3Dmol.js: 3D structure visualization for the modern web

Jasmine Collins[1,2], Matthew Ragoza[1,2], Justin Jensen[3], David Ryan Koes[1]

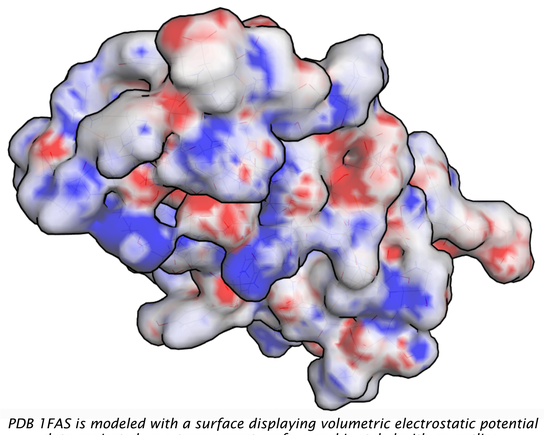
[1]*Department of Computational and Systems Biology, University of Pittsburgh,* [2]*Kenneth P. Dietrich School of Arts and Sciences, University of Pittsburgh,* [3]*Pittsburgh Science & Technology Academy*

http://3dmol.csb.pitt.edu

Introduction

3Dmol.js is an open source, hardware-accelerated, object-oriented JavaScript library for fast and accessible molecular visualization. Without the use of plugins, users can visualize and interact with molecular data in any modern desktop or mobile web browser with near native performance.

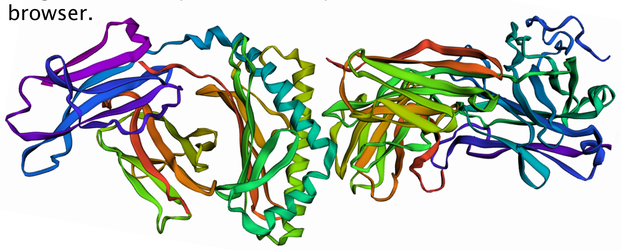
Supported file formats include cube, mmCIF, mol2, pdb, pqr, sdf, and xyz. Visualization styles include lines, crosses, sticks, spheres, and cartoons. Atoms can be colored by type or property. Users can generate molecular surfaces, create shapes such as arrows and lines, annotate the view with text and image labels, and install callback handlers for interactivity with the viewer contents.



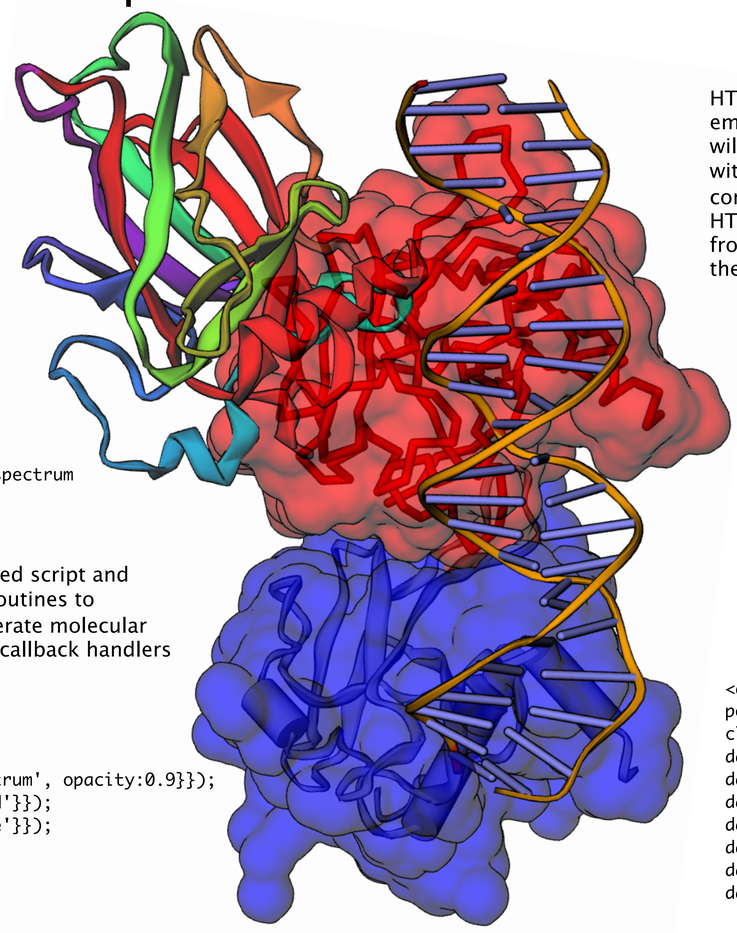
PDB 1FAS is modeled with a surface displaying volumetric electrostatic potential data projected on a transparent surface and is styled with an outline.

