

# jV User's Guide

version 4.5.4

3 February 2018

## Contents

1. Introduction.....	3
2. System requirements.....	4
3. Tutorials .....	5
Starting up.....	5
Displaying molecules.....	6
Operating molecules.....	7
Displaying polygons .....	9
Working with PDBMLplus .....	10
Animation .....	10
Saving your work.....	11
Terminating the application .....	11
4. GUI .....	12
Menu bar.....	12
Open-Local dialog.....	15
Open-Remote dialog.....	16
Animation control dialog.....	19
5. Document type of the polygon file .....	20
6. Functional site information for molecules .....	22

## 1. Introduction

- jV is an application to display three-dimensional structure of protein and nucleic acid molecules. jV is developed in Java and uses JOGL API (<http://jogamp.org/jogl/www/>) which is a Java binding for OpenGL for 3D display processing. Therefore, jV runs on a platform on which the Java Runtime Environment (JRE) and JOGL are available (includes Windows, Mac OS X, and some typical Linux distributions). It is also possible to use jV on a web browser as a Java applet.
- The molecule file to be read is a Protein Data Bank (PDB) format file, a PDBML file which is a XML version of PDB, and a mmCIF file. In addition to molecules, three-dimensional image of arbitrary polygons can be displayed together, where the polygon file to be read is a predefined XML file. Multiple molecular and polygon images are displayed at the same time, and can be operated individually. It is also possible to process an animation for molecules.
- The interactive mouse operations such as rotation or translation to molecular and polygon images are possible and display style of their image can be operated from the GUI menu bar. In addition, a command line interface is also available. GUI design and command syntax follows those in Rasmol (<http://www.openrasmol.org/>) with some extensions. The command line interface provides more detailed operations than GUI.

## 2. System requirements

jV runs on the Java Runtime Environment (JRE) with JOGL API. The version requirements are as follows.

- JRE (includes Java Plug-in) 1.6 or later  
(<http://java.com/download/>)
- JOGL API 2.2.4 or later  
(<http://jogamp.org/jogl/www/>)

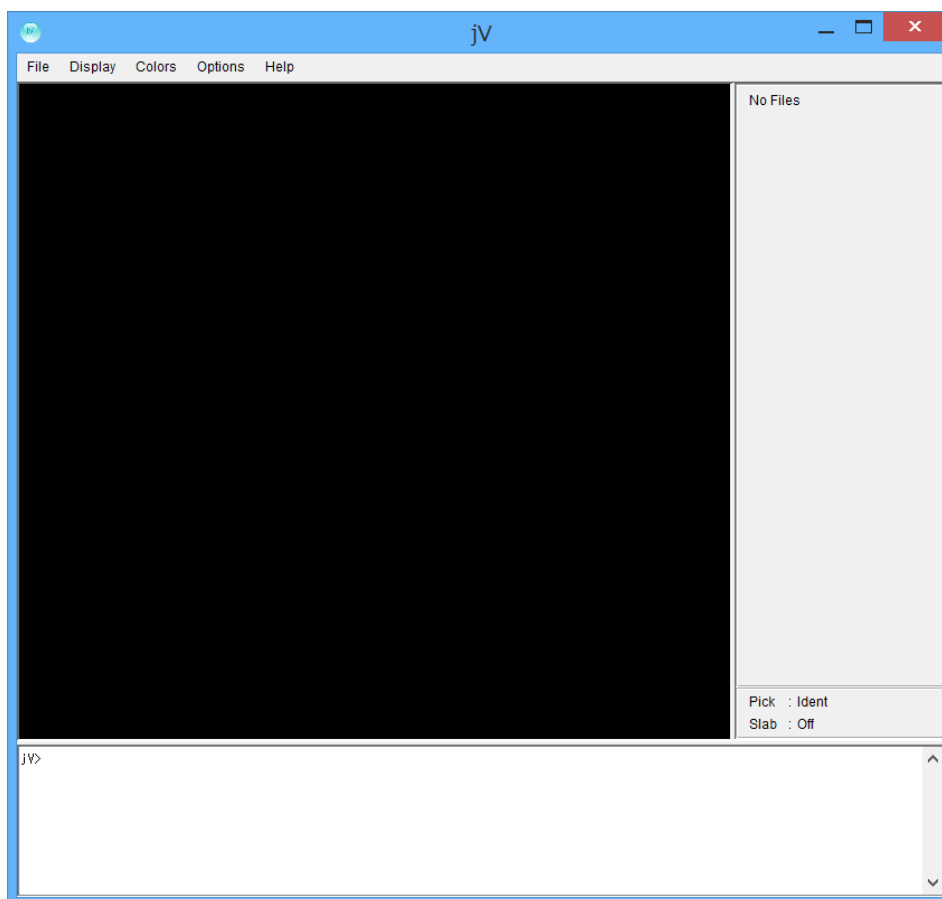
These softwares are available on typical operation systems such as Windows (2000/XP/Vista/7/8/8.1), Mac OS X 10.6 or later, and Red Hat Linux 7.3 or later.

### 3. Tutorials

#### Starting up

It is assumed that Java Runtime Environment has been properly installed. You can start up jV by downloading the binary distribution and running a start-up script contained in the distribution. Another way to use jV is to visit the URL <https://pdbj.org/eF-site/servlet/stable/jV.jnlp> with a web browser, which launch jV via Java Web Start technology. Figure 1 shows the start-up screen. The window has a 3-panel structure, and the upper-left panel with the black background is a 3D image display area. The status of files that have been loaded to the application is shown in the upper-right panel and the lower panel is a command-input and a message-output area.

