Received: 4 November 2021 | Revised: 29 August 2022 | Accepted: 30 August 2022

DOI: 10.1002/bmb.21676

ARTICLE

Biochemistry and Molecular Biology Education

WILEY

Python scripting for biochemistry and molecular biology in Jupyter Notebooks

Paul A. Craig[1,2] | **Jessica A. Nash**[2] | **T. Daniel Crawford**[2,3]

[1]School of Chemistry & Materials Science, Rochester Institute of Technology, Rochester, New York, USA

[2]Molecular Sciences Software Institute, Blacksburg, Virginia, USA

[3]Department of Chemistry, Virginia Tech, Blacksburg, Virginia, USA

Correspondence
Paul A. Craig, Rochester Institute of Technology, Rochester, NY, USA.
Email: pac8612@rit.edu

Funding information
National Science Foundation, Grant/Award Numbers: CHE-2136142, IUSE-2142033

Abstract

A programming workshop has been developed for biochemists and molecular biologists to introduce them to the power and flexibility of solving problems with Python. The workshop is designed to move users beyond a "plug-and-play" approach that is based on spreadsheets and web applications in their teaching and research to writing scripts to parse large collections of data and to perform dynamic calculations. The live-coding workshop is designed to introduce specific coding skills, as well as provide insight into the broader array of open-access resources and libraries that are available for scientific computation.

KEYWORDS

Jupyter Notebook, live-coding, MolSSI, Python, workshop

A programming workshop focused on Python and its associated libraries has been implemented in Jupyter Notebooks.[1] Python is free and open source, has many scientific packages, and is widely used. The workshop was developed at the Molecular Sciences Software Institute (MolSSI)[2–4] as a contribution to the MolSSI Education Initiative.[5,6] The MolSSI is an NSF-funded institute that works to improve software development in the computational molecular sciences. MolSSI Education has a wide variety of educational material, ranging from workshops that focus on new programmers[4] to graduate students working on software development projects.[7] This manuscript answers questions about getting started with Python in Jupyter Notebooks and briefly summarizes the contents of the workshop, providing a list of resources to help you move in this direction.

1 | WHAT IS A JUPYTER NOTEBOOK?

The Jupyter Notebook is a browser based development environment that enables bundling of code, text, and images (Figure 1). Jupyter Notebooks can be installed on Windows, Macintosh and Linux computers that have Python installed using either pip or conda. This workshop recommends installation of the Jupyter Notebook and the necessary Python packages using the Anaconda Data Science platform.[8] It is also possible to use Jupyter Notebooks by setting up a dedicated JupyterHub server on your campus (JupyterHub – JupyterHub 1.4.2 documentation). You can also gain access to Jupyter Notebooks through ChemCompute[1] or NanoHub[9] using an academic email address.

2 | WHAT IS THE ADVANTAGE OF USING JUPYTER NOTEBOOKS AND PYTHON SCRIPTING?

Publishers provide Excel spreadsheets with textbooks now and there are powerful web applications like

This article reports a session from the virtual international 2021 IUBMB/ASBMB workshop, "Teaching Science with Big Data."

