www.editada.org

International Journal of Combinatorial Optimization Problems and Informatics, 17(3), 2026, 211-218. ISSN: 2007-1558.
https://doi.org/10.61467/2007.1558.2026.v17i3.1274

Interactive 3D Visualization Engine for Molecular Docking and Structural Complex Analysis

Post-grade Student: Ernesto Alan Borjas Torres[1],

Advisors: Carlos A. García Pérez[2]*, Adolfo Josué Rodríguez Rodríguez[1]*,

Alfredo Juárez Saldivar[3], Wenceslao Eduardo Rodríguez Rodríguez[1]

[1] *Reynosa Rodhe Multidisciplinary Academic Unit, Computer Systems Engineer, Autonomous University of Tamaulipas (UAT), Reynosa-San Fernando Highway 88779, Reynosa, Tamaulipas, Mexico*

[2] *Digitalization & Transformation, Helmholtz Zentrum München, Neuherberg, Germany*

[3] *Reynosa Aztlan Multidisciplinary Academic Unit, Autonomous University of Tamaulipas (UAT), Laguna de Chapala 88740, Reynosa, Tamaulipas, Mexico*

Email address(es): carlos.garcia@helmholtz-munich.de,

arodriguez@docentes.uat.edu.mx , alfredo.juarez@uat.edu.mx , wrodriguez@docentes.uat.edu.mx

✉Corresponding author: Carlos A. García Pérez, Adolfo Josué Rodríguez Rodríguez

Abstract. This paper presents the initial development of a new docking analysis tool leveraging WebGL for real-time molecular visualization directly in the browser eliminating the installation burden and offering a fresh level of interactivity and modularity in docking result interpretation. The methodology consists of developing a web API in the Python programming language, implementing the interaction between the React Framework and the underlaying Python script to process data, cluster the docked conformation, calculate the secondary structure of the protein PDB file and retrieve the results to the React Framework to visual representation. The tool renders visual components of the 3D molecular structure viewer in react-three/fiber, react-three/drei, and WebGL, providing a user-friendly and modern interface; and leveraging the computational pipeline capabilities to reduce CPU load to the WebGL engine.

Keywords: Molecular Docking; Webserver molecular visualizers; 3D molecular structure viewer.

Article information
Received: Dec 11, 2025
Accepted: Feb 13, 2026

1 Introduction

Molecular docking is an essential computational strategy in drug discovery, allowing scientists to predict the interactions between small molecules (ligands) and their biological targets (proteins). Tools like AutoDock and AutoDock Vina (Morris et al., 2009; Trott & Olson, 2010) are widely used for this purpose due to their flexibility and robust scoring functions. However, a persistent bottleneck remains: the interpretation and visualization of the vast number of dockings poses these tools generate; especially when analyzing multiple compounds across multiple binding sites or targets. To address these challenges, two tools have been developed in recent years: JADOPPT (Java-based AutoDock Preparing and Processing Tool) and PyDRA (García-Pérez et al., 2017; Varela-Salinas et al., 2017) (Python Docking Results Analyzer). Both were designed to enhance visual analytics, enable clustering-based pose refinement, and streamline multi-ligand comparison workflows. JADOPPT focuses on interactive pose analysis and AutoDock grid modification, while PyDRA extends PyMOL (Schrödinger, LLC, 2015) with treemap and dendrogram-based clustering visualizations compatible with both AutoDock and Vina outputs. Together, these tools introduced innovative solutions that have significantly improved the docking analysis pipeline, offering enhanced usability and deeper structural insights to computational chemists and structural biologists. However, despite their utility, both tools come with technical constraints. Since JADOPPT is built in Java and PyDRA in Python, users must install and configure both environments properly, which can pose a barrier to non-technical users or those working in restrictive computing environments. Additionally, JADOPPT's 3D engine, based on Jmol, is now deprecated, raising concerns about its

future compatibility and support. Conversely, while PyMOL offers broad functionality, its 3D renderig engine is difficult to customize, making it unsuitable for more dynamic or interactive workflows like those we previously achieved with JADOPPT. Recognizing these limitations, we began exploring a modern, web-native alternative to overcome the challenges of platform dependency and visualization rigidity. In this paper, we present the initial development of a new docking analysis tool, whose name is yet to be finalized, though it will likely incorporate a reference to JavaScript (JS) to reflect its technical foundation. This new tool is powered by a custom-built 3D engine written entirely in JavaScript, leveraging WebGL (Schrödinger, LLC, 2015) for real-time molecular visualization directly in the browser. It aims to provide platform-independent, easily deployable, and extensible molecular docking analysis, eliminating the installation burden and offering a fresh level of interactivity and modularity in docking result interpretation.