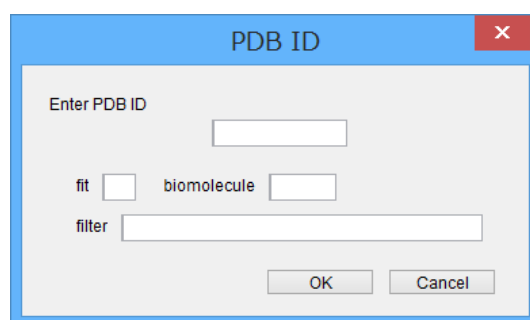
Figure 1. Start-up screen.



## Displaying molecules

If you are working in the environment that is getting on the Internet and know the PDB ID of the molecule that you want to display, the simplest way is as follows. A dialog shown in Figure 2 is opened by selecting the [File] – [Open – Remote] – [PDB ID] menu item. Enter a PDB ID '12as', for example, to the text field at the top of the dialog and click the 'OK' button, then the PDBML file is read through the Internet and the molecule is displayed. A molecule is displayed in a wire frame model with colored by atom type in the initial state.

Figure 2. Dialog to open the molecule specified by PDB ID.



Let us next attempt to load an arbitrary file on the Internet. Here we use 'https://pdj.org/eF-site/jV/1yec.pdb' as an example. The file is a conventional PDB file whose PDB ID is 1yec. In this case we choose the [File] – [Open – Remote] – [PDB] menu item. Then the open-remote dialog appears. Enter the above URL to the text field at the top of the dialog and click the 'OK' button and the molecule is read in and displayed. In jV, each molecule is displayed according to the space coordination described in its file. Therefore, when loading and displaying more than one molecule together, they can be viewed with an appropriate separation in some cases; however, they may be displayed far separately or may be overlapped depending on the combination. In any case, when adding a new file, the position of the viewpoint is automatically controlled for the whole to be displayed.

The coordinate system is the right hand system with the x-y plane being on the screen (the x-axis is a horizontal line with positive values to the right and the y-axis is a vertical line with positive values up). Although, the above examples are about operations for remote files, a PDB file on your local hard disk can also be opened from the [File] – [Open – Local] – [PDB] menu item.

### Operating molecules

Interactive mouse operations are possible for 3D images. By left-dragging the mouse on the 3D-display panel, molecules are rotated around the x-axis or y-axis. By left-dragging the mouse while holding the Shift key on the keyboard, molecules are translated along the z-axis. All mouse operations are as follows (as for the Mac OS X environment, the Command key + left-drag substitutes for the right-drag).

Mouse operations	Actions
Left-drag	The image is rotated around the x (for a vertical motion) and y (for a horizontal motion) axes.
Right-drag	The image is translated on the x-y plane.
[Shift] + left-drag	The image is translated along the z-axis.
[Shift] + right-drag	The image is rotated around the z-axis.
[Alt or Ctrl] + left-drag	The z-clipping plane is shifted when [Option]-[Slab] is checked.
Left-click	An atom of the molecule displayed on the screen is picked.

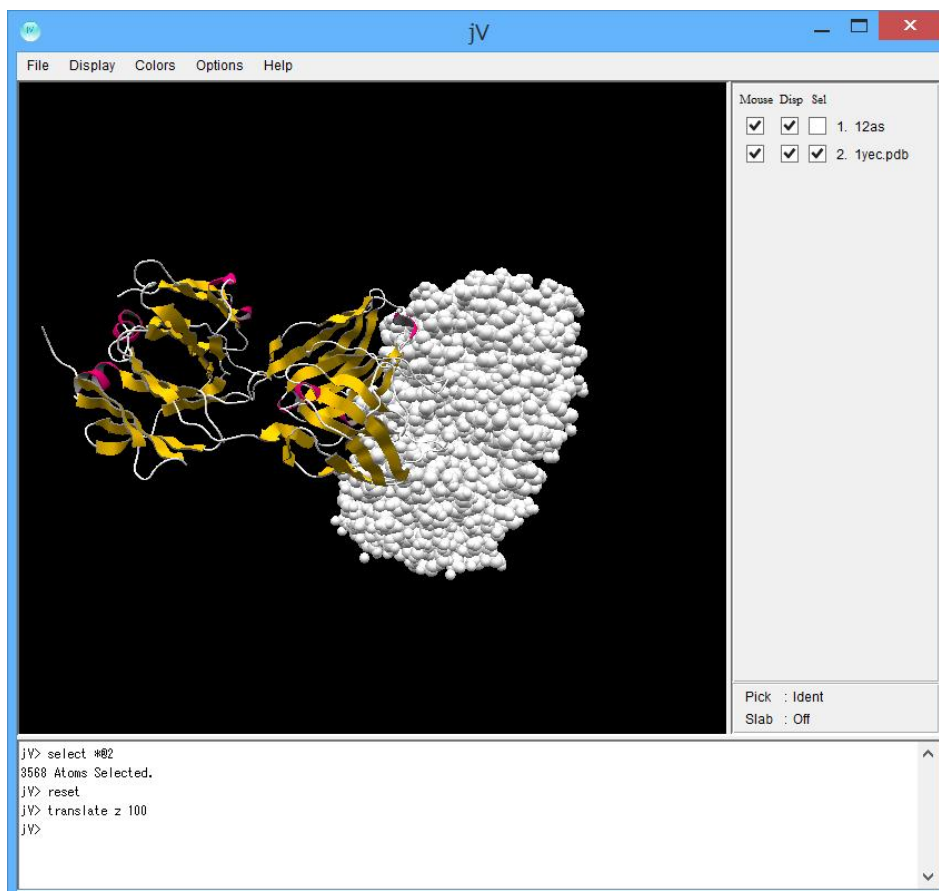
In the upper-right panel in the application window (see Figure 3), the file names that have been loaded to the application are listed, where sequential integers are automatically assigned to the individual file. Each integer is a file ID that can be used to specify the file in the command line interface. In addition, three checkboxes are attached to each file. If you change the status of a 'Disp' checkbox, the corresponding molecule is switched to be displayed or not to be displayed. If you change the status of a 'Mouse' checkbox, the corresponding molecule is switched to be a target of the mouse operations or not. For example, set the both 'Disp' checkboxes on, and the both 'Mouse' checkboxes off. In this case, either molecule does not move against mouse operations.

