Anannotated bibliography: Molecular graphics

Graduate Student[1], Senior Collaborator[2], Staff Scientist[3], and Blaine Mooers[*1,2,3]

[1]Department of Biochemistry and Molecular Biology, University of Oklahoma Health Sciences Center, Oklahoma City, Oklahoma, United States 73104
[2]Stephenson Cancer Center, University of Oklahoma Health Sciences Center, Oklahoma City, Oklahoma, United States 73104
[3]Laboratory of Biomolecular Structure and Function, University of Oklahoma Health Sciences Center, Oklahoma City, Oklahoma, United States 73104

2024-10-24

Contents

Illustrated and annotated bibliography

Mooers2020ShortcutsForFasterImageCreationInPyMOL

Summary

PyMOL is a tool for generating images of biomolecular structures, offering extensive control over their appearance through numerous parameters. To enhance and simplify its use, 241 Python functions, termed “shortcuts,” were developed. These shortcuts, organized into 25 functional groups, allow users to perform tasks such as creating new molecular representations, saving files with unique time-stamped names, and conducting web searches directly from PyMOL.

Context in Molecular Graphics and Human-Computer Interactions

To streamline and enhance user interactions with PyMOL, 241 Python functions, known as “shortcuts,” were developed. These shortcuts, categorized into 25 functional groups, facilitate tasks such as creating innovative

*blaine-mooers at ouhsc.edu, phone: 405-271-8XXX, FAX: 405-271-3X3X

molecular representations, saving files with unique time-stamped names to prevent overwriting, and conducting web searches directly from PyMOL. The help function provides documentation and reusable PyMOL commands, significantly improving user efficiency by reducing the time spent searching for code fragments. This integration of shortcuts exemplifies the synergy between molecular graphics and user-friendly interfaces, optimizing the workflow for researchers and scientists. This is an excellent of good Human-computer interactions (HCI) .

Mooers2021TemplatesForWritingPyMOLScripts

Summary

PyMOL commands offer precise control over the visualization of molecular models, making PyMOL a favored tool for creating images of protein structures for publications and presentations. However, many users struggle to remember these commands due to infrequent use, complicating writing new scripts. One practical approach to address this issue is using code fragments as templates for different parts of the task. These fragments can be accessed from a library while coding in text editors like Visual Studio Code, Vim, and Emacs.

To facilitate this, we developed a library of PyMOL code templates, known as pymolsnips, which simplifies the process of writing PyMOL scripts. Pymolsnips is available on GitHub in formats compatible with 18 popular text editors, supporting Mac, Windows, and Linux operating systems. The GitHub repository also includes animations to guide users through the installation process for each text editor. This library will significantly enhance the productivity of PyMOL users when scripting.

Context in Molecular Graphics and Human-Computer Interaction

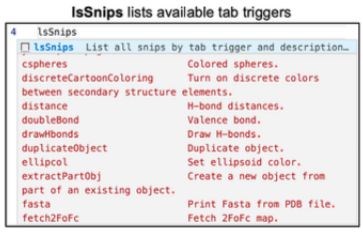

FIGURE 1 The lsSnip tabtrigger and preview of the table of snippets in VSC text editor. The user can find the tabtrigger of interest without finishing the execution of the **lsSnip** tab trigger and thus without inserting the table of snippets into the script file

Figure 1: Text wrapped image

Tools like PyMOL and libraries like pymolsnips play a crucial role in the broader context of molecular graphics and human-computer interaction. Molecular graphics software allows scientists to visualize complex molecular structures, aiding in the understanding and communicating biochemical processes. The ability to precisely control these visualizations is essential for producing high-quality images for research and educational purposes.

Human-computer interaction (HCI) principles are integral to the design of these tools, ensuring they are user-friendly and accessible. Visual Studio Code provide excellent usr interface to snippets (See Fig 1). By providing code snippets and templates, pymolsnips enhances the usability of PyMOL, making it easier for users to create and modify scripts without needing to recall specific commands. This approach aligns with HCI goals of improving efficiency and reducing cognitive load, ultimately fostering a more productive and intuitive user experience.

Snippet interfaces in Text Editors

Pymolsnips can be used at all skill levels (Table **??**). PyMOL users can be divided into five levels of skill **Dreyfus1980AFiveStageModelOfTheMentalActivitiesInvolvedInDirectedSkillAcquisition**. Beginner users may represent 40% of PyMOL users (Table 1). Their use of PyMOL is generally limited to viewing and comparing structures. Beginner users are not yet willing to invest in learning the commands of the PyMOL macro language. They prefer the intuitive nature of the PyMOL GUI and its pulldown menus.

