EZ-Viz movie

Laura Lynn Grell

Follow this and additional works at: http://scholarworks.rit.edu/theses

Recommended Citation

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.
EZ-Viz Movie

Approved: Paul A. Craig, Ph.D.
Thesis Advisor

Gary R. Skuse, Ph.D.
Director of Bioinformatics

Submitted in partial fulfillment of the requirements for the Master of Science degree in Bioinformatics at the Rochester Institute of Technology.

Laura Grell
June 2007
# Table of Contents

Abstract ............................................................................................................................. iii
Thesis/Dissertation Author Permission Statement ................................................................ iv

**Introduction** .................................................................................................................. 1
  What is PyMOL? ............................................................................................................ 2
  Using PyMOL ............................................................................................................... 2
    File Types ................................................................................................................ 2
    Objects & Selections ............................................................................................... 3
    Creating Movies in PyMOL ..................................................................................... 4
  EZ-Viz .......................................................................................................................... 6

**EZ-Viz Movie** ............................................................................................................... 7
  Necessity ....................................................................................................................... 7
  EZ-Viz ‘Make Movies’ Tab ........................................................................................ 8
    Creating a New Movie ............................................................................................ 9
    Editing Frames ....................................................................................................... 11
    Saving Movies ....................................................................................................... 14
    Loading Movies .................................................................................................... 14
    Additional Features .............................................................................................. 15
    Limitations ............................................................................................................. 16

**Results & Discussion** .................................................................................................. 17
  Development .............................................................................................................. 17
    Overview ................................................................................................................. 17
    User Testing ........................................................................................................... 17
    Interacting with PyMOL ....................................................................................... 19
    Revisions ............................................................................................................... 20
  Future Improvements ............................................................................................. 23
    Frame Editor ......................................................................................................... 23
    ‘Make Movies’ Tab ............................................................................................... 24

**Materials & Methods** ................................................................................................. 26

**Conclusions** ............................................................................................................... 27

**Deliverables** .............................................................................................................. 28

**Time Line** .................................................................................................................. 29

**References** ................................................................................................................ 30

**Appendix A** .................................................................................................................. 31

**Appendix B** .................................................................................................................. 36
Abstract

The molecular modeling program PyMOL [1], developed by DeLano Scientific, has the capabilities to produce high-quality molecular images and animations. Although it is a very powerful tool, working with PyMOL requires the use of complex, syntactical commands that can be difficult for users to learn. To address this issue, a plugin coined EZ-Viz was developed in the summer of 2005 to provide an easy-to-use interface for interacting with PyMOL. While the interface replaced many PyMOL commands with buttons and menus, it lacked functionality that was dedicated to aiding users who wanted to create their own custom, molecular animations in PyMOL. The development of EZ-Viz Movie looks to address this issue by implementing a new tab of the EZ-Viz interface where users can create movies. The capabilities of this tab include creating movies from scratch, appending to previously created animations, saving movies for later playback, and editing individual frames.
Thesis/Dissertation Author Permission Statement

Title of thesis or dissertation: EZ-Viz Movie

Name of author: Laura Lynne Grell
Degree: Master of Science
Program: Bioinformatics
College: Gosnell College of Science

I understand that I must submit a print copy of my thesis or dissertation to the RIT Archives, per current RIT guidelines for the completion of my degree. I hereby grant to the Rochester Institute of Technology and its agents the non-exclusive license to archive and make accessible my thesis or dissertation in whole or in part in all forms of media in perpetuity. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation.

Print Reproduction Permission Granted:

I, Laura Lynne Grell, hereby grant permission to the Rochester Institute Technology to reproduce my print thesis or dissertation in whole or in part. Any reproduction will not be for commercial use or profit.

Signature of Author: ___________________________ Date: ____________

Inclusion in the RIT Digital Media Library Electronic Thesis & Dissertation (ETD) Archive

I, Laura Lynne Grell, additionally grant to the Rochester Institute of Technology Digital Media Library (RIT DML) the non-exclusive license to archive and provide electronic access to my thesis or dissertation in whole or in part in all forms of media in perpetuity.

I understand that my work, in addition to its bibliographic record and abstract, will be available to the world-wide community of scholars and researchers through the RIT DML. I retain all other ownership rights to the copyright of the thesis or dissertation. I also retain the right to use in future works (such as articles or books) all or part of this thesis or dissertation. I am aware that the Rochester Institute of Technology does not require registration of copyright for ETDs.

I hereby certify that, if appropriate, I have obtained and attached written permission statements from the owners of each third party copyrighted matter to be included in my thesis or dissertation. I certify that the version I submitted is the same as that approved by my committee.

Signature of Author: ___________________________ Date: ____________
Introduction

Computer aided molecular modeling plays an essential role in the research of biological structures. We have come a long way from the time when we speculated what proteins and other macromolecules looked like, to now, when we can view them three dimensionally in different colors and representations. The images we are able to create have made understanding the relationship between structure and function of these molecules much simpler. While static images provide researchers a means to present snapshots of findings to colleagues, they lack the versatility and movement that can be demonstrated in a movie. In some instances, molecular movies are created to show the structural changes that occur when a ligand binds to the active site or alternate conformations the molecule takes on, while in other cases they can be used to introduce the structural features of the molecule that are important for proper function. For this project we will be focusing on the latter.

