Computational Skills Training for Undergraduate Researchers in Molecular Engineering

Kristen Finch
University of Washington
finchkn@uw.edu

Ryan Beck
University of Washington
rbeck4@uw.edu

Xiaosong Li
University of Washington
xsli@uw.edu

Nam Pho
University of Washington
npho@uw.edu

Xiao Zhu
University of Washington
xiaozhu@uw.edu

ABSTRACT

In June 2024, the University of Washington's (UW) Clean Energy Institute (CEI) and Molecular Engineering and Materials Center (MEMC) in partnership with UW Research Computing (RC) prepared complimentary training for a group of 25 Research Experience for Undergraduates (REU) participants. Workshop participants had completed zero to four years of post-secondary education and came from 17 colleges and universities across eight states with 29% currently attending 2-year programs. On average, 14 students attended a given workshop. The program included four targeted workshop offerings, spanning essential skills in computational science and advanced topics: (1) Python via Jupyter, (2) Command Line Interface (CLI) and high performance computing (HPC), (3) Gaussian and Quantum Espresso, and (4) data analysis using linear and logistic regression as well as neural networks. The program's effectiveness was evaluated with a post-workshop survey. Survey results indicated most participants had little prior experience in these topics but indicated the content was relevant for their current and future aspirations. The survey showed some students agreed with statements indicating that learning objectives were met, but overall scores and open responses indicated areas for improvement. In the future, the CLI and HPC session will be converted from one to two sessions and the material in the applied Gaussian and Quantum Espresso demonstrations reduced. The program's materials are reproducible and publicly accessible, compatible with most academic HPC clusters. Our program addressed a wide range of training and education needs within computational science, emphasizing practical skills and interdisciplinary applicability.

KEYWORDS

Undergraduate Research, REU Workshop, Student Training, High Performance Computing, HPC Education, Jupyter Notebook, Linux, Command Line, SLURM, Gaussian, Quantum Espresso, Molec-ular Engineering, Machine Learning

https://doi.org/10.22369/issn.2153-4136/6/1/10

1 INTRODUCTION

The UW Clean Energy Institute (CEI) and Molecular Engineering and Materials Center (MEMC) are combining innovative research with comprehensive training to foster broader engagement in STEM fields. As part of their grant-funded activities, CEI and MEMC partnered with UW Research Computing (RC) to provide training for REU students residing on campus in Seattle, WA, USA for an immersive 12-week research experience. Students were recruited for their interest and promise in chemical and molecular engineering. Four workshops were delivered in June 2024, covering a range of topics from fundamental computing and data analysis to more advanced and applied techniques (Figure 1). The topics were carefully curated to support on-boarding and build technical capacity relevant to the students' research projects, as well as to equip them with skills that will be valuable in their future coursework and careers. The workshops primarily utilized the UW RC High-Performance Computing (HPC) cluster, Hyak, and Hyak's implementation of the Open OnDemand [14] platform, ensuring hands-on experience with widely adopted tools. The effectiveness of the workshops was evaluated through a post-workshop survey, which provided the RC team with valuable insights for refining future iterations of the workshop series. These lessons will inform the development of future workshops aimed at enhancing HPC skills across the UW research community. Additionally, the materials and methods developed for this series offer a template that can be adapted for other academic research computing centers, paving the way for broader implementation and outreach in STEM education.

Figure 1: A photo from Workshop 3 of the series, "Using Gaussian for Computational Chemistry."

2 OBJECTIVES

The objective of these workshops was to introduce data analysis tools and HPC, as well demonstrate how more advanced and scientific domain-specific software utilizes the job scheduling infrastructure and computing power delivered through HPC. The workshops offered were as follows, and the team developed specific objectives for each workshop (WS):

WS 1 *Basic Python Programming*

- Understand the basics and importance of Jupyter Notebooks [12] towards reproducible and transparent scientific research.
- Navigate the basics of Python as a scripting and programming language.

WS 2 *Using Hyak, Linux Operating System (OS)*

- Navigate a file system and explore files using Linux commands.
- Understand the purpose of a job scheduler like SLURM in HPC [28].

WS 3 *Using Gaussian for Computational Chemistry*

- Create and modify input file(s) for quantum chemistry calculations, and become familiar with output files from Gaussian [7] and Quantum Espresso [9, 10].
- Submit and monitor computational chemistry jobs to an HPC cluster.

WS 4 *Data Analysis in Python with scikit-learn and PyTorch*

- Apply linear and logistic regression models to data and interpret the results.
- Train a neural network in Python and understand training parameters.
- Tune network architecture parameters to improve performance and articulate the various metrics to evaluate a machine learning model.

Our final objective was to make the materials available for future iterations and independent study by the student participants.