Online Viewer

3Dmol.js provides a hosted viewer where the molecular data is set and styled through a URL specification. Data may be retrieved from a remote URL, such as a publicly accessible shared folder on cloud storage. This allows users to easily share complex scenes without requiring that the recipients have any software other than a modern web browser.



http://3dmol.csb.pitt.edu/viewer.html?pdb=4WW2&style=cartoon:color~spectrum



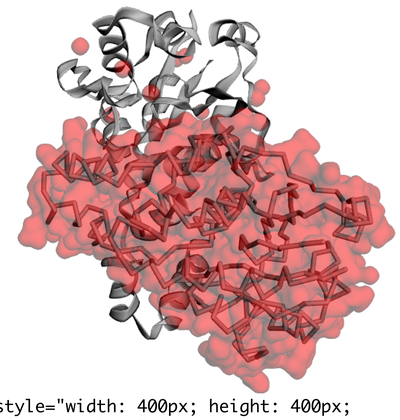
JavaScript API

JavaScript developers can use 3Dmol.js by including a single minified script and using the routines provided in the $3Dmol namespace. There are routines to manipulate molecular data, create isosurfaces from grid data, generate molecular surfaces, annotate the view with text and image labels, and install callback handlers for when a user interacts with the viewer (e.g., clicks on an atom).

```
$3Dmol.download("pdb:1TUP", viewer, {});
 viewer.setViewStyle({style:"outline",color:"black", width:0.1});
 viewer.setStyle({}, {cartoon:{ribbon:true}});
 viewer.setStyle({chain:'A'}, {cartoon:{arrows:true, color:'spectrum', opacity:0.9}});
 viewer.setStyle({chain:'B'}, {cartoon:{style:"trace", color:'red'}});
 viewer.setStyle({chain:'C'}, {cartoon:{tubes:true, color:'purple'}});
 viewer.addSurface({},{opacity:0.85, color:'red'},{chain:'B'});
 viewer.addSurface({}, {opacity:0.8, color:'blue'},{chain:'C'});
 viewer.render();
```

Embeddable Viewer

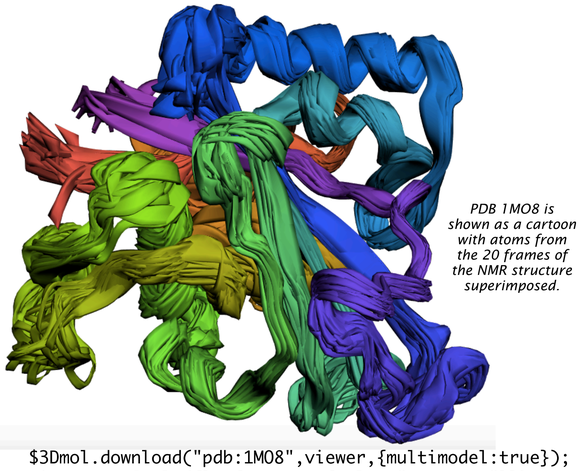
HTML authors do not need to use JavaScript to embed 3D viewers within their websites. 3Dmol.js will automatically turn any HTML element annotated with the `viewer_3Dmoljs` class into a viewer. The contents of the viewer are set and styled through HTML data tags. Molecular data can be retrieved from a remote URL or an element embedded within the web page.



```
<div style="width: 400px; height: 400px;
position: relative;"
class='viewer_3Dmoljs'
data-element='moldata_sdf'
data-backgroundcolor='0xffffff'
data-select1='chain:A'
data-style1='cartoon:style=trace'
data-surface1='opacity:.7;color:red'
data-select2='chain:B'
data-style2='cartoon:thickness=0.1'></div>
```

Multiple-model Support & Animation

A multiple-model file can be parsed into separate frames and be animated forward, backward or back and forth. Alternatively, all atoms can be read into one frame and displayed at once. The user can easily set the speed, direction, and number of loops of the animation.