PyMOL, a molecular modeling program developed by DeLano Scientific, provides users with the basic tools needed to generate molecular movies. Although PyMOL is a very powerful program capable of producing high-quality images, it has a steep learning curve that can be difficult to overcome. In the summer of 2005, a notebook style, user interface named EZ-Viz [2] was developed to work alongside PyMOL and provided users with an easier way of interacting with the program. The original plugin contained a ‘Movies’ tab that featured four pre-made movies that were generated by extracting data from the PDB file and using that information to issue commands. These simple movies provided users with the means to create a simple and informative molecular animation for any PDB file at the click of the button.
Unfortunately, the pre-made movies have a limited range of use and may leave out important structural components that users may otherwise want to point out. With this in mind, we have created an enhancement to the current version EZ-Viz interface that will allow users to create customized molecular movies to display the many important structural features of their molecule. The ‘Make Movies’ tab maintains the same easy-to-use interface but provide users with options for zooming, rotating and altering structural representations.

**What is PyMOL?**

PyMOL is a powerful, robust molecular modeling program developed by Warren Delano of DeLano Scientific that has the capabilities to produce high quality images, analyze crystallographic data and produce animations. It is used throughout the scientific and educational community to explore structures and produce molecular images that are capable of portraying relevant structural information. Although PyMOL offers its users a great deal of control when it comes to creating images, the user interface is not intuitive and has a steep learning curve.

**Using PyMOL**

**File Types**

When starting to work with a structure in PyMOL, the atomic coordinates must be loaded into the program through a file. The most commonly used file type is the Protein Data Bank or PDB format. Each file from the Protein Data Bank represents an “experimentally determined three-dimensional structure of a biological macromolecule.” [3] Within each file is an abundance of knowledge about a specific structure including
the atomic coordinates, the names of bound ligands, the names and numbers of peptide chains, primary citation records, and additional structural data. Additional file types that can be loaded into PyMOL include MOL and Macromodel files, XPLOR and CCP4 maps, Raster 3D input, and PyMOL session files. Since the original EZ-Viz interface was developed using the information extracted from PDB files, the movie maker tab will do the same.

**Objects & Selections**

To better understand how molecules are manipulated in PyMOL, two concepts must be explained further: objects and selections.

The *Introduction to PyMOL* tutorial [4] describes a PyMOL object as “… a primary representation of a molecule with all the atoms, bonds, colors and display modes in PyMOL’s memory." Each object has its own copy of the atoms it contains stored in memory. When a molecule is loaded into the PyMOL viewing window, a new object, specific to that molecule, is created. The initial object appears in the PyMOL viewer and the name of the object, usually the PDB code, appears in the object list displayed in the internal GUI. (Figure 1)

In the same tutorial, a selection is described as “… a pointer to a defined set of atoms in one or several objects.” [4] Selections do not have their own copy of the atoms they contain and instead simply reference atoms contained within a PyMOL object. It follows that each atom can belong to more than one selection, but only one object.
While selections are commonly used to select a subset of atoms for manipulation using the \textit{select} command, objects can also be created for a subset of atoms using the \textit{create} command. There is not a lot of documentation on the differences between creating an object vs. a selection, however clicking an objects’ name will hide/show it completely while clicking a selections’ name (displayed in parentheses) will only select/deselect it.

Either by creating an object or making a selection, the users have the ability to manipulate them in many ways: applying colors, altering representations and zooming in and out. It is through the manipulation of selections and objects that users can create high-quality molecular images.

\textbf{Creating Movies in PyMOL}

Unlike many other molecular modeling programs, PyMOL has the capabilities to create molecular movies along with static images. Unfortunately, creating movies in PyMOL is an involved process that requires an advanced understanding of PyMOL commands that the typical user may not possess. Although simple movies, like rocking
or rotating a molecule, can be created using only three commands, more complex movies require a series of commands to generate multiple frames with differing contents. Currently the movie-making capabilities of PyMOL focus on having the user understand two essential concepts; states and frames. The PyMOL manual [5] describes these two concepts as follows:

- **States**: “States most directly correspond to particular arrangements of atoms at a point in time. For example, one state could consist of one step in a molecular dynamics simulation or individual points of a coordinate interpolation. If you are making a movie of a static coordinate set (such as a single crystal structure) then you have only one state. All objects in PyMOL can potentially consist of multiple states.”

- **Frames**: “Frames are like the individual images you'd find on a movie reel, except that in PyMOL, frames are composed of states instead of images, and frames can have additional actions associated with them (such as rotation of the camera”).

By using several states, a user can animate the conformational changes within a structure that occur over a short period of time. An example where this type of animation may be useful would be to demonstrate the structural changes that occur in an active site when a ligand binds within. Molecular states are often generated from the experimental results of repeated x-ray diffraction studies on a series of molecular crystals. The structural changes that occur in the molecule to shift from one conformation to the next are many times simulated by a computer program. An example of one such program is
The Yale Morph Server [6] which provides “animations of a plausible or semi-plausible pathway between two submitted protein subunit conformations.” However, since our focus for this project is to allow users to create movies that will introduce viewers to important structural components of a molecule as opposed to demonstrating conformational changes, EZ-Viz Movie will focus on manipulating frames.

