Making it rain: cloud-based molecular simulations for everyone

Pablo R. Arantes,[*,†,⊥] Marcelo D. Polêto,[*,‡] Conrado Pedebos,[*,¶,§] and Rodrigo Ligabue-Braun[*,‖]

†*Department of Bioengineering, University of California, Riverside, CA 92521, United States*

‡*Department of Biochemistry, Virginia Tech, Blacksburg, VA 24061, United States*

¶*School of Chemistry, University of Southampton, Highfield Campus, Southampton SO17 1BJ, United Kingdom*

§*Present Address: Department of Biochemistry, University of Oxford, Oxford OX1 3QU, United Kingdom*

‖*Department of Pharmacosciences, Federal University of Health Sciences of Proto ALegre (UFCSPA), Porto Alegre 90050-170, RS, Brazil.*

⊥*Corresponding author*

E-mail: pabloa@ucr.edu; mdpoleto@vt.edu; conrado.pedebos@bioch.ox.ac.uk; rodrigolb@ufcspa.edu.br

Abstract

We present a user-friendly front-end for running molecular dynamics (MD) simulations using OpenMM toolkit on the Google Colab framework. Our goals are: 1) to highlight the usage of a cloud-computing scheme for educational purposes for a hands-on approach when learning MD simulations and 2) to exemplify how low-income research

groups can perform MD simulations in the microsecond timescale. We hope this work facilitates teaching and learning of molecular simulation throughout the community.

Introduction

Reliable structural depictions of biomolecules are essential for understanding how the natural world works.[1] Experimental techniques for obtaining such structures are mainly represented by X-ray crystallography, nuclear magnetic resonance spectroscopy, and cryo-electron microscopy, with the latter facing an explosive increase in application.[2,3] In addition, computational methods for biomolecular modeling (particularly proteins) have reached an unprecedented quality, with AlphaFold2[4] and RoseTTAFold[5] being the most recent examples. All these advances, combined with the availability of the human structural proteome[6] usher the demand for dynamical, time-dependent methods for biological structure analysis that are essential for holistic inspections of protein function in the organism-level context.[7,8]

Molecular dynamics (MD) simulations, a classic physics approximation for describing atomic trajectories, is widespread among the methods for describing biomolecular motion.[9,10] These simulations, however, require progressively more computation power, and efforts are increasing to allow massive runs of MD.[10] Thus, one alternative to in-house computing facilities is the use of cloud computing for molecular modeling projects, especially due to its scalability, reliability, and lower costs.[11] However, the use of cloud computing requires from the user the ability to perform technical or tedious tasks, such as configuring remote computer nodes, installing the required software under specific conditions, and running calculations (by launching, monitoring, and terminating them).[12] Some propositions for facilitating the use of cloud computing for MD simulations have been made, including variations on AMBER,[13] CHARMM-GUI,[14] QwikMD,[15] and NAMD.[16] All these implementations, however, leave some intermediate task to be performed or need to be set up on a cloud structure by the end user. An exception is described by Engelberger et al.[17] which reported a collection of

Jupyter notebooks for educational purposes that performs phylogenetic analysis, molecular modeling, molecular docking, molecular dynamics, and coevolutionary analysis. Although those protocols were not focused on MD calculations, they greatly inspired the present work.

Considering the ever-growing need for faster, affordable computer resources for novice users, here we present a user-friendly front-end for running molecular dynamics simulations using OpenMM toolkit[18] on the Google Colab framework.[19,20] For educational purposes, the user is only required to provide minimal input files, such as an initial biomolecular tridimensional structure (a PDB file), and the following MD simulation tasks are facilitated from there on. Usual simulation analyses (e.g. RMSD, RMSF, PCA) are also implemented in an automated fashion. Despite its click-and-go appeal, the protocols presented here can be customized for other specific cases, while also allowing students to have a hands-on approach when learning MD simulations.

Google Colab framework and MD simulations

Google Colaboratory, or Colab for short, is a product from Google Research organization.[19,20] According to their own website,[19] "*Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.*" In practice, Colab uses virtual machines to execute Jupyter notebook codes on its server using GPUs and each notebook connection (a session) can run up to 12 hours in Colab free version, while the Colab paid version (Colab Pro) has a time limit of 24 hours per session. Currently, only the United States, Canada, Japan, Brazil, Germany, France, India, United Kingdom, and Thailand have access to Colab Pro. Moreover, Colab is equipped with Nvidia K80, T4, P4 and P100 GPUs and Colab Pro users have access to their fastest GPUs.

