The Journal of Open Source Education

Course Materials for an Introduction to Computational Chemistry Techniques

Sarah Victoria Stewart [1,2], **Hannah Pollak** [1], **Timothy J. Spankie** [1,3,4], **Audrey Ngambia** [1], **Angela Chitzanidi**[5], **and Valentina Erastova** [1,2]

1 School of Chemistry, University of Edinburgh, Joseph Black Building, David Brewster Road, Edinburgh EH9 3FJ, United Kingdom **2** UK Centre for Astrobiology, School of Physics and Astronomy, University of Edinburgh, James Clerk Maxwell Building, Peter Guthrie Tait Road, Edinburgh EH9 3FD, United Kingdom **3** Computational Biology, School of Life Sciences, University of Dundee, Dow Street, Dundee DD1 5EH, United Kingdom **4** Biological Chemistry and Drug Discovery, School of Life Sciences, University of Dundee, Dow Street, Dundee DD1 5EH, United Kingdom **5** Research Services, Information Services, University of Edinburgh, Argyle House, 3 Lady Lawson Street, Edinburgh EH3 9DR, United Kingdom

DOI: 10.21105/jose.00299

Software

- Review
- Repository
- Archive

Submitted: 29 May 2025
Published: 29 September 2025

Summary

An Introduction to Computational Chemistry Techniques is an open educational resource originally developed for postgraduate taught masters chemistry students at the University of Edinburgh to introduce practical aspects of molecular dynamics simulations. Aimed at students with no prior experience in programming or computational chemistry, this module is delivered over a month as a series of three 3-hour workshops in a computer-lab setting, and a final session dedicated to individual projects.

The module introduces foundational concepts in Unix command-line navigation, setting up, running and analysing molecular simulations, using high-performance computing systems. The material is provided to the students through virtual machines with the content delivered via HTML-based guides, opened on local web browsers, and open-source tools. Students work through materials at their own pace, supported by demonstrators. Hosted on GitHub, this resource promotes the reuse and adaptation of accessible, computation-rich learning materials for chemistry education worldwide.

Statement of Need

Computational techniques are integral to modern chemical research, enabling precise simulation of chemical systems and prediction of molecular properties. Molecular dynamics (MD) simulations facilitate the study of complex systems, such as protein complexes, biological membranes, and material interfaces, accelerating advancements in drugs, materials, and technologies. Despite its importance, many chemistry curricula still lack hands-on computational training (Grushow & Reeves, 2019). Without foundational skills in Linux, high-performance computing (HPC), and simulation software, postgraduate chemistry students may struggle to engage with computational literature or methods. This module addresses this gap by providing practical MD simulation training in an accessible format, empowering students to perform computational research independently. By sharing these resources openly, we aim to support global educators in adopting or adapting computationally-enabled teaching in chemistry and related fields. Feedback from students highlights that the learners found the material engaging, recognised the importance of the skills being taught, and felt they had gained valuable and sufficient knowledge aligned with the learning objectives.

Content and Instructional Design

The course comprises three 3-hour-long workshops, run weekly. Each session begins with a short lecture to contextualise the material. The remainder of the session is set for independent work using HTML guides with demonstrators available. The students are also supported by weekly drop-in sessions. A final workshop session is used to introduce individual projects, assessment criteria and allow students to get started. The students have two weeks to work on the project and submit a summative report. (Note, we are unable to share the individual project materials.) The course is designed for taught postgraduate masters students taking degrees in 'Materials Chemistry', 'Analytical Chemistry', and 'Medicinal and Biological Chemistry', and therefore the content for the third workshop and individual projects is adjusted to suit these diverse backgrounds.

Module Goals

The module aims to equip students with the practical skills necessary to perform meaningful molecular dynamics simulations, including:

- Understand the practical aspects of molecular simulations.
- Use basic command-line interfaces and high-performance computing resources.
- Operate common computational chemistry packages to tackle real chemical problems.
- Prepare systems for molecular dynamics simulations and troubleshoot the set-up, simulations and analysis steps.
- Understand the limitations of the computational chemistry techniques used.
- Report the methodology and observations in a condensed written format.
- Perform group work, encouraged and developed through the practicals.