1.1 Standalone molecular visualizers

The following describes software widely used in the field of molecular biology and molecular modeling. These tools do not require additional programs or external servers to operate and can be installed and run directly on a device.

1.1.1. PyMOL

PyMOL is a molecular visualization software used in biological and medical research to represent and animate three-dimensional structures of proteins, compounds, and other biological molecules. It allows researchers to explore, analyze, and generate high-quality images for publications in the field of molecular and structural biology (Ahmad, 2023; PyMOL, n.d.). In addition, PyMOL includes plugins that allow molecular docking studies, such as those performed with AutoDock and AutoDock-Vina, making it a powerful tool for studying drug targets and medicinal chemistry (Seeliger & de Groot, 2010). PyMOL allows us to visualize molecular structures in three dimensions with representations such as individual atoms, beta sheets, helices, bonds, and more. This software is written in the Python programming language. In conclusion, PyMOL is an essential tool for visualization and analysis of molecular structures such as electrostatic potential, molecular dynamics trajectories, calculations and measurements, and more (Seeliger & de Groot, 2010; Yuan et al., 2017).

1.1.2. AutoDockTools

AutoDockTools (ADT), part of the MGLTools suite, is a software package written in the Python programming language (for the graphical interface) that is part of the AutoDock suite and used to prepare, analyze, and visualize input and output files for AutoDock, a program that facilitates molecular docking and screening of small molecules to macromolecular receptors (Morris et al., 2009). The main functions of AutoDockTools are: molecule preparation, simulation setup, 3D molecule visualization, docking execution, and result analysis (Morris et al., 2009). AutoDockTools is a useful tool in drug development by exploring protein-ligand interactions, contributing to drug target prediction (Dey et al., 2025). In conclusion, AutoDockTools is essential for data preparation and visualization for AutoDock, a simulation program for predicting the binding of small molecules (e.g., drugs) to macromolecular receptors (e.g., proteins) (Dey et al., 2025).

1.1.3. VMD

Visual Molecular Dynamics (VMD) is a 3D molecular visualization software used to view and analyze molecular docking results (Humphrey et al., 1996) including those obtained with AutoDock and AutoDock Vina, considering the representation of the conformations of the docked ligands and their interactions with the receptor protein. VMD is free and open source. Users can run their own TCI and Python scripts, as VMD also includes interpreters for these programming languages, which facilitates the automation of tasks and the implementation of custom algorithms (NIH Resource for Macromolecular Modeling and Bioinformatics & Theoretical and Computational Biophysics Group, 2012). In conclusion, VMD is a powerful and versatile tool for visualizing, analyzing, and understanding the results of molecular docking simulations, with applications in research, education, and scientific outreach (NIH Resource for Macromolecular Modeling and Bioinformatics & Theoretical and Computational Biophysics Group, 2012).

1.1.4. ChimeraX

ChimeraX is a free software developed by the University of California, San Francisco (UCSF) and is the successor to UCSF Chimera, used for the visualization of macromolecules such as proteins, as well as for the analysis and manipulation of molecular structures. ChimeraX stands out for its ability to efficiently handle large volumes of data, and its high-quality visualization and interaction of structures and molecular data, including the visualization of molecular surfaces (Goddard et al., 2018).

1.2 Webserver molecular visualizers

The following describes four tools used for the exploration and understanding of molecular structures and their dynamics, particularly in the fields of structural biology and biochemistry, in a 3D environment within a web browser.

1.2.1. NGL Viewer

NGL Viewer is a web-based collection of molecular graphics tools that allows the visualization of molecular structures such as proteins and nucleic acids (DNA/RNA) in 3D (Rose et al., 2018). As a web-based tool, it runs in a browser without the need for special plugins. It is flexible, offering different ways to represent molecules and allowing interaction with them (Rose & Hildebrand, 2015). NGL Viewer supports formats like “Macromolecular Transmission Format” (MMTF) for efficient data handling. It is a useful tool that allows the scientific community to explore structural data through a graphical interface and can be integrated into other applications (Nguyen et al., 2018).