Advanced beginners (30% of users) are able to navigate the GUI quickly. They also use simple commands like the “fetch” command to retrieve coordinate files from the Protein Data Bank. They may make images of global views and structure superpositions. They also make simple close-up views of protein-ligand interactions and subunit-subunit interfaces. They rely on session files to save work in progress. Competent users (20% of users) use scripts to assemble images for publication. Proficient users (9% of users) know most of the frequently used

Table 1: Tools by target skill level.

Skill level	Shortcuts	Snippets	Polyglot docs	Quizzes
Beginner	✓			✓
Adv. Beginner	✓	✓		✓
Competent	✓	✓	✓	✓
Proficient.	✓	✓	✓	✓
Expert	✓	✓	✓	✓

parameters and are willing to invest time in learning new commands. They have been using text editors for years but may not have discovered the power of snippet libraries.

The expert users (1%) understand the PyMOL macro language syntax and remember many of the commands. They extend the capability of PyMOL by importing functions from modules outside of PyMOL. They may even be involved in the development of new plugins. They are likely expert users of one or more text editors. They would welcome a snippet library and the shortcuts. The tools developed in this proposal will benefit users at all levels of expertise and help them to move to higher levels of expertise (Table 1).

Acronyms

HCI Human-computer interactions. 2

Glossary

annotated bibliography An annotated bibliography gives a summary of each bibliographic entry. The annotation also provides the entry's central idea(s) and a general idea of the source's content. It also provides an evaluation of the source's credibility and its relevance to the field. It might contain an assessment of the relevance to own's own work when written for personal use. The following list of elements can be included.. 1

Mathematical Notation

$\mathcal{A}$ Asymmetric unit of the crystal. 6

$A(\mathbf{q})$ Scattering amplitude. 6

$\alpha(\mathbf{h})$ Phase of the structure factor. 6

α, β, γ Angles between unit cell edges. 6

B Temperature factor or B-factor. 6

$n\lambda = 2d\sin\theta$ Bragg's law. 6

$e^{-2B\sin^2\theta/\lambda^2}$ Debye-Waller factor. 6

$\Delta\phi$ Phase shift. 6

$d_{\textbf{lim}}$ Resolution limit. 6

$d_{\textbf{min}}$ Resolution of the crystal structure. 6

$d_{\textbf{shell}}$ Resolution shell. 6

$\mathcal{F}$ Fourier transform. 6

$|F(\mathbf{h})|$ Amplitude of the structure factor. 6

$|F(\mathbf{h})|$ Magnitude of the structure factor. 6

$F(\mathbf{h}) = F'(\mathbf{h}) + iF''(\mathbf{h})$ Real and imaginary parts of the structure factor. 6

$F(\mathbf{h})$ Structure factor for reflection $\mathbf{h}$. 6

$f(\theta)$ Atomic scattering factor as a function of angle θ. 6

f_0 Atomic form factor. 6

$\mathbf{h}$ Reciprocal lattice vector. 6

(hkl) Miller indices for crystal planes. 6

$\mathbf{hkl}$ Reciprocal lattice point. 6

$I(\mathbf{h}) \propto |F(\mathbf{h})|^2$ Diffraction intensity proportional to the square of the structure factor amplitude. 6

$I(\mathbf{h})$ Intensity of reflection $\mathbf{h}$. 6

$\mathbf{k} - \mathbf{k}_0 = \mathbf{G}$ Laue equation for diffraction. 6

P Crystallographic symmetry group. 6

ϕ Phase angle of the structure factor. 6

$\phi(\mathbf{h})$ Phase problem in crystallography. 6

$\phi(\mathbf{h}) = \arg(F(\mathbf{h}))$ Phase retrieval from structure factor. 6

$\mathbf{q}$ Scattering vector. 6

$\rho(\mathbf{r})$ Electron density distribution. 6

$\rho(\mathbf{r}) = \frac{1}{V}\sum_{\mathbf{h}} F(\mathbf{h})e^{2\pi i \mathbf{h}\cdot\mathbf{r}}$ Electron density as a Fourier series. 6

$\rho(\mathbf{r})$ Electron density map at position $\mathbf{r}$. 6

$\rho(\mathbf{r})$ Electron density at position $\mathbf{r}$. 6

$\mathcal{S}$ Crystallographic symmetry operations. 6

$\mathcal{S}$ A set. 6

θ Diffraction angle. 6

U Atomic displacement parameter. 6

$\langle u^2 \rangle$ Mean square displacement of atoms. 6

a, b, c Unit cell dimensions. 6

V Volume of the unit cell. 6

Index