# The 3Dmol.js learning environment: a classroom response system for 3D chemical structures

Keshavan Seshadri,[†] Peng Liu,[‡] and David Ryan Koes[*,¶]

[†]*Center for Computational Natural Sciences and Bioinformatics (CCNSB), International Institute of Information Technology, Gachibowli, Hyderabad, Telangana 500032, India*

[‡]*Department of Chemistry, University of Pittsburgh, Pittsburgh, PA 15260*

[¶]*Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA 15260*

E-mail: dkoes@pitt.edu

**Abstract**

Classroom response systems are an important tool in many active learning pedagogies. They support real-time feedback on student learning and promote student engagement, even in large classrooms, by allowing instructors to solicit an answer to a question from all students and show the results. Existing classroom response systems are general purpose and not tailored to the specific needs of a chemistry classroom. In particular, it is not easy to deploy molecular representations except as static images. Here we present the 3Dmol.js learning environment, a classroom response system that uses the open source web-based 3Dmol.js JavaScript framework to provide interactive viewing and querying of 3D molecules. 3Dmol.js is available under a BSD 3-clause open source license, and the learning environment features are all available through `http://3dmol.csb.pitt.edu/` without any software installation required.

## Keywords

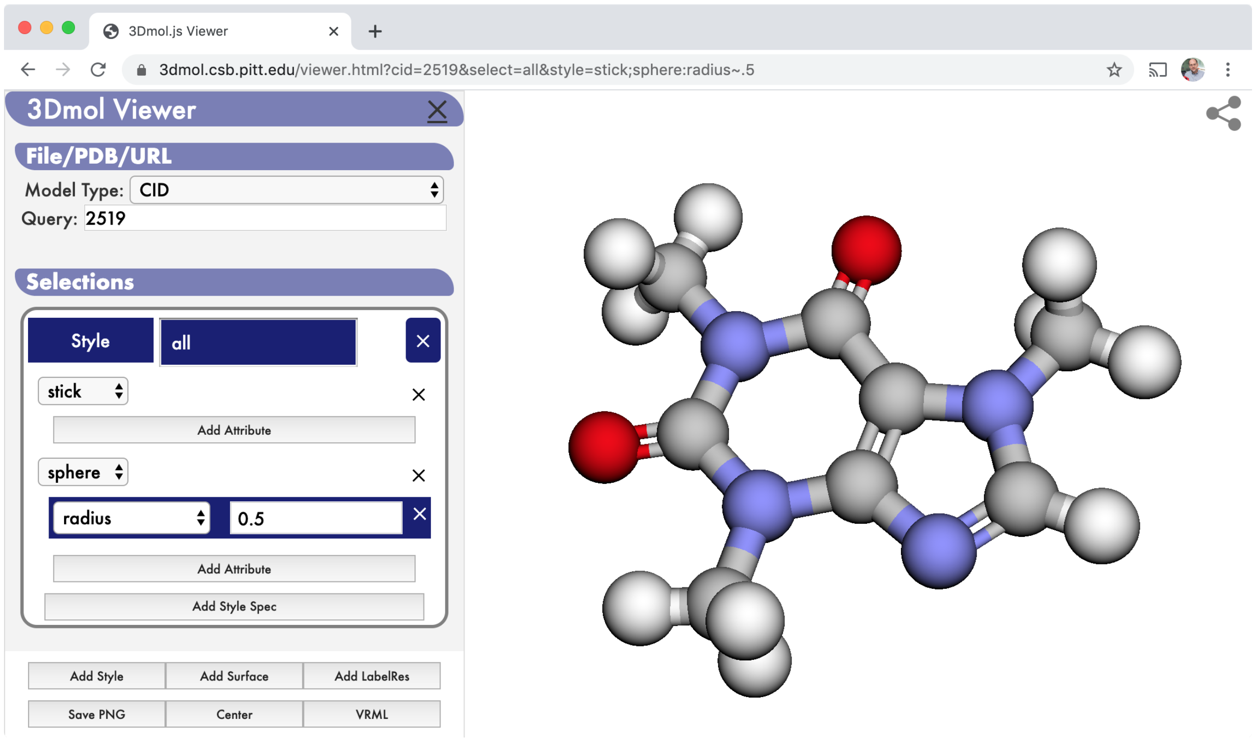
audience response system, clickers, active learning