Subsequently set the 'Mouse' checkbox of file 1 on. In this case, mouse operations work for the molecule 1. Thus, each file is assigned either selected or not selected, and transform operations, such as rotation or translation, act only on selected files. On the other hand, every atom in a molecule is also assigned either selected or not selected, independently of whether the molecule itself is selected or not. Operations that modify display models or colors work for selected atoms. To see this, choose the [Display] – [Spacefill] menu item. Then the both molecules are displayed with the space fill model because all atoms are initially selected. In the same way, if you choose the [Colors] – [Monochrome] menu item, the both molecules are colored white. A 'Sel' checkbox selects or disselects all atoms in each file.

Let us next try a more detailed operation. Type 'select \*@2' in the command line area and press Enter key. This command means to select all atoms contained in the file 2 (the usage of the select command is summarized in the reference manual). Subsequently, choose the [Display] – [Cartoon] menu item. As a result, only the molecule 2 is displayed in the cartoon model. Furthermore select the [Colors] – [Structure] menu item, and each residue of the molecule 2 is colored according to its secondary structure type (see Figure 3).

In order to reproduce the same view as that in Figure 3, first execute 'reset' command. It restores the position of the viewpoint and the transform of every file to the initial state. Note that the command line interface tries to complement a command word when the Tab key is pressed. If you enter 'res' to the command line and subsequently press the Tab key, the command 'reset' is automatically entered. Next set the both 'Mouse' checkboxes on, and execute 'translate z 100' command. This command means to translate the molecules along the z-axis by 100 angstroms.

Figure 3. An example of molecular images.



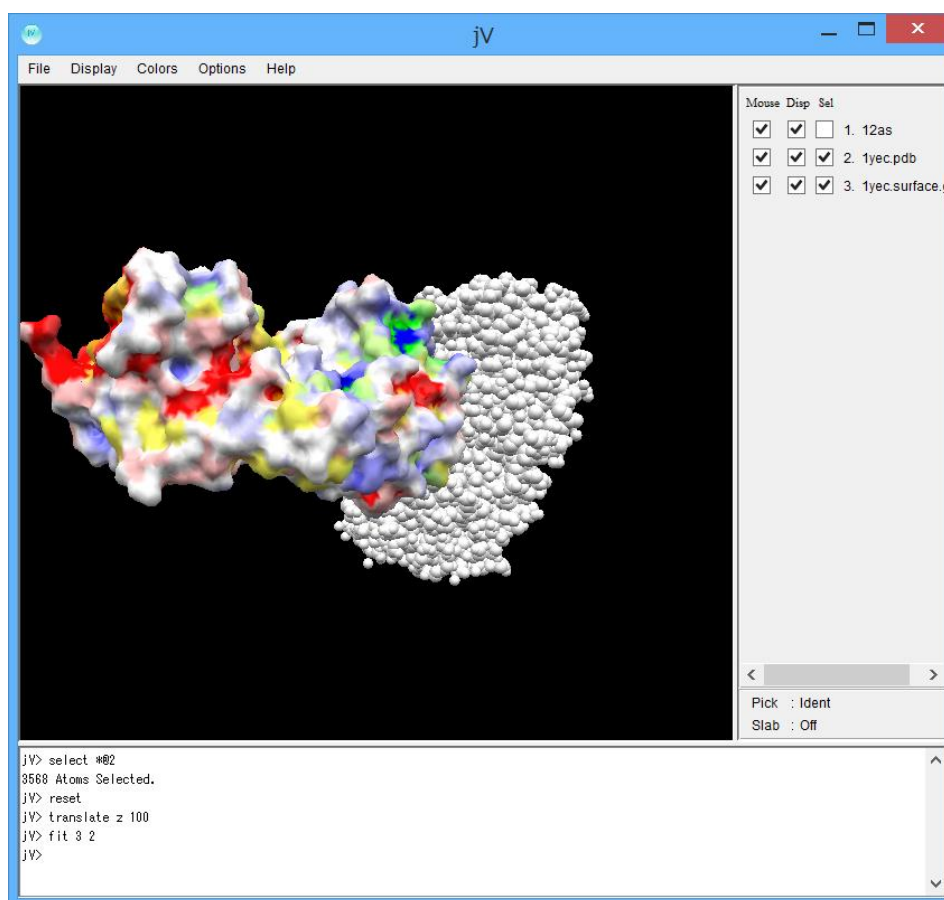


## Displaying polygons

In addition to molecules, arbitrary shapes described by polygons can be read in and displayed together, where the polygon file is an XML file, whose document type is described in the section 6. Here, we use 'https://pdj.org/eF-site/jV/1yec.surface.gz' as an example. This polygon file represents a molecular surface of the molecule 2 (1yec) and the surface is colored according to the electrostatic potential and the hydrophobicity. It is assumed that the current image is the same as that in Figure 3. Then, open the open-remote dialog by choosing the [File] – [Open – Remote] – [Polygon] menu item, enter the above URL to the text field at the top of the dialog, and click the 'OK' button. As a result, the polygon is loaded, displayed and assigned file ID 3.

