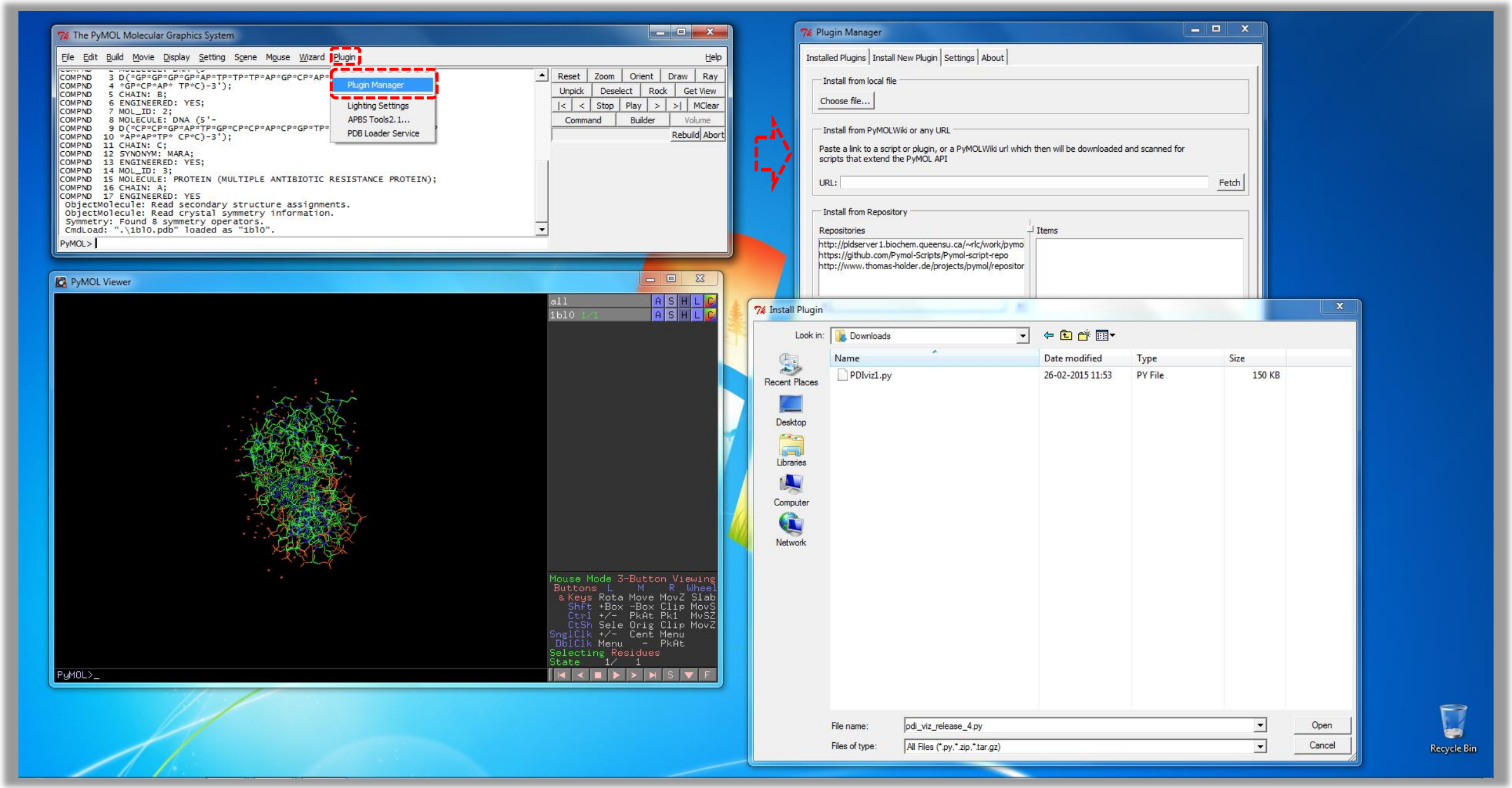
PDIviz 1.2

Quick Start Guide

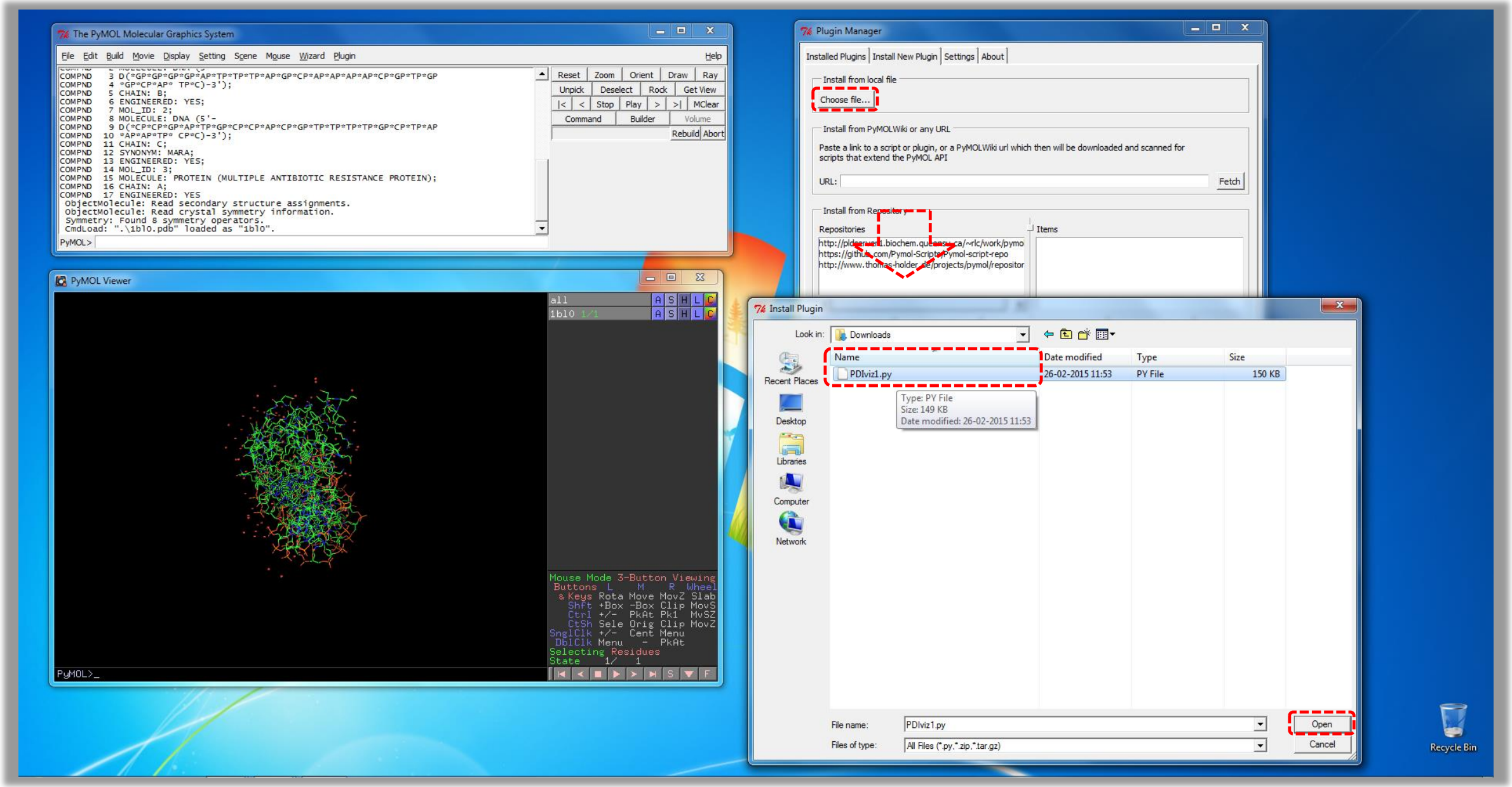
Judemir Ribeiro, Francisco Melo and Andreas Schüller

2016-04-07

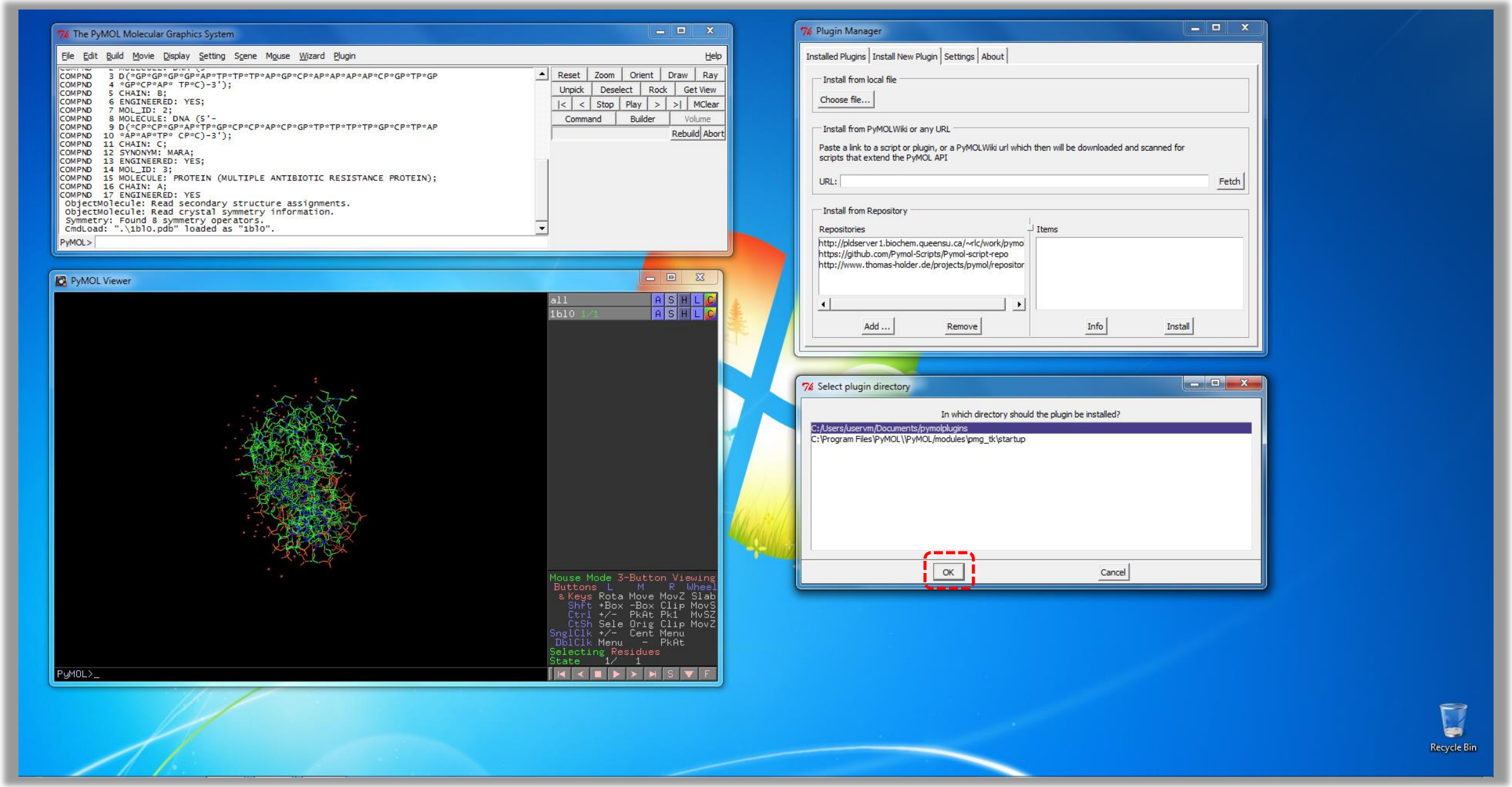
Windows Installation



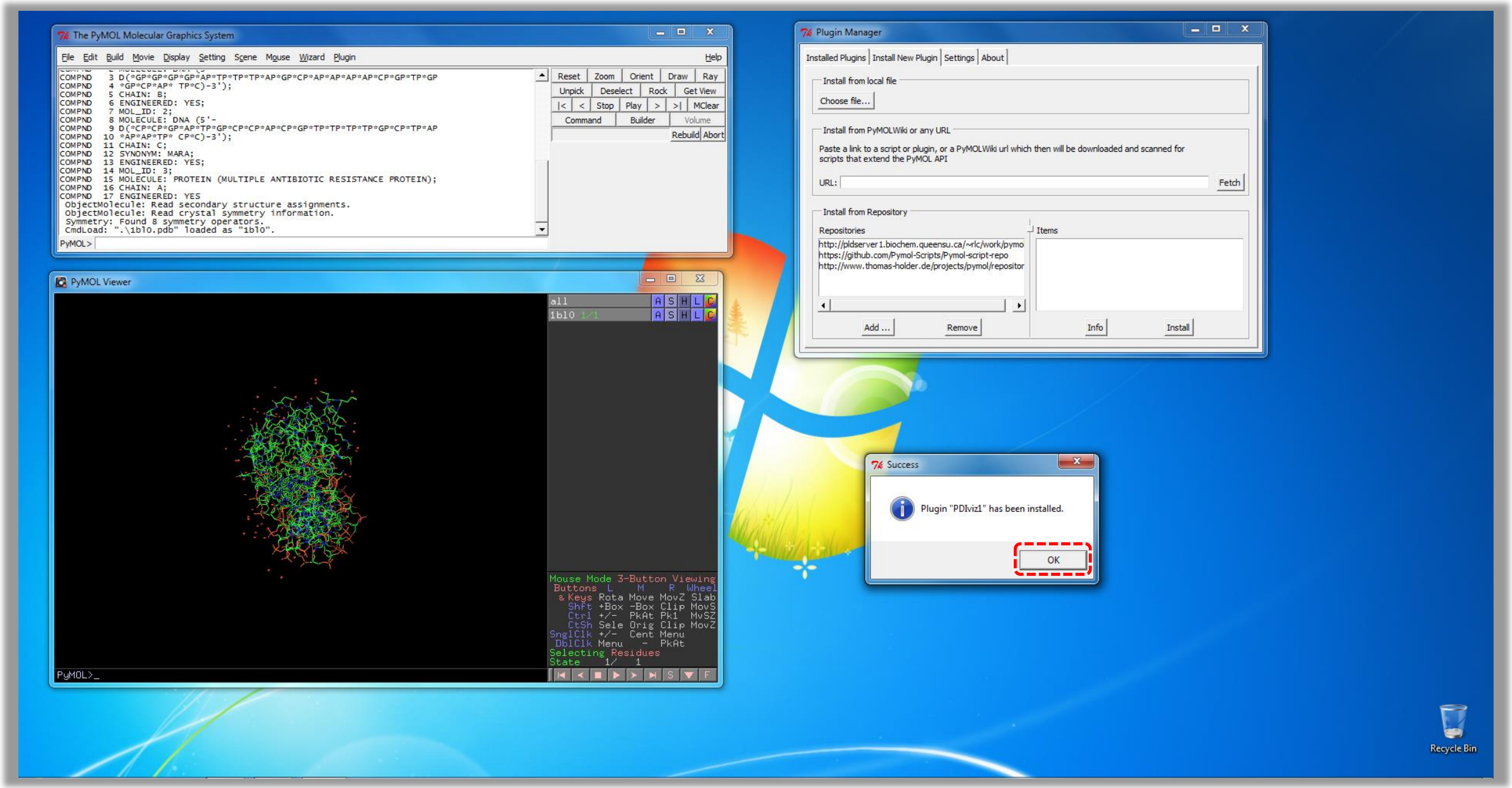
Step 1: Open PyMOL, then open the “Plugin” menu and click on “Plugin Manager”, as indicated by the red dashed rectangle. This will open the “Plugin Manager” window, as indicated by the dashed arrow.