1.2.2. IcmJS

ActivelcmJS, also known as IcmJS, is a JavaScript/HTML5 library that displays molecular models in an interactive 3D environment within web browsers without the need for plugins or extensions, allowing users to explore the structure of molecules visually (Molsoft, n.d.). Being a JavaScript and HTML5 library, it allows users to operate directly in the browser, making it easy to use and deploy. This tool is developed by Molsoft L.L.C. Although Icm.JS and NGL Viewer are both web-based molecular viewers, they have significant differences: Icm.Js is a library emphasizing a comprehensive set of tools for biological and chemical data visualization with desktop quality in modern browsers, while NGL Viewer is an open-source library emphasizing visualization of large molecular structures.

1.2.3. 3Dmol.js

3Dmol.js is an open-source library that allows users to manipulate and explore 3D molecular models within a web browser [19]. This library uses WebGL, taking advantage of the graphics card's capabilities to efficiently render molecules (Rego & Koes, 2015). It provides a user-friendly interface for end users and an API for developers to embed and control molecular visualizations within their own web applications (Rego & Koes, 2015). 3Dmol.js is an open-source library available under the BSD license for use in a variety of projects, including integrating molecular views into web applications, creating educational tools, sharing and embedding molecular data (Rego & Koes, 2015).

1.2.4. Mol Viewer

The web-native Mol Viewer (part of the Mol open-source project) is an open-source web-based toolkit hosted on GitHub, for the visualization and streaming of large-scale macromolecular and experimental coordinate data (Sehnal et al., 2021). This library is developed by an international group of contributors and supported by the Research Collaboratory for Structural Bioinformatics (RCSB) Protein Data Bank (PDB) and Protein Data Bank in Europe (PDBe) (Sehnal et al., 2021). Examples of its application are experimental/validation-related data for macromolecular models and the creation of visually interesting, engaging, and interactive educational materials (Sehnal et al., 2021). Some features of Mol viewer are advanced user interface, high-quality rendering, sequence view and molecular component focus tools and animation export, among others (Sehnal et al., 2021).

In summary, NGL Viewer, IcmJS, 3Dmol.js, and Mol Viewer are all web-based tools or libraries for visualizing molecular structures in 3D, however, they present distinctives features: NGL viewer support for various file formats, diverse molecular representations, and advanced rendering techniques; IcmJS offers tools for displaying docking hitlists, ligand interaction diagrams, and alignments, all linked to the 3D structure; 3Dmol.js offers a comprehensive JavaScript API for developers and a straightforward declarative interface for embedding and sharing of molecular visualizations in web applications; and Mol Viewer supports high-performance graphics and data handling, enabling the visualization of complex biological assemblies (Rose & Hildebrand, 2015; Molsoft, n.d.; University of Pittsburgh, n.d.; Sehnal et al., 2021).

In summary, webserver molecular visualizers present advantages about standalone molecular visualizers such as: No installation needed, cross-browser compatibility, easy collaboration & sharing, handles large data, among others. However, standalone molecular visualizers also show features to consider such as advanced rendering techniques, high performance, they can be used without an internet connection, and users have full control over all aspects of the visualization and analysis pipeline.

2 Related Works

2.1 JADOPPT

Java based Autodock Preparing and Processing Tool (JADOPPT) is a tool that allows interactive visual analysis providing means for refinement of docking studies (García-Pérez et al., 2017). JADOPPT is mainly used for three tasks. First task: Dimensionality reduction through hierarchical selection of representative poses for docked molecules. JADOPPT calculates the root mean square deviation (RMSD) between poses and automatically clusters them. The cluster size is determined by the RMSD threshold. The pose with the minimum energy within each cluster is selected as its representative. JADOPPT offers two types of clustering algorithms: hierarchical and partitional, also known as K-RMSD. Clustering results are presented as dendrograms or box views for K-RMSD. JADOPPT adds a graphical interface to the Jmol molecular viewer that allows analysis and comparison of clusters by visually inspecting their representatives, their binding energies, and automatically changing the selected representatives (García-Pérez et al., 2017). Second task: to conjoint clustering of multiple ligands from the poses selected in the previous task, allowing a visual comparison of different molecules (García-Pérez et al., 2017). Third task: interactively modifying the map files from AutoDock to prevent the appearance of non-relevant branches in subsequent docking exercises [3]. In conclusion, JADOPPT is a visual tool for analyzing and comparing multiple docking results. Its focus is to reduce the analysis time and validation of docking trials. JADOPPT is a visual analytical tool that complements the available tools for analyzing and refining the docking results (García-Pérez et al., 2017).