## Introduction

Active learning is "the process of having students engage in some activity that forces them to reflect upon ideas and how they are using those ideas."[1] The application of active learning pedagogies results in significant improvements in learning outcomes in science classrooms,[2–4] including organic chemistry classes.[5–7] One popular active learning method is the use of classroom response systems,[8] also known as audience/student/personal response systems or "clickers." In these systems, the instructor poses a question and students answer using a personal device, either a dedicate piece of hardware (a "clicker") or a web-enabled device such as a smartphone or laptop.[7] Students answer anonymously and the collated answers are shown to the class to enable further discussion. In a meta-analysis[9] of 53 studies, the use of classroom response systems was found to produce a small, but statistically significant, benefit on cognitive outcomes and a larger benefit on non-cognitive outcomes, such as the students' enjoyment of the class and ranking of the instructor.

The use of classroom response systems in the chemistry classroom is growing, particular for large classes.[10,11] However, general-purpose software not designed specifically for chemistry is often used, reducing the efficacy of the system.[7] Importantly, we are not aware of any system that seamlessly integrates 3D chemical structures, which are important for engaging students and enabling spatial learning,[12] for example when visualizing stereo-selective reaction pathways.[13] Advances in web technologies have mode visualizing 3D structures online a seamless experience. It is no long necessary to install plugins[14] or use Java[15] as there are multiple JavaScript libraries that use the hardware-accelerated 3D graphics WebGL standard to provide high performance, in-browser 3D molecular visualizers.[16–19] To address the need for a molecule-focused classroom response system, we have extended the 3Dmol.js[16]

JavaScript library for online molecular visualization to include a learning environment with classroom response system features. We next describe how these features work and discuss future directions.


3Dmol.js Viewer

3dmol.csb.pitt.edu/viewer.html?cid=2519&select=all&style=stick;sphere:radius~.5

Model Type: CID
Query: 2519

Style all
stick
sphere
radius 0.5

Figure 1: The 3Dmol.js viewer. The edit panel (left) is shown and is used to load a molecule, in this case compound id 2519 from PubChem, and set the style. The result can be accessed at the URL `https://3dmol.csb.pitt.edu/viewer.html?cid=2519&select=all&style=stick;sphere:radius~.5`

## 3Dmol.js Learning Environment

3Dmol.js[16] is a WebGL-based JavaScript library for hardware-accelerated online molecular graphics. It supports most molecular file formats and visualization styles, including support for volumetric data (e.g. cube files) and simulation data (e.g. AMBER[20] or GROMACS[21] data). In addition to its full-featured JavaScript API, 3Dmol.js offers an embedding API, where a molecular view can be added to a web page by inserting a single `div` statement, and a URL-based hosted viewer API. The hosted viewer (`http://3dmol.csb.pitt.edu/viewer.`

`html`) is specifically designed so that all information to display the current molecular view is included in the URL. That is, having constructed a scene by loading the appropriate molecular data and applying the desired styles using the provided edit panel (Figure 1), it is possible to share the URL with students (e.g. as a QR code) so they can see the identical scene and interact with it. Molecular data is loaded into the viewer by specifying a PubChem compound id (the PubChem entry must have an associated 3D structure), a Protein Data Bank identifier, or a URL to an externally hosted file. We have added functionality to the 3Dmol.js API and hosted viewer to create a learning environment that can be used as a classroom response system where students answer queries by clicking on atoms of a 3D molecule.