Step 2: Click on the “Choose File” button in the Plugin Manager window, which will open the “Install Plugin” window, Indicated by the dashed arrow. Navigate to the folder where you downloaded the plugin, select it and click “Open”



Step 3: When asked to select the installation directory, confirm the default selection by clicking “OK”.



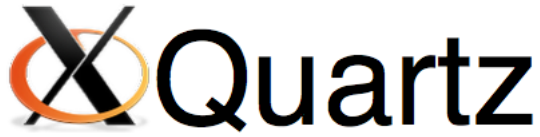
Step 4: Click “OK” to dismiss the success popup and it’s done! You may now close the “Plugin Manager” window, too. It is recommended to restart PyMOL, just to be sure. Earlier versions of PyMOL required this.

Mac OS X Installation

On Mac OS X the X11 version of PyMOL is required in order to run the PDIviz plugin.

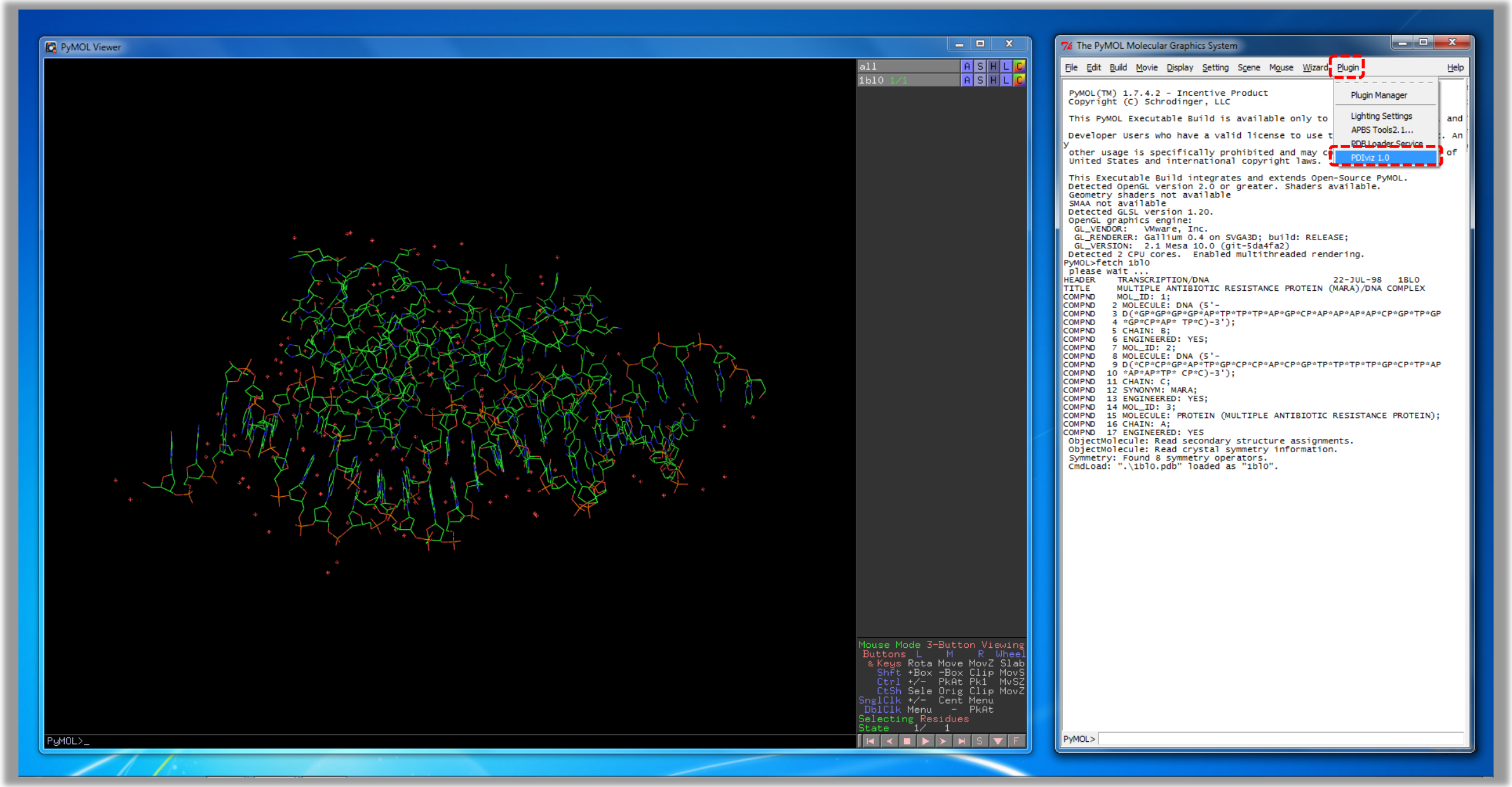
- If you have access to the latest incentive versions of PyMOL you may install the PyMOL version "Mac alternative X11-only build".
- In addition, there is a trick to enable the X11 mode in normal MacPyMOL. All you need to do is to rename the application from "MacPyMOL" to "MacPyMOLX11Hybrid" or from "MacPyMOLEdu" to "MacPyMOLX11Hybrid":

- To run X11 applications on Mac you need to install XQuartz, if you haven't done so, yet. It is available free of charge from http://xquartz.macosforge.org/.

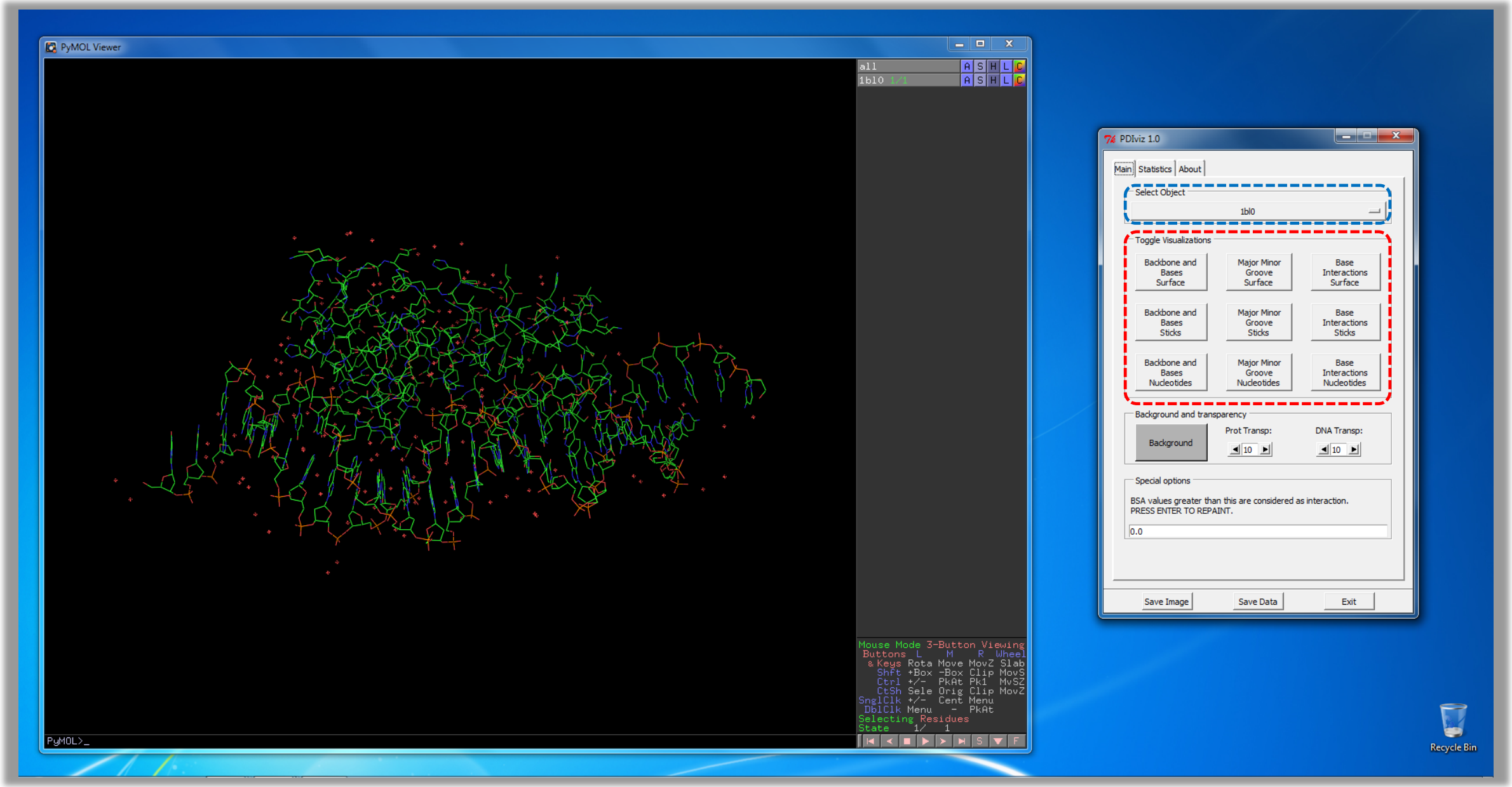


- Once PyMOL runs with its X11 GUI on Mac, the installation of the plugin is equivalent to the Windows installation.