Note that the polygon is displayed at the different position from the molecule 2 since the molecule 2 is translated. However, because the polygon and the molecule 2 are described by the same coordinates in their files, their images can be exactly superposed by executing 'fit 3 2' from the command line, where the 'fit' command performs the copy of the transform matrix. Finally, the image in Figure 4 appears.

Figure 4. An example of polygon images.



You can control the transparency rate of polygons. Try 'set transparency 0.5' from the command line, where 0.5 stands for 50% transparency. To restore the image, execute 'reset transparency' or 'set transparency 0'. In the same sense as each atom of a molecule, each vertex of a polygon is assigned either selected or not selected. All vertices are selected in the initial state. Execute a command 'displayvertex off' and all vertices become invisible. Next execute 'selectvertex within(10.0, 1&\*H@2)', where this means to select vertices that exist within 10 angstroms from each atom that belongs to the first residue of the 'H' chain of the molecule 2. Subsequently execute 'displayvertex on'. As a result, only the selected vertices appear. The usage of the selectvertex command is summarized in the reference manual.

### Working with PDBMLplus

Functional site information on a molecule stored in PDBMLplus is available in jV. Before trying this, let us close the molecule 2 and the polygon 3 to make the situation simple. Select the [File] – [Close] – [2. 1yec.pdb] and [File] – [Close] – [3. 1yec.surface.gz] menu items sequentially. Then execute a 'reset' command and the transform of the molecule 1 and the position of the viewpoint are initialized.

Next, confirm that the 'Mouse' checkbox of the file 1 is checked, and try a 'show pdbj' command. The functional site information of the molecule '12as' is read from the PDBMLplus and 'type' and 'subtype' keywords for each functional site are listed. These keywords can be used to select a set of atoms. For example, if a keyword 'CATRES' exists (probably it does), the command 'select pdbj:CATRES' selects the corresponding atoms.

In addition, gene ontology information can be obtained from the PDBMLplus. The gene ontology information for the molecule 12as is displayed by the command 'show godata'. Similar operations can be applied to other external databases on the Internet provided that the external database supports the interface that jV requires. See the 'Functional site information for molecules' section for details.

### Animation

The animation of molecules can be performed in jV. The animation file takes the PDB format, where the MODEL line divides each frame. Let us open an animation file 'https://pdbj.org/eF-site/jV/anim.pdb.gz' as an example from the open-remote dialog shown by selecting the [File] – [Open – Remote] – [Animation] menu item. When an animation file is loaded to the application, the first frame is automatically displayed in the wire frame model. You can start the animation by using the animation-control

dialog that is shown by choosing the [Options] – [Animation] menu item. The dialog provides operations such as start and stop of the animation, selection of the animation mode, and adjustment of the speed. Some sample figures of the dialog are shown in the ‘GUI’ section.

#### Saving your work

While in your work, you can create a script file that contains a set of commands that reproduces the current work by using the [File] – [Save] – [Script] menu item. The script file can be loaded by the [File] – [Open – Local] – [Script] menu item. On the other hand, the [File] – [Save] – [mmCIF] menu item and [File] – [Save] – [PDB] menu item enable to save a molecule to a mmCIF and PDB format file, respectively, with the coordinates of the atoms being transformed according to the current image. Here, only one molecule must be selected and only selected atoms in the molecule are saved. The other two menu items in the [File] – [Save] menu, [PNG] and [JPEG], save the current screen image to an image file in the PNG and JPEG format, respectively. The created image file has the same size as that of the 3D-display panel in the application.

#### Terminating the application

The application is terminated when the [File] – [Exit] menu item is chosen or the application window is closed by the way your operating system provides. You can also quit the application by executing ‘exit’ or ‘quit’ command.

## 4. GUI

### Menu bar

The organization of the menu bar and the function of its components are shown below.

#### 1) File menu

Component		Function
Open - Local	mmCIF	A local mmCIF file is opened.
	PDBML	A local PDBML file is opened.
	PDB	A local PDB file is opened.
	Polygon	A local polygon file is opened.
	Polygon CGO	A local polygon CGO file is opened.
	Script	A local script file is opened.
	Animation	A local animation file is opened.
Open - Remote	PDB ID	A PDBML file at the PDBML FTP site is opened.
	eF-site ID	A set of molecular and polygon files stored in eF-site database are opened.
	mmCIF	A remote mmCIF file is opened.
	PDBML	A remote PDBML file is opened.
	PDB	A remote PDB file is opened.
	Polygon	A remote polygon file is opened.
	Polygon CGO	A remote polygon CGO file is opened.
	Script	A remote script file is opened.
Animation	A remote animation file is opened.	
Information		Information about the selected files are shown.
Close		The specified file is closed.
Save	mmCIF	A mmCIF format file that contains the current atom coordinates is created.
	PDB	A PDB format file that contains the current atom coordinates is created.
	Script	A script by which the present condition is reproduced is created.
	PNG	The current image is saved as PNG.
	JPEG	The current image is saved as JPEG.

Exit	The application is terminated.
------	--------------------------------

2) Display menu

Component	Function
Wireframe	The selected atoms are displayed in a wire frame model.
Backbone	The selected atoms are displayed in a backbone model.
Sticks	The selected atoms are displayed in a stick model.
Spacefill	The selected atoms are displayed in a space-fill model.
Ball&Stick	The selected atoms are displayed in a ball & stick model.
Ribbons	The selected atoms are displayed in a ribbon model.
Cartoon	The selected atoms are displayed in a cartoon model.