2.2 PyDRA

PyDRA (Python Docking Results Analyser) is a plugin for the molecular visualizer PyMOL allows to simultaneously process more than one docking file for the dlg and pdbqt file format from AutoDock and Vina. Also, this plugin facilitates the visualization of conformations through two clustering methods: K-RMSD algorithm and a hierarchical clustering through an agglomerative algorithm (AGNES agglomerative nesting). PyDRA performs three processes: a matrix construction which handles file parsing from AutoDock, clustering method selection and visual analyses of the clustering results. In summary, PyDRA allows to load more than one file at a time recognizing results from AutoDock and Vina, complementing the manipulation and visualization of the molecular structure through PyMOL in an easy way (Varela-Salinas et al., 2017).

2.3 Our 3D engine solution

In this work we propose the development and implementation of a web application for a 3D molecular structure viewer using a WebGL-based web tool that has a simple interface and takes advantage of future implementations of the technology. The methodology involved developing a web API in the Python programming language and implementing the interaction between the React Framework and the underlying Python scripts. These scripts process data (drug components and protein structure information), cluster the docked conformation, calculate the secondary structure of the protein PDB file and retrieve the results to the React Framework for visual representation. This tool renders visual components of the 3D molecular structure viewer in react-three/fiber, react-three/drei, and WebGL, providing a user-friendly and modern interface. Implementing the use of WebGL for rendering three-dimensional structures of molecules, the aim is to provide better compatibility with modern GPUs, support for general-purpose GPU calculations that remove load from the CPU, faster operations and access to more advanced

GPU functions. Likewise, a simpler interface will make the program a more accessible tool for those interested in molecular visualization without the need to learn how to use complex interfaces such as PyMOL or VMD, thus having a better reach for introductory users.

2.4 3D Engine Results

Fig. 1 shows the general view of the tool. The 3D visualizer engine displays a protein structure loaded from a PDB file, along with docking results from AutoDock Vina. On the right side, the top panel presents clustering results in a scatterplot visualization, while the bottom panel shows a hierarchical clustering result. Both visualizations are linked to the 3D visualizer. The user can control the number of conformations displayed by selecting a tag (e.g., Cluster 1, Cluster 2, etc.) in the scatterplot or by clicking on a branch in the dendrogram.

