MDAnalysis: A Python Package for the Rapid Analysis of Molecular Dynamics Simulations

Submission subgroup: **Mini Symposium: Biology and Medicine**

Abstract

MDAnalysis (http://mdanalysis.org) is an object-oriented library for structural and temporal analysis of molecular dynamics (MD) simulation trajectories and individual protein structures. MD simulations of biological molecules have become an important tool to elucidate the relationship between molecular structure and physiological function. Simulations are performed with highly optimized software packages on HPC resources but most codes generate output trajectories in their own formats so that the development of new trajectory analysis algorithms is confined to specific user communities and widespread adoption and further development is delayed. The MDAnalysis library addresses this problem by abstracting access to the raw simulation data and presenting a uniform object-oriented Python interface to the user. It thus enables users to rapidly write code that is portable and immediately usable in virtually all biomolecular simulation communities. The user interface and modular design work equally well in complex scripted workflows, as foundations for other packages, and for interactive and rapid prototyping work in IPython / Jupyter notebooks, especially together with molecular visualization provided by nglview and time series analysis with pandas. MDAnalysis is written in Python and Cython and uses NumPy arrays for easy interoperability with the wider scientific Python ecosystem. It is widely used and forms the foundation for more specialized biomolecular simulation tools. MDAnalysis is available under the GNU General Public License v2.

Long Description

BACKGROUND

Molecular dynamics (MD) simulations of biological molecules have become an important tool to elucidate the relationship between molecular structure and physiological function. Simulations are performed with highly optimized software packages on HPC resources but most codes generate output trajectories in their own formats so that the development of new trajectory analysis algorithms is confined to specific user communities and widespread adoption and further development is delayed. Typical trajectory sizes range from gigabytes to terabytes so it is typically not feasible to convert trajectories into a range of different formats just to use a tool that requires this specific form. Instead, a framework is required that provides a common interface to raw simulation data.

RESULTS

The MDAnalysis library [1] addresses this problem by abstracting access to the raw simulation data and presenting a uniform object-oriented Python interface to the user. MDAnalysis is written in Python and Cython and uses NumPy arrays for easy interoperability with the wider scientific Python ecosystem. It currently supports more than 25 different file formats and covers the vast majority of data formats that are used in the biomolecular simulation community, including the formats required and produced by the most popular packages NAMD, Amber, Gromacs, CHARMM, LAMMPS, DL_POLY, HOOMD. The user interface provides "physics-based" abstractions (e.g. "atoms", "bonds", "molecules") of the data that can be easily manipulated by the user. It hides the complexity of accessing data and frees the user from having to implement the details of different trajectory and topology file formats (which by themselves are often only poorly documented and just adhere to certain "community expectations" that can be difficult to understand for outsiders).

The user interface and modular design work equally well in complex scripted workflows, as foundations for other packages [2] [3] [4], and for interactive and rapid prototyping work in IPython/Jupyter notebooks, especially together with molecular visualization provided by nglview [5] and time series analysis with pandas [6]. Since the original publication [1], improvements in speed and data structures make it now possible to work with terabyte-sized trajectories containing up to ~10 million particles. MDAnalysis also comes with specialized analysis classes in the MDAnalysis.analysis module that are unique to MDAnalysis such as the LeafletFinder graph-based algorithm for the analysis of lipid bilayers [1] or the Path Similarity Analysis for the quantitative comparison of macromolecular conformational changes [2].

MDAnalysis is available in source form under the GNU General Public License v2 from GitHub https://github.com/MDAnalysis/mdanalysis and PyPi; conda packages are also available. The documentation is extensive http://docs.mdanalysis.org including an introductory tutorial http://www.mdanalysis.org/MDAnalysisTutorial/ and a very friendly and welcoming user and developer community.

CONCLUSIONS

MDAnalysis provides a uniform interface to simulation data, which comes in a bewildering array of formats. It enables users to rapidly write code that is portable and immediately usable in virtually all biomolecular simulation communities. It has a very active international developer community with researchers that are expert developers and users of a wide range of simulation codes. MDAnalysis is widely used (the original paper [1] is cited more than 180 times) and forms the foundation for more specialized biomolecular simulation tools [2] [3] [4]. Ongoing and future developments will improve performance further, introduce transparent parallelisation schemes to utilize multi-core systems efficiently, and interface with the SPIDAL library [7] for high performance data analytics algorithms.

BIOGRAPHY

Oliver Beckstein is an Assistant Professor in the Department of Physics at Arizona State University and leads the Computational Biophysics research group http://becksteinlab.physics.asu.edu/. He is one of the founders of MDAnalysis (together with Naveen Michaud-Agrawal and Elizabeth Denning) and an active core developer. He obtained his undergraduate degree in physics from the University of Erlangen-Nuremberg, Germany, and his DPhil in Biochemistry from the University of Oxford, UK. He was a Junior Research Fellow at Merton College, Oxford, and a postdoctoral fellow at Johns Hopkins University. Research in his group centers on using computational approaches to elucidate molecular mechanisms of transport across the cell membrane and development of new analysis and sampling algorithms. The lab is also part of a large NSF-funded cooperative effort (http://spidal.org/) to create middleware and analytics libraries for data science at large scale on high-performance computing systems. Some of the algorithms for efficiently analyzing large simulations will be implemented as part of the MDAnalysis package.

1(1, 2, 3, 4)

N. Michaud-Agrawal, E. J. Denning, T. B. Woolf, and O. Beckstein. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. J Comp Chem, 32:2319–2327, 2011. doi: 10.1002/jcc.21787. URL http://doi.org/10.1002/jcc.21787

2(1, 2, 3)

S.

L. Seyler, A. Kumar, M. F. Thorpe, and O. Beckstein. Path similarity analysis: A method for quantifying macromolecular pathways. PLoS Comput Biol, 11(10):e1004568, 10 2015. doi: 10.1371/journal.pcbi. 1004568. URL http://dx.doi.org/10.1371%2Fjournal.pcbi.1004568

3(1, 2)

M. Tiberti, E. Papaleo, T. Bengtsen, W. Boomsma, and K. Lindorff-Larsen. ENCORE: Software for quantitative ensemble comparison. PLoS Comput Biol, 11(10):e1004415, 10 2015. doi: 10.1371/journal.pcbi. 1004415. URL http://dx.doi.org/10.1371%2Fjournal.pcbi.1004415

4(1, 2) E. Somogyi, A. A. Mansour, and P. J. Ortoleva. ProtoMD: A prototyping toolkit for multiscale molecular dynamics. Computer Physics Communications, 202:337 – 350, 2016. ISSN 0010-4655. doi: 10.1016/j.cpc. URL http://www.sciencedirect.com/science/article/pii/S0010465516300030

5 https://github.com/arose/nglview

6 http://pandas.pydata.org/

7 http://spidal.org