

# Molywood: streamlining the design and rendering of molecular movies

Miłosz Wieczór<sup>1,2,\*</sup>, Adam Hospital<sup>2</sup>, Genis Bayarri<sup>2</sup>, Jacek Czub<sup>1</sup> and Modesto Orozco<sup>2,\*</sup>

<sup>1</sup>Department of Physical Chemistry, Gdańsk University of Technology, Narutowicza 11/12, 80-233 Gdańsk, Poland

<sup>2</sup>Institute for Research in Biomedicine (IRB Barcelona), Barcelona Institute of Science and Technology, Baldiri Reixac 10, Barcelona 08028, Spain

\*To whom correspondence should be addressed.

## Abstract

**Motivation:** High-quality dynamic visuals are needed at all levels of science communication, from the conference hall to the classroom. As scientific journals embrace new article formats, many key concepts – particularly in structural biology – are also more easily conveyed as videos than still frames. Notwithstanding, the design and rendering of a complex molecular movie remain an arduous task. Here, we introduce Molywood, a robust and intuitive tool that builds on the capabilities of VMD to automate all stages of movie rendering.

**Results:** Molywood is a Python-based script that uses an integrated workflow to give maximal flexibility in movie design. It implements the basic concepts of actions, layers, grids and concurrency and requires no programming experience to run.

**Availability:** The script is freely available on GitLab ([gitlab.com/KomBioMol/molywood](https://gitlab.com/KomBioMol/molywood)) and features an extended documentation, tutorial and gallery hosted on [mmb.irbbarcelona.org/molywood](https://mmb.irbbarcelona.org/molywood).

**Contact:** [miłosz.wieczor@pg.edu.pl](mailto:miłosz.wieczor@pg.edu.pl), [modesto.orozco@irbbarcelona.org](mailto:modesto.orozco@irbbarcelona.org)

## 1 Introduction

Since the groundbreaking resolution of the 3-dimensional structure of myoglobin in 1960 (Kendrew *et al.*, 1960), our understanding of processes of life has increasingly relied on the visualization of molecules. The two revolutions in high-resolution structural biology – first in X-ray crystallography, second in cryo-EM imaging (Stuart *et al.*, 2016) – resulted in tens of thousands of unique protein structures being deposited in public databases, ultimately providing us with an atomistic-level insight into the peculiar world of molecular shapes and conformational changes. However, the scientific communication of these findings primarily relies on still images and figures, often obscuring or distorting the structural insights.

This situation could be rectified through widespread adoption of molecular movies as a complementary visual aid in the dissemination of scientific findings (McGill, 2008). Not only are publishers starting to actively encourage the use of multimedia in the new enhanced, web-only article formats, but also science communication has become increasingly visual due to the proliferation of science-oriented blogs, vlogs, and social media channels (Welbourne and Grant, 2016). On the other hand, researchers often find existing tools inadequately suited to these challenges. In fact, a very limited choice of such tools exists today: VMD’s Movie Maker remains limited to simplest cases, BioBlender and ePMV require programming and/or Blender skills (Johnson *et al.*, 2011; Andrei *et al.*, 2012), while PyMOL’s eMovie effectively vanished from the Web (Hodis *et al.*, 2007; Hsin *et al.*, 2008). Simultaneously, platforms such as Molecular Maya are commercial and thus not accessible to many researchers. While molecular movies of excellent quality do exist in

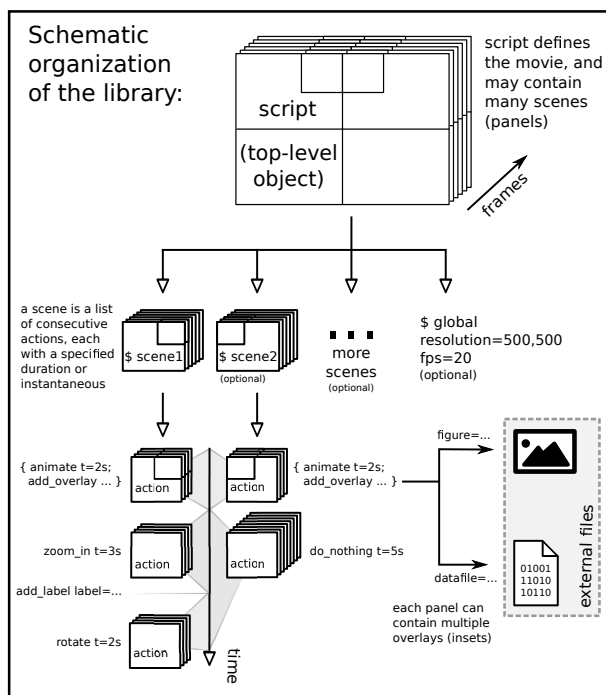
the literature, they are usually a result of a lengthy work on integrating workflows, automatizing, and post-processing.

These shortcomings prompted us to create Molywood, a Python tool capable of managing complex moviemaking workflows across a range of utilities, including the open-source utilities FFmpeg and Imagemagick, Python’s Matplotlib, and VMD with its extended Tcl scripting capabilities (Humphrey *et al.*, 1996). With an easy-to-follow syntax and flexible composition rules, the tool greatly simplifies movie design, allowing the user to focus on conceptual rather than technical issues. Finally, we provide a library of minimal working examples and featured full-fledged movies – hosted at [mmb.irbbarcelona.org/molywood](https://mmb.irbbarcelona.org/molywood) – to guide users through the supported options.