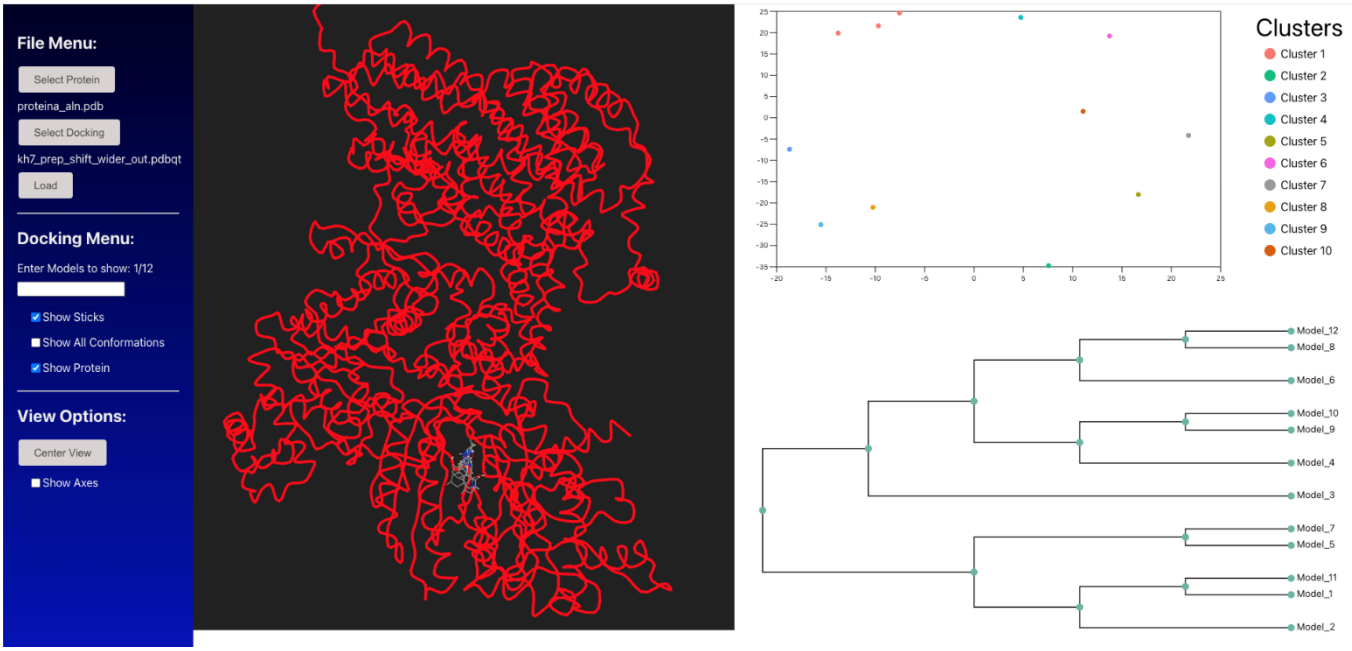

Fig. 1. General view of the tool.

On the left side of the interface is the toolbox control. In the first section of the toolbox, the user can select the input files: the protein and the docking results. After selecting a protein file and a docking file (either .dlg or .pdbqt), the user can load and process both by clicking the “Load” button. The second section allows the user to manipulate the visual representation of the docking results. The menu provides options to adjust how the conformations are displayed: either as sticks, or as sticks and spheres. The checkbox “Show All Conformations” and the input field control the number of conformations displayed (ligands). In other words, the checkbox enables visualization of all conformations, while the input field allows the user to filter conformations by index. For example, the user can display conformations 1, 2, and 6 by entering them separated by commas. Finally, the checkbox “Show Protein” serves as a toggle to show or hide the protein. The “Center View” button recenters the view to the origin, and for better orientation, a “Show Axes” checkbox is provided to display a 3D axis.

