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

Richard J. Gowers[||][**][†], Max Linke[‡‡][†], Jonathan Barnoud[§][†], Tyler J. E. Reddy[‡], Manuel N. Melo[§], Sean L. Seyler[¶], Jan Domanski[‡], David L. Dotson[¶], Sébastien Buchoux[††], Ian M. Kenney[¶], Oliver Beckstein[¶][*]

DRAFT

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.

Index Terms—molecular dynamics simulations, science, chemistry, physics, biology

Introduction

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 [MADWB11] 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 [SKTB15], [TPB+15], [SMO16], 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. Since the original publication [MADWB11], 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 [MADWB11] or the Path Similarity Analysis for the quantitative comparison of macromolecular conformational changes [SKTB15].

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.

† *These authors contributed equally.*
|| *University of Manchester, Manchester, UK*
** *University of Edinburgh, Edinburgh, UK*
‡‡ *Max Planck Institut für Biophysik, Frankfurt, Germany*
§ *University of Groningen, Groningen, The Netherlands*
‡ *University of Oxford, Oxford, UK*
¶ *Arizona State University, Tempe, Arizona, USA*
†† *Université de Picardie Jules Verne, Amiens, France*
* *Corresponding author: oliver.beckstein@asu.edu*

Copyright © 2010 Richard J. Gowers et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

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 [MADWB11] has been cited more than 180 times) and forms the foundation for more specialized biomolecular simulation tools. Ongoing and future developments will improve performance further, introduce transparent parallelisation schemes to utilize multi-core systems efficiently, and interface with the SPIDAL library for high performance data analytics algorithms.

References

[MADWB11] Naveen Michaud-Agrawal, Elizabeth Jane Denning, Thomas B. Woolf, and Oliver Beckstein. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *J Comp Chem*, 32:2319–2327, 2011.

[SKTB15] Sean L. Seyler, Avishek Kumar, M. F. Thorpe, and Oliver Beckstein. Path similarity analysis: A method for quantifying macromolecular pathways. *PLoS Comput Biol*, 11(10):e1004568, 10 2015.

[SMO16] Endre Somogyi, Andrew Abi Mansour, and Peter J. Ortoleva. ProtoMD: A prototyping toolkit for multiscale molecular dynamics. *Computer Physics Communications*, 202:337 – 350, 2016.

[TPB$^+$15] Matteo Tiberti, Elena Papaleo, Tone Bengtsen, Wouter Boomsma, and Kresten Lindorff-Larsen. ENCORE: Software for quantitative ensemble comparison. *PLoS Comput Biol*, 11(10):e1004415, 10 2015.

DRAFT