A tool for exploratory data analysis of time series data from molecular dynamics simulations

Arushi Prakash
Department of Chemical Engineering,
University of Washington, Seattle
arushi3@uw.edu

ABSTRACT
In this paper, I present a visualization tool for exploring molecular simulation data. This tool addresses the design concern that quantitative data - like energy, distance, and pressure - and structural data are visualized separately and thereby lose context. This tool contains a 3D rendering of the structure at each time step and 2D scatter plot distribution of points denoting relationships between physical data at each time step. This web-based GUI platform for visualizing both types of data will decrease user interface with too many software, decrease scripting time, promote collaboration and increase exposure of data for exploratory analysis. I also tested this tool with domain researchers to understand how they used it and the extensions that they would prefer.

Author Keywords
D3, Three.js, molecular simulation, multi-dimensional scaling.

INTRODUCTION
Current visualization tools for molecular dynamics simulations focus on representing structural aspects - like secondary structure of protein, charge, atom type - at every time step of the simulation. This approach does not take into account changes in physical quantities at each of those time steps. Relating crucial physical/quantitative information – like energy, pressure – to the structure is necessary for a holistic assessment of the simulation trajectory. Moreover, simulation scientists often go back and forth between tools like Visual Molecular Dynamics (VMD) and PyMol for structure, and gnuplot and xmgrace for plotting. When quantitative data is analyzed separate from structural data, the user loses context, and vice versa. Understanding both forms of data is important to understand the mechanism, detect unphysical behavior and identify other variables that might be important.

This design study aims to create a tool that allows molecular simulation scientists to explore quantitative data from their simulation while maintaining structural context. I use two types of data from simulations – time series of structure and time series of quantitative data and represent them by a 3D model and dimensionally-reduced scatter plot, respectively. By intra- and inter- plot interaction, I hope to design a system that promotes data exploration among simulation scientists. The design decisions for this toolkit are given below with an informal usability study of domain scientists. To my knowledge this is the first molecular modelling tool which retains context of physical data from simulations and removes the need for extensive scripting for these analyses tasks.

DATA CHARACTERISTICS
Data from simulations are voluminous, and is often a time series. This data fluctuates rapidly (see Figure 1), according to statistical mechanics principles. Both average values and degree of fluctuation are important to analyze data. Small multiple representation line plots is the standard visualization but it has limited use for understanding relationships between variables. Since the data fluctuates, these plots are dense limiting accurate comparisons.

Structural data from simulations is in the form of x, y, and z coordinates of all atoms in the system. The number of atoms are usually constant in the system while the positions change. Molecular modelling software also add data about atom sizes, and calculate possible bonds between atoms in order to represent structural data as accurately as possible in 3D.

RELATED WORK
Web-based molecular modelling visualization – Most visualization tools for molecular simulation require installing client software. Among web-based approaches, Jmol – a popular choice - uses a Java Applet to render 3D molecules and enable user interaction[14]. Others have utilized the power of WebGL[15], generally using the Threejs graphics library, for visualization. Some examples include bioweb3D for biological visualization [9] and iview for protein-ligand visualization[7]. Some javascript libraries have also been developed specifically to tackle challenges in visualizing molecular modelling – SpiderGL[3] and ChemDoodle Web Components[16]. These approaches have historically concentrated on modelling the 3D structure without considering associated data from simulations.

Temporal visualization – This is a prolific area but I will focus on the visualizations used in this paper. If the temporal data is thought of as a space-time cube, I use 3 major visualization techniques as defined by Bach, 2014[1] – time coloring, time juxtaposing, and space flattening.