## 2 Application

Molywood is organized around three hierarchical concepts: Scripts, Scenes, and Actions. A Script is a collection of Scenes, and defines the entire movie; similarly, a Scene is a collection of Actions and can be thought of as a single panel. Most movies will consist of a single Scene, so here we focus on Actions first.

An Action is the main primitive that abstracts a single event – a rotation, zoom, highlight, etc. – either with a specific duration or effected instantaneously. In Molywood, many Actions can overlap in time, e.g. one can rotate the view while simultaneously zooming in and animating the trajectory. Moreover, Actions can be asynchronous (i.e., overlap only partially with one another): following the above example, it is possible to start the rotation when the zoom-in and animation are already underway. All Actions are controlled by key-value pairs that specify all tunable parameters, such as action duration, transition smoothness or color of the highlight. For more complex behavior, they can interact with other



**Fig. 1.** The conceptual organization of Molywood. A Script is divided into possibly multiple Scenes, which in turn consist of instantaneous, consecutive or simultaneous Actions. Individual Actions can access external resources, such as numeric data, figures or other movies. A number of simple global parameters controls video quality.

Actions: for instance, a structural highlight can be labeled upon creation, so that the same highlight can be later altered or removed. Arbitrarily long extracts from audio files (e.g. voice narratives) can also be easily aligned with specific Actions.

A large portion of Molywood is structured around Overlays – graphical elements that can be overlaid on the basal scene. Overlays support dynamically changing text, 1D and 2D graphs with custom formatting, external figures and movies, as well as other Scene objects (i.e., one molecular movie can be used as an inset in another one). One can also dynamically change the position, relative size and opacity of the Overlays, with no limit to their number or the stacking of layers. Overlays are hence indispensable in augmenting the bare visuals with additional data, such as labels, analysis results, insets or zooms, logos, or ending credits.

In turn, the concept of Scenes was primarily introduced to enable multi-panel movies, e.g., to contrast related systems side-by-side. Their use, however, is very flexible: they might as well be treated as overlays, or joined consecutively to yield movie-like cuts and transitions. Moreover, different Scenes can visualize different molecular systems. This flexibility in design fosters users' creativity in making appealing videos targeted at the general public.

Finally, several global options exist that make the tool even more easily customized. Each Scene can be loaded from a saved VMD visualization state, preserving user-defined viewpoints, selections and settings. Alternatively, a quick start is also possible from an existing PDB structure and/or trajectory, with an option to download PDB structures on-the-fly from the online database. Movie quality can be easily set by specifying frame-per-second (fps) values and Scene resolution. In case of resource-intensive renders, frames can be preserved to allow for restarting when only minor changes are introduced. Last but not least,

four hierarchical rendering modes exist to speed up the design process: (i) no rendering, with frames only displayed in VMD; (ii) quick-and-dirty rendering via VMD's Snapshot; (iii) proper rendering with Tachyon, possibly on an external machine; (iv) professional rendering with shading and Tachyon's Ambient Occlusion. By changing render modes, one can quickly get through the design process and offload the most resource-heavy rendering step to an external workstation, where all dependencies can be conveniently batch-installed in a separate conda environment.

### 3 Discussion

Due to the evolving character of Molywood, finer details of the implementation, an extended documentation of all keywords, a set of mini-tutorials and the examples gallery are hosted on <https://mmb.irbbarcelona.org/molywood> and regularly updated. The gallery, in particular, is intended as an ever-growing collection of highlights to serve other users as both guidelines and inspiration; users may submit their movies along with self-contained input datasets if they want to share their work. Meanwhile, the tutorials provide a detailed overview of all available keywords, parameters and defaults, serving as an extended and more intuitive documentation. We also encourage user feedback to help Molywood better address the needs of the community, hoping the tool will eventually define new standards in movie generation, as well as inspire further developments of open-source visualization software. With this common goal, we shall see the future of the field as yet more visually appealing.

### Acknowledgements

#### Funding

This work has been supported by the Polish National Agency for Academic Exchange, the Instituto Nacional de Bioinformática, the European Union's Horizon 2020 research and innovation program [BioExcel-2 project], the Biomolecular and Bioinformatics Resources Platform (ISCIII PT 13/0001/0030) as well as the MINECO Severo Ochoa Award of Excellence (Government of Spain) (awarded to IRB Barcelona). MO is an ICREA Research Professor.

### References

- Andrei, R. M. *et al.* (2012). Intuitive representation of surface properties of biomolecules using BioBlender. *BMC Bioinformatics*, **13**(SUPPL.4), S16.
- Hodis, E. *et al.* (2007). eMovie: a storyboard-based tool for making molecular movies.
- Hsin, J. *et al.* (2008). Using VMD: An introductory tutorial.
- Humphrey, W. *et al.* (1996). VMD: Visual molecular dynamics. *Journal of Molecular Graphics*, **14**(1), 33–38.
- Johnson, G. T. *et al.* (2011). ePMV embeds molecular modeling into professional animation software environments. *Structure*, **19**(3), 293–303.
- Kendrew, J. C. *et al.* (1960). Structure of myoglobin: A three-dimensional fourier synthesis at 2 Å resolution.
- McGill, G. (2008). Molecular Movies... Coming to a Lecture near You.
- Stuart, D. I. *et al.* (2016). The democratization of cryo-EM.
- Welbourne, D. J. and Grant, W. J. (2016). Science communication on YouTube: Factors that affect channel and video popularity. *Public understanding of science (Bristol, England)*, **25**(6), 706–18.