




Molearn: a Python package streamlining the design of generative models of biomolecular dynamics

Samuel C. Musson ¹ and Matteo T. Degiacomi ¹✉

¹ Department of Physics, Durham University, United Kingdom ✉ Corresponding author

DOI: [10.21105/joss.05523](https://doi.org/10.21105/joss.05523)

Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Summary

We present `molearn`, a Python package streamlining the implementation of machine learning models dedicated to the generation of protein conformations from example data obtained via experiment or molecular simulation.

Editor: Richard Gowers 

Reviewers:

- [@rmeli](#)
- [@JoaoRodrigues](#)

Submitted: 17 May 2023

Published: 05 September 2023

License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).

Statement of need

Most biological mechanisms directly involve proteins. The specific task each of these biopolymers carries out is linked to their three-dimensional shape, enabling them to bind to designated binding partners such as small molecules, ions, or other biopolymers. Crucially though, biomolecules are flexible and so are continuously jostling and reconfiguring due to Brownian motion. Thus, their function emerges from characteristic conformational dynamics. Characterising the structure and dynamics of biomolecules at the atomic level provides us with a fundamental understanding of the mechanisms underpinning life and is the first step in numerous technological applications. Progress in these areas has been spearheaded by the development of a diverse palette of dedicated experimental techniques (Dobson, 2019). Unfortunately, none is singlehandedly capable of routinely reporting on the full fine structure of biomolecular conformational spaces. As such, our understanding of life at the molecular level is inherently biased by the techniques we adopt to observe it (Marsh & Teichmann, 2015). Molecular dynamics (MD) simulations yield atomistic insights into the conformational landscape of biomolecules, complementing and extending data gathered experimentally. MD simulations estimate the true conformational landscape of biomolecules by iteratively generating new conformers based on an initial, known atomic arrangement and physical models of atomic interactions. While MD enables obtaining key insight into biomolecular function, it is not a silver bullet: exhaustive sampling of processes such as folding or ligand binding usually lay beyond what can be routinely observed.

Generative Neural Networks have been shown to be effective predictors of a protein's 3D structure from its sequence (Baek et al., 2021; Jumper et al., 2021). Several efforts have also demonstrated that a neural network trained with MD conformers can learn a meaningful dimensionality reduction of the data, usable for reaction coordinate definition (Chen et al., 2018; Frassek et al., 2021), or driving conformational space sampling (Mehdi et al., 2022; Noé et al., 2019; Sidky et al., 2020). In this context, we have previously presented generative neural networks capable of producing protein conformations based on small pools of examples produced by MD (Degiacomi, 2019; Ramaswamy et al., 2021). The issue is that developing a Machine Learning model to study biomolecular dynamics is a lengthy process. This requires setting up means of transforming conformational space data into tensor data submittable to a model, as well as assessing a model's quality (e.g., in terms of their energy or according to structural descriptors).

Here we present `molearn`, a Python framework facilitating the implementation of generative

neural networks learning protein conformational spaces from examples obtained via experiments or MD simulations.

Package Description

Classes available provide support for the following tasks:

- *Data loading.* `mollearn` provides methods to parse molecular conformers using `biobox` (Rudden et al., 2022), and convert them into a PyTorch (Paszke et al., 2019) tensor format suitable for training.
- *Model design.* `mollearn` comes with a range of pre-implemented models, ready to be trained on any desired datasets or subclassed to create custom models.
- *Loss function definition.* While the classical loss function in a generative model typically builds upon a mean square error between input and output, here we provide the capability of directly interacting with the OpenMM molecular dynamics engine (Eastman et al., 2017). Specifically, we have implemented means of transferring PyTorch Tensor data directly into OpenMM's backend on GPU (without data transfer via the CPU). This enables quickly evaluating the energy of a generated model according to any force field accepted by OpenMM. This also enables directly running MD simulations with generated conformers while the model trains.
- *Model analysis.* Once a model is trained, it is important to gather metrics defining the quality of the protein structures it can generate. We provide tools to quickly quantify structure quality in terms of root mean square deviation between the coordinates of atoms in input and output, DOPE score (Shen & Sali, 2006), Ramachandran plot, and user-defined functions. Analysis data is returned in the form of numpy arrays (Harris et al., 2020), for ease of manipulation and plotting. We also provide a graphical user interface (GUI), enabling the visualisation of neural network latent space, and generation of interpolations between states visualised in an interactive 3D panel by a combination of MDAnalysis (Michaud-Agrawal et al., 2011) and NGLview (Nguyen et al., 2018).

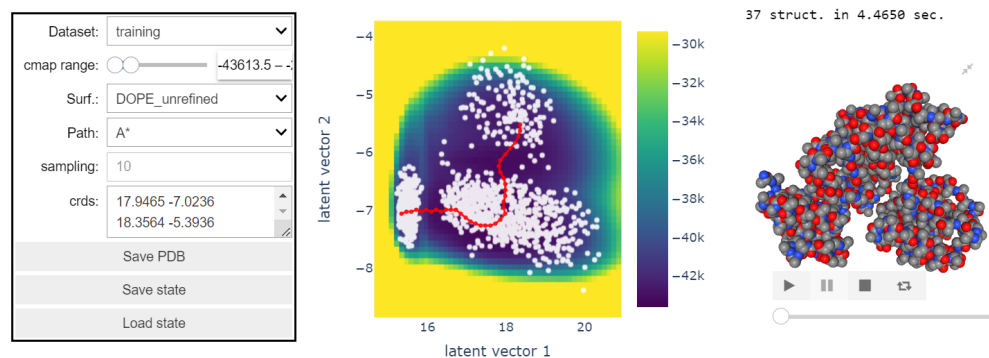


Figure 1: `mollearn` analysis tools include a graphical user interface, enabling the on-demand generation of protein conformations. The panel on the left controls how the neural network latent space is presented, the central panel is a Plotly interactive graph displaying the latent space, and the panel on the right is a 3D representation of an interpolation through the latent space supported by NGLview.