3 METHODS

3.1 Format

Each workshop was three hours, live, hands-on, and delivered in Seattle, WA, USA on the main UW campus. Students were sent calendar invitations and reminder emails with links to training materials at least 24 hours in advance of each workshop. In addition, virtual office hours were held the day prior to each workshop to offer support, whether workshop related or beyond.

Workshops utilized a variety of tools to deliver content. WS 1 used Jupyter Lab and Notebooks via Hyak's Open OnDemand [14] environment delivered through each participant's browser. Additionally, each participant was guided through the installation of Anaconda Navigator for later independent study during their REU. WS 2 was delivered using participants' local shell terminals. All commands were demonstrated on Hyak, which runs Rocky Linux 8. Hyak is the UW tri-campus supercomputer with over 30K compute cores, 500 GPU accelerators, and 2.5 PB of parallel file systems storage operating on a shared condo model. The CEI has dedicated resources that are primarily used for research throughout the year and were temporarily reallocated for education and outreach during these workshops. WS 3 used a combination of submitting jobs and editing scripts in the Hyak shell, then generating inputs and viewing Gaussian and Quantum Espresso results with Hyak's OnDemand virtual desktop. WS 4 used Google Colab, demonstrating an additional platform for using Python-based computational notebooks in the cloud.

3.2 Training Materials

Training materials were prepared and made publicly accessible via a GitHub repository or as a web-based tutorial on Hyak's documentation website. The program's materials are reproducible, compatible with most academic HPC clusters, and in-line with FAIR computational workflow specifications [11]. The format is flexible and could be altered to focus on any HPC-compatible discipline-specific software.

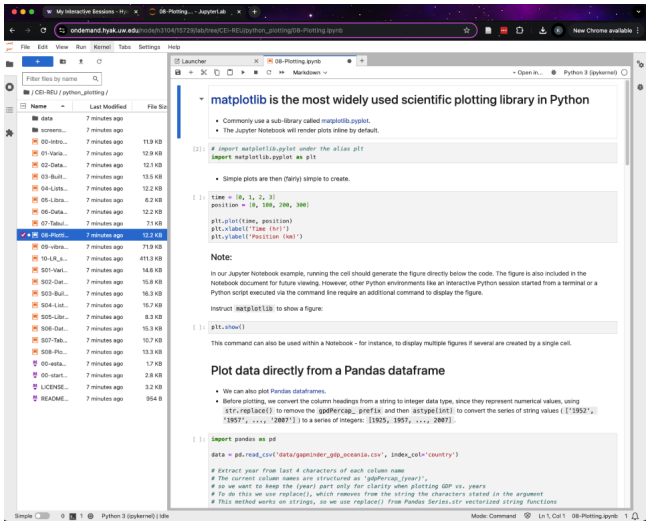

Figure 2: Jupyter Lab and Notebooks prepared for WS 1 accessed via Hyak Open OnDemand. Python was taught via live demonstration with a blank notebook. The photo shows notebooks that were prepared from Software Carpentry's lesson, "Plotting and Programming in Python" for the participant's independent study.

WS 1 sampled exercises from Software Carpentry's [26, 27] lesson, "Plotting and Programming in Python." Selected sections were made into Jupyter Notebooks with and without solutions for participant independent study, but the demonstration was live from a fresh Notebook without solutions. WS 1 was delivered via Hyak's OnDemand platform, which ran a containerized version of Jupyter's datascience-notebook. (Figure 2).

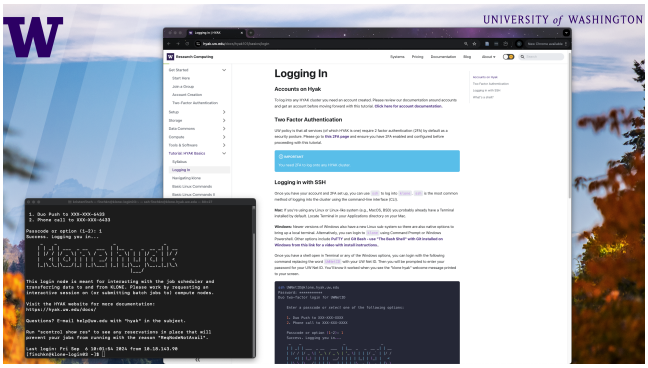

Figure 3: Hyak's documentation website showing materials for WS 2 as well as a shell logged into Hyak's third generation cluster, klone.

WS 2 introduced Hyak's file system (Figure 3), and some exercises to demonstrate Linux commands were sampled from Software Carpentry's [26, 27] lesson, "The Unix Shell." SLURM interactive jobs, batch jobs, and array jobs were demonstrated using Apptainer [16, 17] and a containerized version of Locator neural network [1] and publicly available data from black cottonwood trees [8].