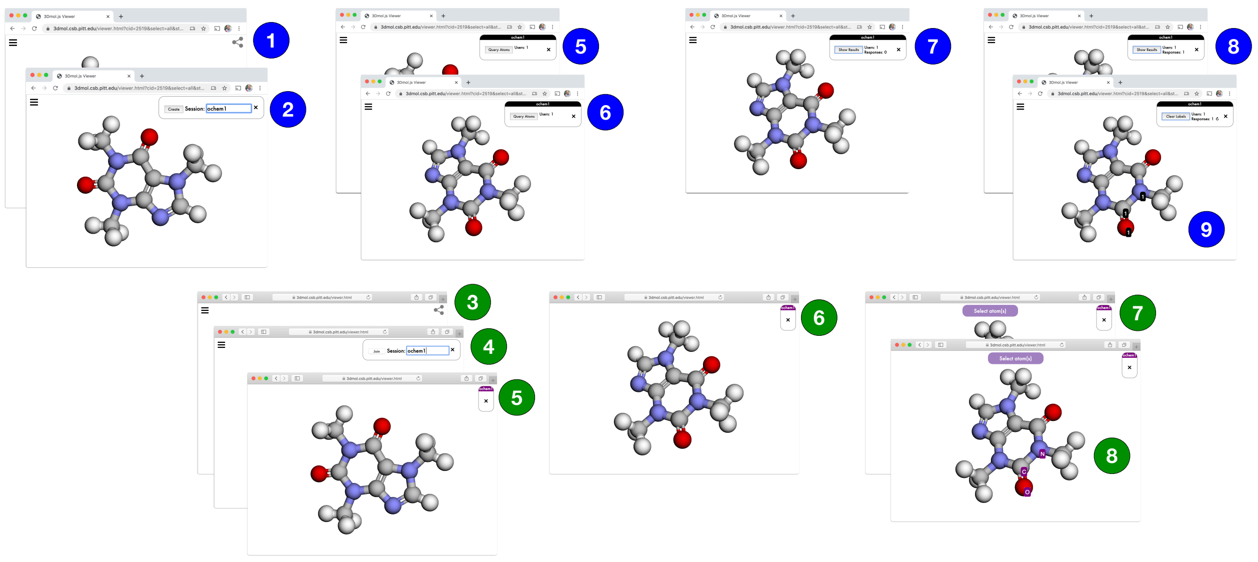


Figure 2: Flow of instructor (top) and student (bottom) interactions with the 3Dmol.js classroom response system.

## User Interface

Usage of the 3Dmol.js learning environment is shown in Figure 2 and a video demonstration is available at `http://3dmol.csb.pitt.edu/doc/tutorial-learning_environment.html`. (1) The instructor clicks on the 'share' icon in the upper right of the hosted viewer. (2) In the resulting text box, the instructor creates a session by specifying its name. Once created, a session lasts as long as the creating browser window is left open and only that browser window has the ability to manipulate the session contents. The session will be initialized with the current molecular view. If a session is already available, the 'Create' button will change to a 'Join' button. (3) When students click on the share button (4) they enter the name of the already created session and click 'Join.' Alternatively, students can be provided with a URL of the form `http://3dmol.csb.pitt.edu/viewer.html?session=NAME` where NAME is the name of the session. Note that the session must be extant for this to be a valid URL. (5) This populates their viewer with the instructor's molecular view and the instructor's view is updated to show that an additional user has connected. Monitoring the number of connections is important for ensuring the whole class is participating. (6) When the instructor modifies the view of the molecule, the student's view is updated to match. As long as the instructor is not changing the view, students can freely interact with the molecule by rotating, translating, or zooming the view. (7) When the instructor clicks on 'Query Atoms' every student is prompted to 'Select atom(s).' (8) As students click on atoms, the number of responses is reported to the instructor. When students select atoms they are labelled. Students can deselect an atom by clicking on it again. The number of responses is the number of students who have made any selection, not the number of selected atoms. (9) Once satisfied with the number of responses, the instructor can click 'Show Results' and atoms in the instructor view will be labeled with the number of students who have clicked on them. The opacity of the label is scaled proportionally to the popularity of the selection. There is a refresh button available to load the most recent results, for example after further explaining a concept in response to the previously shown selections. Once the answers have been discussed, the results can be cleared to pose a new question, or a different molecule can be loaded using the edit panel.

