# PyXlinkViewer User Manual

## Requirements

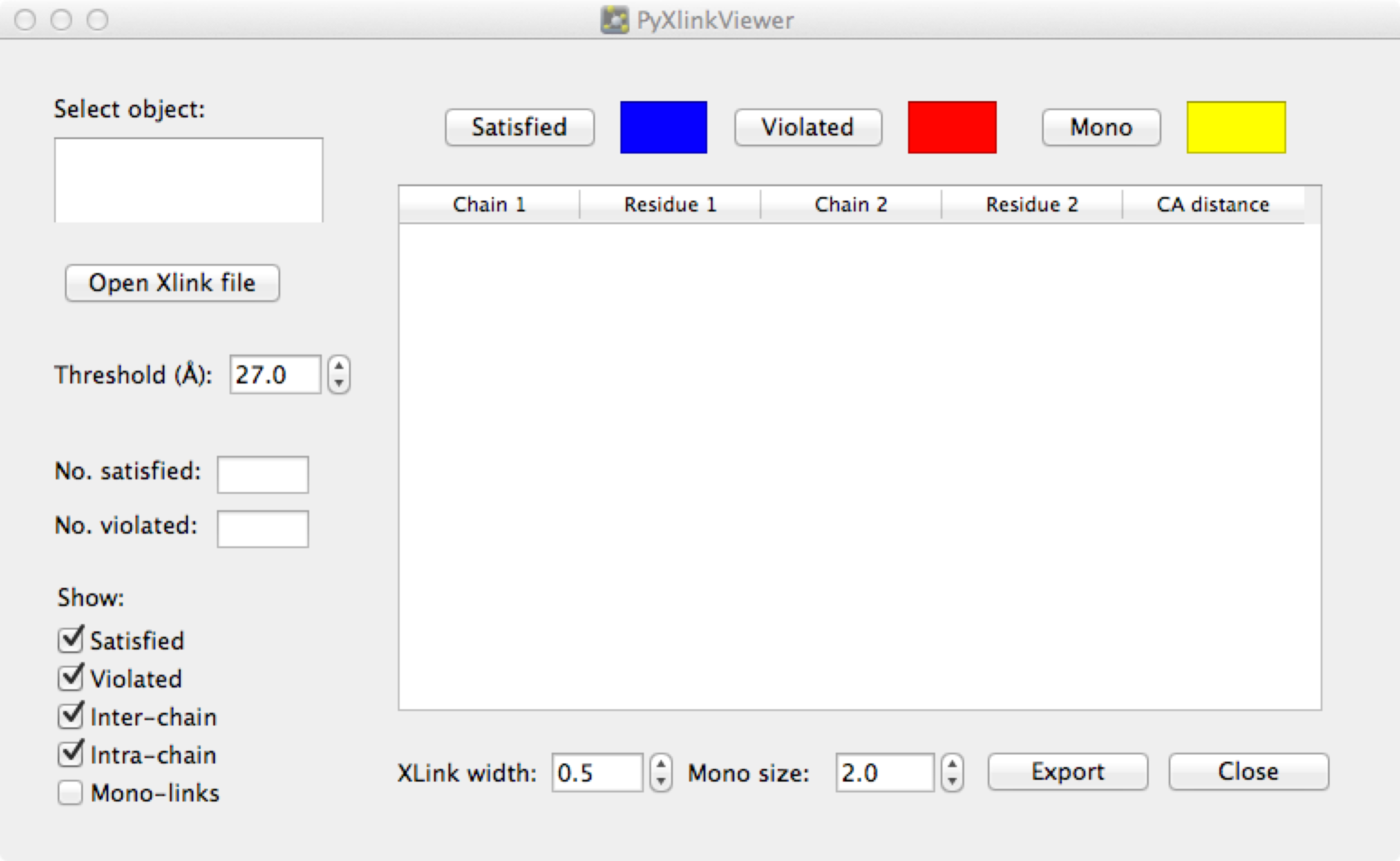
The PyXlinkViewer plugin requires PyMOL v2 to be installed (available at <https://pymol.org/2/>). It can be run on Linux, macOS or Microsoft Windows operating systems. The plugin is downloaded and installed as a ZIP file (‘*pyxlinkviewer\_install.zip’*) available at <https://github.com/BobSchiffrin> under a GNU General Public License.

## Installation

* Open PyMOL v2.
* From the menu bar select *‘Plugin->Plugin Manager’*.
* In the Plugin Manager dialog, select ‘*Install New Plugin’* from the tab list at the top of the dialog.
* In the *‘Install New Plugin’* tab click on the *‘Choose file…’* button.
* In the open file dialog navigate to and select the ‘*PyXlinkViewer\_install.zip’* file and press the ‘*Open*’button.
* A *‘Select plugin directory’* dialog will appear with the default directory ‘<*user-home-directory*>/.pymol/startup’. Click OK to install here and the plugin will load each time PyMOL is started up.
* A message box with the message ‘Plugin "PyXlinkViewer" has been installed.’ should appear. Click OK.
* A new menu item ‘*PyXlinkViewer’* should now be visible in the PyMOL *‘Plugin’* menu.

## Usage and dialog controls

To use PyXlinkViewer, first one or more PDB files, including the PDB file associated with the crosslinking data, are loaded into PyMOL. Next, PyXlinkViewer is started by selecting *’PyXlinkViewer’* from the PyMOL *‘Plugin’* menu. The PyXlinkViewer main dialog will appear which consists of the following controls and displays.