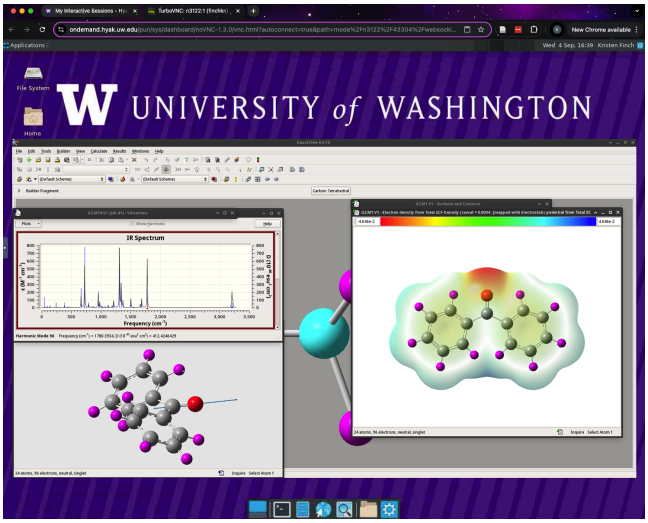

Figure 4: GaussView accessed via Hyak Open OnDemand virtual desktop feature, TurboVNC, showing an example molecule used during WS 3 from the photochemCAD database (benzophenone). The upper left panel shows the calculated infrared absorption spectrum (and the "normal mode" responsible for the highlighted peak). The right panel shows the electrostatic potential and the electronegativity of the oxygen atom.

WS 3 introduced quantum chemistry software on Hyak. The GaussView [4] graphical user interface was delivered via Hyak's OnDemand platform (Figure 4), and used to set up both Gaussian [7] and Quantum Espresso [9, 10] calculations. Job types included geometry optimizations and band plots were shown. In addition, the photochemCAD database [24] was used to compare experimental and calculated absorption spectra. Exercises for WS 4 were prepared as Jupyter Notebooks and delivered with Google Colab (Figure 5).

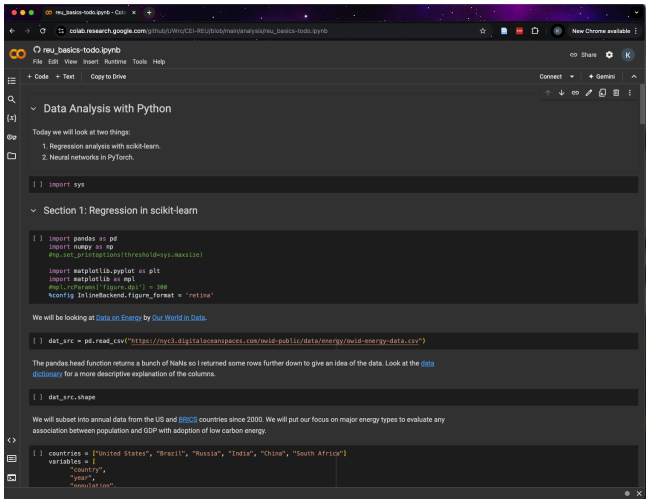

Figure 5: Google Colab view of Jupyter Notebook prepared for WS 4 to demonstrate some data analysis basics as well as linear and logistic regression and training a neural network.

WS 4 utilized a data set from Our World in Data on statistics regarding country-level clean energy adoption [23]. Trends were predicted using linear regression to predict future clean energy utilization and classification done using logistic regression with scikit-learn [21] to predict what factors would contribute towards a country adopting clean energy (or not). A shallow, three-layer artificial neural network (ANN) was generated with PyTorch [20] but the framework allows for the attendee to modulate ANN depth, layer width, and test optimization algorithms to experiment with different ANN architectures. All these aspects help to demonstrate the effort that goes into tuning an ANN to assess performance improvements relative to logistic regression. Model performance metrics were explored, including R^2 and Receiver Operating Characteristic (ROC) curves [6], as quantitative measures for model comparison.

3.3 Accessibility

With the exception of one student, all students brought a personal computer to the workshop and installed the required software to participate. One student was provided a laptop for use by UW Information Technology. No other accommodations were requested.

3.4 Evaluation and Analysis

All workshop objectives were translated into post-workshop survey statements to evaluate if the objective was met, and we refer this category of survey response as, "objective agreement ranking scores" because participants were asked to rank their level of agreement with the statements from 0 (strongly disagree) to 5 (strongly agree). In addition, the survey collected information about the participants

educational experience, prior knowledge about workshop topics, and their assessment of the relevance of the topics towards their current and future studies. We also requested positive and negative feedback short responses.

Our analysis included data visualization and rudimentary statistical analysis of participant agreement rankings using R [22] via a Tidyverse container (V4.4.1) from the Rocker Project [2, 19]. For the purposes of this discussion, we evaluated a ranking ≥ 2.5 as indicating that an objective had been met.