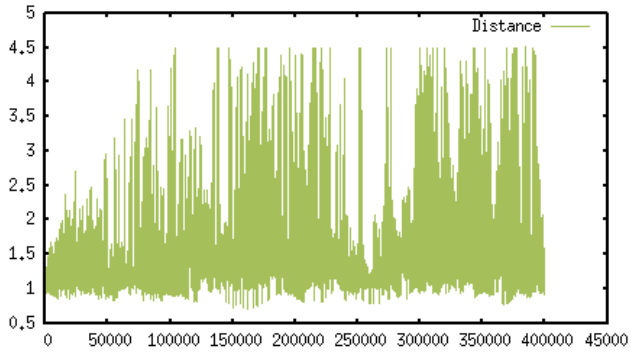

Figure 1. Sample quantitative data from simulations. Distance on the y-axis is in nm. Frame numbers are plotted on the x-axis

Examples of time coloring include distinguishing old links from new[4] and old character forms from new[17]. Examples of time juxtaposing – placing slices of time domain together – are the famous small multiple analysis of horse gait[5], map timeline[11] and dynamics maps[8]. The space flattening approach in this paper has previously been used to visualize document histories[12] and dynamic networks[6,10]

METHODS

Quantitative Data

Processing Algorithms - Inspired by time curves[2], I encode spatial distance between points to represent difference between data values, and encode colors to represent time. Up to three data columns/attributes are selected for plotting. Each column is scaled down to [0,1]. I populate the distance or similarity matrix by calculating the Euclidean distance between points.

$$d_{ij}{}^2 = \sum_{n=1}^{3} \left(P_{in} - P_{jn}\right)^2$$

where, i,j are different time points, P is the data from the 3 selected attributes/columns.

Manhattan distance also gives similar results for my data. I use classical multi-dimensional scaling to project these points on a 2-dimensional plane (see Figure 2). Points are colored according to the ordinal time variable (blue representing earlier time frames, and red represents later time frames). This is also known as time coloring.

In order to remove overlaps between points and facilitate better comprehension of data, a randomized algorithm moves points away. Points within a minimum distance of separation are chosen to move in the direction of a random angle. Looping over this multiple times (n=400 for this study) ensures removal of overlap but introduces small errors in spatial encoding between points.

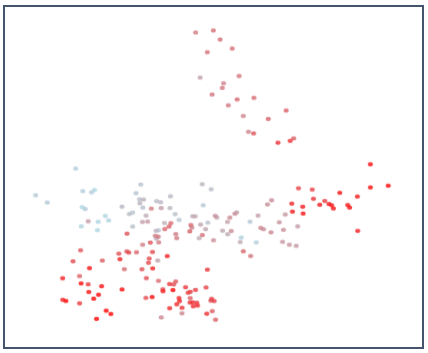

Figure 2. Plot of 3 variables on a 2D plane using multi-dimensional scaling approach (the color scale changes from blue to red, denoting early to later time points)

Interactions – Since this is an exploratory data analysis toolkit, interactions are designed to access as much data as possible. Tooltips give information about the selected data attributes for users to access underlying quantitative data. This helps users to acquaint themselves with values in clusters or outliers. On accessing tooltip information, other points are "dimmed" to highlight this point and prevent distraction (see Figure 3).

Point selection is implemented by drawing an invisible point-in polygon on dragging the cursor across the canvas. Points within the polygon are colored black to show the users their selection. The structural and quantitative data bound to these points can be printed out and used with this or other tools. The selection can also be reset via a button to revise selections or work with other interactions. Clicking on a point updates the 3-dimenstional structure so that it corresponds to the same time point as the selected data point.

Filters – The user can filter the data with respect to all data columns. Data outside the filters are invisible. Filtered data maintain their position with respect to visible and invisible data points so as to not distort the spatial relationship between points and relationship between colors. Filters can also be reset.

All the above features are implemented using Javascript and D3.js[18] library.