Session 1

The first session is an *"Introduction to Linux and command-line"*. This workshop is primarily based on material from Software Carpentry ("The Unix Shell" (Wilson et al., 2019), "Using Shell with HPC" (HPC Carpentry development team, 2021), and "Introduction to HPC" (Koch et al., 2025)). At the start of the workshop, a lecture introduces the course, the motivation, as well as explaining the nature of independent learning and setting up the assessment deadlines. The lecture also overviews computational techniques in chemistry, detailing when each may be beneficial. This is followed by a walkthrough on how to connect to the virtual machines used to run the course (details in the Computational Resources section).

The hands-on portion of the workshop is divided into seven parts. The exercises in six parts introduce the Unix shell and simple Bash commands — such as navigating directories, manipulating files, and writing scripts. Unlike the Software Carpentry model, we teach Vim as the text editor due to its advanced functionality (such as quickly navigating through files). The seventh part introduces high-performance computing, focusing on how to interact with the University of Edinburgh's Eddie compute cluster (University of Edinburgh, n.d.), which the students will use in subsequent sessions. Only the first three parts (working with files and directories and using HPC) are required to be completed, the other parts are given for the more keen or advanced students.

Session 2

The second session, *"Introduction to molecular dynamics simulations"*, guides students through the process of running simple MD simulations using GROMACS. This workshop is adapted from a GROMACS tutorial by Bergh, Majdolhosseini and Villa (Bergh et al., 2025). A short lecture at the start of the session introduces theoretical background on molecular simulations, including the concepts of force fields, timesteps, thermodynamic ensembles (e.g., isothermal-isobaric and canonical), periodic boundary conditions, system preparation and resource requirements for simulation.

The practical component uses a protein-in-water system (allowing students to choose from a selection of small protein structures) to introduce students to key stages of a simulation workflow:

- Pre-processing: Converting input files (e.g., PDB structures) into GROMACS formats, generating topologies, defining the simulation box, and adding water and ions.
- Simulation Execution: Performing energy minimisation followed by equilibration in the canonical (NVT) and isothermal-isobaric (NPT) ensembles.
- Post-processing and Analysis: Using built-in GROMACS tools to extract thermodynamic quantities (e.g., temperature, pressure, potential energy), trajectory information, and structural properties, such as root-mean-square deviation (RMSD) and radius of gyration.

Students visualise the structure and dynamics using VMD (Humphrey et al., 1996) and plot analysed data using Xmgrace (Turner, 2005). Students are prompted to think critically about their simulation results, discussing with demonstrators.

Session 3

The third session builds upon the developed skills to perform advanced simulations that are domain-specific to the three masters programs.

For Medicinal and Biological Chemistry students, this session applies molecular dynamics to a biologically relevant protein–ligand system. A short lecture introduces protein structure and the requirements for modelling them. Students prepare a dimer system as before, with a focus on handling incomplete PDB files and selecting suitable force fields. They perform energy minimisation, equilibration, and production runs using GROMACS, with analysis focusing on system stability and ligand binding through interpreting RMSD and interaction metrics from trajectories. This session and lecture materials have been partially adapted from BioSim Analysis Workshop (Degiacomi et al., 2025).

For Materials Chemistry and Analytical Chemistry students, the focus of the session is on simulating a clay–organic interface, relevant to materials and environmental sciences. A short lecture introduces layered minerals, interfacial interactions, and their applications in catalysis, environmental remediation, and nanotechnology. Students prepare a hydrated clay system containing organic guest molecules. Hands-on tasks include setting up a 2D layered system across the periodic boundary, modifying topology files, solvating and charge-balancing the system, running NPT simulation in semi-isotropic ensemble to account for the layered system and ensuring simulation stability. Students visualise structures using VMD and analyse layer spacing, molecular diffusion, and guest–host interactions with GROMACS tools.