Individual frames can be thought of as a single sketch for an animated film, except that our sketch is the molecule’s orientation. Between two sequential frames there is only a small difference in the molecule’s positioning and when shown at a high speed the transition from one position to the next occurs smoothly. Aside from position, we can also change the appearance of the molecule in each frame. From one frame to the next we can alter the color and representation of a given feature to make it stand out more or less than it did in the previous frame.

**EZ-Viz**

Although PyMOL is a powerful tool capable of producing high-quality images, it requires the use of complex commands to manipulate images in the viewing window, making it sometimes difficult to use. The PyMOL User’s Manual readily admits that “development has been focused on capabilities, not on ease-of-use for new users.” To encourage the use of PyMOL and aid new users, a new tabbed interface named EZ-Viz was developed as a plugin for the already existing PyMOL interface. The EZ-Viz interface (Figure 2) limits the number of syntactical commands that the user has to issue and instead allows them to manipulate images by using drop-down menus and buttons. Other features of the interface include nine preset molecular views, a Chime/PyMOL command converter, molecular sequence and hetero atom information, and four
molecular movies. Although the movies show interesting aspects of molecules such as bound ligands or individual chains, they are pre-made, leaving the user with no control over what is shown in the movie. This is the issue we look to address by developing a new tab for the interface where users can create and edit their own molecular movies in PyMOL.

**Figure 2. EZ-Viz Interface**
The Welcome tab of the EZ-Viz interface. This is one of nine tabs contained in the interface that features easy to use buttons, nine preset molecular views, and molecular sequence information.

**EZ-Viz Movie**

**Necessity**

In creating molecular movies, researchers and educators are trying to illustrate some aspect of a structure that is unique or plays an important role in its function. The truth of the matter is that static images simply cannot represent the dynamic nature of protein structure and function as well as a movie can. Because of this, the possibility of creating molecular movies to illustrate these features holds great promise. For example, Molecules in Motions [7] is a company that produces custom-made, interactive molecular
animations that are used in education and research presentations. While the animations they produce are eye-catching and informative, they are made by a professional staff. As it is, no simple method exists for users of PyMOL or other molecular modeling programs to quickly generate their own useful animations.

**EZ-Viz ‘Make Movies’ Tab**

The current version of EZ-Viz has the capabilities to generate four pre-made movies that are automatically made for the user at the click of a button (Figure 3). Although the current movies feature some key aspects of the loaded structure by extracting data from the PDB file, they do not allow the user any control over what is shown in the movie. To address this issue, we have developed an additional tab for the EZ-Viz interface titled ‘Make Movies’. This enhanced version of the interface features a suite of tools that can be used to generate unique, self-made movies.

**Figure 3. Movies Tab of EZ-Viz**

The current Movies tab of the EZ-Viz interface features four pre-made molecular movies. Also included are a play, stop, and rewind button for the user to control playback of the movie.
Upon opening the ‘Make Movies’ tab, the user can choose one of two options; creating a new movie or loading a previously made movie. Clicking the ‘New Movie’ button enables the ‘Add Frames’ features of the interface that will allow users to begin creating a movie. Alternatively, by clicking the ‘Load Movie’ button a file dialog appears asking the user to select the movie they want to load.

Creating a New Movie

The movie making process can be thought of as having two stages; the filming phase, where new frames are added, and the editing phase, where the filmed frames are altered. As is customary with movie production, the filming phase must occur prior to the editing phase. The same is true when using the ‘Make Movies’ tab of the EZ-Viz interface. The user must film their molecule, by adding action frames, prior to editing the appearance of the molecule in those frames.
The ‘Add Frames’ box in the middle of the ‘Make Movies’ tab, Figure 5, contains all the buttons necessary to incorporate actions into the movie. Users can start the process by selecting one of six actions by clicking the corresponding radio button on the left side of the box. The six actions that can be incorporated are: ‘Zoom In’, ‘Zoom Out’, ‘Rotate Y 180’, ‘Rotate X 180’, ‘Rotate Z 180’, and ‘Hold On View’. The ‘Zoom In’ function allows users to include a close up of a chosen ligand, active site, or chain. The feature that is to be “zoomed in on”, can be selected from the drop down menu of the same name. Alternatively, the ‘Zoom Out’ action returns the user’s view to the entire molecule. The three rotate functions generate a 180° rotation of the current view on the specified axis. The vertical axis is represented by Y, the horizontal axis by X, and the diagonal axis by Z. Finally the ‘Hold on View’ option, allows the users to keep the molecule in its current orientation for a specified set of frames.

Included in the ‘Frames’ dropdown are six intervals, ranging from 1 to 200, that can be used to set the duration of the selected action. The number of frames played per second is dependent on the frame speed setting which can be adjusted on the ‘Movies’ tab. The default speed of 15 frames per second would require the user to make 900 frames to have a movie that lasts one minute.
After selecting an action and its duration, the user can then hit the ‘Add Frames’ button to append the frames to the movie. Upon clicking the button, the PyMOL viewer is updated to reflect the added action and the ‘Current Frame’ field is incremented by the number of frames the user added. This field shows the total number of frames that have been successfully added to the movie. Frames can be added dynamically, with no limit on the number that can be added. Additionally if the user adds frames by mistake, clicking the ‘Undo’ button will remove the frames and revert the molecule to the position it was in prior to the frames being added. At any point during the frame adding process, the user may view their current progress by pressing the ‘Preview’ button. This will show a rough version of the users movie in the viewing window. When the frame addition phase is finished, the user can complete their movie by clicking the ‘Make Movie’ button. This will stitch the movie together and enable the ‘Edit Frames’ button.