In terms of MD simulation, trajectories files can be very large and proper storage space is a fundamental feature for computational resources. While the temporary storage system

can be limited when connected to a Colab computing node, users are allowed to mount their personal Google Drive account so all output files are directly written to a more permanent storage system. Conversely, users can also use the Google Drive system to upload input files that can be used in their notebooks, which gives much flexibility to Colab computational structure. Google Drive free version allows users to store up to 15GB of data, while paid versions can store more than 2TB of data.

Understanding the potential use of such a computing environment and the educational importance of introducing MD simulation concepts to new users, we combined Google Colab framework and the OpenMM MD toolkit[18] in order to create notebooks containing entire workflows for MD simulations. Here, we report 3 notebooks (Figure 1): **A)** exemplifying a complete pipeline for MD simulations, from structure preparation to MD simulation and basic analyses; **B)** Running MD simulations using input files imported from a Google Drive folder; and **C)** combining molecular modeling techniques and MD simulations.

General Notebook structures

All workflows presented here follow the Jupyter Notebook structure[21] and have 2 initial configuration steps that should take less than 7 minutes. The first step is for software and dependency installation, which includes the OpenMM engine,[18] MDanalysis,[22] PyTraj,[23] Numpy,[24] Ambertools[25] and others. The second step is the mounting of the user's Google Drive filesystem in order to store output files. The remaining tasks are the usual ones observed in any MD simulation protocol, namely: topology generation, solvation and neutralization of a simulation box, equilibration run, production run and trajectories analyses. Users are allowed to modify the OpenMM code to adapt each of these steps as needed.

The main difference between the 3 notebooks presented here is the origin of the input file (i.e. the solute molecule). Notebook **A)** was designed to fetch a protein structure directly from the Protein Data Bank, to create its topology using AmberTools along with the actual simulation box itself containing solvent and neutralizing ions. Notebook **B)** was designed to