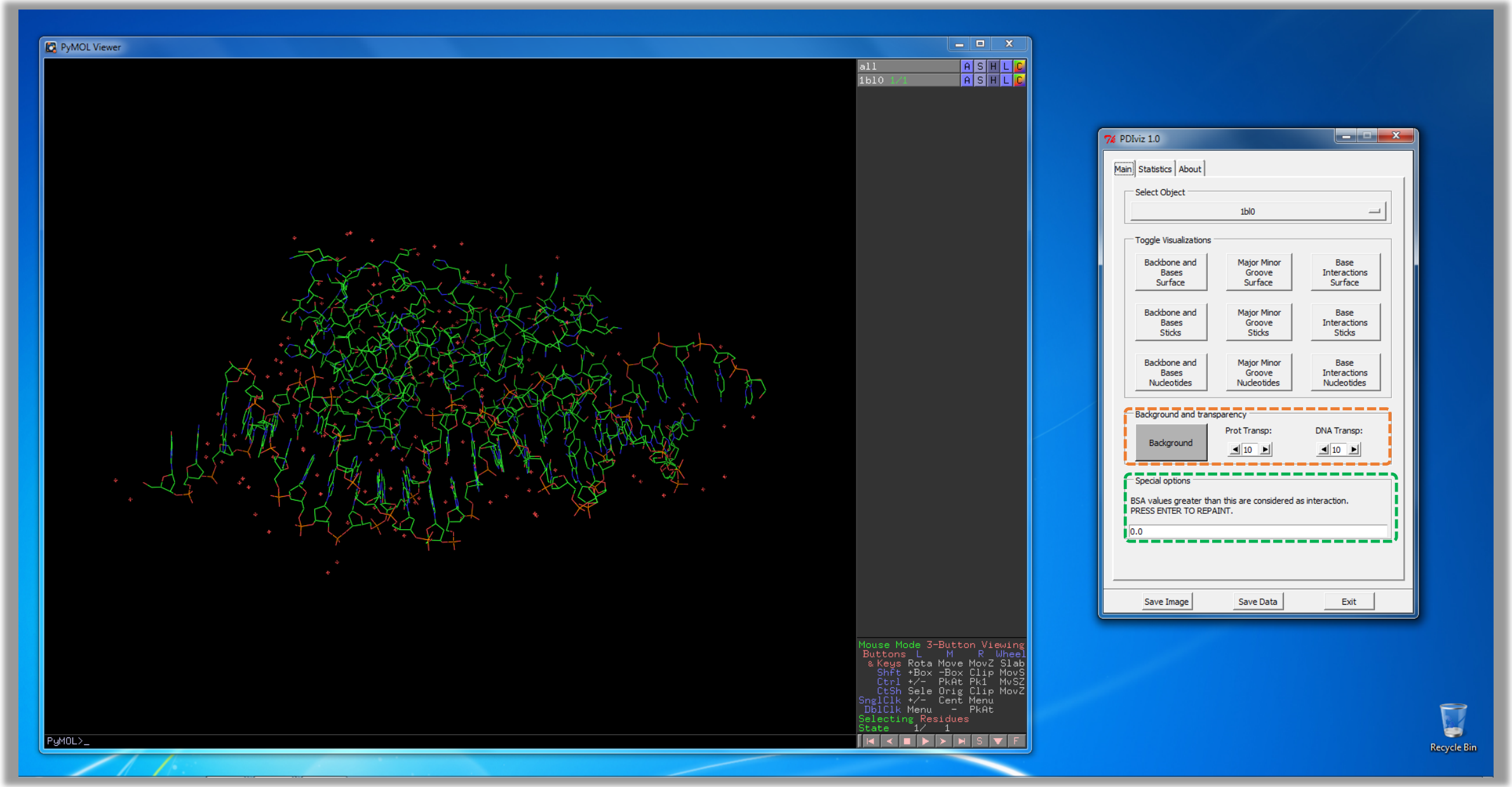
Basic Usage



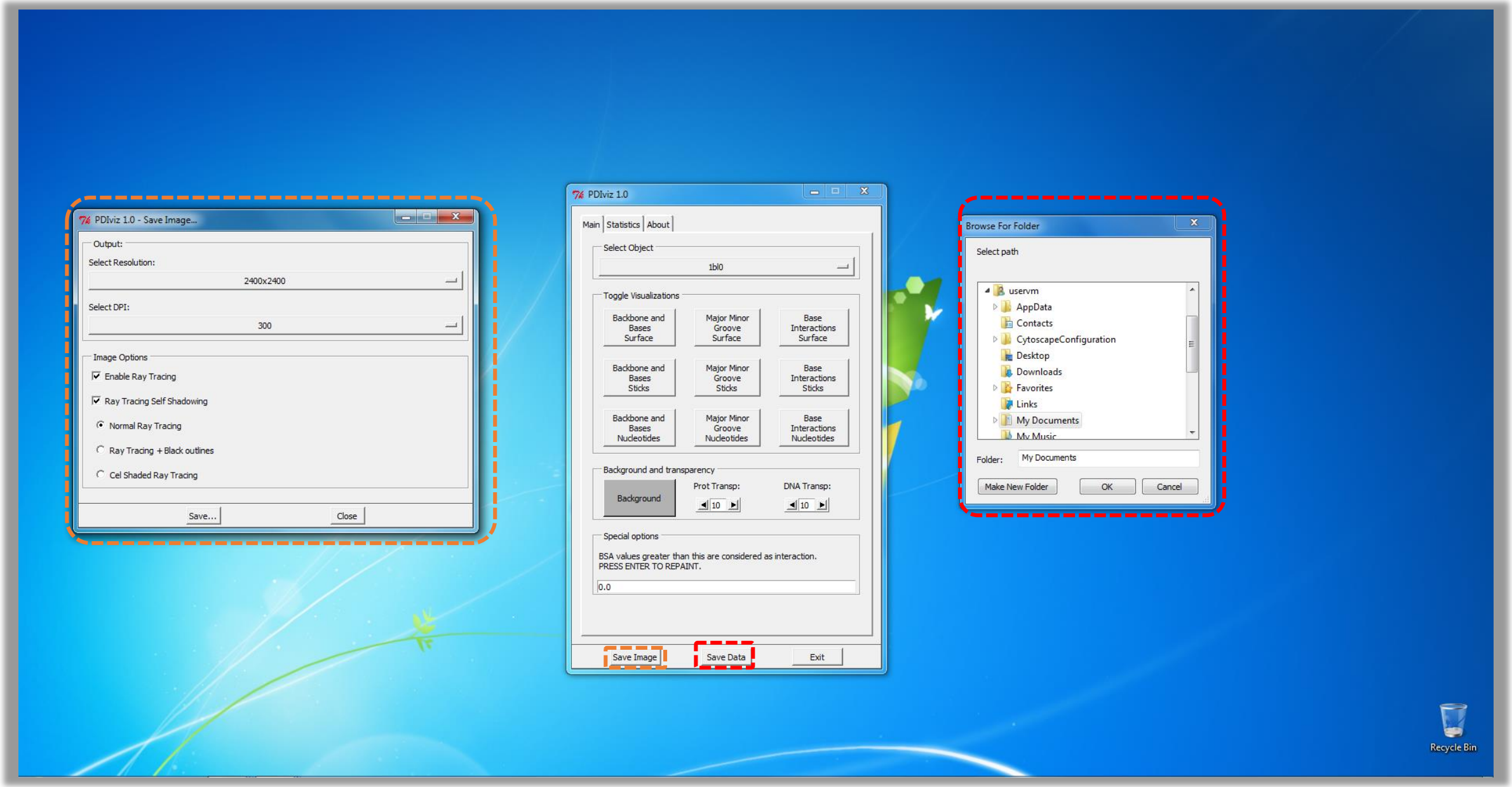
To start PDIviz, open the “Plugin” menu and select “PDIviz 1.0”.



You have now opened the PDIviz main window. The blue rectangle highlights the object selector, which lets you choose from a list of protein-DNA complexes currently loaded in PyMOL. *Note that objects with multiple states must be split first (command: split_states).* The red rectangle highlights the visualization selector buttons. They automatically begin the calculation once pressed.

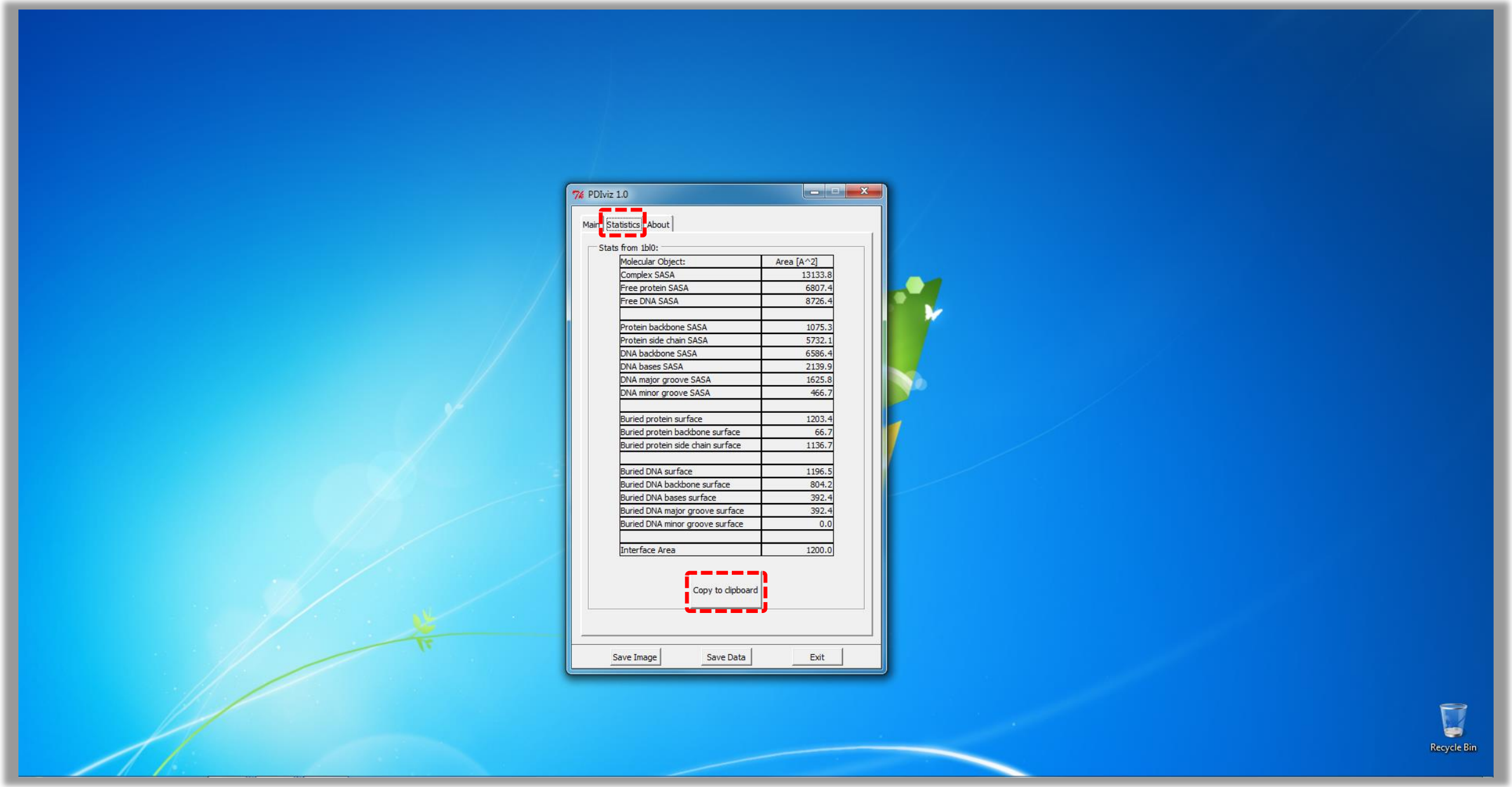


The orange rectangle highlights a background color setter, and two controls for the transparency of surface representations in proteins or DNA. Finally, the green rectangle emphasizes the buried surface area (BSA) cut-off field. Only atoms with a BSA value greater than the cutoff (A^2) will be considered as interaction.



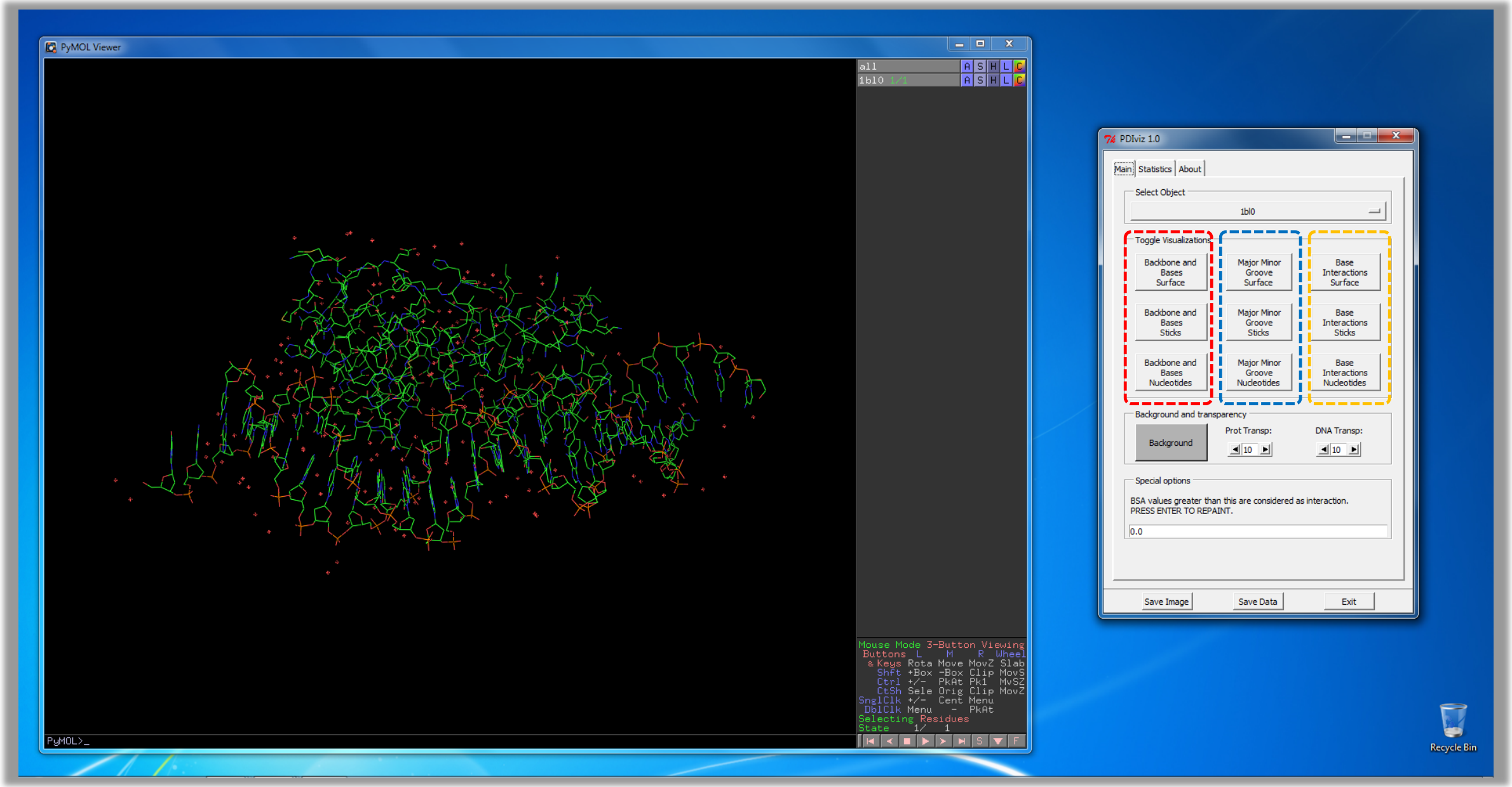
The orange boxes highlight the “Save Image” button and the “Save Image…” window. This provides more options for saving high-quality images than the native PyMOL menu.

The red boxes emphasize the “Save Data” button. This allows you to save the SASA/BSA calculation files in aselected folder.



The “Statistics” tab shows the SASA and BSA values of the various surface areas that compose the protein-DNA complex. This data can be copied to the clipboard with the “Copy to clipboard” button. It is also saved to a file when using “Save Data”.