**Editing Frames**

Once the movie is stitched together, the user can then go frame by frame and make any changes they deem necessary by clicking the ‘Edit Frames’ button. This will open the ‘Frame Editor’ window, shown in Figure 6, which has a series of dropdown menus that allow the user to select molecular features and alter their color and appearance. The selection menu contains all of the same molecular selections featured on the ‘Commands’ tab of the original EZ-Viz interface, along with some additional choices that have been specifically been added for editing movies, illustrated in Figure 7. Included in this expanded list are potential active sites and individual bound ligands.
To edit a frame, the user must first drag the slider to the frame they want to change. Next, the user can choose an object to alter by selecting it from the ‘Select’ menu. Once an object is selected, the user can then use the ‘Colors’, ‘Show’, and ‘Hide’ dropdowns to alter the appearance of the molecule. When making alterations, the edit
that will be applied appears in the ‘Current Edit’ box. If the user agrees this is the change to be made, they can click the ‘Accept Change’ button. This will change the appearance of the molecule in the viewing window and the commands for the edit to appear in the ‘Completed Edits’ box found on the left side of the Frame Editor window. When the user has finished making edits to a specific frame, they must then push the ‘Frame Complete’ button. This will integrate the changes into the movie and clear the ‘Completed Edits’ box. Up until the time the ‘Frame Complete’ button is pressed, users can remove unwanted changes from the ‘Completed Edits’ box by using the ‘Undo’ button on the ‘Frame Editor’ window. Although the change will still appear in the viewing window, it will not occur when the movie is played back. The user may also overwrite changes to a frame after the ‘Frame Complete’ button has been pressed. For example, let’s say that the user edited frame 55 of their movie by changing all the protein in their movie yellow and hit the ‘Frame Complete’ button. After previewing their movie, the user decides that the protein would look better green. By navigating to frame 55 on the ‘Frame Editor’ window and executing the color change to green and pressing the ‘Frame Complete’ button, the previous edits to the frame will have been overwritten. In this case, if the user made any other edits to frame 55, besides coloring the protein yellow, in the first round of editing, they will have to re-execute those changes to have them appear in their movie, as they will have been overwritten as well.

When editing a movie it is important to keep in mind that PyMOL doesn’t know what your molecule should look like in every frame unless you tell it. Unless there are edits specified for a given frame, the molecule will retain the same appearance as it had in the previous frame. This point is especially important when we have to consider that
any movie created with the EZ-Viz interface will be played back in a looping fashion. This means that when the play button is pressed, the movie will repeat itself over and over again, until the stop button is pressed. As a result, the first frame of the movie will sequentially follow the last frame of the movie. While the molecule’s orientation in the viewing window will be set to the default position in this transition, the appearance of the molecule will be retained from the last frame of the movie to the first. Therefore it is highly recommended that edits be made to the first frame of the movie so that the molecule appears as the user intends it to each time the movie cycles. It should also be noted that the user does not need to make changes to the frames in sequential order. Additionally editing can also be done in movies that were previously saved to file and then loaded back into EZ-Viz Movie.

**Saving Movies**

EZ-Viz Movie also has the capabilities to save movies for the purposes of either making revisions at a later time or playback. During the movie creation process, a log is kept on the main ‘Make Movies’ tab that stores all the commands used to generate the movie. When the ‘Save Movie’ button is clicked, all the commands are written to a user specified file. This movie file can then be retrieved at a later time by pressing the ‘Load Movie’ button on the ‘Make Movies’ tab.

**Loading Movies**

Since movies can be saved for later playback, the option for re-loading those movies is also included on the EZ-Viz ‘Make Movies’ tab. This could prove useful for opening a previously made movie for playback or appending frames onto an already
existing movie. Prior to loading a movie file, the molecule that the movie was created for must be loaded in the viewing window. The molecule should be loaded through the EZ-Viz interface using either the ‘Open’ or ‘Fetch PDB’ button. This step is necessary to ensure correct movie creation and playback. Therefore, if there is no molecule loaded or the incorrect molecule is loaded, an error message will be displayed. After the molecule is loaded the movie file can then be loaded using the ‘Load Movie’ button. Once the movie is loaded, the user can add more frames, make edits to those frames, or if the movie is complete, play it back.

**Additional Features**

Aside from all the tools mentioned above, there are a few additional features of the EZ-Movie interface that should be mentioned.