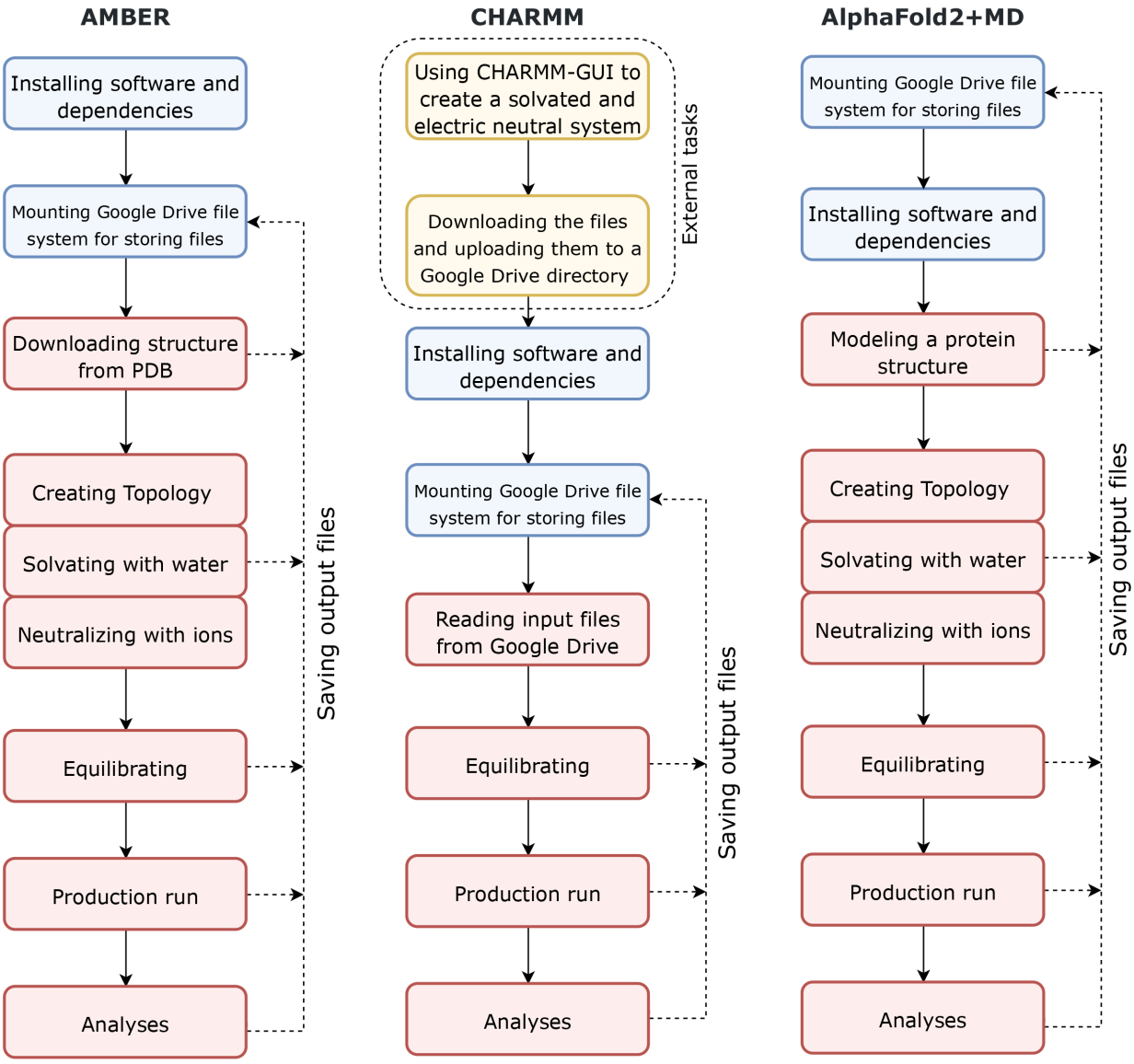

Figure 1: Workflow of the 3 presented Jupyter notebooks. Blue boxes are configuration steps required by the Colab environment, red boxes are actual MD related tasks and yellow boxes represent tasks that are external to the Jupyter notebook. During every MD related task, output files are directly written at the user's Google Drive storage filesystem.

show how users can provide their own input files for the OpenMM engine via Google Drive storage system. As demonstrated, inputs generated by CHARMM-GUI[14] server can be easily uploaded to a Google Drive folder, allowing the user to use them to start a simulation. Moreover, notebook **C)** was designed to demonstrate how molecular modeling techniques and MD simulations can be combined to generate structurally relevant information. In this case, we combined our MD notebook with the AlphaFold2 algorithm that was recently ported to Google Colab.[4,26] Together, these protocols are meant to cover most of the basic needs of MD approaches.

Usage examples

In order to show the feasibility of using such a cloud-based computational resource for research purposes, we carried out a 1 microsecond long simulation of lysozyme using the Amber FF19sb force field in explicit OPC water, totalizing 26,033 atoms in the simulation box. For this, we used a Colab Pro version, which allowed us to use P100 GPUs to yield an average simulation speed of ≈230 ns/day. Due to the limiting 12/24h job limit in Colab, jobs are expected to be terminated when they reach that time. In order to circumvent that limitation, we divided the entire trajectory in 50 strides of 20 ns each. At the end of each stride, OpenMM generates a .xml state file containing all atom's positions and velocities, which are read at the beginning of the next stride. All these steps can be easily defined by the user in all notebooks reported here and restart routines are automatically set up. Thus, in order to complete our 1 microsecond trajectory, we simply ran notebook **A)** 5 times, which took us 6 days of work. Structural analyses of our lysozyme simulation are shown in Figure 2 below.