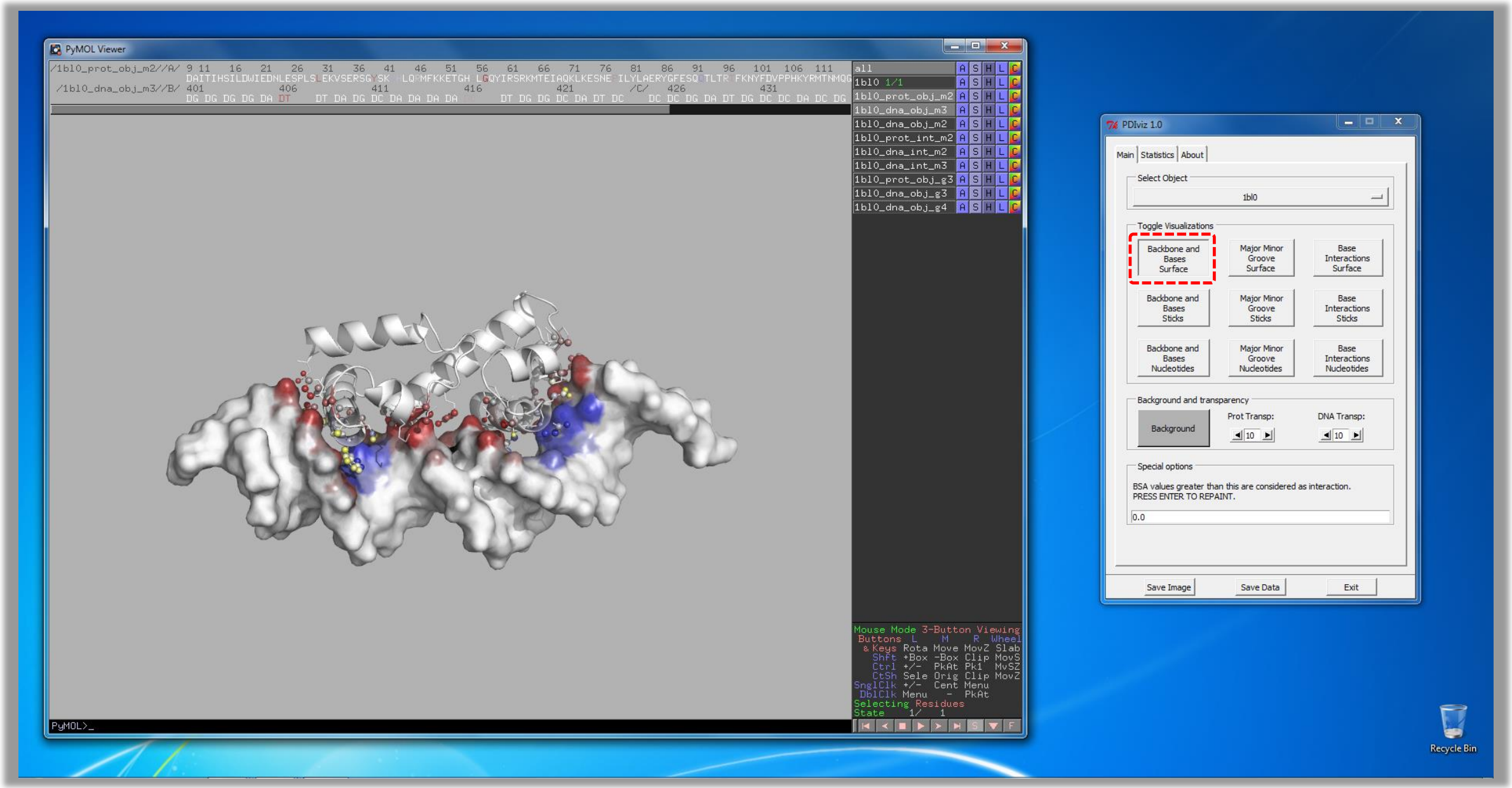
The Visualization Modes



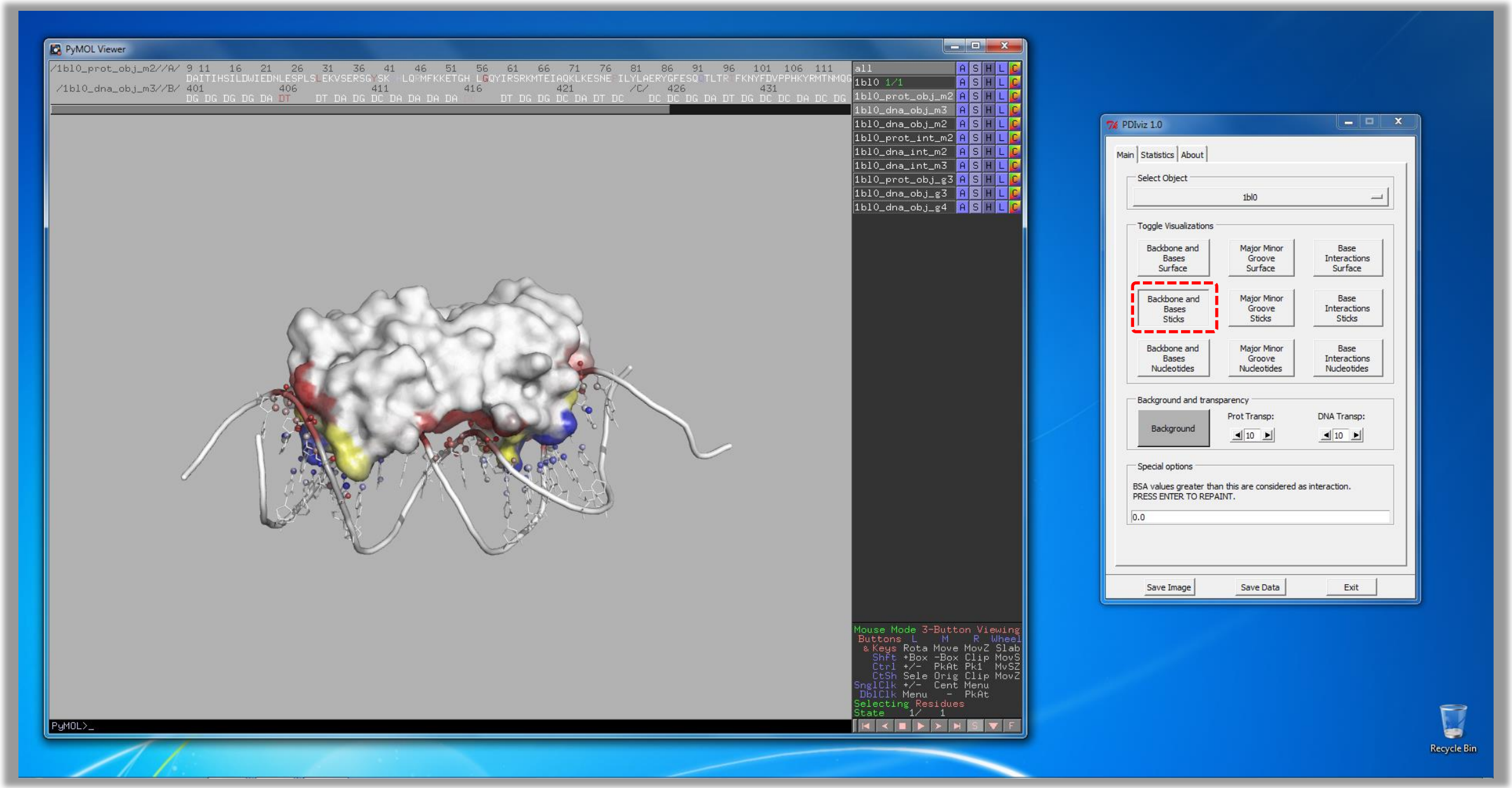
PDIviz has nine visualization modes, organized in 3 columns and 3 rows. The column marked in red shows the interactions of the protein with the DNA backbone and bases, the blue column shows the interactions with the minor and major grooves, and the orange column shows a pharmacophoric interaction map.

The Visualization Modes

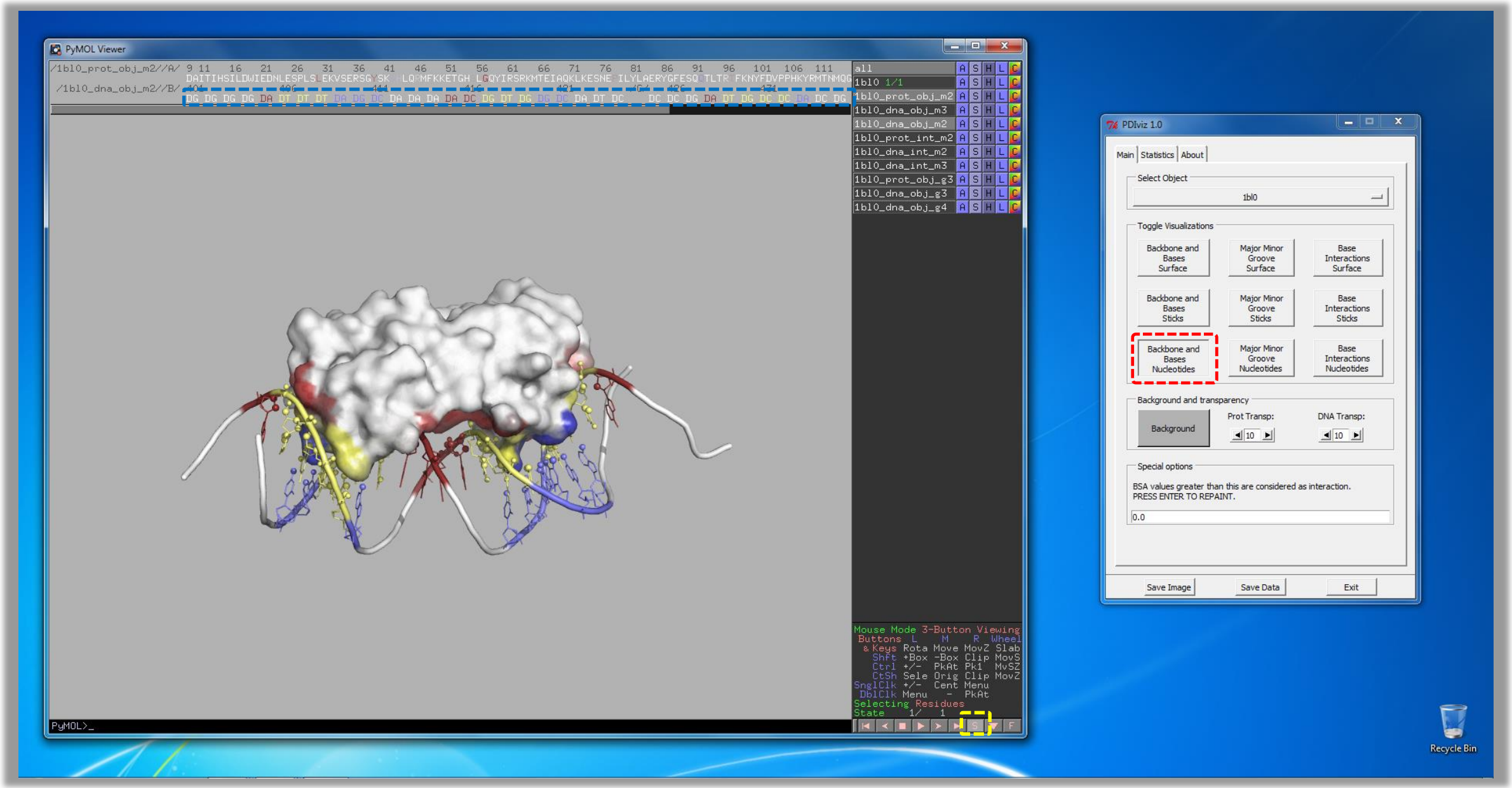
DNA bases and backbone



In this mode, the DNA is shown in a surface representation. Each atom is painted red (backbone atom) or blue (base atom) if they have a BSA greater than the cut-off. In the protein, atoms are painted red if they only interact with backbone atoms, blue if only with base atoms, and yellow if they interact with both types. Color intensities correspond with the BSA of the atoms. All atoms with surface contacts are shown as colored spheres.



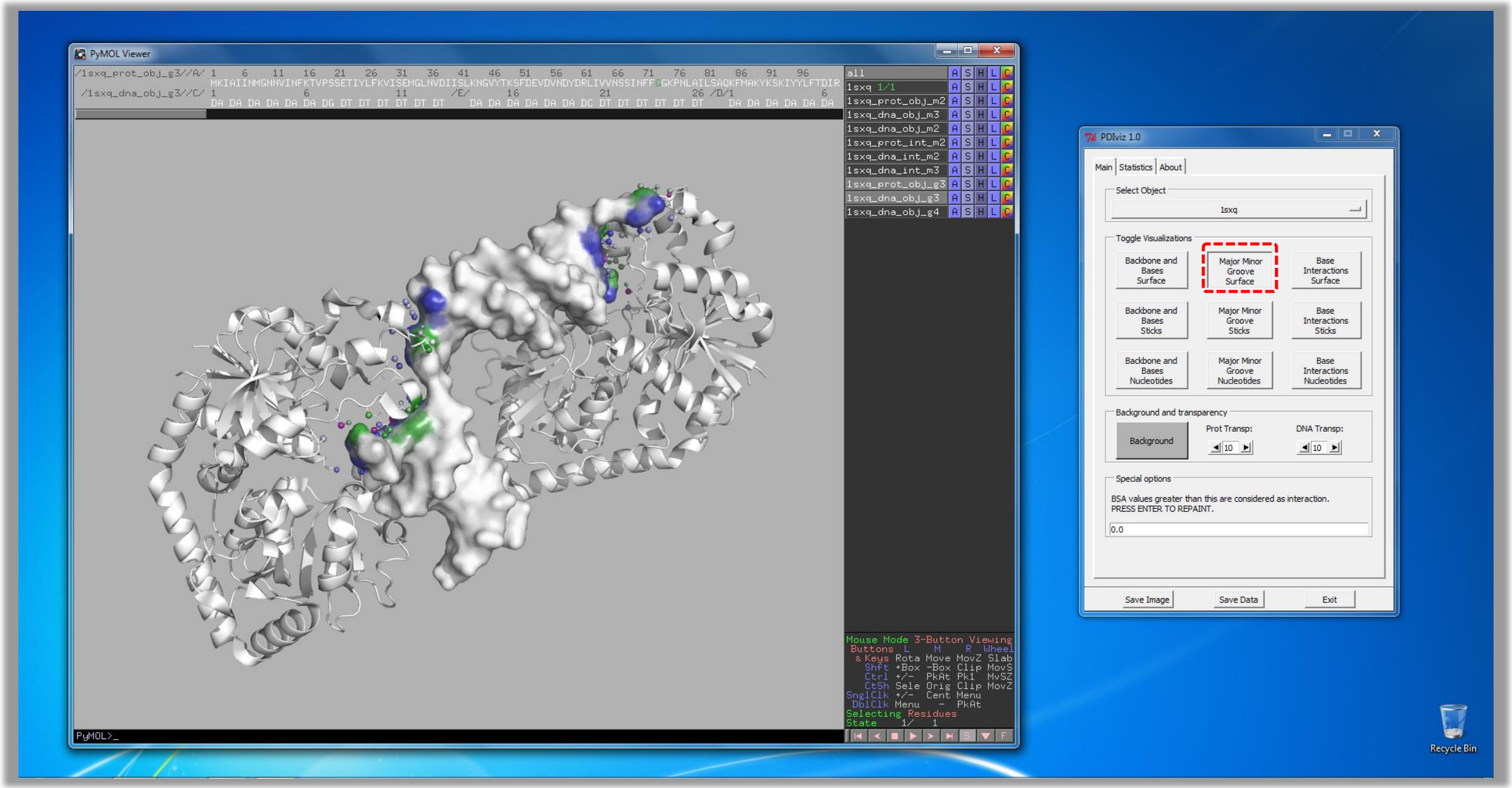
This mode keeps the same coloring pattern, but changes the protein to the surface representation and the DNA to stick representation. Colored spheres show interacting atoms.