The original version of EZ-Viz contained a ‘Select’ menu on the ‘Commands’ tab that listed several molecular features that the user could choose from including individual chains, bound hetero atoms, protein, and nucleic acids. This impressive list was the output of a PDB file parser that was built in to the EZ-Viz plugin. While the original list of molecular features has not been altered, potential active sites have also been added to the ‘Zoom In On’ menu found on the ‘Make Movies’ tab and the ‘Select’ menu on the ‘Frame Editor’ window. The addition of the active sites to the list of molecular features is the result of the addition of another file parser, this time for the Catalytic Site Atlas [8]. According to its website, the Catalytic Site Atlas or the CSA “… is a database documenting enzyme active sites and catalytic residues in enzymes of 3D structures.” Using a local copy of the CSA database the file parser that has been incorporated into the EZ-Viz plugin, the user is able to extract the predicted catalytic residues for a molecule
loaded in PyMOL’s viewing window. A list of these residues is provided to the user via the ‘Ligands & Chains’ menu so that they may be incorporated into a movie. Unfortunately, the catalytic residues for many 3D structures remains uncertain, therefore, this feature is not be available for all molecules.

Another feature that we believe to be helpful to users is the ‘Presets’ menu that was added to the ‘Frame Editor’ window. Currently the menu allows the user to select the EZ-Viz created preset view of ‘Default’ as a display option for incorporation in a movie. This preset view was created in the initial version of EZ-Viz as an informative representation where users could easily distinguish secondary structure, protein, DNA/RNA, and bound ligands. Ultimately, additional preset views could be added to the list of potential views for a movie.

Limitations

As with all interfaces, there are a few limitations of the ‘Make Movies’ tab that may restrict the user.

The first of these limitations occurs when the user loads a movie using the ‘Load Movie’ button. The ‘Undo’ button used for removing frames will only remove frames that were added after the movie was loaded. All the frames that were added in a previous session are not able to be undone.

The only other limitation will most likely only effect advanced users who understand PyMOL movie commands. There is currently no way the manually edit the commands generated by the ‘Make Movies’ tab within the interface. If users wish to delete commands from their movie script file they will have to do so in a text editing program.
Results & Discussion

Development

Overview

The addition of the ‘Make Movies’ tab to the EZ-Viz interface was a long process that brought about many unique developmental challenges. Aside from implementing a suite of movie production tools, the interface also needed to have features that were specific for working with biological molecules. The ultimate goal for this project was successfully combining these two components, while maintaining a user-friendly format. To complete this task a preliminary version of the interface was developed and tested by a group of 18 students. Using the students’ feedback as a guide, the ‘Make Movies’ tab was revised to incorporate their suggestions and create a more complete version of EZ-Viz Movie.

User Testing

To gain some feedback, a preliminary version of the EZ-Viz Movie interface, shown in Figure 8, was tested at RIT in the Molecular Modeling and Proteomics (1001-494, 1001-794) class of 18 students. At the time of testing, the interface was moderately different from the final version, as it lacked both ‘Undo’ buttons, the ‘Preview’ button, several features on the ‘Frame Editor’ window, and required the user to input the total number of frames prior to making the movie. Students were given a task, Appendix A, where they were asked to create a few simple movies using the interface. During the course of the assignment students gave comments, both written and orally, describing their experience using the interface.
After reviewing the students’ comments it was apparent that students endured the greatest amount of trouble while trying to edit frames with the ‘Frame Editor’. Many of the students struggled to edit frames without having an ‘Undo’ button to clear unwanted changes and were unsure if the changes they made had in fact been applied to their movie. These two problems led to confusion and ultimately the movies the students created did not turn out as they intended.

Aside from using the ‘Frame Editor’ window, students also labored when trying to use the rest of the interface. Many students were frustrated with having to set the length of their movie at the beginning of the process and by not having the capability to remove frames they added by mistake. These limitations forced many of the students to start over when creating their movie instead being able to rectify the problem.

Figure 8. Preliminary version of the ‘Make Movies’ tab
Overall, a majority of the students felt that EZ-Movie interface made the movie making process within PyMOL much simpler than issuing commands. Several of the students included ideas for future improvements which will be mentioned later.

**Interacting with PyMOL**

Before any buttons or menus could be implemented, we needed to determine how the new tab would execute PyMOL commands. Because the EZ-Viz interface is a PyMOL plugin it can easily execute commands and communicate with PyMOL in a way that a standalone program cannot. We took advantage of this link and developed the original EZ-Viz interface so that when users clicked a button on EZ-Viz, the corresponding PyMOL command was executed immediately. Developing the ‘Make Movies’ tab, we tried to mimic the interaction that already existed between PyMOL and the plugin.

Unlike the rest of the PyMOL capabilities implemented in the EZ-Viz interface, movie making requires the commands to be executed in a specific order with no way of undoing implemented changes. While trying to maintain the interaction that already existed between PyMOL and the plugin, some of the irksome features of making movies using the command line were transferred to the ‘Make Movies’ tab of the EZ-Viz interface. For example, when creating a movie in PyMOL, it is required that the number of frames be set before any other steps are taken. Initially, this became a mandatory requirement in the ‘Make Movies’ tab. Additionally, there was no way to include an ‘Undo’ button in the interface as PyMOL currently does not have a complementary command to support it. During the user test for the preliminary version of the interface,
many of the students found both these problems to be extremely frustrating and needed to be resolved before the final version of the interface was released.