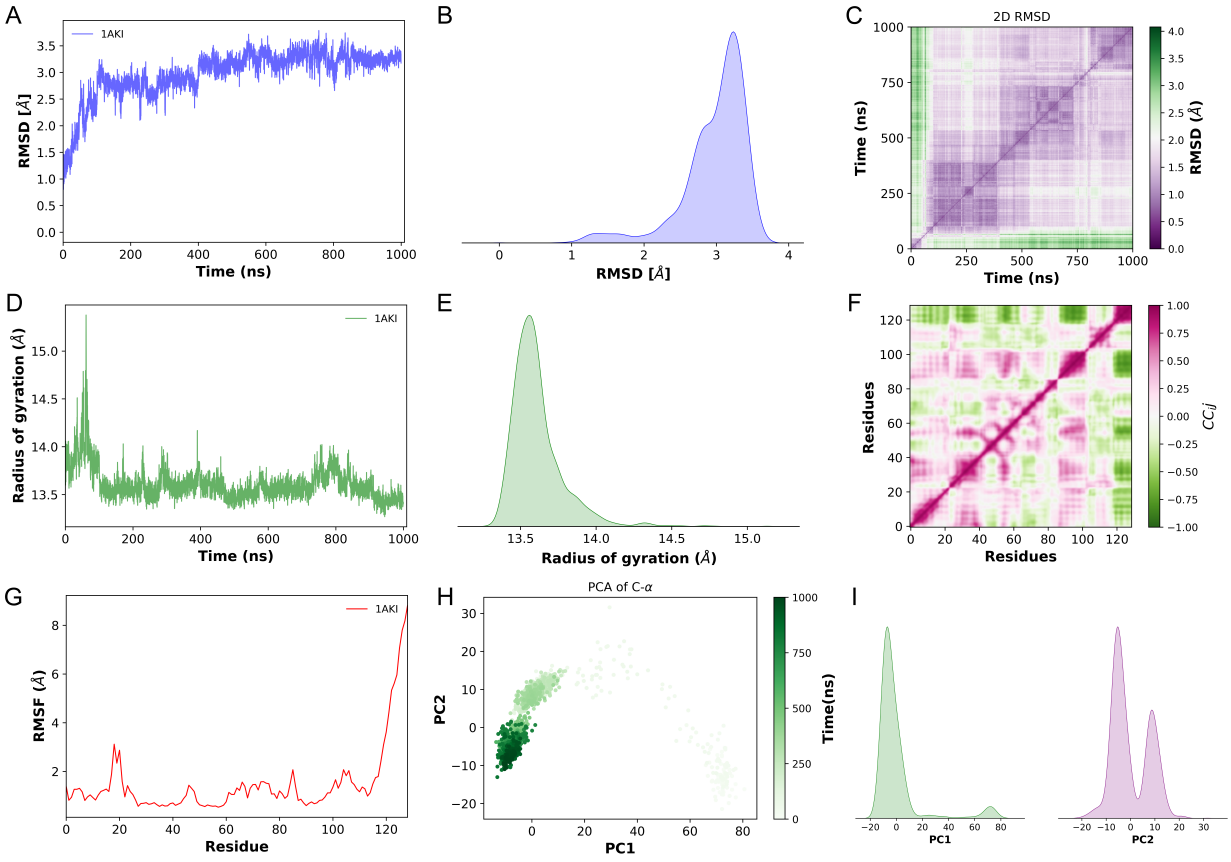

Figure 2: Structural analyses of MD simulations using PyTraj embedded in the notebooks: RMSD values over time (A), its distribution (B) and a 2D RMSD plot (C); radius of gyration over time (D) and its distribution (E); Cross-correlation analysis (F); RMSF per residue (G); Principal Component Analysis (H) and its eigenvector distribution (I).

Applicability

The approach shown here is specially suited for educational purposes. Professors can share the notebook with students and each one can run an entire MD simulation, from system preparation to production runs and analyses in one day. Such hands-on classes do not require local compilation of MD software since all dependencies are installed directly at the computing node via Anaconda module. The usage of OpenMM engine and its well known compatibility with GPU processing is especially well suited for the Colab computing infrastructure.

Another benefit is that students are not required to have much previous computational background. In fact, using the notebooks presented here would require only a Google Drive account for each student and the appropriate storage space on it. An important feature we made sure was present is the structure/trajectory visualization that is embedded in the notebooks. Students can also choose to see, study and edit the OpenMM code behind each task.

Moreover, considering that we were able to use the Colab structure to produce 1 microsecond long MD simulations at a relatively low cost, this approach could be used for underprivileged investigators around the world to conduct their research. In order to demonstrate this, we evaluated both Pro and Free versions of Google Colab in terms of MD simulation speed of system of different sizes, namely PDB ID 7CI3 (18K atoms), 6ZCT (25K atoms), 7AEH (46K atoms), 6VYO (66K atoms) and 7BTF (195K atoms). For these simulations, we solvated the systems using explicit OPC water model and used a timestep of 2 femtosecond for production simulations. We used a Tesla T4 GPU for our Free version calculations and a Tesla P100-PCIE GPU for our Pro Calculations. Performances for each system are shown in Figure 3. Assuming no data loss due to resubmissions and only one active session per user, one could produce nearly 1 microsecond long trajectories for systems with ≈200K using the Pro version for ≈1 month. A cost-free alternative would require ≈50 days for the same system. Despite the necessity of ressubmitting a job 1-2 times a day, we believe useful

research can be done using this approach.