## Case Study

We effectively deployed the 3Dmol.js learning environment as part of an organic chemistry recitation attended by approximately 80 students. The major goal of the recitation was to practice concepts related to nomenclature and stereochemistry of carbohydrate molecules, which require the student to have a good understanding of the 3D structures and the ability to analyze these structures. Students were provided with a paper study guide that included questions and a QR code they could scan to load the 3Dmol.js session. The handout asked a series of questions about the nomenclature and stereochemistry regarding the 3D structures of several monosaccharide molecules. The students were led through the questions using the learning environment to guide their thinking. For example, the first question asked them to classify the sugar as a triose, tetrose, pentose, hexose, or heptose, so the students were asked to click all the carbons to count the number of carbon atoms in the molecule. This was a simple question to introduce the interface and the interactive features to the students. In another example, when answering whether the sugar was an aldose or ketose and whether it is an alpha or beta anomer, students were asked to click on the anomeric carbon of the cyclic structure of a furanose. A minority of students clicked on an incorrect carbon that is also attached to the oxygen atom in the ring but not a hemiacetal carbon, which prompted a discussion about how these carbons differ. The majority of students (50-60) provided timely responses to queries. Students had no difficulty using the interface, despite receiving no explicit training on interacting with the 3D viewer. Interestingly, the number of connected users steadily rose over the course of the recitation to more than one hundred active connections. Based on in-class observations, this was likely due to students switching from using their mobile device to a laptop. Although the learning environment is usable on a mobile device, there are definite benefits to a device with a large screen and precise pointing device. Student feedback on using the tool was positive, although one student provided the insightful comment that they would not have the luxury of manipulating 3D molecules when they took the test, pointing out the difficulty in assessing a student's ability to understand stereochemistry separately from their ability to process 2D notations for stereochemistry.

## Implementation

Like many web applications, the 3Dmol.js learning environment is divided into a frontend, the part that is run in the browser, and a backend, the part that is run on a server. The frontend consists of the 3Dmol.js molecular viewer. 3Dmol.js relies on the availability of WebGL, a JavaScript interface to OpenGL that enables hardware accelerated 3D graphics in the browser. All modern browsers support WebGL 1.0, although currently some browsers (e.g. Internet Explorer, some mobile browsers) do not fully support WebGL 2.0 which is required for volumetric rendering in 3Dmol.js. 3Dmol.js supports touch controls (drag to rotate, pinch to zoom, three finger drag to translate) and so can be used on mobile devices, although some older devices with limited graphics processing power may not support WebGL.

The backend is a Python script that uses the Flask[22] web framework to manage the learning environment sessions and communicate with the frontend. Sessions are implemented using WebSockets and SocketIO,[23,24] which is a low overhead communication protocol for maintaining a full duplex connection between a web browser and a server. It does not require polling, reducing the computational load on both the client and server, although the Flask-SocketIO library we are using will gracefully degrade to higher overhead communication protocols to support older browsers. The server stores the state of the session (the molecular data, the applied styles, and the viewer orientation) as updates are sent from the instructor as well as responses to queries as updates are sent from the students. Sessions end when the instructor disconnects, freeing the memory occupied by the session state.