To resolve these issues, the way that the plugin executed PyMOL commands for the ‘Make Movies’ tab was reevaluated. Instead of immediately executing the commands when a button is pressed, they are printed to a log file. When the user has completed adding features to their movie and clicked the ‘Make Movie’ button, the commands in the log file are executed and the movie is made. The addition of the log file effectively resolved the inability to add an ‘Undo’ button and removed the need to set the total number of frames. These resolutions will be described in more detail below.

Revisions

To incorporate the revisions suggested by the students, the interface underwent a major overhaul. Improvements included redesigning the ‘Frame Editor’ window, adding the ability to undo changes, and removing the need to set the number of frames.

Redesigning the ‘Frame Editor’ window involved adding a few features that were not present in the preliminary version (Figure 9). The first of those features was a text box named ‘Current Edit’ that would be used to display the change that would be made to the molecule if the ‘Accept’ button was pushed. Additionally, a larger text area (Completed Edits) was added to display all the edits that have been accepted for a specified frame. The addition of this text area allows users to see a list of all the changes they have already made to a frame. Although the user still has to push the ‘Accept’ button and the ‘Frame Complete’ button to incorporate the edits into their movie, hopefully the addition of the two text boxes and the ‘Undo’ button which is described below will help to alleviate the confusion that was initially associated with this window.
Aside from the aesthetic changes made to the ‘Frame Editor’ window, alterations also needed to be made to the underlying code that controlled the functionality of each button on the interface. In the preliminary version of the interface, all the buttons in the interface executed a PyMOL command when they were clicked. Unfortunately, having the code structured in this way severely limited the flexibility of the interface by restricting users to execute their button pushes in a given order. Essentially, using the interface was no less constraining than having to create a movie using commands. To

Figure 9. Preliminary (top) and final (bottom) versions of the ‘Frame Editor’ window
resolve this problem, the code for the interface was restructured. In the revised version, all the commands needed to make a movie are stored in a log file and are not executed until the ‘Make Movie’ button is pressed. By not executing the PyMOL commands immediately, users are provided with greater control and flexibility.

The restructuring of the code made it possible to add desperately needed functionality to the interface that could not have previously be incorporated. The most critical of the additions were two ‘Undo’ buttons; one on the main ‘Make Movies’ tab for removing mistakenly added frames and the other in the ‘Frame Editor’ window for undoing unwanted edits. At the movie level, pressing the ‘Undo’ button found on the ‘Make Movies’ tab will remove the commands for most recent frame addition from the log file. At the frame level, pressing the ‘Undo’ button on the ‘Frame Editor’ window will remove the commands for the most recent edit from a given frame log. Therefore since the commands are deleted from the log file, they will not be incorporated into the movie. However, once the ‘Frame Complete’ button is pressed, changes to a given frame cannot be undone and instead must be overwritten.

Additionally, streamlining the code made it possible to remove the need for users to specify the length of their movie. Instead, users can dynamically add frames to their movie without specifying a maximum prior to starting. The number of frames is then calculated by the interface when the ‘Make Movies’ button is pressed and the PyMOL command for setting the number of frames is then appended to the beginning of the log file. It is also possible to load a previously made movie and add more frames. In this case a dialog will popup notifying the user that they are about the add more frames and will have to click the ‘Make Movies’ button to finish their movie.
Future Improvements

The current version of the ‘Make Movies’ tab of the EZ-Viz interface is a simple tool that will allow users to easily create molecular animations. However since this is the first version there are several areas that could be improved upon to make the tool even better. We have listed below our own suggestions for features that could be added to the interface as well as some recommendations from the student test group.

Frame Editor

Although it has been significantly improved through several revisions that were made following the usability study, the ‘Frame Editor’ window could still be upgraded. To begin, the ‘Presets’ dropdown currently only has one option, the default view. Adding functionality so that the other eight views from the ‘Presets’ tab could be incorporated into a movie would greatly increase the value of this menu.

Another upgrade could be to expand the editing capabilities for movies beyond changing the color and structural representation. The ‘Advanced’ tab of the EZ-Viz interface provides users with options for transparency, size, width, and thickness; none of which are implemented on the ‘Frame Editor’ window. Including these options would allow users to take their movies to more advanced level.

A separate log file for each frame would also be a helpful feature that could be implemented in the ‘Frame Editor’ window. Currently when the user navigates from one frame to the next, the large text area remains unchanged. It would be beneficial to the user if this text area was updated with the edits that have been made in the specified frame. This way the user has a written account of the changes that are being made in each frame.
‘Make Movies’ Tab

A feature that is not currently implemented in the ‘Make Movies’ tab is the ability to position the molecule in a user defined starting position with the mouse. While making the movie, the interface automatically resets the molecule to its original position for the start of the movie. It would be beneficial to include a way for the user to set the starting position of the molecule and have the movie remember it.

The current capabilities of the EZ-Viz interface allow users to add actions to their movie by using the check buttons on the ‘Make Movies’ tab, however this method completely ignores the ability of the users to position the molecule with their mouse. A potential upgrade would be to allow users to position the molecule using their mouse and then storing that position to a given frame. This process could be repeated to create an entire movie.

Another additional feature that could be added to the interface is a movie format converter. Unfortunately, without some file conversion, PyMOL movies can only be played back in the PyMOL viewing window. Incorporating a conversion tool into the EZ-Viz interface would enable users to generate PyMOL movies that could easily be placed into presentations or web pages. Possible movie formats include: AVI, Flash, and animated GIFs. Although this feature has not yet been added to the EZ-Viz movie interface, methods for converting PyMOL movies to a more usable format are currently under investigation.