Usage

`mollearn` comes with a series of examples, usable as-is, to train and analyse a neural network. Tutorials on neural network analysis are also available, including a GUI to directly interact with a trained neural network (Figure 1). Results obtainable via `mollearn` are exemplified in (Ramaswamy et al., 2021). There, we designed and trained a 1D convolutional autoencoder against protein molecular dynamics simulation data. The neural network was trained via a loss

function directing the neural network to both faithfully reconstruct training data, and produce low-energy interpolations between them, whereby the internal energy of produced models is assessed according to the Amber ff14SB force field (Maier et al., 2015).

Acknowledgements

We thank Cameron Stewart, Ryan Zhu, and Marco Mattia for their feedback. Matteo T. Degiacomi acknowledges support from the Engineering and Physical Sciences Research Council (EP/P016499/1).

References

- Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D., Millán, C., Park, H., Adams, C., Glassman, C. R., DeGiovanni, A., Pereira, J. H., Rodrigues, A. V., Dijk, A. A. van, Ebrecht, A. C., ... Baker, D. (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, 373(6557), 871–876. <https://doi.org/10.1126/science.abj8754>
- Chen, W., Tan, A. R., & Ferguson, A. L. (2018). Collective variable discovery and enhanced sampling using autoencoders: Innovations in network architecture and error function design. *The Journal of Chemical Physics*, 149(7), 072312. <https://doi.org/10.1063/1.5023804>
- Degiacomi, M. T. (2019). Coupling Molecular Dynamics and Deep Learning to Mine Protein Conformational Space. *Structure*, 27(6), 1034–1040.e3. <https://doi.org/10.1016/j.str.2019.03.018>
- Dobson, C. M. (2019). Biophysical techniques in structural biology. *Annual Review of Biochemistry*, 88, 25–33. <https://doi.org/10.1146/annurev-biochem-013118-111947>
- Eastman, P., Swails, J., Chodera, J. D., McGibbon, R. T., Zhao, Y., Beauchamp, K. A., Wang, L.-P., Simmonett, A. C., Harrigan, M. P., Stern, C. D., & others. (2017). OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLoS Computational Biology*, 13(7), e1005659. <https://doi.org/10.1371/journal.pcbi.1005659>
- Frassek, M., Arjun, A., & Bolhuis, P. (2021). An extended autoencoder model for reaction coordinate discovery in rare event molecular dynamics datasets. *The Journal of Chemical Physics*, 155(6), 064103. <https://doi.org/10.1063/5.0058639>
- Harris, C. R., Millman, K. J., Van Der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., & others. (2020). Array programming with NumPy. *Nature*, 585(7825), 357–362. <https://doi.org/10.1038/s41586-020-2649-2>
- Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., ... Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873), 583–589. <https://doi.org/10.1038/s41586-021-03819-2>
- Maier, J. A., Martinez, C., Kasavajhala, K., Wickstrom, L., Hauser, K. E., & Simmerling, C. (2015). ff14SB: Improving the accuracy of protein side chain and backbone parameters from ff99SB. *Journal of Chemical Theory and Computation*, 11(8), 3696–3713. <https://doi.org/10.1021/acs.jctc.5b00255>
- Marsh, J. A., & Teichmann, S. A. (2015). Structure, Dynamics, Assembly, and Evolution of Protein Complexes. *Annual Review of Biochemistry*, 84(1), 551–575. <https://doi.org/10.1146/annurev-biochem-060614-034142>

- Mehdi, S., Wang, D., Pant, S., & Tiwary, P. (2022). Accelerating all-atom simulations and gaining mechanistic understanding of biophysical systems through state predictive information bottleneck. *Journal of Chemical Theory and Computation*, 18(5), 3231–3238. <https://doi.org/10.1021/acs.jctc.2c00058>
- Michaud-Agrawal, N., Denning, E. J., Woolf, T. B., & Beckstein, O. (2011). MDAAnalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry*, 32(10), 2319–2327. <https://doi.org/10.1002/jcc.21787>
- Nguyen, H., Case, D. A., & Rose, A. S. (2018). NGLview—interactive molecular graphics for jupyter notebooks. *Bioinformatics*, 34(7), 1241–1242. <https://doi.org/10.1093/bioinformatics/btx789>
- Noé, F., Olsson, S., Köhler, J., & Wu, H. (2019). Boltzmann generators: Sampling equilibrium states of many-body systems with deep learning. *Science*, 365(6457), eaaw1147. <https://doi.org/10.1126/science.aaw1147>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., & others. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32. <https://doi.org/10.48550/arXiv.1912.01703>
- Ramaswamy, V. K., Musson, S. C., Willcocks, C. G., & Degiacomi, M. T. (2021). Deep Learning Protein Conformational Space with Convolutions and Latent Interpolations. *Physical Review X*, 11(1), 011052. <https://doi.org/10.1103/PhysRevX.11.011052>
- Rudden, L. S., Musson, S. C., Benesch, J. L., & Degiacomi, M. T. (2022). Biobox: A toolbox for biomolecular modelling. *Bioinformatics*, 38(4), 1149. <https://doi.org/10.1093/bioinformatics/btab785>
- Shen, M., & Sali, A. (2006). Statistical potential for assessment and prediction of protein structures. *Protein Science*, 15(11), 2507–2524. <https://doi.org/10.1110/ps.062416606>
- Sidky, H., Chen, W., & Ferguson, A. L. (2020). Molecular latent space simulators. *Chemical Science*, 11(35), 9459–9467. <https://doi.org/10.1039/d0sc03635h>