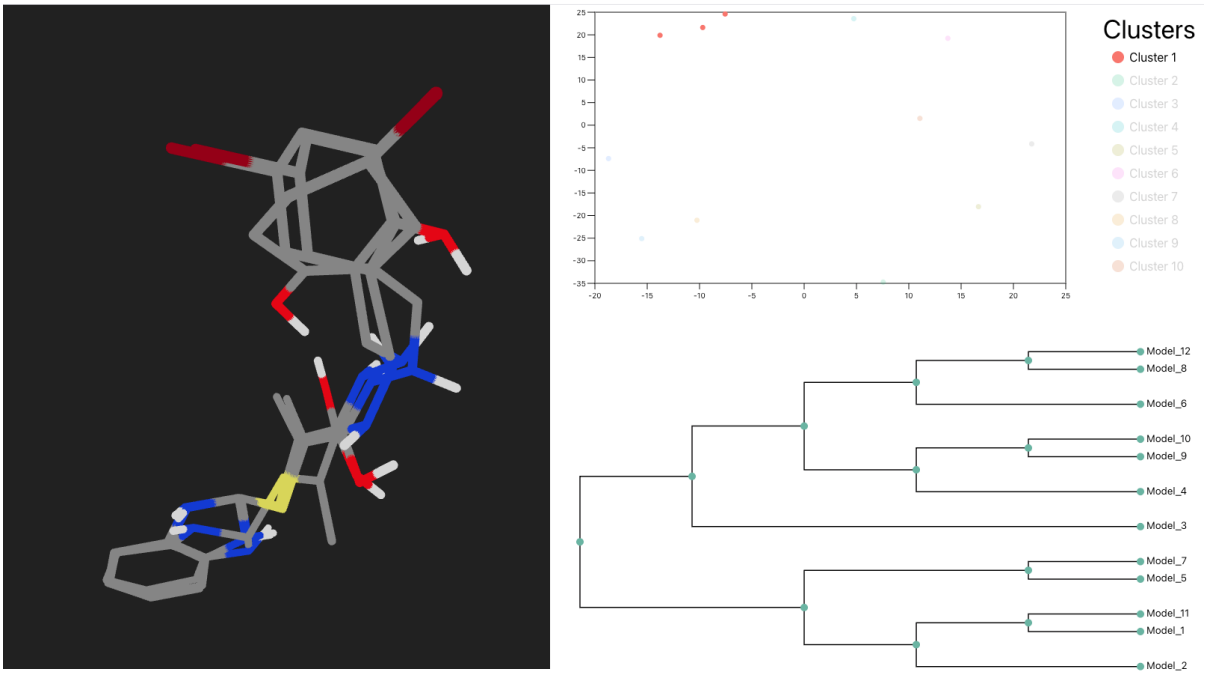

Fig. 2. User interaction with the 3D visualization engine via linked views.

Figure 2 shows the interaction between the visualization plots (scatterplot and dendrogram) and the 3D engine. The integration of linked views plays a crucial role in computer–human interaction, as it enables users to explore complex datasets intuitively and from multiple perspectives. By synchronizing different visual representations (such as structural, spatial, and clustering information), linked views allow users to quickly identify relationships, patterns, and anomalies that might otherwise remain hidden. In the context of bioinformatics tools for drug design, this interactivity significantly enhances decision-making. Researchers can seamlessly correlate docking conformations, cluster behaviors, and protein–ligand interactions within the same analytical environment. This not only accelerates the interpretation of docking results but also supports more informed hypotheses and rational design choices, ultimately improving the efficiency and accuracy of the drug discovery process.

4 Conclusions

The current version of the tool provides an initial framework for visualizing molecular docking results and exploring interactions between proteins and ligands. It offers a foundational 3D representation that helps users observe how docking conformations relate to the overall protein structure and how molecules spatially interact within the binding site. This early implementation demonstrates the potential of interactive visualization for enhancing understanding and communication in bioinformatics and drug design. However, the present visualization is limited to a basic representation of the protein, without detailed secondary structure elements. Future development will focus on extending the 3D engine to include explicit visual representations of α-helices and β-sheets, as well as improved color schemes to distinguish structural and functional regions more clearly. These enhancements will provide a more informative and interpretable depiction of the protein, supporting a deeper understanding of molecular interactions. In addition, future work will explore the integration of machine learning techniques to classify and cluster docking results automatically. By leveraging data-driven analysis, the tool could identify patterns and relationships among conformations more efficiently, supporting more informed decision-making in the drug discovery process. Ultimately, these developments will transform the tool into a more comprehensive and intelligent visualization platform for structural bioinformatics applications.

References

Ahmad, S. (2023, August 19). Exploring PyMOL colors for molecular visualization. LinkedIn. https://www.linkedin.com/pulse/exploring-pymol-colors-molecular-visualization-sunail-ahmad-/

Dey, R., Chaube, U., Bhatt, H., & Patel, B. (2025). Small molecule drug design. In Encyclopedia of Bioinformatics and Computational Biology (2nd ed., Vol. 6, pp. 622–633). Elsevier. https://doi.org/10.1016/B978-0-323-95502-7.00262-1

García-Pérez, C., Therón, R., & López-Pérez, J. L. (2017). JADOPPT: Java based AutoDock preparing and processing tool. Bioinformatics, 33(4), 583–585. https://doi.org/10.1093/bioinformatics/btw677

Goddard, T. D., Huang, C. C., Meng, E. C., Pettersen, E. F., Couch, G. S., Morris, J. H., & Ferrin, T. E. (2018). UCSF ChimeraX: Meeting modern challenges in visualization and analysis. Protein Science, 27(1), 14–25. https://doi.org/10.1002/pro.3235

Humphrey, W., Dalke, A., & Schulten, K. (1996). VMD: Visual molecular dynamics. Journal of Molecular Graphics, 14(1), 33–38. https://doi.org/10.1016/0263-7855(96)00018-5

Khronos Group. (n.d.). WebGL: Low-level 3D graphics API based on OpenGL ES. Retrieved June 4, 2026, from https://www.khronos.org/webgl/

Molsoft. (n.d.). IcmJS: JavaScript 3D molecular viewer. Retrieved June 4, 2026, from https://www.molsoft.com/activeicmjs.html