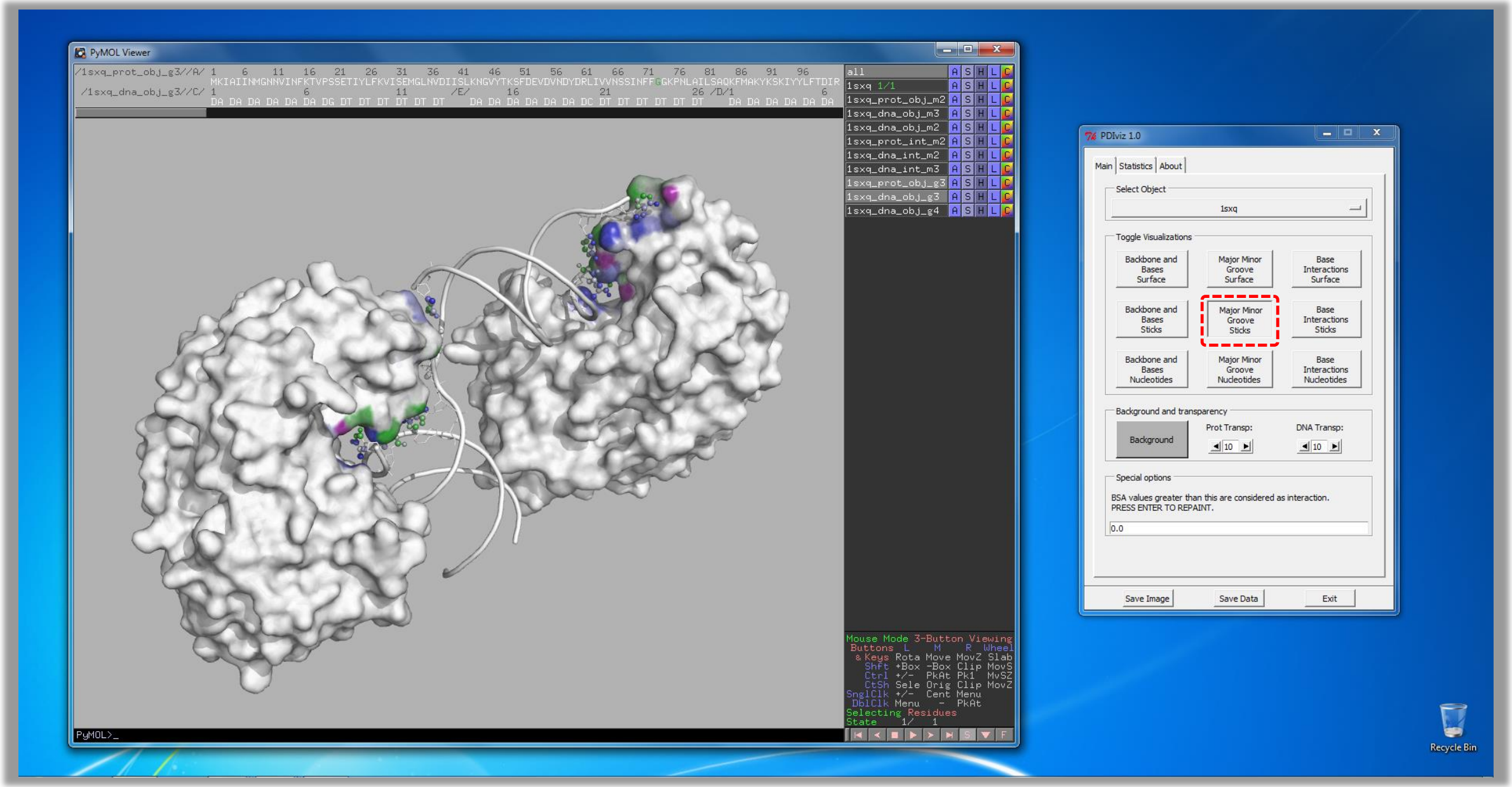
This mode colors the DNA by nucleotide, which allows you to identify the type of interactions of nucleotide without looking at the structure. This can also be seen in sequence viewer (blue highlight), opened in the PyMOL main window by clicking in the “S” button (yellow box). Red nucleotides interact with the protein only via the DNA backbone, blue ones interacts only via DNA base atoms, and yellow nucleotides interact with both types of atoms.

The Visualization Modes

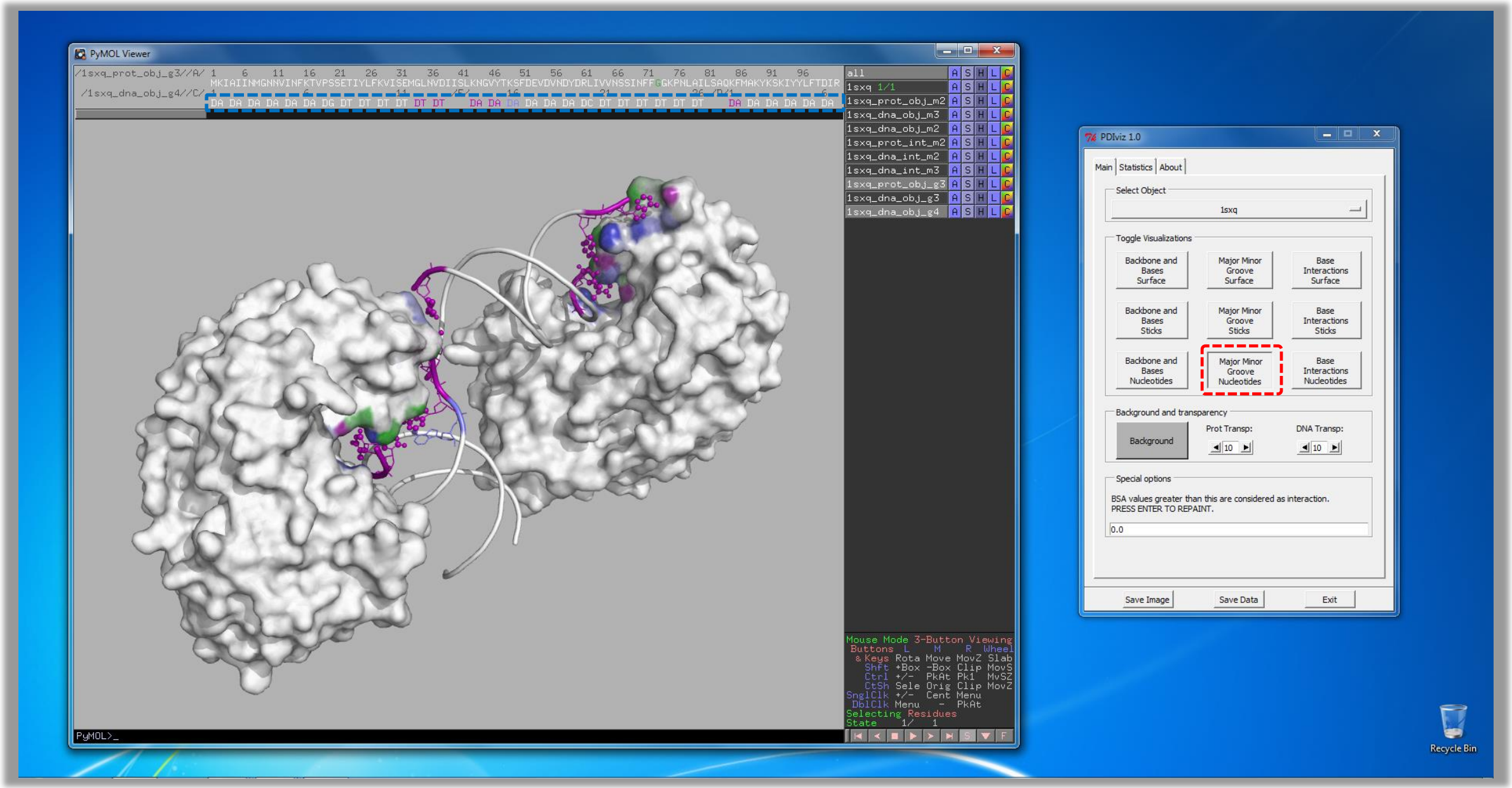
DNA minor and major grooves



In this mode, the DNA is shown in a surface representation. Each atom is painted blue (major groove atom) or green (minor groove atom) if they have a BSA greater than the cut-off. In the protein, atoms are painted blue if they only interact with major groove atoms, green if only with minor groove atoms, and purple if they interact with both types. Colors intensities correspond to the BSA values of the atoms.



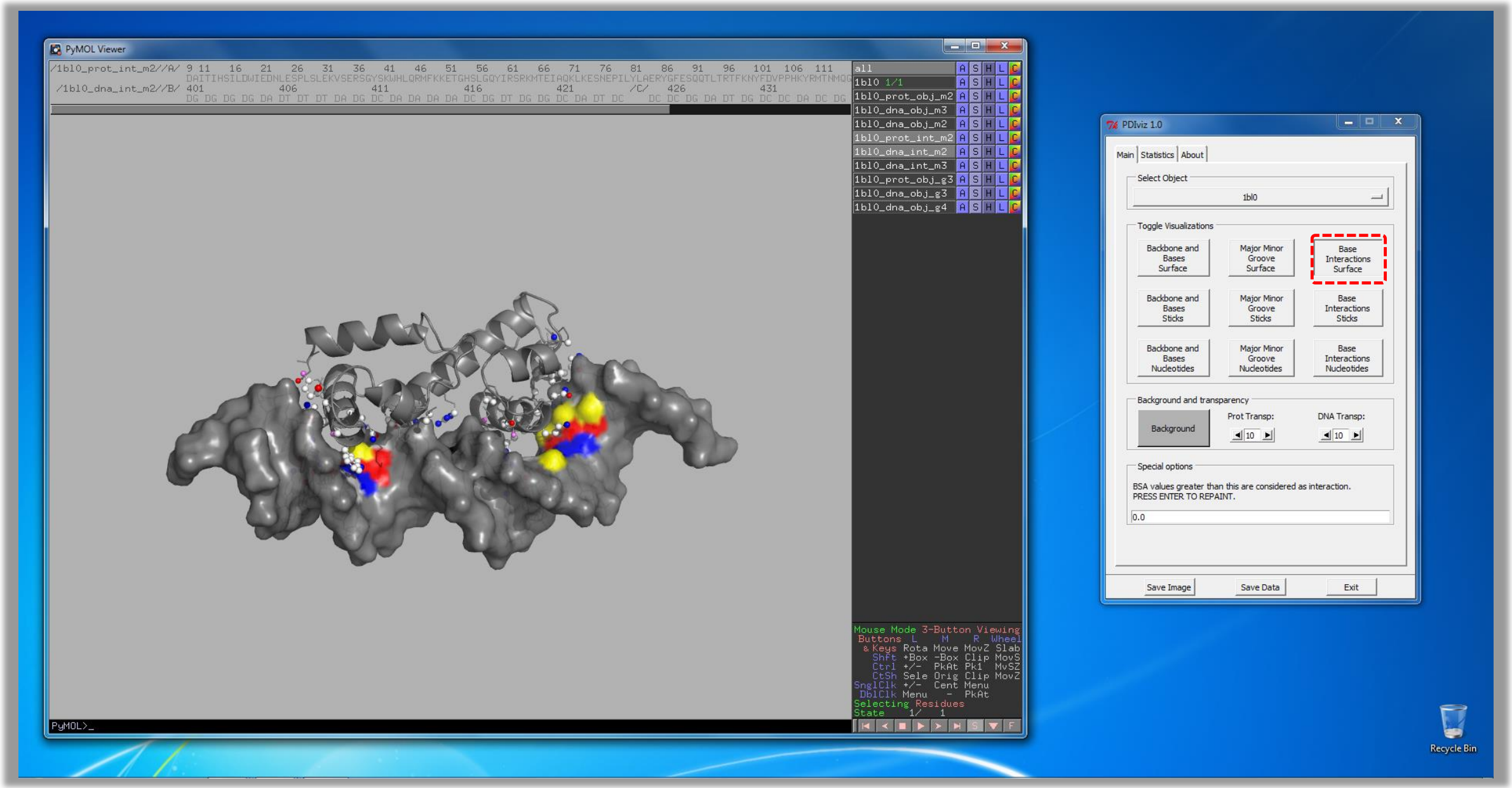
This mode keeps the same coloring pattern, but changes the protein to the surface representation and the DNA to stick representation. Colored spheres show interacting atoms.



