVu geometry associated to the colorMap. This legend will be added to the middle of the viewport.

To move the legend the user will need to do follow the steps from the “Transforming a geometry apart from the others” paragraph. The legend geometry name is the name of your colorMap.

Loading a new molecule

PMV can have multiple molecules loaded at the same time.

The molecular surface representation nicely shows the tunnel forming the active site of the protease.

The user will now load one of the HIV protease inhibitors (indinavir) and display it using a sticks and balls representation.

Display indinavir by sticks and balls.

- File -> readMolecule -> indinavir.pdb
- Select -> Direct Select -> Molecule List ...-> indinavir
- Un/Display -> Lines -> undisplay
- Un/Display -> Sticks and Balls
- Set the cylinder radius at 0.20
- Set the balls radius at 0.40
- Set the balls quality at 20

Adding hydrogen and compute the gasteiger charges on the ligand:

- Select the ligand "indinavir" using one of your favorite selection commands.
- Load the new commands "Add Hydrogens"
- Edit -> hydrogens -> add:
In the "Add hydrogens" panel that apperas on the screen select:
 - All hydrogens
 - noBondOrder because the indinavir was read from a PDB file.
 - Yes to renumber atoms to include the new hydrogens.
- Edit -> Charges -> Compute Gasteiger charges.

Possible problem:

Executing the add Hydrogen may cause the following error:

```
>>> ERROR *****
Traceback (most recent call last):
  File "./ViewerFramework/VF.py", line 562, in tryto
    result = apply( command, args, kw )
  File "./Pmv/editCommands.py", line 889, in doit
    hat = addh.addHydrogens(mol.allAtoms, method=method)
  File "./PyBabel/addh.py", line 92, in addHydrogens
    Hatoms = self.place_hydrogens1(atoms, num_H_to_add)
  File "./PyBabel/addh.py", line 119, in place_hydrogens1
    Hat = Hat + self.add_methyl_hydrogen(a, SP3_C_H_DIST)
```

File `./PyBabel/addh.py`, line 315, in `add_methyl_hydrogen`
`c3 = atom2.bonds[1].atom1.coords`
IndexError: list index out of range

This may be caused by missing bonds in some residues. When a PDB file is loaded in PMV, the bonds are either created from the CONECT records or built by distance. Sometimes the distance between atoms in a residue is superior to the cutoff (covalent radius * 1.1) and the bond is not created. The user then needs to add the missing bonds using the `addBondsGC` command, from the `bondsCommands` module. The `addBondsGC` is a picking command which creates a bond between two picked atoms.

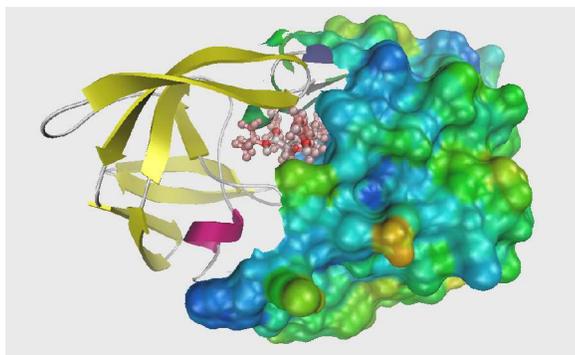
- Load the `bondsCommands` module
- Select the “`addBondsGC`” PCOM
- Then pick on the two atoms to bind.
- Select another PCOM to stop the `addBondsGC` PCOM.

Color the ligand by charges.

The charge property of an atom is not a directly accessible attribute of the atom, therefore the user cannot use the `colorByProperty` command. However the `colorByExpression` allows the user to create a color map using a python expression.

The color by expression commands allows the user to type in a python expression that will be evaluated at the chosen level. The result of the evaluation needs to be a list of numeric values to create a color map which will then be used to color the chosen geometry.

- Color -> By Expression
 - Select Atom level, the charges property being an atom property
 - Delete the content of the text widget and type the following
`lambda(x: x.charges['gasteiger'])`
 - Click on the "eval expression" to evaluate the given expression and make sure that everything is fine
 - Choose the RedWhite ramp to color the indinavir by gasteiger charges
 - Click OK to carry out the color command.