4 RESULTS

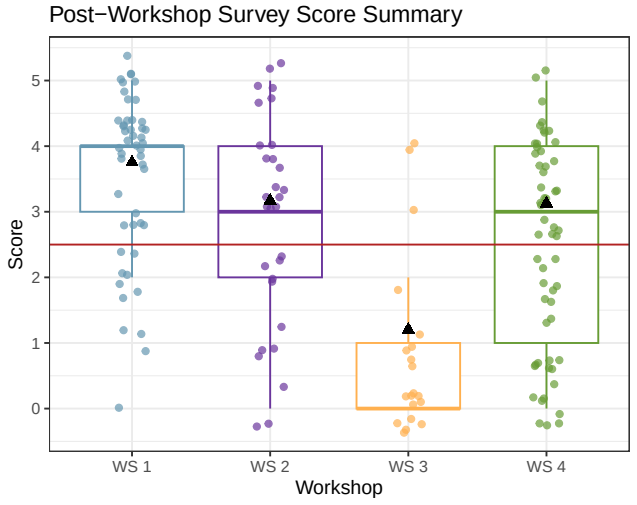

Figure 6: Box plot shows the distribution of objective agreement ranking scores for each workshop (WS). Points indicate individual scores from post-workshop survey respondents, which were used to generate the box plot summaries. These points and the box plots were color coded by workshop. Black triangular points indicate the average overall score for each workshop on a scale from 0 (not helpful) to 5 (extremely helpful). The red line at 2.5 is our threshold for if workshop objectives were met. *NOTE: Objective agreement ranking scores were collected on a discrete scale. In this plot, points were "jittered" for better visualization of the data.*

Thirteen participants responded to the post-workshop survey. The prior education experience breakdown was 23.1% for having completed 0-1 years of post secondary education, 46.2% for 1-2 years, 15.4% for 2-3 years, and 15.4% for 3-4 years. The two workshops demonstrating Python (WS 1 and WS 4) were evaluated with the highest relevance for participants future research and professional development, scoring on average 3.50 and 3.25, respectively on a scale of 0 (not certain of the relevance) to 5 (high relevance or likely frequent and regular usage in the future).

The overall average objective agreement score was 2.66 (SD = 1.64) indicating that some objectives were met yet suggesting some areas for improvement in future iterations. By workshop, the average objective agreement ranking scores were: 3.50 (SD = 1.25) for WS 1, 2.83 (SD = 1.62) for WS 2, 0.90 (SD = 1.33) for WS 3, and 2.48 (SD = 1.53) for WS 4. Participants' overall workshop rating on a scale from 0 (not helpful) to 5 (extremely helpful) was evaluated independent of the objective agreement rankings, and participants ranked each workshop higher than the agreement rankings scores suggest: 3.75 (SD = 0.622) for WS 1, 3.17 (SD = 1.17) for WS 2, 1.20 (SD = 1.30) for WS 3, and 3.12 (SD = 1.64) for WS 4. Additionally, the respondents agreed that the instructors were understanding of their needs (3.65 on average; SD = 1.20), but short responses requesting critical feedback and improvements indicated that the pace at which the material was covered should be decreased for better learning and retention.

5 DISCUSSION

Although the average objective agreement ranking for the workshop series as a whole was leaning in agreement that our objectives were met, we have reflections to share along with improvements for future iterations:

(1) WS 1 was largely successful, but the wrong venue was selected. The classroom was too small and lacked climate control, especially important for a summer course. Being the first workshop, at least 20 participants attended and stayed for the entire three hour session. Subjectively, there were palpable feelings of excitement and energy in this first session. However, due to first time delivery issues and other logistical issues around the infrastructure, it is possible this momentum was blunted as the relatively high attendance did not carry over to subsequent workshops. Learning basic Python with Jupyter Notebooks is generally enjoyable, which was reflected by the positive feedback.

(2) WS 2 was in part successful, but it was clear that combining basic Linux skills with advanced concepts included in demonstrating SLURM was not effective for this group. Like with Python basics, Linux basics are fun and easy to follow, but the jump to SLURM was not well executed. This particular discussion of SLURM also utilized a software container, which exposed the students to additional, advanced content. We have since improved this training with simpler and easier to consume examples. For future iterations, WS 2 should be split into a session focused solely on Linux CLI and a separate session on SLURM and parallel computing to address learning fatigue.

(3) Feedback from participants and instructor observations also indicated that the pacing of WS 3 needed to be slowed and include fewer distinct examples, allowing more time to be spent on each part of the demonstration. One participant commented that while the workshop PDF included a step-by-step guide, it was missing some vital commands that would have helped participants catch up when they missed a step and consequently fell behind the presentation.

(4) Since WS 1 and WS 4 were both using Jupyter Notebooks, in future iterations, we will deliver these workshops in sequence to increase continuity with these tools. During WS 4, in-class engagement in discussion suggesting that the participants were following the course content, but the range of agreement ranking scores was high, indicating not all participants were reached.