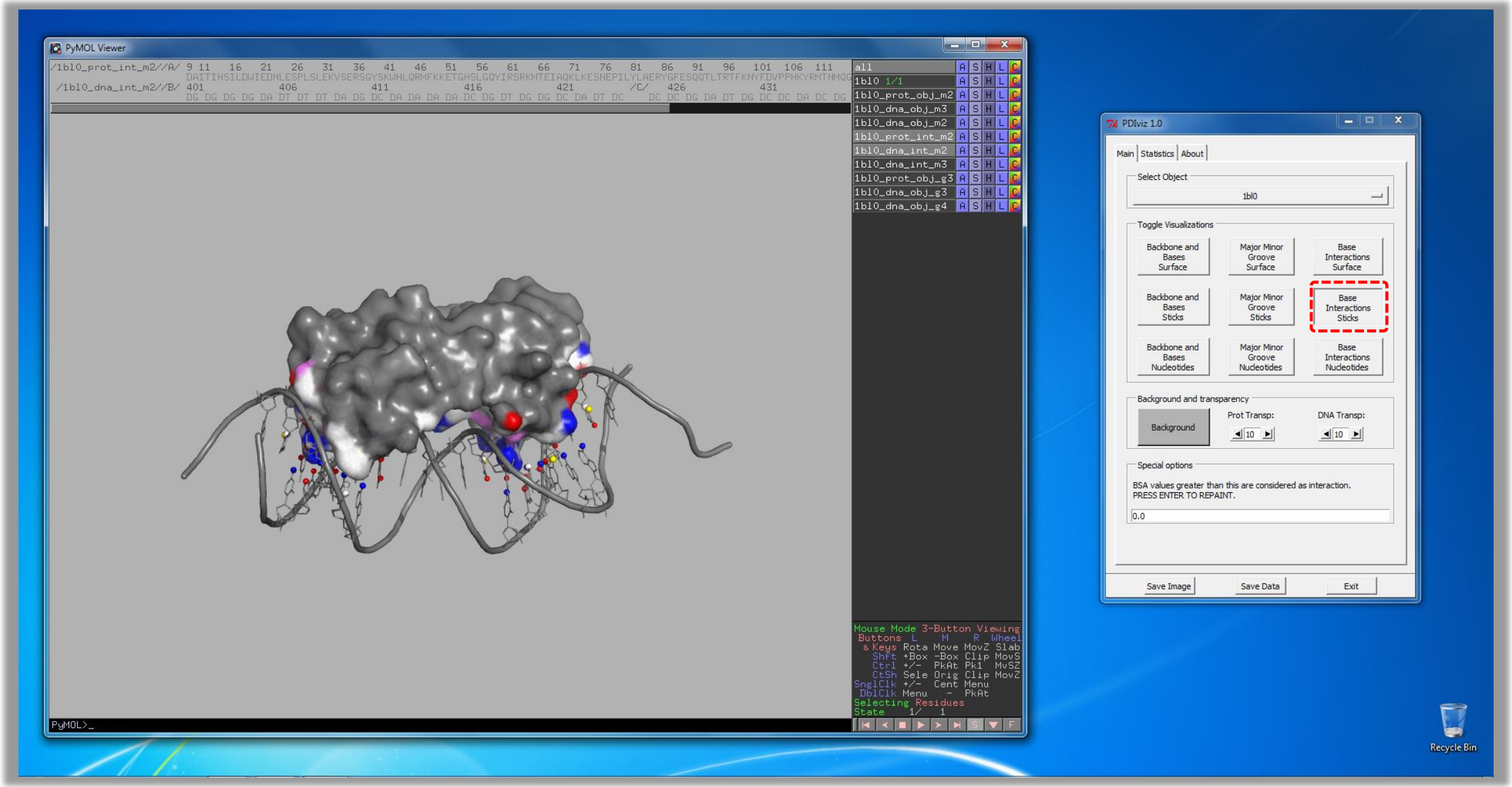
This mode colors the DNA by nucleotide, which allows you to identify the type of interactions of nucleotides without looking at the structure.
This can also be seen in sequence viewer (blue box), opened in the PyMOL main window by clicking in the “S” button (yellow box). Blue nucleotides interact with the protein only via major groove atoms, green nucleotides interacts only via minor groove atoms, and purple nucleotides interact with both grooves.

The Visualization Modes

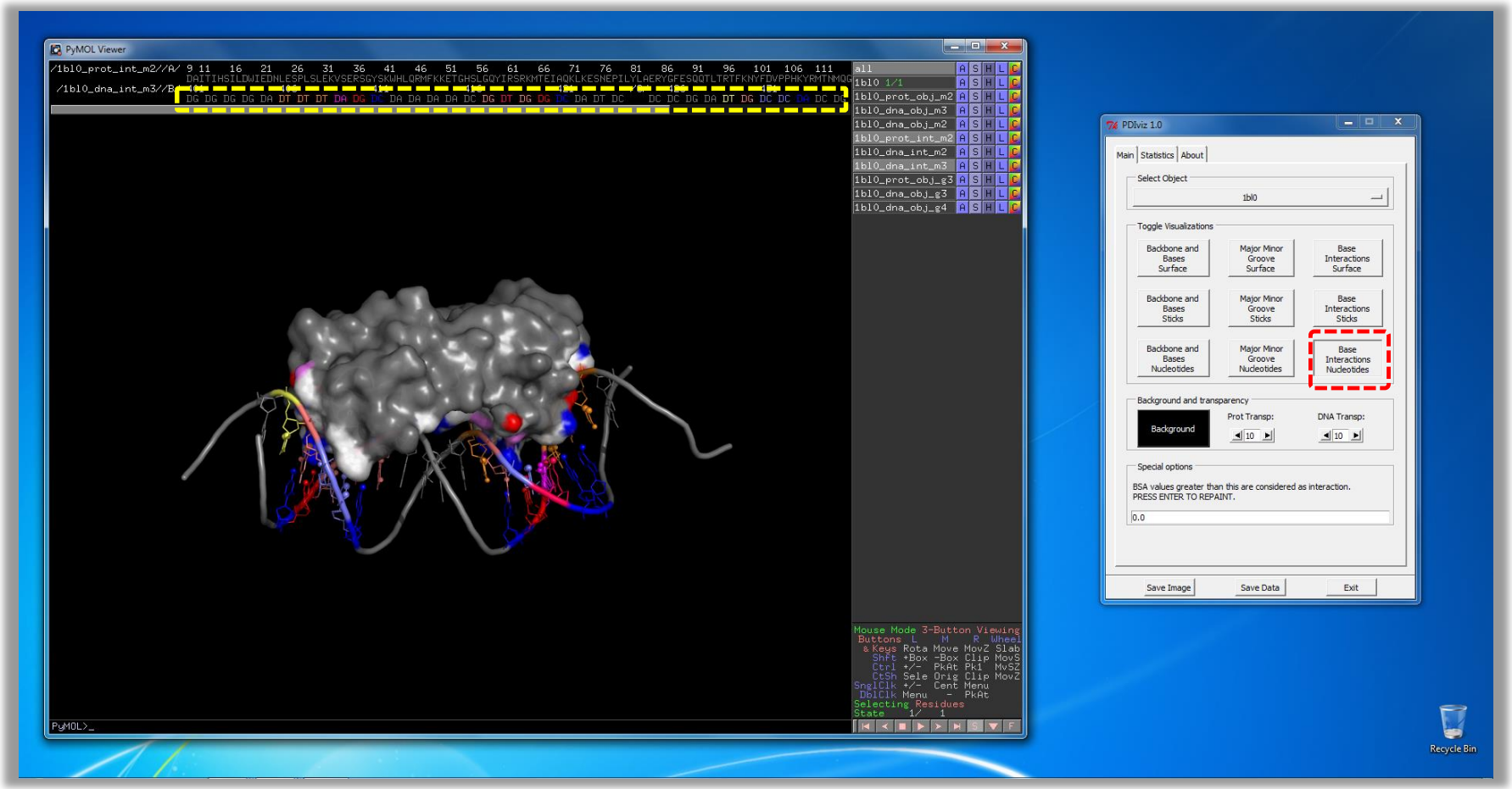
DNA and protein pharmacophoric visualization



In this mode, the DNA is shown with a surface representation. Atoms are painted blue (H-bond donor), red (H-bond acceptor), white (neutral carbon) and yellow (thymine methyl) if they have a BSA greater than the cut-off. In the protein, atoms with a BSA value greater than the cut-off are painted in the same scheme, with the addition of purple for atoms that can be both, donors and acceptors of H-bonds. The colors are shaded according to the BSA value of the atom.



This mode keeps the same coloring pattern, but changes the protein to the surface representation and the DNA to stick representation. Colored spheres show interacting atoms.



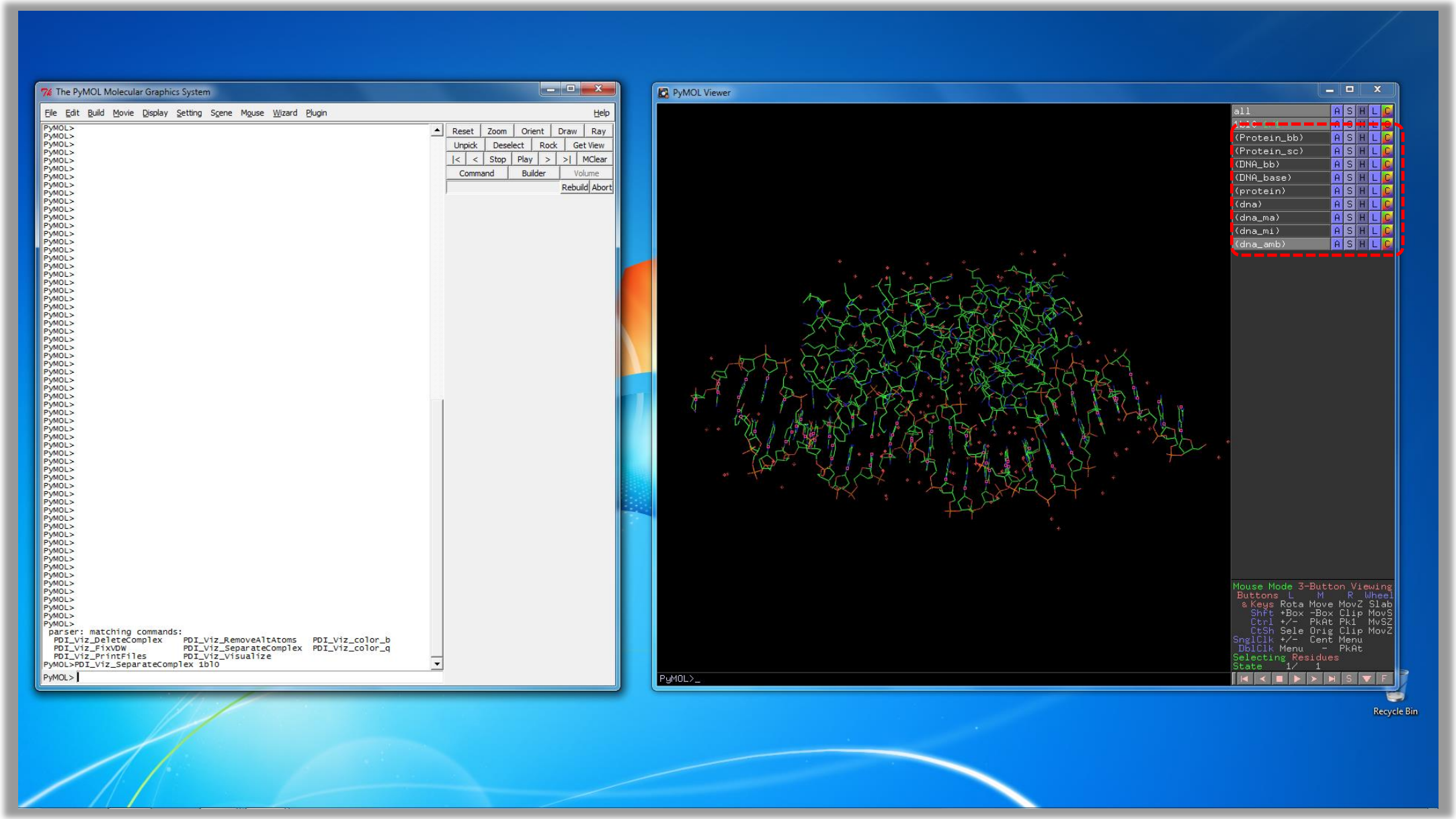
Type	Color
D	
A	
H	
T	
DA	
DH	
AH	
DAH	
DT	
AT	
HT	
ATH	
DAT	
DTH	
DAHT	

In this mode, the nucleotides are painted in one of 15 colors, according to the combination of interactions of its atoms. The types in the table are as follows: D, H-bond donor; A, H-bond acceptor; H, neutral carbon (C-H); and T, thymine methyl. An interacting nucleotide will have one of the combinations in the table, and will be painted with respective color.