To make the interface even easier to use, a student test participant suggested adding tips, as are often seen in help files, to the user when they hover over a certain
feature of the interface. These tips could provide more detailed information about each feature to further explain their purpose.

Finally, a primary citation text mining tool could also be implemented to gain even more information about a given structure. Every PDB file contains the name of the journal article where the structure was first introduced. Many times these articles provide readers with molecular details including residues found in the active site, functional domains, and features unique to the specific structure. Using text mining techniques this information could be extracted from the article and presented to the user in an efficient and informative manner. This information would provide the user with suggestions for the molecular features that would be interesting to portray in a movie.
Materials & Methods

The EZ-Viz interface was developed using Python’s Tkinter [9] and Mega Widget toolkits. Both toolkits provide the programmer with existing Python modules that can be used for adding buttons, checkboxes, drop-down menus, text areas, images and other GUI components. Since we are expanding to the current the interface it is essential to maintain code consistency throughout the entire program and therefore we will continue to use both the Tkinter and Mega Widget toolkit to import the necessary components.

In creating this interface, it is essential that several basic features be implemented to provide the user with a modest amount of control. Therefore it was our objective to implement the more essential components of the interface prior to adding more advanced features. Included in the list of basic features is adding functionality so that the user can effectively produce simple movies that are written to a script file and then can be played in the PyMOL viewing window. Users should be able add actions to their movie such as rotations and zooming as well as being able to alter the appearance of their molecule. Advanced features that were added later included extracting active site information from the CSA [8], undo buttons, and an output window for users to visually view their edits.
Conclusions

The ‘Make Movies’ tab of the EZ-Viz interface is an easy to use tool for creating molecular movies in PyMOL. The final version of the interface included all the tools necessary for creating a new movie, buttons for adding and editing frames, a save function, a preview button to view the current state of the movie, and a feature to load previously created movies for playback or editing.

The development of the original EZ-Viz interface was a significant achievement that simplified using PyMOL and encouraged its use. With the addition of the ‘Make Movies’ tab, the EZ-Viz interface has developed into a powerful counterpart for working with PyMOL while remaining easy to use. The continued development of the EZ-Viz interface will promote the use of PyMOL by both scientists and educators and encourage its application for making high-quality molecular images and movies.
Deliverables

- EZ-Viz interface with new movie tab
- Upgraded EZ-Viz help file with instructions for new tab
- Source code for interface
- Thesis Write-up
Time Line

Winter Quarter

• Design look of interface
• Get features for basic movie working
• Evaluation of ligand interaction programs

Spring Quarter

• Focus on text mining capabilities
• Testing
• Write User Manual
• Thesis Write-up
References


Appendix A

Introduction to EZ-Viz Movie – User Testing

PyMOL Background
PyMOL is a molecular modeling program that was developed by DeLano Scientific to generate high-quality molecular images for presentations and publications. It also has the capabilities to produce molecular animations that can be shown in the classroom or at professional conferences. The steps that must be taken to create an informative movie are numerous, complex and can be cumbersome for new users. Today we will be testing a new user interface with the main purpose of making this task much simpler and user friendly. Please feel free to share your comments about the interface as they will be taken into consideration to make future improvements.

Note: The EZ-Viz Movie interface is still being developed, which means you may encounter error messages or find that part or all of PyMOL crashes at certain points. Please keep a record of any error messages you encounter as you go through these exercises and include these in your final report.

Introduction to PyMOL Movies
Let’s start by having a brief introduction to PyMOL. When you open PyMOL you will find that two windows appear. Within these two windows there are three components: the internal GUI, the external GUI, and the viewing window. The internal GUI serves as a list of objects within PyMOL while the viewing window is where your molecule will be displayed. The external GUI serves as a command line to interact with PyMOL as well as provide you with output from the program.

Next, let’s view a movie that was created using PyMOL. This movie was created by issuing over 160 commands. This movie has close to 2,000 frames and features 4 zoom ins, 4 zoom outs, several rotations, and some advanced frame manipulation. This is a more complex example, but I feel it demonstrates the potential that PyMOL has to create movies. However, let’s try to make a basic movie that consists of 100 frames. Issue the following commands sequentially to create your movie.

<table>
<thead>
<tr>
<th>Description of Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Load the molecule 1D66</td>
<td>Plugins -&gt; PDB Loader -&gt; type ‘1d66’</td>
</tr>
<tr>
<td>Set the number of frames</td>
<td>mset 1, 100</td>
</tr>
<tr>
<td>Store the first frame</td>
<td>mview store, 1</td>
</tr>
<tr>
<td>Zoom in on chain B</td>
<td>orient chain b</td>
</tr>
<tr>
<td>Store this view at frame 25</td>
<td>mview store, 25</td>
</tr>
<tr>
<td>Hold on this view – store view at frame 50</td>
<td>mview store, 50</td>
</tr>
<tr>
<td>Zoom out from chain B</td>
<td>orient</td>
</tr>
<tr>
<td>Store the view at frame 75</td>
<td>mview store, 75</td>
</tr>
<tr>
<td>Rotate the molecule 180° on X-axis</td>
<td>turn x, 180</td>
</tr>
<tr>
<td>Store this view at frame 100</td>
<td>mview store, 100</td>
</tr>
<tr>
<td>Make the movie</td>
<td>mview interpolate</td>
</tr>
</tbody>
</table>
Now we can play our movie by typing ‘mplay’. In short, this movie zooms in on chain B, holds on that view, zooms back out and then rotates 180° on the X-axis. By watching the movie what does it tell us? When we zoom in on chain B can we really distinguish it from the rest of the molecule? We need to issue more commands to edit the frames so the movie is informative. Let’s make it so that chain B is different from the rest of the molecule and flashes different colors while it is zoomed in.