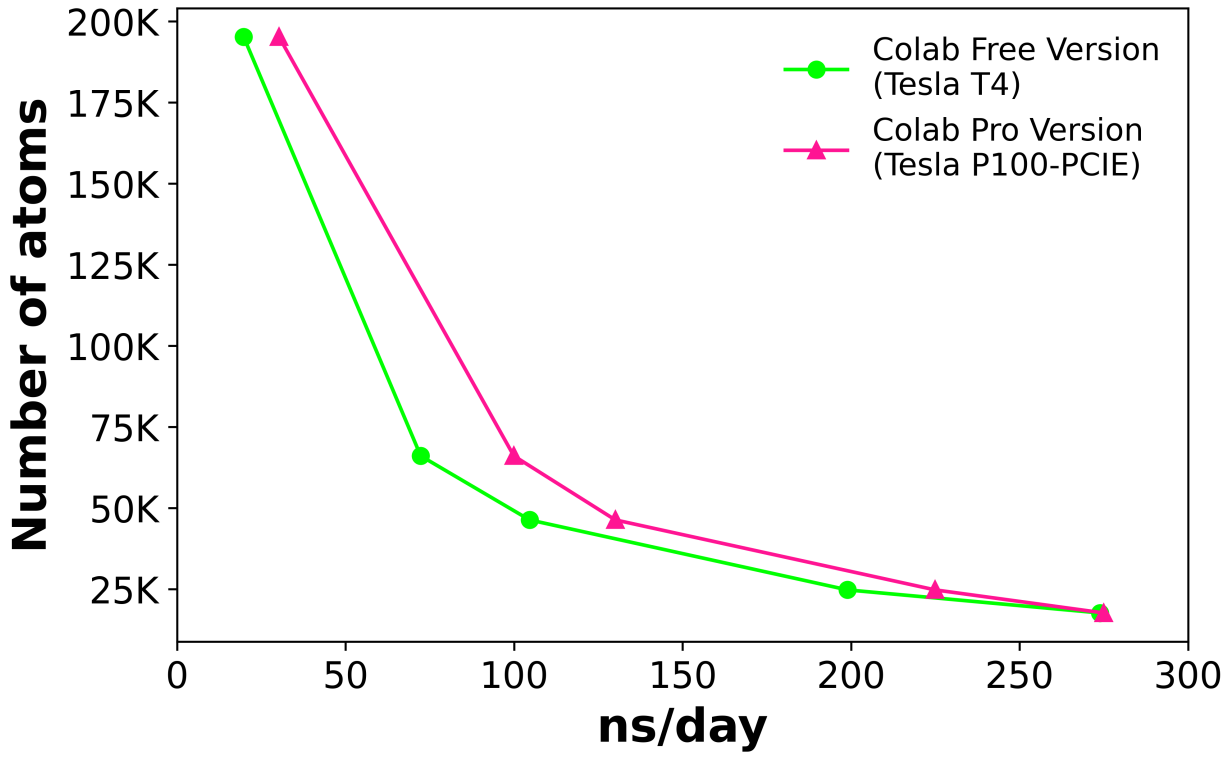

Figure 3: Simulation speed of multiple systems using Colab Pro (magenta) and Free (green) versions. Fully solvated systems simulated, from left to right: PDB ID 7BTF (195K atoms), 6VYO (66K atoms), 7AEH (46K atoms), 6ZCT (25K atoms) and 7CI3 (18K atoms)

Conclusions

Here we present an affordable (cost-free) strategy for performing molecular dynamics simulations using cloud computing resources. It involves shareable, ready-to-use, customizable Jupyter notebooks that guide the end-user in running their calculations using the Google Colab services. We hope this work facilitates teaching and learning of molecular simulations, as well as allowing for low-income research groups to perform MD in the microsecond timescale.

Acknowledgement

We would like to thank the OpenMM team for developing an excellent and open source engine and the AlphaFold team for developing an excellent model and open sourcing the software. We also would like to thank Sergey Ovchinnikov, Milot Mirdita and Martin Steinegger for their fantastic ColabFold and David Koes for the excellent py3Dmol plugin.

Supporting Information Available

All notebooks presented here are freely and publicly available at `https://github.com/pablo-arantes/Making-it-rain`.

References

(1) Boodhun, N. Seeing is believing: structures and functions of biological molecules. *https://doi.org/10.2144/btn-2017-0123* **2018**, *64*, 143–146.

(2) Rout, M. P.; Sali, A. Principles for Integrative Structural Biology Studies. *Cell* **2019**, *177*, 1384–1403.

(3) Callaway, E. Revolutionary cryo-EM is taking over structural biology. *Nature* **2020**, *578*, 201.