3) Colors menu

Component	Function
Monochrome	The selected atoms are colored in white.
CPK	The selected atoms are colored by CPK color scheme.
Shapely	The selected atoms are colored by the color scheme in which each amino acid and nucleic acid is assigned a unique color according to the amino acid and nucleic acid properties.
Group	The atoms of every chain are drawn as a smooth spectrum from red (N-terminal of the molecule) to blue (C-terminal).
Chain	The selected atoms are colored by the color scheme in which each chain is assigned a unique color.
Temperature	The selected atoms are drawn as a smooth spectrum from red (high value) to blue (low value) according to the value of the temperature factor.
Structure	The secondary structures are colored by the color scheme in which each secondary structure is assigned a unique color.
Charge	The selected atoms are drawn as a smooth spectrum from blue (positive) to red (negative) according to the charge.
Amino	The selected atoms are colored by the color scheme in which each amino acid is assigned a unique color.

## 4) Options menu

Component		Function
Hetero Atoms		Users can choose whether the heterogeneous atoms described by 'HETATM' in PDB files are selected or not.
Hydrogens		Users can choose whether the hydrogen atoms are selected or not.
Slab		Users can choose whether the z-clipping plane is located on the default position or is shifted.
Stereo	Off	Disables stereo display.
	Wall-Eyed	Enables side-by-side wall-eyed stereo viewing.
	Cross-Eyed	Enables side-by-side cross-eyed stereo viewing.
	Red-Blue	Enables red-blue anaglyph 3D rendering.
	Red-Cyan	Enables red-cyan anaglyph 3D rendering.
	Red-Green	Enables red-green anaglyph 3D rendering.
Load To Center		Users can choose whether a new image is added to the center of screen or not when a new file is opened.
Pick	Off	The mouse-pick is disabled.
	Ident	The atom and residue names, their serial numbers, chain identifier and file ID of the object picked by mouse-click are represented.
	Coordinates	The atom and residue names, their serial numbers, chain identifier, file ID and coordinate of the object picked by mouse-click are represented.
	Distance	The distance between the first and second clicked atoms is calculated and shown.
	Center	The center of rotation and window are transferred to the clicked position.
	Select	Only clicked files are operated.
Animation		The animation control dialog is opened.

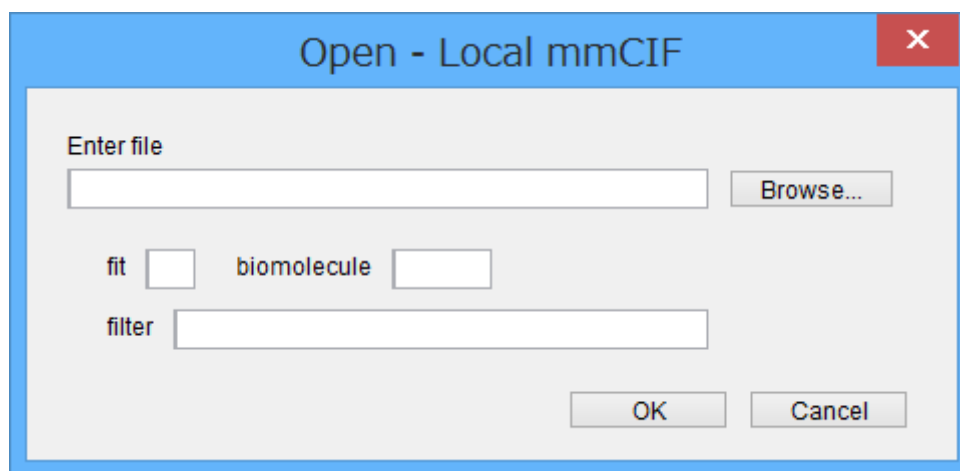
## 5) Help menu

Component	Function
About jV	Information about this application is shown.
Open Manual	A manual page is opened with a web browser.

## Open-Local dialog

Selecting each menu item in the [File] – [Open – Local] menu opens the open-local dialog. For example, Figure 5 shows a dialog opened by the [File] – [Open – Local] – [mmCIF] menu item.

Figure 5. The open-local dialog.



Users can enter a local file path at the 'Enter file' text field, or select a file by using a file chooser dialog that your operating system provides by clicking the 'Browse...' button. If a file ID that is already opened in the application is specified at the 'fit' text field, the transform of the newly opened file is set identical to that of the specified file. A biomolecule structure can be specified by entering assembly ID at the 'biomolecule' text field. For a biomolecule structure, a distinct model ID is assigned to each transform operation, which is listed by 'show information' command. Note that the biomolecule parameter is not available for flat PDB files. When a filter expression is entered at the 'filter' text field, only atoms that matches a given filter are loaded. The filter takes any atom expressions except some keywords (bonded, helix, sheet, turn, within and pdbj:). The atom expression should be double-quoted or enclosed in a parenthesis when it contains spaces.

## Open-Remote dialog

### (1) URL with parameters

Selecting [mmCIF], [PDBML], [PDB], [Polygon], [Polygon CGO], [Script] or [Animation] menu item in the [File] – [Open – Remote] menu opens the open-remote dialog that can handle a URL with parameters. For example, Figure 6 shows a dialog opened by the [File] – [Open – Remote] – [mmCIF] menu item.

Figure 6. The open-remote dialog.

The dialog box is titled "Open - Remote mmCIF" and features a close button in the top right corner. It contains the following elements:

- An "Enter URL" text input field.
- Two checkboxes labeled "fit" and "biomolecule".
- A "filter" text input field.
- A checked checkbox labeled "Attach Parameters".
- A table with three columns: "name", "value", and "type". The table is currently empty.
- Three buttons to the right of the table: "Add", "Remove", and "Clear".
- A "delimiter:" text input field with a comma character.
- Five buttons at the bottom: "Load", "Save", "OK", "Cancel", and "Submit".