We think that learning fatigue was present in all workshops due to the three hour format. An alternative pedagogical delivery suggests balancing our approach with more active self-directed learning [3, 25]. Outside of the workshop series, students were participating in a immersive research experience with new colleagues and independent projects, further adding stress on their time, energy, and focus. Hosting more frequent but shorter sessions could be more effective for this demographic as there has been a generational shift in attention spans and emphasis on brevity [18, 25].

6 FUTURE PLANS

It is worth exploring if a format that connects the workshops around a single dataset or product could improve participant engagement and the overall learning experience. For example, a single dataset or problem would allow any "output" generated from a prior workshop be the "input" to the next. Should a student attend every session, the material would make sense not only in terms of the presented computational skill but it would loosely resemble a journey an experienced researcher might take to explore a hypothesis. A final hackathon could bring together students to write an academic paper unifying the workshop materials and results, and provide a tangible outcome for the participants in addition to capacity building.

In addition, we aim to broaden the applicability of this content and extend this integrated training model beyond the REU event. By applying this approach to various science and engineering disciplines, we can provide a cohesive learning experience that supports students in building computational skills progressively. This template may also be beneficial in reaching fields that lack training advanced computing and data literacy but may benefit from technological advanced in computer vision and AI. Introducing these disciplines to a structured, data-driven learning process can address training gaps and inspire avenues for research. Our goal is to create a versatile training platform at University of Washington that fosters interdisciplinary collaboration and advances students' skills across a wide range of scientific and engineering domains.

7 CONCLUSIONS

Many universities have long had a research computing, high-performance computing, or similar teams for decades and only relatively recently adopted data science analogs [5, 13, 15]. As noted previously in this paper, the research computing team has intentions to refine and redeploy the material used in this REU workshop. The UW RC team is developing a regular training schedule for its research community, drawing from both accepted best-practices nationally as well as custom modules. We explored how packaging this combination of pre-existing and custom educational materials would be received by a diverse undergraduate research audience and provided the lessons and feedback for other teams to learn from. Given the sample size ($n = 25$), we acknowledge there could be limited external validity of any conclusions. However, we believe it is important to provide these voices toward the ongoing national discussions around computational skills training for research trainees in higher education.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation (NSF) through the UW Molecular Engineering Materials Center, a Materials Research Science and Engineering Center (DMR-2308979). Computations were facilitated through the use of advanced computational, storage, and networking infrastructure provided by the shared facility supported by the Clean Energy Institute (CEI) via the UW Hyak Supercomputer.

REFERENCES

[1] CJ Battey, Peter L Ralph, and Andrew D Kern. 2020. Predicting geographic location from genetic variation with deep neural networks. 9 (2020), e54507. https://doi.org/10.7554/eLife.54507

[2] Carl Boettiger and Dirk Eddelbuettel. 2017. An Introduction to Rocker: Docker Containers for R. *The R Journal* 9, 2 (2017), 527–536. https://doi.org/10.32614/RJ-2017-065

[3] Katarzyna Czabanowska, Jos Moust, André Meijer, Peter Schröder-Bäck, and Herma Roebertsen. 2012. Problem-based Learning Revisited, introduction of Active and Self-directed Learning to reduce fatigue among students. 9, 1 (2012). https://doi.org/10.53761/1.9.1.6

[4] Roy Dennington, Todd A. Keith, and John M. Millam. 2019. GaussView Version 6. Semichem Inc. Shawnee Mission KS.

[5] Rae Earnshaw. 2019. *Data Science Institutes and Data Centers.* Springer International Publishing, 93–108. https://doi.org/10.1007/978-3-030-24367-8_7 Series Title: Advanced Information and Knowledge Processing.

[6] Tom Fawcett. 2006. An introduction to ROC analysis. 27, 8 (2006), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

[7] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, G. Scalmani, V. Barone, G. A. Petersson, H. Nakatsuji, X. Li, M. Caricato, A. V. Marenich, J. Bloino, B. G. Janesko, R. Gomperts, B. Mennucci, H. P. Hratchian, J. V. Ortiz, A. F. Izmaylov, J. L. Sonnenberg, D. Williams-Young, F. Ding, F. Lipparini, F. Egidi, J. Goings, B. Peng, A. Petrone, T. Henderson, D. Ranasinghe, V. G. Zakrzewski, J. Gao, N. Rega, G. Zheng, W. Liang, M. Hada, M. Ehara, K. Toyota, R. Fukuda, J. Hasegawa, M. Ishida, T. Nakajima, Y. Honda, O. Kitao, H. Nakai, T. Vreven, K. Throssell, J. A. Montgomery, Jr., J. E. Peralta, F. Ogliaro, M. J. Bearpark, J. J. Heyd, E. N. Brothers, K. N. Kudin, V. N. Staroverov, T. A. Keith, R. Kobayashi, J. Normand, K. Raghavachari, A. P. Rendell, J. C. Burant, S. S. Iyengar, J. Tomasi, M. Cossi, J. M. Millam, M. Klene, C. Adamo, R. Cammi, J. W. Ochterski, R. L. Martin, K. Morokuma, O. Farkas, J. B. Foresman, and D. J. Fox. 2016. Gaussian~16 Revision C.01. Gaussian Inc. Wallingford CT.