**Self Hosting** The easiest way to use the 3Dmol.js learning environment is through the hosted viewer at `http://3dmol.csb.pitt.edu/viewer.html`. However, it may be desirable to host the learning environment on a local server. This can be easily done by running a standalone Flask web server. On a Ubuntu style Linux server the necessary dependencies can be installed with the commands:

```
apt install npm python3-pip git
pip3 install flask flask_socketio eventlet
```

and 3Dmol.js can cloned (or, alternatively, downloaded) from GitHub:

```
git clone https://github.com/3dmol/3Dmol.js.git
```

and built with the node package manager:

```
cd 3Dmol.js
npm install
```

Once 3Dmol.js is installed and built, a standalone web server can be launched:

```
cd learning-environment
python3 server.py
```

This will launch a server on port 5000. Note that services running on this port must be accessible to students (check with your network administrator), and the port can be changed with the `-p` argument. The viewer is at `http://HOSTNAME:5000/static/viewer.html` where `HOSTNAME` is the server's internet address. Files on the web server can be accessed using relative paths to the 3Dmol.js directory (not the learning-environment directory). For example, `http://HOSTNAME:5000/static/viewer.html?url=molecule.sdf&style=stick`, will load `3Dmol.js/molecule.sdf`. Similar commands can be used to install a server on MacOS and Windows, with the main difference being how the Python, npm, and git dependencies are installed.

The standalone Flask-based web server should be sufficient for meeting the demands of a single chemistry class. If a more scalable solution is desired, the 3Dmol.js website includes documentation and links for integrating the learning environment server with a high performance web server such as apache.

## Future Plans

The 3Dmol.js learning environment already provides the necessary features to be fully usable in a chemistry classroom. However, there are number of ways the software could be improved. Currently, the hosted viewer and URL API does not support a grid view where there are multiple independent viewers shown on the same page. In order to have students answer questions that select between different molecules (e.g., "Which molecule is a disaccharide?"), a multi-molecule file must be used. In this case, the molecules are part of the same coordinate system and so can not be independently manipulated. We would like to change this so not only can multiple molecules be independently displayed, but questions can be posed where students select whole molecules, not single atoms. Other ideas for improved selection semantics are to allow students to select whole residues, e.g. amino acids of protein, for biochemistry applications. Currently, if a molecule is not available in a public database, the user must host the molecular data themselves. Although this can easily be done by uploading to public folders in file sharing services, we are considering supporting file uploads natively within the viewer. Uploaded files would be associated with a unique identifier and permanently stored on the server. Another future goal is to integrate with Jupyter notebooks. 3Dmol.js viewers can already be embedded in a Jupyter notebook using the py3Dmol module, but we would like to support learning environment features within notebooks as well. This would allow instructors who create slides using Jupyter to embed the active learning activities directly within the lecture materials. We already provide a small JavaScript plugin that provides this functionality for multiple choice questions (`https://github.com/dkoes/asker.js`). We strongly encourage users of the 3Dmol.js learning environment to provide feedback and suggestions on the project's GitHub Issues page (`https://github.com/3dmol/3Dmol.js/issues`).

## Acknowledgement

We thank the students of CHEM 0320 Organic Chemistry 2 for evaluating the 3Dmol.js learning environment during their recitation. Keshavan Seshadri was supported through Google Summer of Code and the Open Chemistry project. This work is supported by R01GM108340 from the National Institute of General Medical Sciences and CHE-1654122 from the National Science Foundation.

## References

(1) Collins, J. W.; O'Brien, N. P. *The Greenwood dictionary of education*; ABC-CLIO, 2011.

(2) Michael, J. Where's the evidence that active learning works? *Advances in physiology education* **2006**, *30*, 159–167.

(3) Deslauriers, L.; Schelew, E.; Wieman, C. Improved learning in a large-enrollment physics class. *Science* **2011**, *332*, 862–864.

(4) Armbruster, P.; Patel, M.; Johnson, E.; Weiss, M. Active learning and student-centered pedagogy improve student attitudes and performance in introductory biology. *CBE—Life Sciences Education* **2009**, *8*, 203–213.

(5) Jeske, R. C.; Jones, J. A.; Stanford, C. L. *Active Learning in Organic Chemistry: Implementation and Analysis*; ACS Publications, 2019; pp 69–86.