When 'Add' button is clicked, a new parameter line is appended in the table at the center of the dialog. Figure 7 shows an example of a URL with one parameter. By clicking the 'Add' button the necessary number of times, you can attach multiple parameters. When 'Remove' button is clicked, a highlighted line in the parameter table is removed. When 'Clear' button is clicked, all parameters are removed.



Figure 7. Example of a URL with a parameter.

(2) 'type' column and mouse pick

The 'type' column in the parameter table is selected from the follows.

type	values	mouse pick
text	arbitrary value	does not work
atom ID	atom ID	work
atom	set of chain ID, residue ID and atom ID	work
residue	set of chain ID and residue ID	work
orig_coords	atom coordinates written in the molecule file	work
curr_coords	atom coordinates currently displayed	work

For the case of a type to which mouse pick works, selecting an atom in the screen by mouse pick sets the corresponding value to the 'value' column in a highlighted line in the parameter table. The value of the 'delimiter' field is used to delimit atom's x, y, and z coordinate.

(3) Saving and loading the configuration

By clicking the 'save' button, a file chooser dialog appears and the current configuration

can be saved to a file. In the same way, a configuration can be loaded by clicking the 'load' button. The contents of a configuration file are in XML format as follows.

```
<?xml version="1.0"?>

<remote_file url="http://example.com">
  <params attach="true" coordinates_delimiter=",">
    <param name="param1" value="value1" type="text"/>
  </params>
</remote_file>
```

#### (4) fit, biomolecule and filter

Three text fields entitled fit, biomolecule and filter work the same way as opening a local file. How to use them is described at the previous 'Open-Local dialog' section.

#### (5) Execution

When the 'OK' button is clicked, a new file is loaded and the dialog is closed. When you use 'Submit' button, on the other hand, the dialog remains visible even after a new file is loaded.

### Animation control dialog

The animation control dialog is visible only if [Options] - [Animation] menu is checked. When there is no animation file loaded, the animation control dialog is as Figure 8. If some animation files are opened, the animation control dialog is altered according to the number of the files. Figure 9 indicates the dialog in the presence of two files.

Figure 8. The animation control dialog in the absence of files.

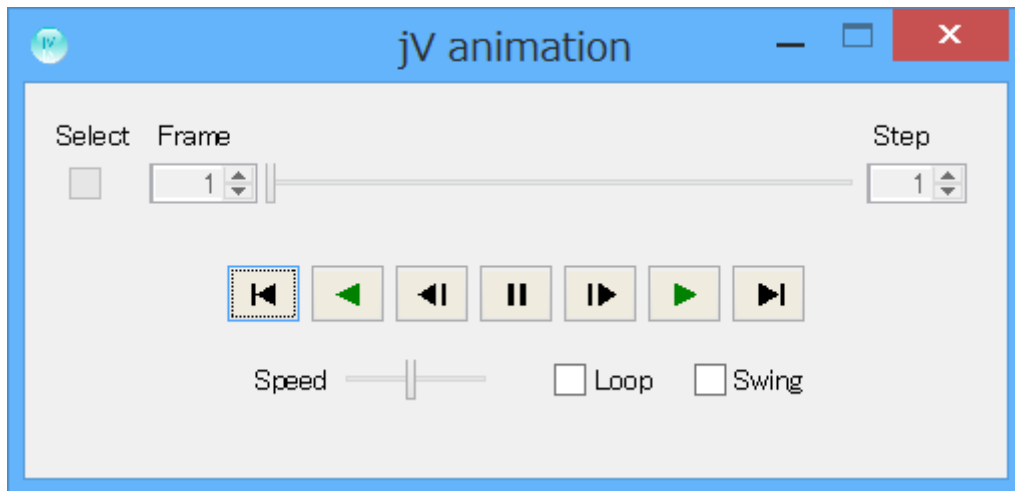
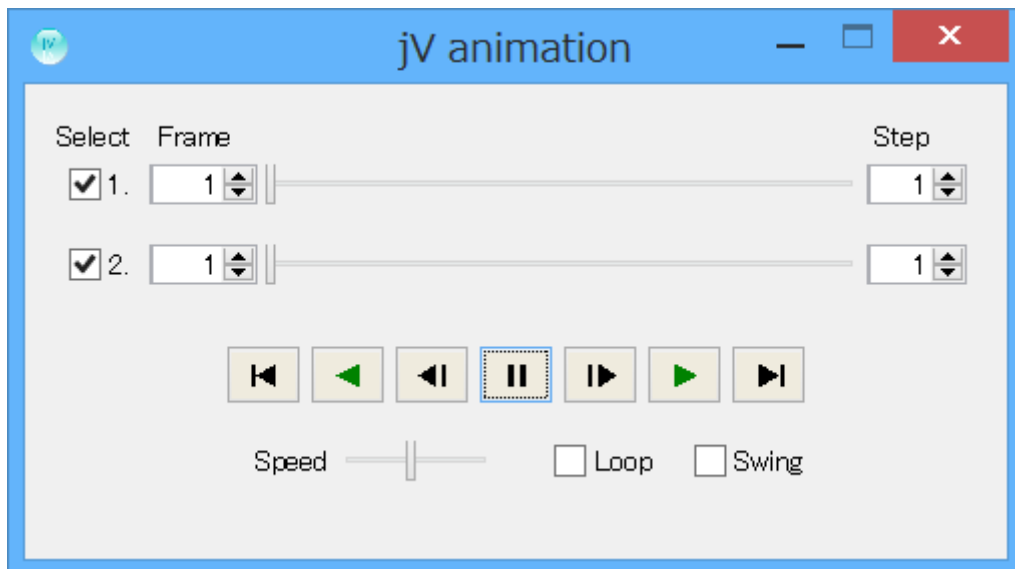


Figure 9. The animation control dialog in the presence of two files.