Session 4

The final session introduces the individual projects and guides students in getting started on their independent simulations and final reports. A short lecture gives tips for scientific writing, provides an example report, highlights expectations for the project, and provides templates and submission guidelines. Additionally, students are introduced to Overleaf for scientific writing in LaTeX, and templates for the report are provided. The remainder of the session is used for hands-on progress toward simulation setup and report writing.

Please note that we are unable to share the materials for this part of the module, since it is used in a summative assessment. Instead, we are including a simplified HTML page linking the templates and an example report.

Assessment

The module was created as part of a Research Techniques course at the University of Edinburgh (SCQF 11, 20 credits), which trains postgraduate taught masters students in practical aspects of chemistry research. This module contributes 40% of the total grade for the course and is assessed in two ways. Overall comprehension of the first two sessions is assessed based on multiple-choice quizzes, each quiz being 5% of the module. The quizzes were run through LEARN, the University's Virtual Learning Environment. Students are first provided a formative quiz to familiarise themselves with how to operate the quiz and what type of knowledge is expected. When scoring over 80%, students can access the summative quiz. The summative quiz can only be attempted once. Both the formative and summative quizzes were neither timed nor invigilated. We share the formative quizzes within GitHub materials.

The largest part of the assessment (30% of the course mark) is based upon individual projects. The projects build upon what students have learned in Session 3, allowing them to set up their individual simulation systems, analyse them and produce a short report in the form of a mini-paper. The project assesses students' ability to independently design, execute, and analyse an MD simulation relevant to their degree program. The report is made to be concise, and the marking scheme rewards the ability to condense the information, create a quality methodology and produce appealing visuals in the results section.

Computational Resources

Each student was given access to an individual Linux virtual machine, created using the University of Edinburgh's Research Cloud Computing Service, Eleanor, deployed on the OpenStack platform (University of Edinburgh, n.d.). The machines were launched from an Ubuntu 22.04.3 LTS with Xfce Desktop Environment image, which has been saved to be re-used for future instances of the course. The machines had GROMACS 2024.4 (Abraham et al., 2015), VMD 1.9.4 (Humphrey et al., 1996) and Xmgrace 5.1.25 (Turner, 2005) software installed. Network storage was also mounted on each machine to enable access to the course materials and the transfer of outputs off the machine. Provisioned with 4 virtual CPU cores, 8GB RAM, and 80GB disk, these resources can easily be increased if required. Local accounts with randomly generated passwords were created for the students on the machines. Each student accessed their assigned machine through Remote Desktop using their assigned username and password.

The resource-heavy simulations (i.e., equilibration and production runs) were performed on NVIDIA A100 20GB Multi-Instance GPUs (MIGs) on the University of Edinburgh's Research Compute Cluster, Eddie (University of Edinburgh, n.d.). The Eddie compute cluster uses the Altair Grid Engine Scheduler and the Rocky Linux 9 operating system. To enable shorter queuing times for the students, 10 MIGs were ringfenced for them for the duration of the course. The ringfenced MIGs were on a single compute node with 64 CPU cores and 768GB of system RAM. On Eddie, each student had access to a 2TB scratch area where they could perform the simulations, and used GROMACS 2024.4 with GPU support and Xmgrace 5.1.25 software.

The course materials are written using HTML, as it offers a portable and cross-browser-compatible framework for delivering accessible content to users with diverse needs. Its semantic structure supports screen readers and assistive technologies, while its integration with CSS enables flexible, user-responsive design. We have added a dyslexia-friendly font option, which can be toggled by a button on each page. The font can be changed by placing the desired font `.ttf` files in `Notes/fonts/`, and altering the font-including section of the relevant `.css` files.

Reuse, implementation and modification

The complete course is available on GitHub under an open license - CC-BY-4.0. Course slides, shell scripts, tutorials, and example data are all included. Implementation requires a computer lab with Unix, GROMACS, and HPC access, though adaptations for local desktops or cloud platforms (e.g., Google Colab) are feasible. The materials have been successfully adapted for undergraduate chemistry students joining our group to introduce MD simulations.