PMV Tutorial

*Fig 7: Indinavir represented by sticks and balls and colored by Gasteiger charges.
The chain A of the protease is represented by traditional ribbon diagram colored by secondary structure elements and the chain B is represented by a MSMS surface colored by temperature Factor.*

Show/Hide a molecule:

When a molecule is represented by several geometries, using the show/hide molecule command will undisplay/display all the geometries at the same time, effectively hiding/showing the molecule.

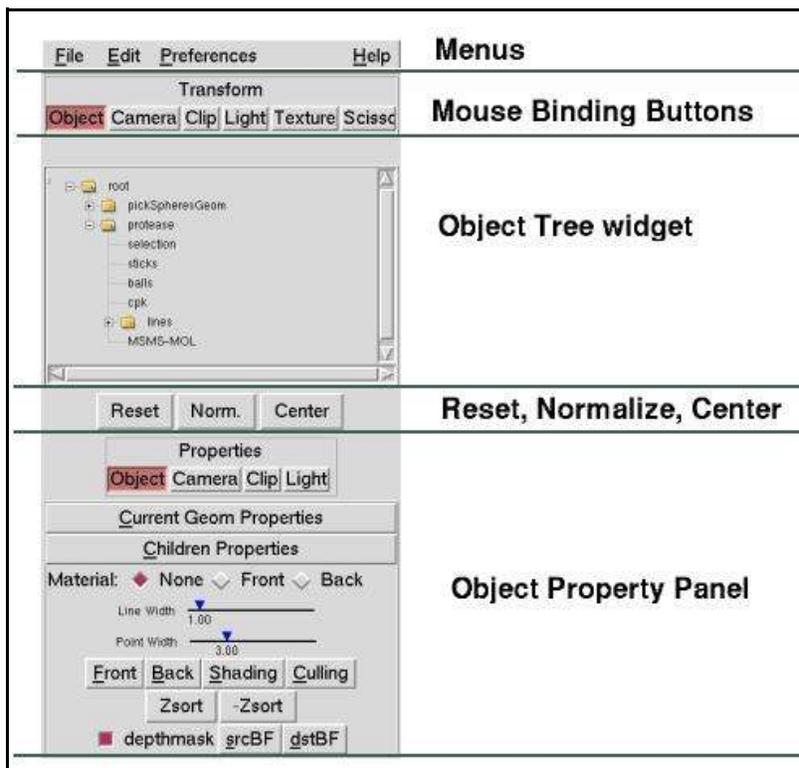
- Menu bar -> Show/Hide Molecule
- protease

Making a publishable picture:

The DejaVuGUI:

Overview

The VieweGUI allows the user to modify properties of the objects in the Viewer, the Camera, the clipping planes, the lights etc...



DejaVu ViewerGUI

Menus:

A set of menu are available to the user.

Under the File menu the user can either “Load Transformation” or “Save Transformation”.

Under the edit menu the user can “Apply Transformation” when available.

Under the preference menu the user can choose to:

- “Transform the root only” by checking this preference. This means that eventhough the current object is not root transformation will be applied to 'root' rather than the current object. When the user wants to move a geometry independently of the others this preference has to be turned off.
- “ Show Picked Vertex”.

Mouse Binding Buttons

The GUI presents a row of buttons under *Transform* which allows the user to direct transformations to the current Object, the current Camera, the current Clipping plane, the current Light or the current Texture. By default "*Object*" is selected (I call transformation a rotation, translation or scale operation).

Object Tree widget:

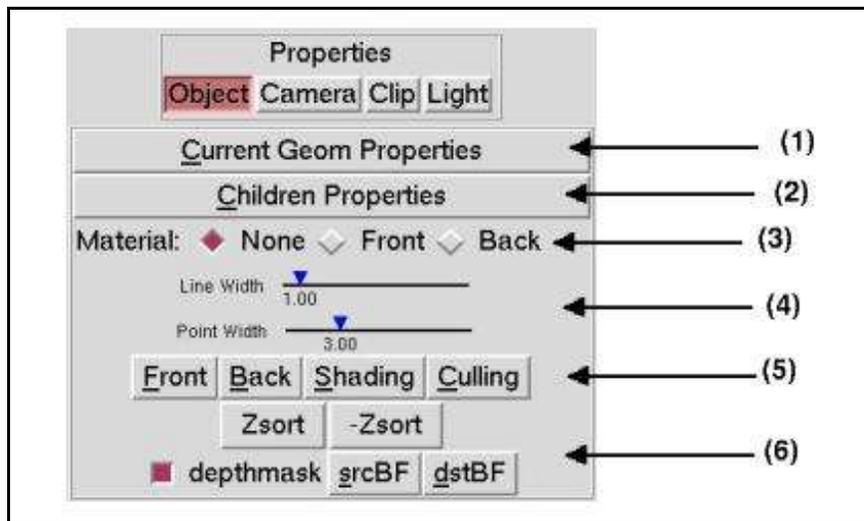
This widget shows the tree-like structure of geometries present in a Viewer. Names have to be unique and if a same name is given to two different geometries, the Viewer will concatenate a '.x' where x is an integer to the object name to make the name unique.

Property Panels:

There is a property panel for the current object, camera as well as for clipping planes and light sources. Only one of these panels is visible at a time. The one shown is selected using the buttons in the "Properties Panel Menu" box.

The Object property panel:

Overview:



(1) Current Geom Properties is a list of all the properties of the current Object that can be

toggled on or off by the user.

- (2) Children Properties is the list of the properties of the children of the current Object that can be toggled on or off by the user.
- (3) This set of radiobutton displays a Material Editor allowing the user to edit material for either the front or the back faces of the current Object.
- (4) Sliders for the user to set the lineWidth and the pointWidth of the current object.
- (5) Set of buttons to display:

the drawing mode menus for the front or back facing polygons. The user can choose inherit, in which case the polygons are drawn in the same mode as their parent, or as points, lines or with a filled representation. There may also be an outline mode which appears only if your OpenGL provides the OffsetPolygon extension.

the shading mode menu. The user can choose an inherited shading mode, with flat shading which shades the entire polygon with one color, with smooth shading which uses gourout shading or not at all.

the culling mode menu to select which polygons to be culled in the same mode as in the current object's parent, to do no culling, to cull the back-facing polygons, to cull the front-facing polygons or to cull all of them (which makes the object disappear!)

- (6) The Zsort or -Zsort button toggle a sorting of the polygon along the Z axis.

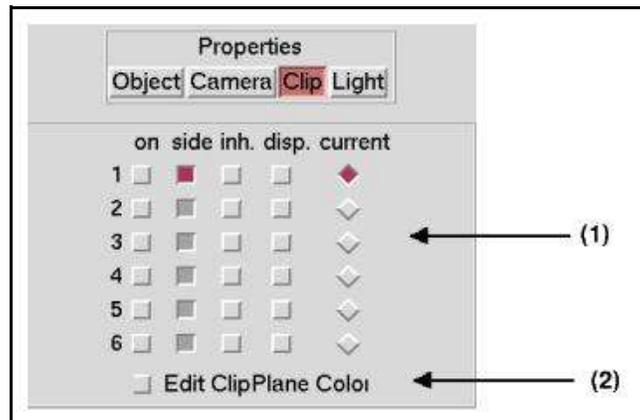
Transforming a geometry independently from the others:

Changing the representation of the front and back faces of the msms surface

Enabling a scissor window on the msms surface

The Clipping plane property panel:

Overview:



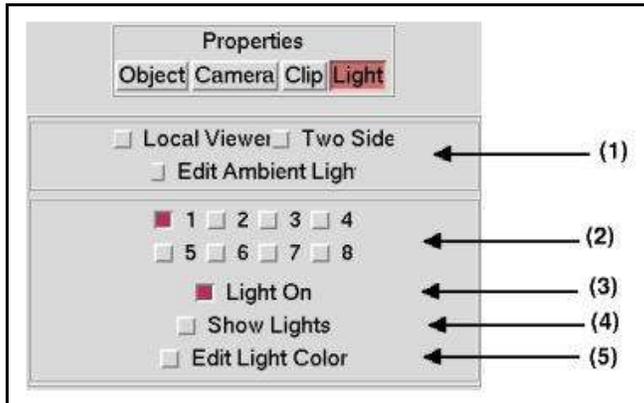
- (1) The on button lets the user turn on or off the current clipping plane on a per object basis. If it is on, the current plane will clip the current object, hiding one side. The side button lets you toggle which side of the current object, as divided by the current clipping plane, is hidden. The inh button sets whether the children of the current object will inherit the type of clipping performed by the current clipping plane. The disp button lets you toggle whether the current clipping plane is displayed in the scene. The radiobuttons under current let you select the current clipping plane.