[8] A. Geraldes, S. P. Difazio, G. T. Slavov, P. Ranjan, W. Muchero, J. Hannemann, L. E. Gunter, A. M. Wymore, C. J. Grassa, N. Farzaneh, I. Porth, A. D. Mckown, O. Skyba, E. Li, M. Fujita, J. Klapste, J. Martin, W. Schackwitz, C. Pennacchio, D. Rokhsar, M. C. Friedmann, G. O. Wasteneys, R. D. Guy, Y. A. El-Kassaby, S. D. Mansfield, Q. C. B. Cronk, J. Ehlting, C. J. Douglas, and G. A. Tuskan. 2013. A 34K SNP genotyping array for Populus trichocarpa: Design, application to the study of natural populations and transferability to other Populus species. 13, 2 (2013), 306–323. https://doi.org/10.1111/1755-0998.12056

[9] P Giannozzi, O Andreussi, T Brumme, O Bunau, M Buongiorno Nardelli, M Calandra, R Car, C Cavazzoni, D Ceresoli, M Cococcioni, N Colonna, I Carnimeo, A Dal Corso, S de Gironcoli, P Delugas, R A DiStasio Jr, A Ferretti, A Floris, G Fratesi, G Fugallo, R Gebauer, U Gerstmann, F Giustino, T Gorni, J Jia, M Kawamura, H-Y Ko, A Kokalj, E Küçükbenli, M Lazzeri, M Marsili, N Marzari, F Mauri, N L Nguyen, H-V Nguyen, A Otero de-la Roza, L Paulatto, S Poncé, D Rocca, R Sabatini, B Santra, M Schlipf, A P Seitsonen, A Smogunov, I Timrov, T Thonhauser, P Umari, N Vast, X Wu, and S Baroni. 2017. Advanced capabilities for materials modelling with QUANTUM ESPRESSO. *Journal of Physics: Condensed Matter* 29, 46 (2017), 465901. http://stacks.iop.org/0953-8984/29/i=46/a=465901

[10] Paolo Giannozzi, Stefano Baroni, Nicola Bonini, Matteo Calandra, Roberto Car, Carlo Cavazzoni, Davide Ceresoli, Guido L Chiarotti, Matteo Cococcioni, Ismaila Dabo, Andrea Dal Corso, Stefano de Gironcoli, Stefano Fabris, Guido Fratesi, Ralph Gebauer, Uwe Gerstmann, Christos Gougoussis, Anton Kokalj, Michele Lazzeri, Layla Martin-Samos, Nicola Marzari, Francesco Mauri, Riccardo Mazzarello, Stefano Paolini, Alfredo Pasquarello, Lorenzo Paulatto, Carlo Sbraccia, Sandro Scandolo, Gabriele Sclauzero, Ari P Seitsonen, Alexander Smogunov, Paolo Umari, and Renata M Wentzcovitch. 2009. QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter* 21, 39 (2009), 395502 (19pp). http://www.quantum-espresso.org

[11] Carole Goble, Sarah Cohen-Boulakia, Stian Soiland-Reyes, Daniel Garijo, Yolanda Gil, Michael R. Crusoe, Kristian Peters, and Daniel Schober. 2020. FAIR Computational Workflows. *Data Intelligence* 2, 1-2 (01 2020), 108–121. https://doi.org/10.1162/dint_a_00033 arXiv:https://direct.mit.edu/dint/article-pdf/2/1-2/108/1893377/dint_a_00033.pdf