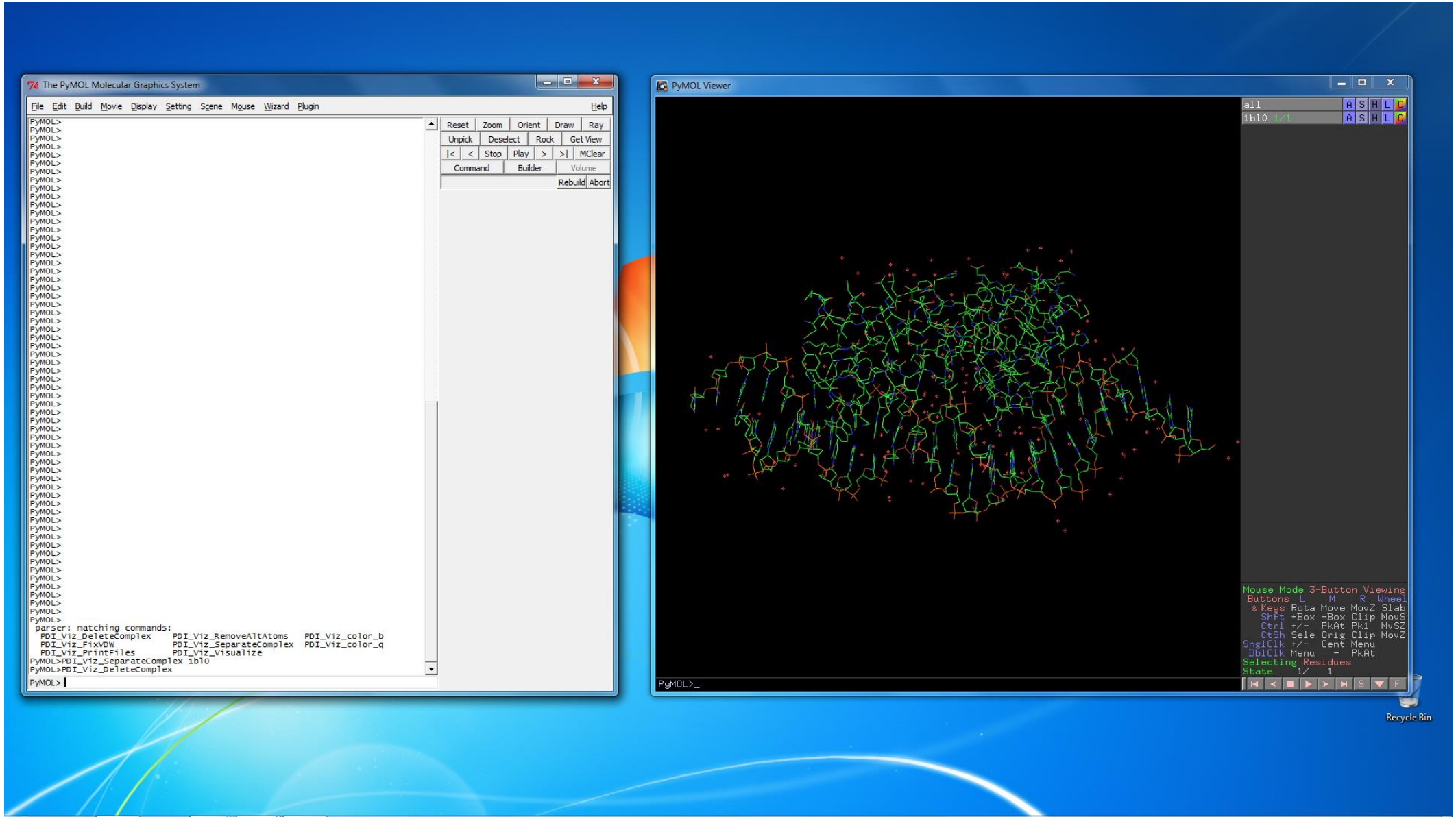
In the yellow box you can see how the DNA sequence is painted according to this scheme. The types highlighted in light yellow only occur on Thymine nucleotides.

Command Line Functions

PDI_Viz_SeparateComplex and PDI_Viz_DeleteComplex



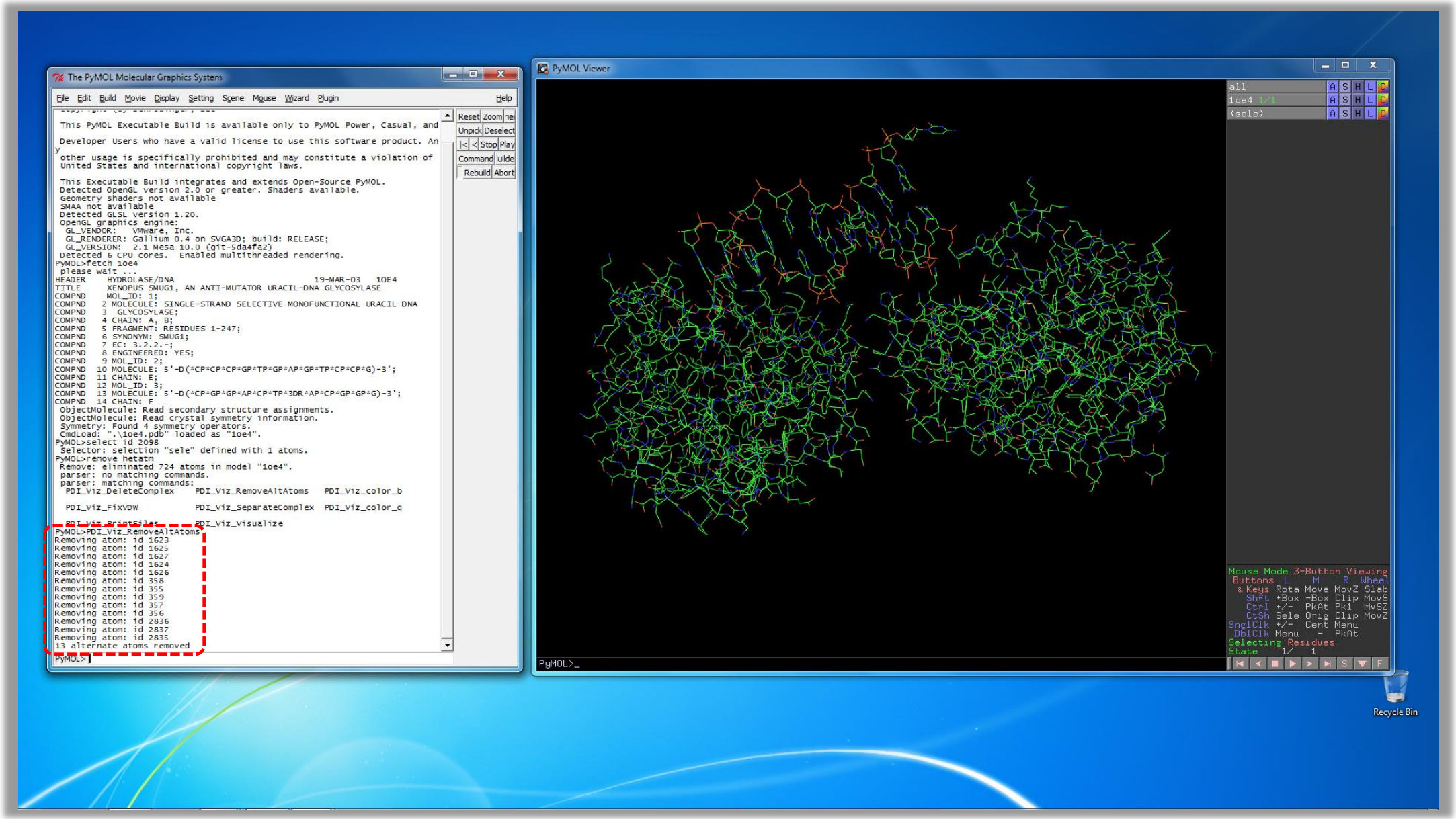
The `PDI_Viz_SeparateComplex` function simply selects the protein backbone, side chains, dna backbone, bases, dna major groove, dna minor groove and dna ambiguous groove. Its only argument is the object name.



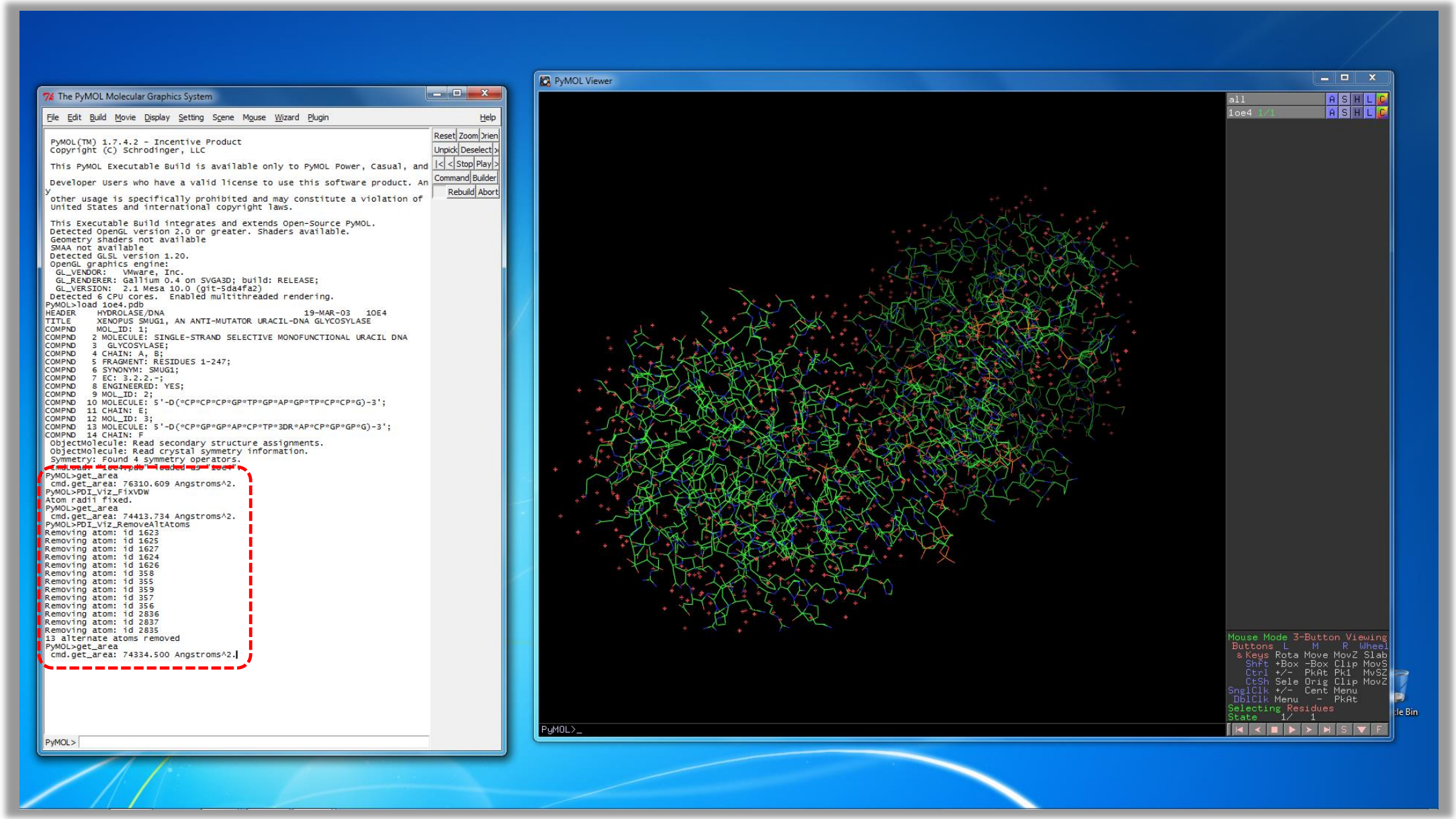
The `PDI_Viz_DeleteComplex` function just deletes the selections created by `PDI_Viz_SeparateComplex`.

Command Line Functions

PDI_Viz_RemoveAltAtoms and PDI_Viz_FixVDW



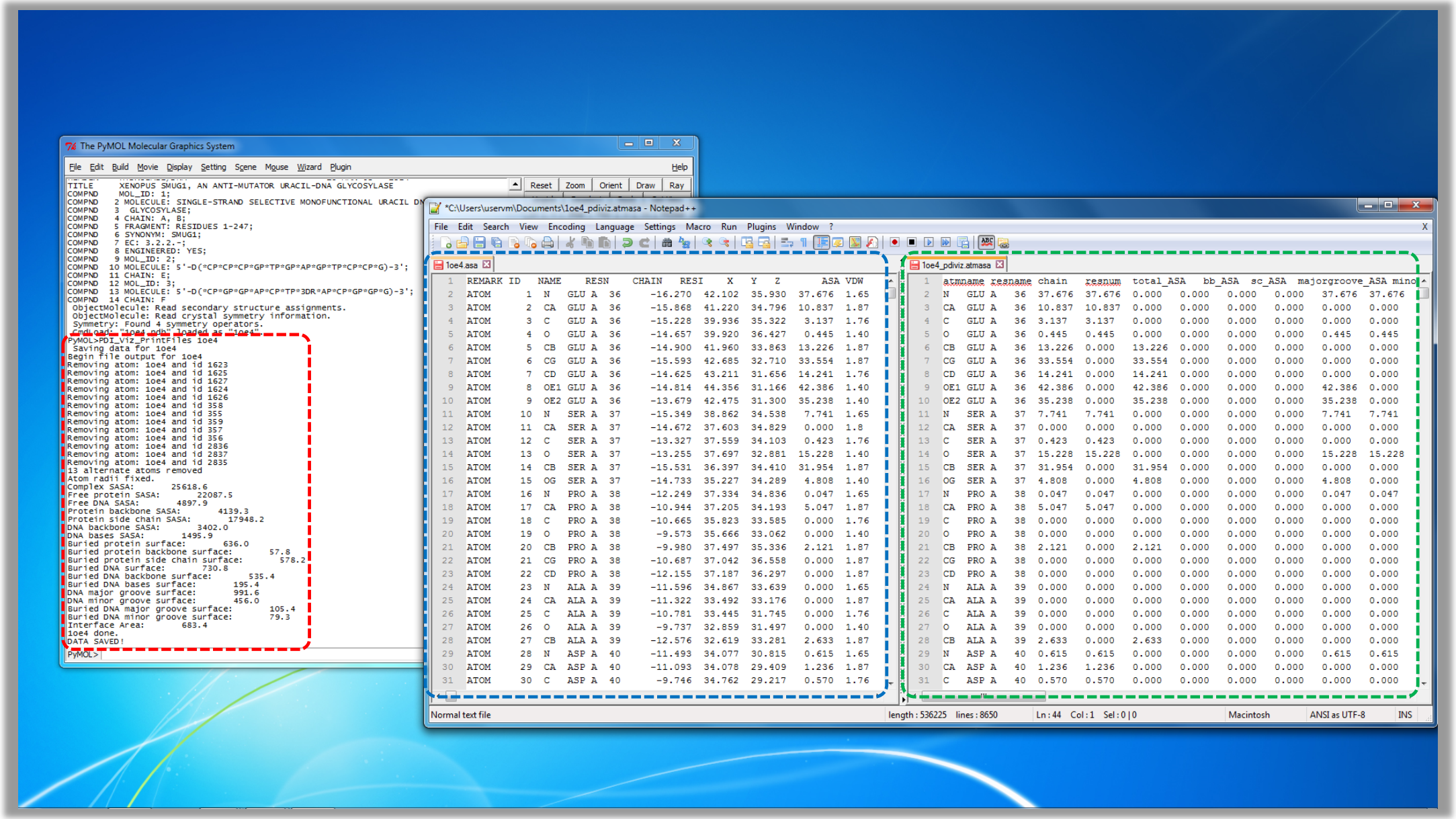
The `PDI_Viz_RemoveAltAtoms` function removes alternative positions (altloc). It removes the standard “alternative atom locations” and non-standard “alternative chain locations”. The red box highlights the ID of the removed atoms. Its only optional argument is an object name.