Structural Data

Structural data is rendered in 3-dimensions using the three.js Javascript library. Each scene is created using spheres, which are colored by atom type, from position data in the trajectory file. This is the "space-filling model" of the molecule. Colors can be changed according to the user's preference from a dropdown menu populated dynamically with atom names from the structure file. A perspective

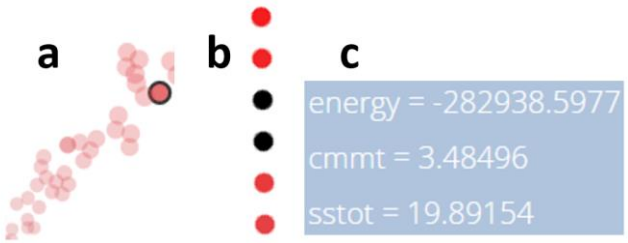

Figure 3. Mouse interactions enabled on the the scatter plot.(a) Point highlighting (b) Point selection (c) Tooltip information

camera is use to provide depth. This scene is animated to allow the user to rotate and zoom the structure to see features in all directions.

Coloring and Size – Colors of atoms follow the CPK coloring scheme. The size of atoms correspond to the ratio of their van der Waals radii, taking the hydrogen atom as a reference.

RESULTS

Using the techniques described above, my web-based visualization tool produced visualization as seen in Figure 2. This plot shows that the system broadly visited two states – visiting one state in the middle of the simulation and staying there for a short time. Having identified this, it would be useful for the user to see the structures (see Figure 4) in these states to assist in reasoning about the states that the system visited.

The user can also select points by dragging the mouse over and/or around points. The structural and quantitative data from these points can be written to file.

DISCUSSION

I conducted an informal usability study of this system by engaging domain researchers in my research laboratory. Being well-acquainted with line plots with defined axes, the users had a learning curve associated with using this tool. It was easy for them to identify clusters that were far apart from each other but clusters close together were confusing. This confusion arose from the color encoding of points which force the user to cluster similarly colored points together rather than proximity of points.

Users were quick to test different combinations of variables and filters which endorsed my aim of designing this as an exploratory visualization tool. Users appreciated the context provided by a 3D structure. The latency in rendering 3D structures and limited rendering options in my system made them prefer selecting points of interest and using standard software to reanalyze structures. They also appreciated the feature of exporting data of choice so that they can send them to other people that might be interested in this data.

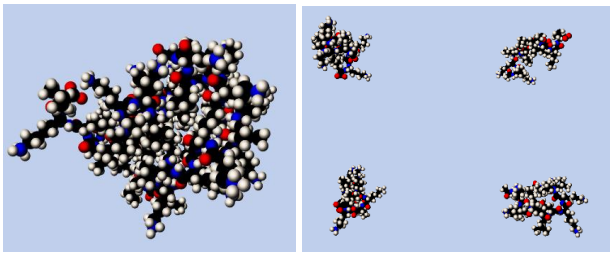

Figure 4. Interactive 3D rendering of molecules in different states with CPK coloring of atoms and sphere size defined by van der Waals radii

The earlier version of the graph has lines and fisheye zooming but these features were deleted because users found that they were messy and distracting from other useful interactions, respectively

Users were more engaged in this tool – since they used many interaction features - than a standard line plot or a separate 3D rendering. Since all dropdown menus are dynamically generated from input data, users could upload their own data and thereby be more engaged in the exploratory data analysis process.

FUTURE WORK

Through the usability study it is evident that my tool can be improved by using faster 3D rendering algorithms, and advanced rendering and querying options at par with standard molecular modelling toolkits. The 2D plotting platform can be expanded to include different plot types. Learning from ManyEyes[13], adding space for comments or annotations can encourage group members to collaborate virtually. Including options to perform calculations on given quantitative data would also improve this tool. Rendering images as PNG for sharing can also be included.