<table>
<thead>
<tr>
<th>Description of Action</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>In frame 26 make chain B appear as cartoon, color chain B green</td>
<td>mdo 26: hide lines, chain B; show cartoon, chain B; color green, chain B;</td>
</tr>
<tr>
<td>In frame 30 color chain B red</td>
<td>mdo 30: color red, chain B</td>
</tr>
<tr>
<td>In frame 35 color chain B green</td>
<td>mdo 35: color green, chain B</td>
</tr>
<tr>
<td>In frame 40 color chain B red</td>
<td>mdo 40: color red, chain B</td>
</tr>
<tr>
<td>In frame 45 color chain B green</td>
<td>mdo 45: color green, chain B</td>
</tr>
<tr>
<td>In frame 50 make chain B appear as lines, color chain B red</td>
<td>mdo 50: color red, chain B; hide cartoon, chain B; show lines, chain B;</td>
</tr>
</tbody>
</table>

Now play the movie again using ‘mplay’, we can now see exactly where chain B is in this molecule. So using 16 commands we have made a simple, informative movie. Now what if I asked you to go find the active sites in this molecule, zoom in on each of the four bound ligands, make them stand out, show the potential interacting residues, and then zoom back out. How many of you think you could do that?

Let’s try a different way…EZ-Viz Movie!

**Movie Exercise #1**
From the PyMOL plugins menu select EZ-Viz Movie
Using the ‘Fetch PDB’ button on the bottom of the EZ-Viz Movie window, load the molecule ‘1D66’.

Use this time to familiarize yourself with the EZ-Viz Make Movies interface and play around with the buttons and features. Do your best to try and re-create the same movie that we created above using the commands with the buttons and menus on the interface. Please write down any aspect of the interface that seems confusing or any comments you have along the way.

*Note – There is a ‘Play’, ‘Stop’, and ‘Rewind’ button on the ‘Movies’ tab. Please feel free to use these instead of typing the play and stop commands.*

**Movie Exercise #2**
Let’s walk through the features of the interface and make the movie together.

**Movie Exercise #3**
HIV-1 protease plays an essential role in the proliferation of HIV viruses in an infected host. Its role cleaving the gag-pol polypeptide into functional proteins leads directly to the creation of infectious HIV-1 viruses. As with many proteins, the function of HIV-1 protease is directly related to its structure. By better understanding the structure of this protease we can gain a better idea of how to inhibit its actions. Make a movie of the HIV-1 protease structure ‘1MSM’. Please include the following features:

- Distinguish the two chains from one another
- Zoom in on the active site
- One or more rotations
- Change the color of the bound ligand
- Change the way the molecule looks

When your movie is complete, please save your movie using the following format for the file name - “YOUR LAST NAME_1MSM”. (You don’t need to worry about the file extension as it will be added for you).

Movie Exercise #4
Now that you have prepared a movie it’s time to play it. To test the full capabilities of the interface we will load the movie using the ‘Load Movie’ button.

Fully close PyMOL and the EZ-Viz Movie interface. After they have closed, re-open PyMOL and the EZ-Viz movie interface. Load the ‘1MSM’ molecule and the movie that you created for it. Play the movie to make sure that it is in fact the same movie you created. If there are any errors, or the movie does not playback as expected, please comment on it below.

Movie Exercise #5
This final exercise is for you to play around with the interface and experiment with the features it provides. Perhaps use this time to create movies or images for your protein structure project. If you have any questions please feel free to ask.

Submission
You need to submit two files to the Movie Maker dropbox on the MyCourses website. The first file will be the movie file you created for 1MSM using EZ-Viz Movie. The second file will be a Word document that includes your answers to the questions in this handout and also any error messages or bugs you encountered, with an explanation of what you were doing when the error/bug occurred.
EZ-Viz Movie Questionnaire
Name:

Exercise 1 Questions
1) Have you had any previous experience with PyMOL? If so, please briefly explain.

2) Did you find the interface easy to follow? How successful were you in creating a movie without any instruction on how to do so?

3) Were the button and menu names easy to interpret? What buttons and/or menus were not self-explanatory? What could they be named instead?

Additional Questions
1) Are there any features that were not included in the interface that you would like to see added? Please elaborate.

2) Do you believe that this interface makes the movie making process in PyMOL simpler than issuing the commands?
Report any errors that you encountered along with a description of what you were doing when it occurred:

Please make any additional comments you have regarding the EZ-Viz Movie interface (comments, suggestions, and critiques):
Appendix B

Source Code – on supplementary CD