(6) DeCicco, R. C. *Active Learning in Organic Chemistry: Implementation and Analysis*; ACS Publications, 2019; pp 57–68.

(7) Shea, K. M. Beyond Clickers, Next Generation Classroom Response Systems for Organic Chemistry. *J. Chem. Educ.* **2016**, *93*, 971–974.

(8) Martyn, M. Clickers in the classroom: An active learning approach. *Educause quarterly* **2007**, *30*, 71.

(9) Hunsu, N. J.; Adesope, O.; Bayly, D. J. A meta-analysis of the effects of audience response systems (clicker-based technologies) on cognition and affect. *Computers & Education* **2016**, *94*, 102–119.

(10) Gibbons, R. E.; Laga, E. E.; Leon, J.; Villafañe, S. M.; Stains, M.; Murphy, K.; Raker, J. R. Chasm crossed? Clicker use in postsecondary chemistry education. *J. Chem. Educ.* **2017**, *94*, 549–557.

(11) Woelk, K. Optimizing the use of personal response devices (clickers) in large-enrollment introductory courses. *J. Chem. Educ.* **2008**, *85*, 1400.

(12) Fatemah, A.; Rasool, S.; Habib, U. Interactive 3D Visualization of Chemical Structure Diagrams Embedded in Text to Aid Spatial Learning Process of Students. *J. Chem. Educ.* **2020**, *97*, 992–1000.

(13) O'Brien, M. Creating 3-Dimensional Molecular Models to Help Students Visualize Stereoselective Reaction Pathways. *J. Chem. Educ.* **2016**, *93*, 1663–1666.

(14) Charistos, N. D.; Tsipis, C. A.; Sigalas, M. P. 3D Molecular Symmetry Shockwave: A Web Application for Interactive Visualization and Three-Dimensional Perception of Molecular Symmetry. *J. Chem. Educ.* **2005**, *82*, 1741.

(15) Hanson, R. M. Jmol–a paradigm shift in crystallographic visualization. *J. Appl. Crystallogr.* **2010**, *43*, 1250–1260.

(16) Rego, N.; Koes, D. 3Dmol. js: molecular visualization with WebGL. *Bioinformatics* **2015**, *31*, 1322–1324.

(17) Rose, A. S.; Hildebrand, P. W. NGL Viewer: a web application for molecular visualization. *Nucleic Acids Res.* **2015**, *43*, W576–W579.

(18) Mol*. `https://molstar.org/`, Accessed: 2020-07-07.

(19) Wang, J.; Youkharibache, P.; Zhang, D.; Lanczycki, C. J.; Geer, R. C.; Madej, T.; Phan, L.; Ward, M.; Lu, S.; Marchler, G. H.; Wang, Y.; Bryant, S. H.; Geer, L. Y.; Marchler-Bauer, A. iCn3D, a web-based 3D viewer for sharing 1D/2D/3D representations of biomolecular structures. *Bioinformatics* **2019**, *36*, 131–135.

(20) Case, D. A.; Cheatham III, T. E.; Darden, T.; Gohlke, H.; Luo, R.; Merz Jr, K. M.; Onufriev, A.; Simmerling, C.; Wang, B.; Woods, R. J. The Amber biomolecular simulation programs. *J. Comput. Chem.* **2005**, *26*, 1668–1688.

(21) Van Der Spoel, D.; Lindahl, E.; Hess, B.; Groenhof, G.; Mark, A. E.; Berendsen, H. J. GROMACS: fast, flexible, and free. *J. Comput. Chem.* **2005**, *26*, 1701–1718.

(22) Flask: web development, one drop at a time. `https://flask.palletsprojects.com/en/1.1.x/`, Accessed: 2020-07-08.

(23) socket.io. `https://socket.io/`, Accessed: 2020-07-08.

(24) flask-SocketIO. `https://flask-socketio.readthedocs.io/`, Accessed: 2020-07-16.

## Graphical TOC Entry