REFERENCES

1. Benjamin Bach, Pierre Dragicevic, Daniel Archambault, Christophe Hurter, and Sheelagh Carpendale. 2014. A Review of Temporal Data Visualizations Based on Space-Time Cube Operations. *Eurographics Conference on Visualization*.
2. Benjamin Bach, Conglei Shi, Nicolas Heulot, Tara Madhyastha, Tom Grabowski, and Pierre Dragicevic. 2016. Time Curves: Folding Time to Visualize Patterns of Temporal Evolution in Data. *IEEE Transactions on Visualization and Computer Graphics* 22, 1: 559–568. http://doi.org/10.1109/TVCG.2015.2467851
3. Marco Di Benedetto, Federico Ponchio, Fabio Ganovelli, and Roberto Scopigno. 2010. SpiderGL. *Proceedings of the 15th International Conference on Web 3D Technology - Web3D '10*, ACM Press, 165. http://doi.org/10.1145/1836049.1836075

4. Christian Collberg, Stephen Kobourov, Jasvir Nagra, Jacob Pitts, and Kevin Wampler. 2003. A system for graph-based visualization of the evolution of software. *Proceedings of the 2003 ACM symposium on Software visualization - SoftVis '03*, ACM Press, 77. http://doi.org/10.1145/774833.774844
5. BY Eadweard Muybridge. AN ELECTRO-PHOTOGRAPHIC INVESTIGATION OF CONSECUTIVE PHASES OF ANIMAL MOVEMENTS.
6. Tanja Falkowski, Jorg Bartelheimer, and Myra Spiliopoulou. 2006. Mining and Visualizing the Evolution of Subgroups in Social Networks. *2006 IEEE/WIC/ACM International Conference on Web Intelligence (WI 2006 Main Conference Proceedings)(WI'06)*, IEEE, 52–58. http://doi.org/10.1109/WI.2006.118
7. Hongjian Li, Kwong-Sak Leung, Takanori Nakane, et al. 2014. iview: an interactive WebGL visualizer for protein-ligand complex. *BMC Bioinformatics* 15, 1: 56. http://doi.org/10.1186/1471-2105-15-56
8. Zhicheng Liu, Shamkant B. Navathe, and John T. Stasko. 2011. Network-based visual analysis of tabular data. *2011 IEEE Conference on Visual Analytics Science and Technology (VAST)*, IEEE, 41–50. http://doi.org/10.1109/VAST.2011.6102440
9. Jean-Baptiste Pettit, John C Marioni, CL Luengo Hendriks, et al. 2013. bioWeb3D: an online webGL 3D data visualisation tool. *BMC Bioinformatics* 14, 1: 185. http://doi.org/10.1186/1471-2105-14-185
10. B. Shneiderman and A. Aris. 2006. Network Visualization by Semantic Substrates. *IEEE Transactions on Visualization and Computer Graphics* 12, 5: 733–740. http://doi.org/10.1109/TVCG.2006.166
11. Alice Thudt, Dominikus Baur, and Sheelagh Carpendale. 2013. *Visits: A Spatiotemporal Visualization of Location Histories*. The Eurographics Association.
12. Fernanda B. Viégas, Martin Wattenberg, and Kushal Dave. 2004. Studying cooperation and conflict between authors with *history flow* visualizations. *Proceedings of the 2004 conference on Human factors in computing systems - CHI '04*, ACM Press, 575–582. http://doi.org/10.1145/985692.985765
13. Fernanda B. Viegas, Martin Wattenberg, Frank van Ham, Jesse Kriss, and Matt McKeon. 2007. ManyEyes: a Site for Visualization at Internet Scale. *IEEE Transactions on Visualization and Computer Graphics* 13, 6: 1121–1128. http://doi.org/10.1109/TVCG.2007.70577
14. JSmol. Retrieved from http://wiki.jmol.org/index.php/JSmol
15. WebGL. Retrieved from https://www.khronos.org/webgl/
16. ChemDoodle Web Components. Retrieved from https://web.chemdoodle.com/
17. Strokes Order Project. Retrieved from https://commons.wikimedia.org/wiki/Commons:Stroke_Order_Project
18. D3.js. Retrieved from https://github.com/d3/d3/wiki