(2) This check button allows the user to edit the current clipping plane's color.

Clipping the msms surface

The Light property panel:

Overview:

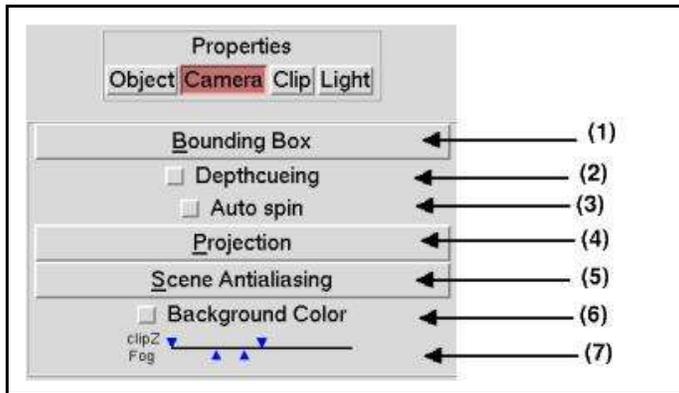


- (1) The default OpenGL Lighting Model (see [OpenGL Programming Guide](#), M. Woo, J. Neider and T. Davis, Reading, MA: Addison-Wesley Developers Press, 1997. pages 193-195) is based on three concepts: global ambient light, whether the viewpoint position is local to the scene or infinite distance away and whether lighting calculations should be done differently for front and back faces of objects. The default ambient light can be changed with the color editor. The location of the viewpoint affects how the highlights on objects are calculated. The Viewer uses the infinite distance viewpoint by default but this can be changed to use a local viewer. The Viewer uses front face lighting calculations as the default. It is slower to calculate lighting for both sides of the polygons, but this may be useful in special cases.
- (2) These buttons allow the selection of the current light. Currently only directional light sources are supported. The bottom three buttons always apply to the currently selected light. The Light On button toggles the light on and off. Show Lights button toggles whether a line is drawn in the Camera showing the direction of the current light. Edit Light Color displays a color editor which can be used to change the color of the current light.

Modifying the lights property

The Camera property panel:

Overview:



- (1) The bounding box of the current object by default is not shown. You can use this menu to show the bounding box alone or with the current object. This bounding box has the special property that it remains orthogonal to the axes of the viewer even if the object it bounds is rotated. If the current object is root, the bounding box encompasses all of the objects in the scene.
- (2) To turn on or off the Depthcueing.
- (3) In auto spin mode the object continues its motion after the mouse button has been released. This motion continues until the next mouse button click.
- (4) Lets the user select either orthographic or perspective (default).
- (5) Scene antialiasing uses the standard OpenGL scene jittering to achieve antialiasing. You can select how much jittering is done from these values: 0, 2, 3, 4, 8, 15, 24 and 66.
- (6) This displays the color editor which allows you to modify the background color of the current camera.
- (7) The line represents the z-axis with the Viewer at the left and infinity at the right. The triangles may be dragged anywhere in this range. The left blue triangle on the top of the line sets the position of the near clipping plane while the top right blue triangle sets the position of the far clipping plane. By changing the positions of these two triangles, you can display selected sections along the z-axis. The bottom left triangle represents the plane along the z-axis where fog starts and the bottom right triangle the plane for 100% fog. Linear fog is implemented between these two.

Changing the background color

Turning on the antialiasing

Saving your scene as an image:

Useful information

My contact information

Sophie COON sophiec@scripps.edu 4 9556

Our web site:

<http://www.scripps.edu/~sanner/python>