Morris, G. M., Huey, R., Lindstrom, W., Sanner, M. F., Belew, R. K., Goodsell, D. S., & Olson, A. J. (2009). AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. Journal of Computational Chemistry, 30(16), 2785–2791. https://doi.org/10.1002/jcc.21256

Nguyen, H., Case, D. A., & Rose, A. S. (2018). NGLview: Interactive molecular graphics for Jupyter notebooks. Bioinformatics, 34(7), 1241–1242. https://doi.org/10.1093/bioinformatics/btx789

NIH Resource for Macromolecular Modeling and Bioinformatics, & Theoretical and Computational Biophysics Group. (2012, January 29). VMD user's guide (Version 1.9.1). University of Illinois at Urbana-Champaign. https://www.ks.uiuc.edu/Research/vmd/vmd-1.9.1/ug/ug.html

Nipype. (2025). pydra: 1.0a2 [Computer software]. Zenodo. https://zenodo.org/records/16671149

PyMOL. (n.d.). Welcome to PyMOL. Retrieved June 4, 2026, from https://pymol.org/dokuwiki/doku.php?id=welcome

Rego, N., & Koes, D. (2015). 3Dmol.js: Molecular visualization with WebGL. Bioinformatics, 31(8), 1322–1324. https://doi.org/10.1093/bioinformatics/btu829

Rose, A. S., & Hildebrand, P. W. (2015). NGL Viewer: A web application for molecular visualization. Nucleic Acids Research, 43(W1), W576–W579. https://doi.org/10.1093/nar/gkv402

Rose, A. S., Bradley, A. R., Valasatava, Y., Duarte, J. M., Prlić, A., & Rose, P. W. (2018). NGL viewer: Web-based molecular graphics for large complexes. Bioinformatics, 34(21), 3755–3758. https://doi.org/10.1093/bioinformatics/bty419

Schrödinger, LLC. (2015). The PyMOL molecular graphics system (Version 1.8) [Computer software].

Seeliger, D., & de Groot, B. L. (2010). Ligand docking and binding site analysis with PyMOL and Autodock/Vina. Journal of Computer-Aided Molecular Design, 24, 417–422. https://doi.org/10.1007/s10822-010-9352-6

Sehnal, D., Bittrich, S., Deshpande, M., Svobodová, R., Berka, K., Bazgier, V., Velankar, S., Burley, S. K., Koča, J., & Rose, A. S. (2021). Mol Viewer: Modern web app for 3D visualization and analysis of large biomolecular structures. Nucleic Acids Research, 49(W1), W431–W437.* https://doi.org/10.1093/nar/gkab314

Seshadri, K., Liu, P., & Koes, D. R. (2020). The 3Dmol.js Learning Environment: A Classroom Response System for 3D Chemical Structures. Journal of Chemical Education, 97(10), 3872–3876. https://doi.org/10.1021/acs.jchemed.0c00579

Spivak, M., Stone, J. E., Ribeiro, J., Saam, J., Freddolino, L., Bernardi, R. C., & Tajkhorshid, E. (2023). VMD as a platform for interactive small molecule preparation and visualization in quantum and classical simulations. Journal of Chemical Information and Modeling, 63(15), 4664–4678. https://doi.org/10.1021/acs.jcim.3c00658

Trott, O., & Olson, A. J. (2010). AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimisation, and multithreading. Journal of Computational Chemistry, 31(2), 455–461. https://doi.org/10.1002/jcc.21334

University of Pittsburgh. (n.d.). Using 3Dmol.js. Retrieved August 1, 2025, from https://3dmol.csb.pitt.edu/doc/

Varela-Salinas, G., García-Pérez, C. A., Peláez, R., & Rodríguez, A. J. (2017). Visual clustering approach for docking results from Vina and AutoDock. In F. Martínez de Pisón, R. Urraca, H. Quintián, & E. Corchado (Eds.), Hybrid Artificial Intelligent Systems. HAIS 2017 (Lecture Notes in Computer Science, Vol. 10334, pp. 342–353). Springer. https://doi.org/10.1007/978-3-319-59650-1_29

Yuan, S., Chan, H. C. S., & Hu, Z. (2017). Using PyMOL as a platform for computational drug design. WIREs Computational Molecular Science, 7(2), Article e1298. https://doi.org/10.1002/wcms.1298