Users can select which files are controlled by their checkbox. The frame position and playback and stop of the animation can be controlled individually to each file. The playback-speed and the selection of 'loop' and 'swing' modes are common for all files.

## 5. Document type of the polygon file

The contents of a polygon file are organized as follows.

```
<?xml version="1.0"?>

<polygon>
  <vertices count="100" id_numbers=" use" >           ←impossible to omit
    <vertex id="1" image="x y z nx ny nz R G B"/>
    <vertex id="2" image="x y z nx ny nz R G B"/>
    . . .
  </vertices>

  <point_array count="2" size=" 3" >                 ←possible to omit
    <point id="1" vertex="v1"/>
    <point id="2" vertex="v1"/>
  </point_array>

  <line_array count="2" width=" 2" >                 ←possible to omit
    <line id="1" vertex="v1 v2"/>
    <line id="2" vertex="v1 v2"/>
  </line_array>

  <triangle_array count="2" transparency=" 0.5" >   ←possible to omit
    <triangle id="1" vertex="v1 v2 v3"/>
    <triangle id="2" vertex="v1 v2 v3"/>
  </triangle_array>

  <quad_array count="2">                             ←possible to omit
    <quad id="1" vertex="v1 v2 v3 v4"/>
    <quad id="2" vertex="v1 v2 v3 v4"/>
  </quad_array>

  <polyline_array count="2" width=" 1" >             ←possible to omit
    <polyline id="1" vertex="v1 v2 v3 v4 v5 . . ."/>
    <polyline id="2" vertex="v1 v2 v3 v4 v5 . . ."/>
  </quad_array>

</polygon>
```

The 'vertices' element contains the sequence of vertices and the number of vertices is specified by the 'count' attribute. If the value of the 'id\_numbers' attribute is 'ignore', the vertex id is automatically numbered starting from 1. If the 'id\_numbers' attribute is set to be 'use', users can assign arbitrary numbers to each vertex id. In the 'vertex' element, x, y and z represent the x, y and z coordinates of the vertex, respectively, and nx, ny and nz represent the x, y and z components of the normal vector at the vertex, respectively. They are assigned real numbers. The color of each vertex is defined according to the RGB color scheme; R, G and B take the real numbers from 0 to 255. The 'point\_array', 'line\_array', 'triangle\_array', 'quad\_array' and 'polyline\_array' elements contain the sequence of polygons such as dots, lines, triangles, quadrangles and polylines, respectively. They can be omitted. The v1, v2, v3, v4 and v5 describe id of the vertices

constructing the polygon. For these elements, the 'transparency' attribute that is omissible can be used to set the transparency of the image. To specify the size of dots and the width of lines, the 'point\_array' element has the omissible 'size' attribute and 'line\_array' and 'polyline\_array' elements have the omissible 'width' attribute, respectively. The XML schema file that describes the type definitions and element declarations is published at <http://www.pdbj.org/jv/polygonSchema>. A simple example of a polygon is shown below. It represents a colored cube. The file is published at <https://pdbj.org/eF-site/jV/cube.xml>. Before you open this polygon, you should close all files currently loaded to the application to get an appropriate viewpoint for the cube. To do this, you can use [File] – [Close] menu, or execute 'zap' in the command line.

```

<?xml version="1.0"?>
<polygon>
  <vertices count="24" id_numbers="use">
    <vertex id="1" image=" 1 1 1 0 0 1 255 0 0"/>
    <vertex id="2" image="-1 1 1 0 0 1 255 0 0"/>
    <vertex id="3" image="-1 -1 1 0 0 1 255 0 0"/>
    <vertex id="4" image=" 1 -1 1 0 0 1 255 0 0"/>
    <vertex id="5" image="-1 1 -1 0 0 -1 0 255 255"/>
    <vertex id="6" image=" 1 1 -1 0 0 -1 0 255 255"/>
    <vertex id="7" image=" 1 -1 -1 0 0 -1 0 255 255"/>
    <vertex id="8" image="-1 -1 -1 0 0 -1 0 255 255"/>
    <vertex id="9" image=" 1 1 1 1 0 0 0 255 0"/>
    <vertex id="10" image=" 1 -1 1 1 0 0 0 255 0"/>
    <vertex id="11" image=" 1 -1 -1 1 0 0 0 255 0"/>
    <vertex id="12" image=" 1 1 -1 1 0 0 0 255 0"/>
    <vertex id="13" image="-1 1 -1 -1 0 0 255 0 255"/>
    <vertex id="14" image="-1 -1 -1 -1 0 0 255 0 255"/>
    <vertex id="15" image="-1 -1 1 -1 0 0 255 0 255"/>
    <vertex id="16" image="-1 1 1 -1 0 0 255 0 255"/>
    <vertex id="17" image=" 1 1 1 0 1 0 0 0 255"/>
    <vertex id="18" image=" 1 1 -1 0 1 0 0 0 255"/>
    <vertex id="19" image="-1 1 -1 0 1 0 0 0 255"/>
    <vertex id="20" image="-1 1 1 0 1 0 0 0 255"/>
    <vertex id="21" image=" 1 -1 -1 0 -1 0 255 255 0"/>
    <vertex id="22" image=" 1 -1 1 0 -1 0 255 255 0"/>
    <vertex id="23" image="-1 -1 1 0 -1 0 255 255 0"/>
    <vertex id="24" image="-1 -1 -1 0 -1 0 255 255 0"/>
  </vertices>
  <quad_array count="6">
    <quad id="1" vertex=" 1 2 3 4"/>
    <quad id="2" vertex=" 5 6 7 8"/>
    <quad id="3" vertex=" 9 10 11 12"/>
    <quad id="4" vertex="13 14 15 16"/>
    <quad id="5" vertex="17 18 19 20"/>
    <quad id="6" vertex="21 22 23 24"/>
  </quad_array>
</polygon>