The `PDI_Viz_FixVDW` function redefines the van der Waals (vdW) radii of all atoms in an object. As can be see in the red box, fixing the vdW radii has a great effect on SASA calculations. Removing alternate locations also has an effect. It only has a single argument, the object name. If no argument is given, the vdW radii of all atoms will be redefined.

Command Line Functions

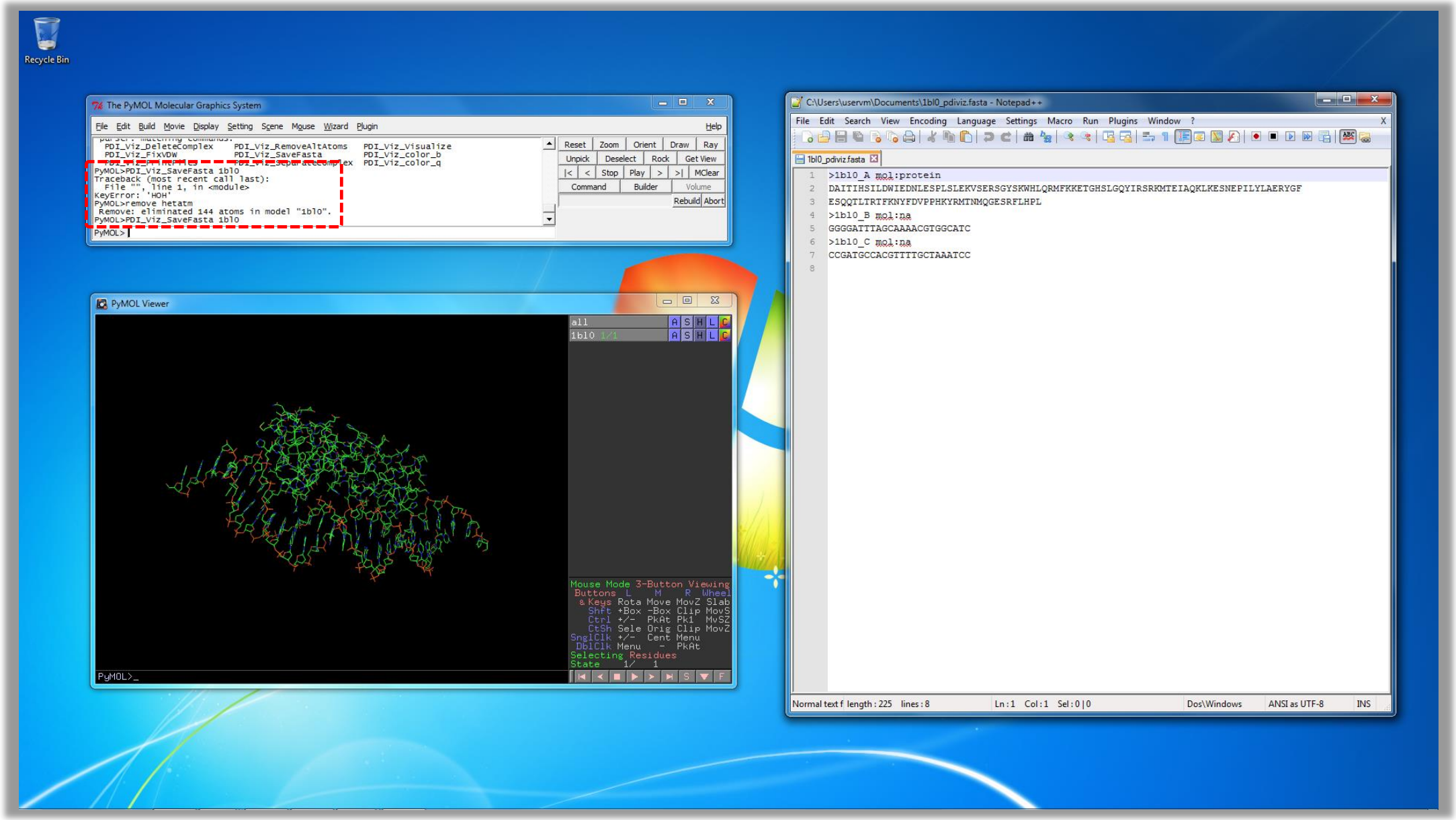
PDI_Viz_PrintFiles



The PDI_Viz_PrintFiles function saves the BSA calculation data in .asa NACCESS compatible files, and in Excel compatible tab-separated .atmasa format. The red box shows that this function automatically calls FixVDW and RemoveAltAtoms. Its arguments are an object name, a save path (optional, defaults on the current directory) and the visualization mode to save as a PyMOL .pse session file (-1 (disable save session), 0 (all) or 1 to 9, optional). The Blue box highlights the contents of an .asa file and the green box the contents of an .atmasa file.

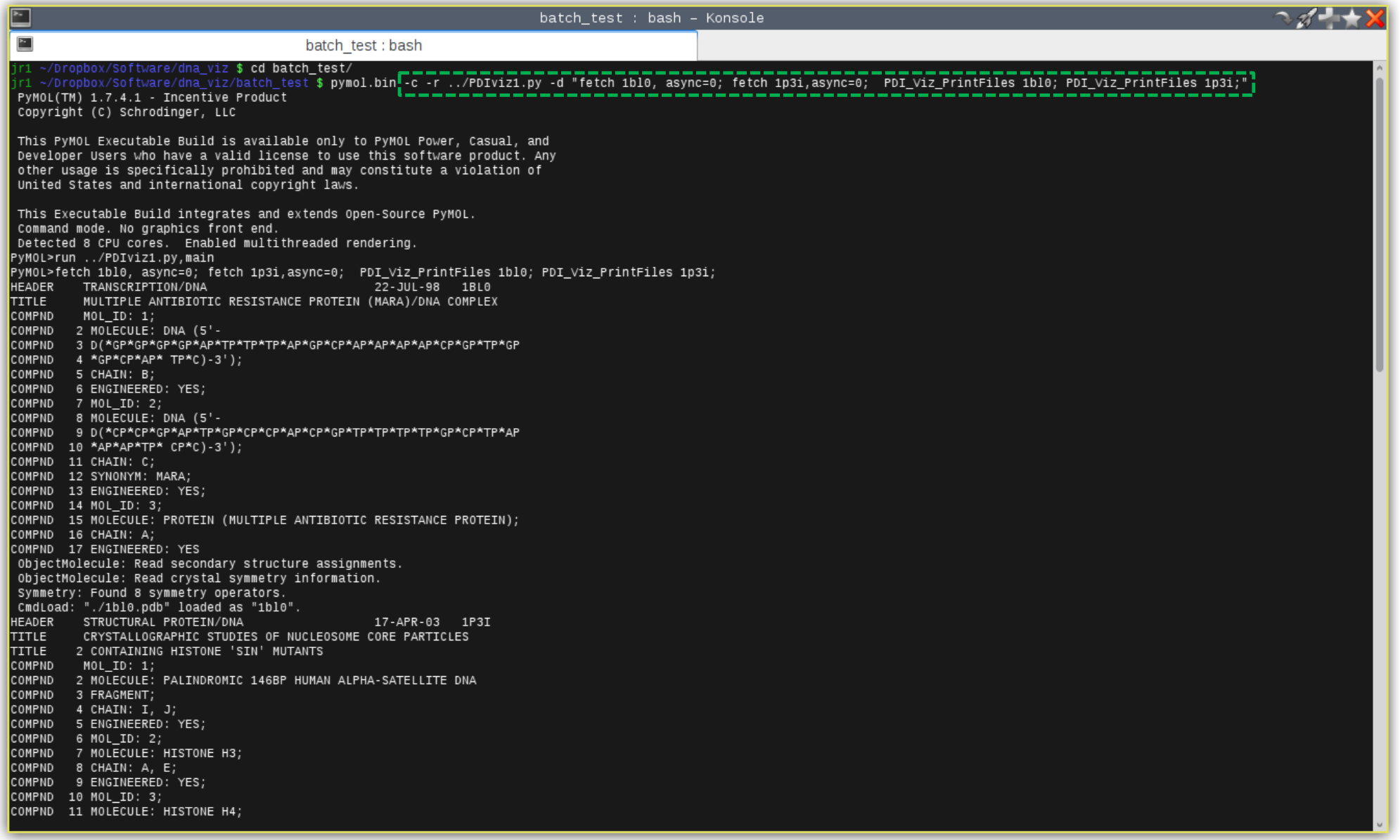
Command Line Functions

PDI_Viz_SaveFasta



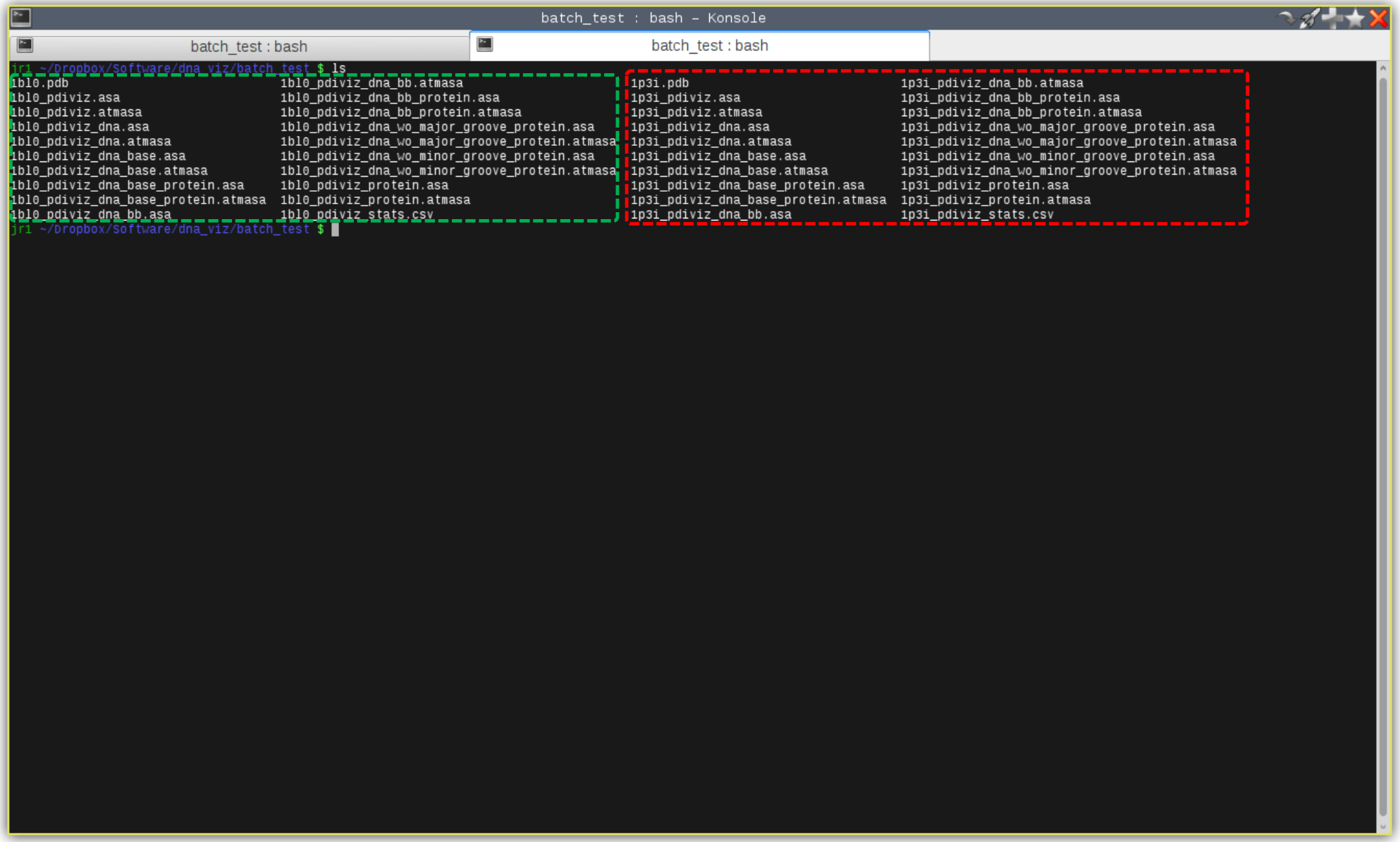
The PDI_Viz_SaveFasta function saves the selected model protein and DNA sequences in a standard FASTA file. Currently it does not support non standard residues or atoms, so they must be removed as shown in the red box. It's only argument is the object name of the protein-DNA complex. The file is saved in the current working directory.

Batch mode



This example will execute the plugin in batch mode from a Unix terminal. We will print the SASA data of two complexes, 1bl0 and 1p3i. In the green box you can see that the plugin should be execute with the `-r` switch, so PyMOL loads the plugin prior to executing any user commands. The `-c` switch tells PyMOL to start in command line mode, without GUI; and the `-d` option tells PyMOL to execute user provided commands. After loading both structures using `fetch`, we call `PDI_Viz_PrintFiles` with the object name as the single argument.

Here we show the console output of the command. Both 1p3i (green box) and 1bl0 (red box) completed without errors.



As a result, a number of output files are generated for 1bl0 (green) and 1p3i (red)

Sample shell script

This following shell script will run PDIviz in batch mode and generate ASA files and PyMOL sessions for all *.pdb files in the current directory.

```
for PDB in *.pdb
    do
        BNAME=$(basename "$PDB")
        PDBID="${BNAME%.*}"
        pymol -c -r PDIviz1.py -d "load $PDB; PDI_Viz_PrintFiles $PDBID,savesession=0"
    done
```