(4) Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature 2021* **2021**, 1–11.

(5) Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **2021**, eabj8754.

(6) Tunyasuvunakool, K. et al. Highly accurate protein structure prediction for the human proteome. *Nature 2021* **2021**, 1–9.

(7) Bershtein, S.; Kleiner, D.; Mishmar, D. Predicting 3D protein structures in light of evolution. *Nature Ecology & Evolution 2021* **2021**, 1–4.

(8) Fleishman, S. J.; Horovitz, A. Extending the New Generation of Structure Predictors to Account for Dynamics and Allostery. *Journal of Molecular Biology* **2021**, 167007.

(9) Liu, X.; Shi, D.; Zhou, S.; Liu, H.; Liu, H.; Yao, X. Molecular dynamics simulations and novel drug discovery. *https://doi.org/10.1080/17460441.2018.1403419* **2017**, *13*, 23–37.

(10) Hollingsworth, S. A.; Dror, R. O. Molecular Dynamics Simulation for All. *Neuron* **2018**, *99*, 1129–1143.

(11) Ebejer, J.-P.; Fulle, S.; Morris, G. M.; Finn, P. W. The emerging role of cloud computing in molecular modelling. *Journal of Molecular Graphics and Modelling* **2013**, *44*, 177–187.

(12) Wong, A. K. L.; Goscinski, A. M. The Design and Implementation of the VMD Plugin for NAMD Simulations on the Amazon Cloud. *International Journal of Cloud Computing and Services Science (IJ-CLOSER)* **2012**, *1*.

(13) Purawat, S.; Ieong, P. U.; Malmstrom, R. D.; Chan, G. J.; Yeung, A. K.; Walker, R. C.; Altintas, I.; Amaro, R. E. A Kepler Workflow Tool for Reproducible AMBER GPU Molecular Dynamics. *Biophysical Journal* **2017**, *112*, 2469–2474.

(14) Jo, S.; Kim, T.; Iyer, V. G.; Im, W. CHARMM-GUI: A web-based graphical user interface for CHARMM. *Journal of Computational Chemistry* **2008**, *29*, 1859–1865.

(15) Ribeiro, J. V.; Bernardi, R. C.; Rudack, T.; Stone, J. E.; Phillips, J. C.; Freddolino, P. L.; Schulten, K. QwikMD - Integrative Molecular Dynamics Toolkit for Novices and Experts. *Scientific Reports 2016 6:1* **2016**, *6*, 1–14.

(16) Nicolas-Barreales, G.; Sujar, A.; Sanchez, A. A Web-Based Tool for Simulating Molecular Dynamics in Cloud Environments. *Electronics 2021, Vol. 10, Page 185* **2021**, *10*, 185.

(17) Engelberger, F.; Galaz-Davison, P.; Bravo, G.; Rivera, M.; Ramírez-Sarmiento, C. A. Developing and Implementing Cloud-Based Tutorials That Combine Bioinformatics Software, Interactive Coding, and Visualization Exercises for Distance Learning on Structural Bioinformatics. *Journal of Chemical Education* **2021**, *98*, 1801–1807.

(18) 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.; Wiewiora, R. P.; Brooks, B. R.; Pande, V. S. OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. *PLOS Computational Biology* **2017**, *13*, e1005659.

(19) Google Colaboratory. `https://colab.research.google.com/notebooks/intro.ipynb`.

(20) Bisong, E. Google Colaboratory. *Building Machine Learning and Deep Learning Models on Google Cloud Platform* **2019**, 59–64.

(21) Project Jupyter. `https://jupyter.org/`.

(22) Michaud-Agrawal, N.; Denning, E. J.; Woolf, T. B.; Beckstein, O. MDAnalysis: A toolkit for the analysis of molecular dynamics simulations. *Journal of Computational Chemistry* **2011**, *32*, 2319–2327.

(23) Roe, D. R.; Thomas E. Cheatham, I. PTRAJ and CPPTRAJ: Software for Processing and Analysis of Molecular Dynamics Trajectory Data. *Journal of Chemical Theory and Computation* **2013**, *9*, 3084–3095.

(24) Harris, C. R. et al. Array programming with NumPy. *Nature* **2020**, *585*, 357–362.

(25) Case, D. A.; et al., `https://ambermd.org/AmberTools.php`.

(26) Ovchinnikov, S.; Mirdita, M.; Steinegger, M. ColabFold - Making Protein folding accessible to all via Google Colab. **2021**,