```

## 6. Functional site information for molecules

A simple XML file is defined in jV in order to describe functional sites of a molecule. If an external database server that returns this type of XML file is prepared, you can let jV read the file through the network and obtain the functional site information. An example of the XML file is as follows.

```
<?xml version=" 1.0" ?>

<site_list>
  <site id=" CATRES1" db=" CATRES" category=" catalytic"
    description=" a catalytic site defined by CATRES, Medline 98100076" >
    <region chain_id=" A" beg_seq_id=" 100" end_seq_id=" 100" />
    <region chain_id=" A" beg_seq_id=" 46" end_seq_id=" 46" />
    <region chain_id=" A" beg_seq_id=" 116" end_seq_id=" 116" />
  </site>
  <site id=" ASN" db=" pdb_hetatom" category=" binding"
    description=" ASPARAGINE binding site" >
    <region chain_id=" A" beg_seq_id=" 48" end_seq_id=" 48" />
    <region chain_id=" A" beg_seq_id=" 72" end_seq_id=" 74" />
  </site>
  <site id=" 0006529" db=" godata" category=" biological_process"
    description=" asparagine biosynthesis" >
    <region chain_id=" A" />
  </site>
  <site id=" 0016874" db=" godata" category=" molecular_function"
    description=" ligase activity" >
    <region chain_id=" A" />
  </site>
</site_list>
```

The XML schema file that defines the document type of the above XML file is published at <https://pdj.org/eF-site/schema/sitelist10.xsd>. When a text file 'properties.txt' exists in the same directory as the application jar file, jV reads the file as a configuration file for the application. The current configuration file, which is attached to the binary distribution, contains a set of URL necessary to connect to the PDBMLplus and eF-site system as follows.

```
# PDBML files are retrieved from the following site.
pdbml_noatom=ftp://ftp.pdbj.org/XML/all-noatom/
pdbml_extatom=ftp://ftp.pdbj.org/XML/all-extatom/
pdbml_plus=http://service.pdbj.org/mine/pdbmlplus/noatom/

# eF-site data are retrieved from the following site.
efsite=https://pdj.org/eF-site/
```

If a database server that returns the functional site file is prepared, its URL should be appended to the configuration file. Because one functional site file is assumed to exist for one PDB ID, the URL written in the configuration file depends on the PDB ID. Therefore, the URL is represented with the use of substitute characters {0}, {1}, {2} and {3} that are substituted by each column of the PDB ID sequentially. An example is as follows.

```
# works with my database.
mydb=http://myhost.jp/mydb/jv/{0}{1}{2}{3}_jv.xml
```

Here, the property name, 'mydb' in the above example, is arbitrary. Note that a PDB ID {0}{1}{2}{3} is substituted as lowercases and available protocols are http:, ftp: and file:. For simplicity, it is assumed that only one molecule file that corresponds to the above example XML file is opened in the application. Then functional site information can be displayed by the 'show' command as follows.

```
jV> show site mydb
File 1:
  db          category
  -----
  CATRES      catalytic
  pdb_hetatom binding
  godata      biological_process
  godata      molecular_function
jV> show site mydb:CATRES
File 1:
  id=CATRES1  category=catalytic  region=A:100-100,A:46-46,A:116-116
  description=a catalytic site defined by CATRES, Medline 98100076
jV> show site mydb:pdb_hetatom:binding
File 1:
  id=ASN      category=binding  region=A:48-48,A:72-74
  description=ASPARAGINE binding site
jV> show site mydb:godata
File 1:
  id=0006529  category=biological_process  region=A
  description=asparagine biosynthesis
  id=0016874  category=molecular_function  region=A
  description=ligase activity
jV> show site mydb:godata:molecular_function
File 1:
  id=0016874  category=molecular_function  region=A
  description=ligase activity
jV>
```

With a similar syntax, the 'select' command can be used to select the relevant set of atoms. To demonstrate the operations described in this section, let us add a following line to the configuration file 'properties.txt'.

```
ex=http://ef-site.hgc.jp/jv3site/servlet/Site?pdb={0} {1} {2} {3}
```

The above URL provides functional site information whose contents are equivalent to those of PDBMLplus for all PDB. To reload the configuration file, quit the application (if it runs) and restart it. Then open the molecule 'lyec' as an example. To do this, you can use [File] – [Open – Remote] – [PDB ID] menu item as in the 'Displaying molecules' section, or equivalently execute 'load ftp lyec' in the command line. Subsequently, execute 'show site ex' and the following keywords will be listed (the contents may change in the future).

```
jV> show site ex
File 1:
  db          category
  -----
  prosite     prosite
  pdb_hetatom binding
```

Detail site information can be shown as follows.

```
jV> show site ex:prosite
File 1:
  id=PS00290 db=prosite category=prosite region=L:192-198
  description=Immunoglobulins and major histocompatibility complex proteins signature.
  [FY]-[L]-C-x-[VA]-[LC]-H
```

Atoms in the region 'L:192-198' can be selected with a similar syntax.

```
jV> select ex:prosite
56 Atoms Selected.
```

In the same way, the following atom selections are possible.

```
jV> select ex:pdb_hetatom
298 Atoms Selected.
jV> select ex:
354 Atoms Selected.
```

Here the last command selects both regions specified by 'prosite' and 'pdb\_hetatom' keywords, respectively.