### Object selection list box

This is populated with the current PyMOL objects when PyXlinkViewer is started. Click to select the object associated with the crosslink data.

### Open Xlink file button

After object selection, click on this button to open an Open File dialog and select the file containing the crosslinking data. The expected file format is the same as that used for crosslink data by Jwalk [1]. Each line of the file contains information about the protein chain and residue number involved in the crosslink in the following format:

<residue 1>|<chain1>|<residue 2>|<chain 2>|

For example, a line in the file for a crosslink between residue 25 in chain A and residue 31 in chain B would appear in the file as:

25|A|31|B|

For monolinks (dead-ends), the format is the same but leaving out the second residue. For example, a monolink found at residue 180 in chain C would appear as:

180|C|

The file format is fairly permissive. Empty lines in the file are ignored by the parser, allowing consecutive groups of crosslinks to be separated by blank lines. Also, extra spaces can be present within a line, and the final ‘|’ at the end of a line can be left out. However, note that files with lines that don’t follow the format specified above, and/or contain monolink data will be incompatible with Jwalk.

The chain identifiers should match the chains of subunits in the PyMOL object. If one or both of the crosslinked residues are not present in the PDB file, then a warning message is displayed in the PyMOL command window, no crosslink is created in the PyMOL viewer, and in the table entry (see below) the distance column is set to zero.

### Crosslink and monolink table

On opening a crosslink file, on the right of the main dialog a table is populated with the crosslink and monolink data. For crosslinks, the chain and residue position of the crosslinked residues are shown. In addition, Euclidean C-C distances are calculated for each crosslink and displayed in the right-most column. If one or both of the residues is missing in the PDB file then this distance is displayed as zero. Additionally, a warning message is displayed in the PyMOL command window that one or both residues is not present in the PDB file so the crosslink/monolink will not be displayed in the PyMOL viewer. For monolinks, the chain and residue information is given in the Chain 1 and Residue 1 columns with a dash appearing in the other three columns. The table is updated dynamically to reflect whether the user has chosen to show/hide satisfied, violated, intra-protein, or inter-protein crosslinks, or monolinks. The data populated in the table can be exported at any time using the *‘Export’* button.

### Threshold double-spin box

The current threshold value in Angstroms for judging whether a crosslink is satisfied or violated is displayed. The user may edit the value by either clicking on the arrows on the right hand side to raise or lower the threshold, or by clicking in the box and editing the value. On changing the threshold value the colours of the crosslinks in the PyMOL viewer are updated to reflect whether each crosslink is satisfied or violated, at the new threshold value.

### Number satisfied and number violated text boxes

In the number satisfied text box, for the currently set threshold, the number of crosslinks with inter-residue C-C distances less than (or equal to) than the threshold value is shown. Similarly, in the number violated text box, the number of crosslinks with inter-residue C-C distances greater than the threshold value is shown. These numbers are dynamically updated as the user changes the threshold value. Note that the numbers displayed do not include crosslinks for which residues are missing in the PDB file, as for these crosslinks the inter-residue distances cannot be calculated.

### Show satisfied, violated, inter-chain, intra-chain and monolink checkboxes

If the satisfied/violated/inter-chain/intra-chain/monolink checkboxes are checked or unchecked the corresponding crosslinks/monolinks in the PyMOL viewer are shown or hidden, respectively. For a crosslink to be displayed, both its satisfied/violated status and its type (inter- or intra-chain) must be checked. For example, if a crosslink is between residues in two different chains and the inter-residue C-C distance is less than the current threshold, both the inter-chain and satisfied checkboxes must be checked for it to be displayed. In addition to changing the crosslinks/monolinks seen in the PyMOL viewer the table of crosslinks in the dialog is updated dynamically to contain rows corresponding to the crosslinks/monolinks displayed in the PyMOL viewer.

### Change satisfied, violated and monolink colour buttons

In the top right of the PyXlinkViewer dialog three buttons allow the user to change the colours of the displayed crosslinks and monolinks. On clicking one of these buttons a colour dialog appears allowing the user to select any colour. On selection of a colour, the colour of the crosslinks/monolinks displayed in the PyMOL viewer is updated. Additionally, the colour displayed in the box to the right of the button is updated to reflect the colour selected.

### Xlink width double-spin box

Crosslinks are displayed in the PyMOL viewer as straight lines between the C atoms of the crosslinked residues. This control allows the user to choose the thickness of the lines displayed.

### Monolink size double-spin box

Monolinks are displayed as spheres at the residue Cposition in the PyMOL viewer. This control allows the user to control the size of the sphere.

### Export button

On clicking the ‘Export’ button a ‘Save As’ file dialog appears allowing the user to save the data currently populated in the table as a CSV file. As the show/hide checkboxes on the left of the dialog affect which crosslinks/monolinks appear as rows in the table, the user has control over which crosslinks (satisfied/violated/inter-chain/intra-chain) or monolinks are exported. Besides the data in the table, two additional columns are exported. The first gives the current threshold value, and the second contains either ‘S’ or ‘V’ depending on whether the crosslink is satisfied or violated.

### Close button

Closes the dialog and ends the PyXlinkViewer session.

## References

1. Bullock, J.M.A., Thalassinos, K. and Topf, M. Jwalk and MNXL web server: model validation using restraints from crosslinking mass spectrometry. *Bioinformatics* 2018;34(20):3584-3585.