[12] Brian E. Granger and Fernando Perez. 2021. Jupyter: Thinking and Storytelling With Code and Data. *Computing in Science & Engineering* 23, 2 (March 2021), 7–14. https://doi.org/10.1109/MCSE.2021.3059263
[13] Tony Hey and Anne Trefethen. 2003. e-Science and its implications. 361, 1809 (2003), 1809–1825. https://doi.org/10.1098/rsta.2003.1224
[14] Dave Hudak, Doug Johnson, Alan Chalker, Jeremy Nicklas, Eric Franz, Trey Dockendorf, and Brian McMichael. 2018. Open OnDemand: A web-based client portal for HPC centers. *Journal of Open Source Software* 3, 25 (May 2018), 622. https://doi.org/10.21105/joss.00622
[15] Nicholas W. Jankowski. 2007-01. Exploring e-Science: An Introduction. 12, 2 (01 2007-01), 549–562. https://doi.org/10.1111/j.1083-6101.2007.00337.x
[16] Gregory M. Kurtzer, cclerget, Michael Bauer, Ian Kaneshiro, David Trudgian, and David Godlove. 2021. *hpcng/singularity: Singularity 3.7.3.* https://doi.org/10.5281/zenodo.4667718
[17] Gregory M. Kurtzer, Vanessa Sochat, and Michael W. Bauer. 2017. Singularity: Scientific containers for mobility of compute. 12, 5 (2017), e0177459. https://doi.org/10.1371/journal.pone.0177459 Publisher: Public Library of Science.
[18] Michael Z. Newman. 2010. New media, young audiences and discourses of attention: from *Sesame Street* to 'snack culture'. 32, 4 (2010), 581–596. https://doi.org/10.1177/0163443710367693
[19] Daniel Nüst, Dirk Eddelbuettel, Dom Bennett, Robrecht Cannoodt, Dav Clark, Gergely Daróczi, Mark Edmondson, Colin Fay, Ellis Hughes, Lars Kjeldgaard, Sean Lopp, Ben Marwick, Heather Nolis, Jacqueline Nolis, Hong Ooi, Karthik Ram, Noam Ross, Lori Shepherd, Péter Sólymos, Tyson Lee Swetnam, Nitesh Turaga, Charlotte Van Petegem, Jason Williams, Craig Willis, and Nan Xiao. 2020. The Rockerverse: Packages and Applications for Containerisation with R. *The R Journal* 12, 1 (2020), 437–461. https://doi.org/10.32614/RJ-2020-007
[20] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. *PyTorch: an imperative style, high-performance deep learning library.* Curran Associates Inc., Red Hook, NY, USA.
[21] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. 2011. Scikit-learn: Machine learning in Python. *the Journal of machine Learning research* 12 (2011), 2825–2830.
[22] R Core Team. 2024. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org/
[23] Hannah Ritchie, Max Roser, and Pablo Rosado. 2020. Renewable Energy. *Our World in Data* (2020). https://ourworldindata.org/renewable-energy.
[24] Masahiko Taniguchi, Hai Du, and Jonathan S. Lindsey. 2018. PhotochemCAD 3: Diverse Modules for Photophysical Calculations with Multiple Spectral Databases. 94, 2 (2018), 277–289. https://doi.org/10.1111/php.12862 _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/php.12862.
[25] Shirine Voller, Eddie Blass, and Vicki Culpin (eds.). 2011. *The Future of Learning.* Palgrave Macmillan UK. https://doi.org/10.1057/9780230306356
[26] G. Wilson. 2006. Software Carpentry: Getting Scientists to Write Better Code by Making Them More Productive. 8, 6 (2006), 66–69. https://doi.org/10.1109/MCSE.2006.122 Conference Name: Computing in Science & Engineering.
[27] Greg Wilson. 2025. Software Carpentry web site. Retrieved December 2010 from http://software-carpentry.org Main web site for Software Carpentry, replacing http://swc.scipy.org..
[28] Andy B. Yoo, Morris A. Jette, and Mark Grondona. 2003. SLURM: Simple Linux Utility for Resource Management. In *Job Scheduling Strategies for Parallel Processing*, Gerhard Goos, Juris Hartmanis, Jan Van Leeuwen, Dror Feitelson, Larry Rudolph, and Uwe Schwiegelshohn (Eds.). Vol. 2862. Springer Berlin Heidelberg, Berlin, Heidelberg, 44–60. https://doi.org/10.1007/10968987_3 Series Title: Lecture Notes in Computer Science.

A ARTIFACT DESCRIPTION

This appendix includes artifact associated with **Computational Skills Training for Undergraduate Researchers in Molecular Engineering**.

The artifacts described there are training materials prepared for the workshop series described, and *do not include computational results*. To access the workshop materials, follow the links presented below. During the workshop series held at University of Washington, participants were given a sponsored UWNetID, which is required to use UW's research computing cluster Hyak. Although the workshop series focused on UW resources and infrastructure, the materials are transformable to similar systems.

A.0.1 Artifact Check-list.