Conclusion

This module equips postgraduate students with practical computational chemistry skills, emphasising reproducibility and open science. It is particularly suited to courses seeking to integrate practical computational methods skills with experimental and theoretical chemistry teaching.

For those interested in learning more about simulating clay surfaces, we have also developed a workshop introducing the use of ClayCode (Pollak et al., 2024) software for set-up and simulation of realistic clay mineral systems.

Acknowledgements

We thank the additional demonstrators, Finlay Clark, Chenfeng Zhang and Ryan Zhu, who supported student learning and Rosa Brauer for feedback on the instructional materials. We also thank Mike Wallis for assistance with setting up the original computational resources, David August for organising the course this module was part of, and Matteo Degiacomi for providing protein lecture material for session 3. This work builds on contributions from Software Carpentry and GROMACS community tutorials. Student feedback mentioned here is based on informal feedback collected as part of routine course delivery and not as part of a formal research study.

References

Abraham, M. J., Murtola, T., Schulz, R., Páll, S., Smith, J. C., Hess, B., & Lindahl, E. (2015). GROMACS: High performance molecular simulations through multi-level parallelism from laptops to supercomputers. *SoftwareX*, *1-2*, 19–25. https://doi.org/10.1016/j.softx.2015.06.001

Bergh, C., Majdolhosseini, M., & Villa, A. (2025). *GROMACS tutorial: Introduction to molecular dynamics*. https://doi.org/10.5281/zenodo.14803238

Degiacomi, M. T., Gowers, R. J., Micaela Matta, & Mey, A. S. J. S. (2025). A course on the setup, running, and analysis of biomolecular simulations. *Journal of Open Source Education*, *8*, 265. https://doi.org/10.21105/jose.00265

Grushow, A., & Reeves, M. S. (2019). Using computational methods to teach chemical principles: overview. In *Using computational methods to teach chemical principles* (pp. 1–10). American Chemical Society. https://doi.org/10.1021/bk-2019-1312.ch001

HPC Carpentry development team. (2021). *Hpc-carpentry/hpc-shell: Introduction to using the shell in a high-performance computing context*. https://hpc-carpentry.github.io/hpc-shell/

Humphrey, W., Dalke, A., & Schulten, K. (1996). VMD – Visual Molecular Dynamics. *Journal of Molecular Graphics*, *14*, 33–38. https://doi.org/10.1016/0263-7855(96)00018-5

Koch, C., Stafford, J., Steinbach, P., Simpson, J., & Keller, T. (2025). *Carpentries-incubator/hpc-intro: An introduction to high performance computing*. https://carpentries-incubator.github.io/hpc-intro/

Pollak, H., Degiacomi, M. T., & Erastova, V. (2024). Modeling realistic clay systems with ClayCode. *Journal of Chemical Theory and Computation*, *20*, 9606–9617. https://doi.org/10.1021/acs.jctc.4c00987

Turner, P. J. (2005). *Grace: A WYSIWYG 2D plotting tool for the x window system*. https://plasma-gate.weizmann.ac.il/Grace/

University of Edinburgh. (n.d.). *Edinburgh compute and data facility*. https://www.ecdf.ed.ac.uk.

Wilson, G., Capes, G., Devenyi, G. A., Koch, C., Silva, R., Srinath, A., Morris, C., Jackson, M., Boughton, A., Emonet, R., Gacenga, F., Nederbragt, L., csqrs, Irving, D., Becker, E. A., Deniz, F., Stimberg, M., Beagrie, R. A., McCloy, D., … Chhatre, V. (2019). *Swcarpentry/shell-novice: Software carpentry: The UNIX shell, june 2019* (G. A. Devenyi, G. Capes, C. Morris, & W. Pitchers, Eds.; Version v2019.06.1). Zenodo. https://doi.org/10.5281/zenodo.3266823