PDB 1MO8 is shown as a cartoon with atoms from the 20 frames of the NMR structure superimposed.

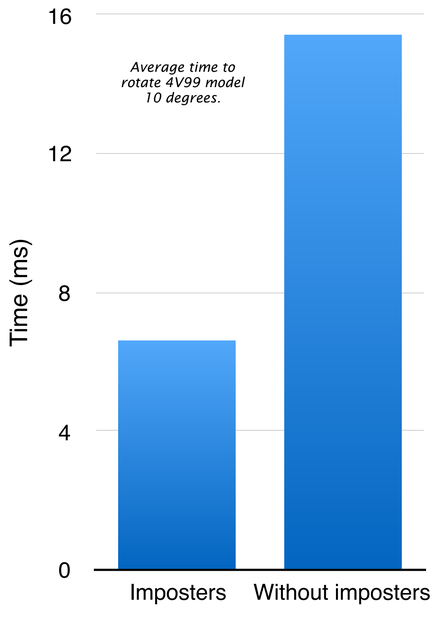
```
$3Dmol.download("pdb:1MO8",viewer,{multimodel:true});
viewer.setStyle({}, {cartoon:{color:"spectrum"}});
viewer.animate({interval: 75, loop: “backAndForth”);
viewer.render();
```

XYZ files with vibration information are visualized through animation. Frames are automatically populated given the user's desired distortion intensity and number of frames.

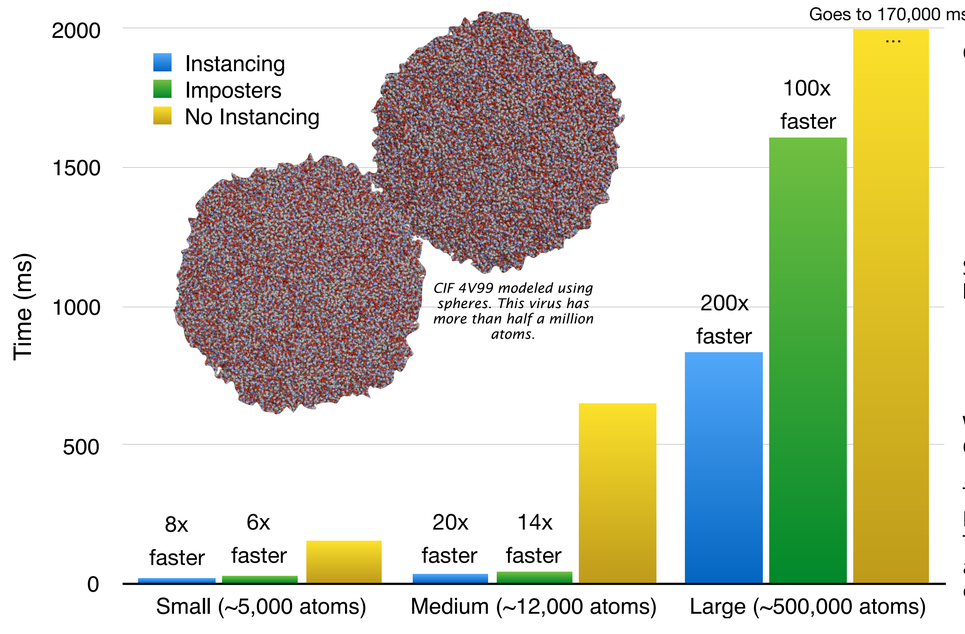
WebGL 2.0 Optimization

WebGL 2.0 provides functionality to accelerate the drawing of several shapes, such as spheres. 3Dmol takes advantage of these features for improved performance through instancing and the use of atom imposters.

Rotation Performance



Model Creation Performance



Multiple Cartoon Styles

3Dmol supports several cartoon rendering styles, such as oval, outline, tubes, arrows, nucleic acids, trace and transparent cartoons.

Future Plans

Currently planned features include

- support for pse files,
- a full-featured hosted viewer,
- improved transparency rendering modes
- expanded file format support through OpenBabel compiled with emscripten,
- integration with JSmol,
- better support for volumetric data.

Submit feature requests to our issue tracker:
http://github.com/dkoes/3Dmol.js/issues

Acknowledgements

We are grateful to Takanori Nakane for creating GLmol which is the basis of 3Dmol.js.

This work was supported by the National Institute of Health [R01GM108340].
The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.