- **Publicly available workshop series repository on GitHub:** https://github.com/UWrc/CEI-REU.git
  - python_plotting sub-directory - these materials accompany WS 1 "Basic Python Programming" held on June 20, 2024.
    - Exercises and explanations used to build the Jupyter notebooks were sampled from Software Carpentry's "Plotting and Programming in Python."
      - Notebooks starting 00-08 can be followed workbook style by running the code blocks to demonstrate presented concepts. This set of Notebooks excludes the solutions to the Exercises.
      - Notebooks starting S00-S08 are duplicates of the Notebooks starting **00-08**, but this set of Notebooks include the solutions to the Exercises.
      - Notebooks starting 09-10 provide plotting examples from Gaussian outputs.
  - using_hyak sub-directory - these materials accompany WS 2 "Using Hyak, Linux Operating System" (OS) held on June 25, 2024. The walk-through tutorial is available on Hyak's documentation website discussed below, but we also share relevant files used in the tutorial via the repository so that they could be more broadly available.
    - locator_NN_job.slurm and locator_NN_array.slurm - template single and array job submission scripts.
    - data sub-directory
      - potr_genotypes.txt - sample data (adapted from [8].
      - potr_m_pred(0-4).txt - sample data (adapted from [8])
  - Using_Quantum_Chemistry_Software sub-directory - these materials accompany WS 3 "Using Gaussian for Computational Chemistry" held on June 27, 2024.
    - Chem_Software_Hyak.pdf - walk through tutorial that demonstrates how to: (1) use the GaussView graphical user interface, (2) create and submit Gaussian jobs, and (3) create and submit Quantum Espresso jobs.
    - ethane.sh file - this file will allow slurm submission of the ethane.gjf file.
    - Pseudopotential .upf files - extra files used in the Quantum Espresso portion of the workshop.
  - analysis sub-directory - these material accompany WS 4 "Data Analysis Basics in Python with scikit-learn and PyTorch" held on July 1, 2024.
    - Complete Jupyter Notebook
    - Jupyter Notebook with "TODO" or in-session code boxes for demonstration.
- **Publicly available tutorial on UW RC's Hyak documentation website:**
  - https://hyak.uw.edu/docs/hyak101/basics/syllabus provides a walk through tutorial with the following sections: Syllabus, Logging In, Navigating Klone, Basic Linux Commands, Basic Linux Commands II.
  - https://hyak.uw.edu/docs/hyak101/basics/syllabus_slurm provides a walk through tutorial for SLURM, which has been updated with additional examples. The "Advanced SLURM" sections were what was originally presented in addition to Linux CLI Basics for WS 2.
- **Additional publicly available resources:**
  - WS 2 used a containerized version of Locator Neural Network [1]. Locator Neural Network is available on GitHub: https://github.com/kr-colab/locator.git
  - The Dockerfile for the containerized version of Locator Neural Network used during WS 2 is available on GitHub:

https://github.com/finchnSNPs/Docker_kr-colab_locator
  - The Docker container used during WS 2 to run Locator Neural Network is available on DockerHub: https://hub.docker.com/repository/docker/finchnsnps/locator/general
- **Software and Code licenses:**
  - Bash, SLURM, and Quantum Espresso are published under the terms of the GNU General Public License.
  - Gaussian16 and Gaussview 6.1.1 are Copyright (c) 1988-2017, Gaussian, Inc. All Rights Reserved, and were used under a license held by the Li Lab at University of Washington.
  - Jupyter is a Copyright (c) Project Jupyter Contributors, and distributed under the terms of the 3-clause BSD License (https://jupyter.org/governance/projectlicense.html).
  - Locator Neural Network [1] is a copyright 2019 of C. J. Battey and released under a Non-Profit Open Software License 3.0 (NPOSL-3.0); (https://github.com/kr-colab/locator/blob/master/LICENSE.txt).
  - matplotlib's license is based on the Python Software Foundation license (https://matplotlib.org/stable/project/license.html).
  - NumPy's is a Copyright (c) 2005-2024, NumPy Developers. The license is publicly available (https://numpy.org/doc/stable/license.html).
  - Python is under the Python Software Foundation license agreement (https://docs.python.org/3/license.html#psf-license).
  - Pytorch is a Copyright (c) 2016-present, Facebook Inc. (https://github.com/pytorch/pytorch/blob/main/LICENSE)
  - scikit-learn was used under Copyright (c) 2007-2024 The scikit-learn developers, and distributed under the terms of the 3-clause BSD License (https://github.com/scikit-learn/scikit-learn/blob/main/COPYING).
  - scipy was used under Copyright (c) 2001-2002 Enthought, Inc. 2003-2024, SciPy Developers. It can be used under a BSD 3-clause (https://github.com/scipy/scipy/blob/main/LICENSE.txt).
  - Some of the data and exercises used in WS 1 and WS 2 are under the Copyright of Software Carpentry and are made available under the Creative Commons Attribution license (CC BY 4.0).
- **Data licenses:**
  - Our adaptation of *Populus trichocarpa* genotype data and locations [8] are licensed under a CC0 1.0 Universal (CC0 1.0) Public Domain Dedication license. Original genotyping results available on DRYAD (https://doi.org/10.5061/dryad.1051d).
  - All visualizations, data, and code produced by Our World in Data are completely open access under the Creative Commons BY license.