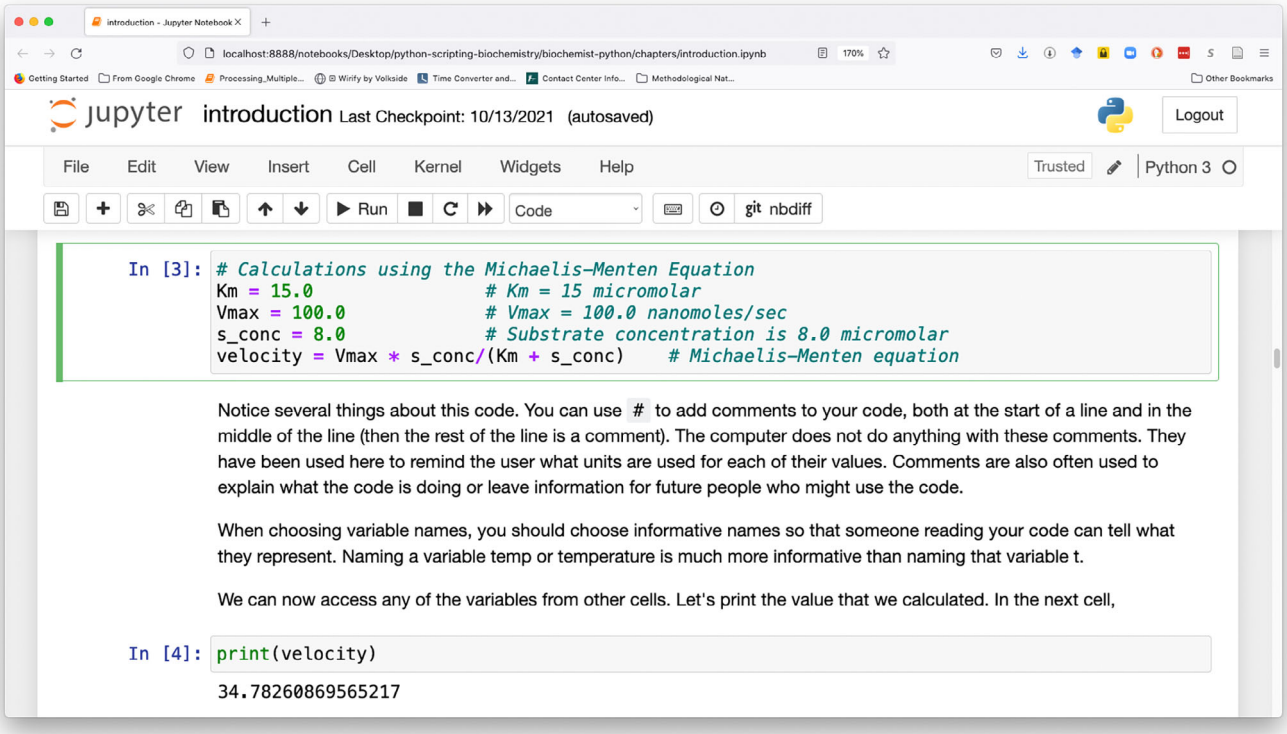

FIGURE 1 Screen capture of the Jupyter Notebook for the Introduction module.

BLAST,[10] Pfam[11] and Clustal Omega[12] that can readily be applied in your classroom and research lab. The reason is simple: The computational power and flexibility of Python scripting in Jupyter Notebooks exceeds what can be done in Excel. These notebooks are flexible and fully customizable to do exactly what you want. Python is a community-based, open source project,[13] with existing libraries of code and functions that are ideal for biochemistry and molecular biology. Plus the community is expanding the code base constantly. You may find that someone else has already written the exact code that you need or, more likely, code that you can readily adapt to your purposes, with the requirement that you must use an appropriate open-source license.

3 | WHAT ARE THE BARRIERS TO CHANGE?

You may be very comfortable with your existing resources, programs like Excel or Origin. There is a learning curve, particularly if you are new to coding in Python. If you decide to move into Jupyter Notebooks, you may find restrictions at your institution. Perhaps everything on your campus has to be built into a course management system, such as Moodle[14] or D2L Brightspace,[15] so you may need to work with your campus information technology team before you can even begin. Established courses on your campus may have adopted textbooks that include their own online resources and the publisher or your colleagues in a multi-section course may resist the move to more active computation in the biochemistry and molecular biology (BMB) curriculum. This may be particularly challenging for early adopters who have yet to receive tenure and are experiencing resistance from tenured faculty members. There may be some reluctance to take time from an already packed curriculum to have your students learn how to code. This workshop was designed to help you overcome at least some of these barriers.

4 | WORKSHOP DESIGN

Material for the workshop came from two sources. The first three modules are drawn in large part from the MolSSI Python Scripting for Computational Molecular Sciences[16] workshop, except that the computation focuses on BMB topics (Table 1). The remaining modules were based on interviews with multiple BMB faculty colleagues that focused on two questions.

TABLE 1 Python scripting for biochemistry and molecular biology modules.

Module	Topics
Introduction	Write text and perform calculations in Jupyter Notebooks, assign variables, datatypes, lists, slices, for-loops and use logic statements
File parsing	Use the os library to work with file paths in Jupyter Notebooks, read Protein Data Bank (PDB) files, search for patterns in multiple PDB files, search files by line number
Processing multiple files and writing files	Use the glob library to find collections of files, read multiple files with nested for loops, print to a file, string formatting
Working with pandas	Import csv data files into pandas dataframes (https://pandas.pydata.org/about/citing.html), find and sort data in the dataframe, perform calculations with functions from the NumPy library (https://numpy.org/citing-numpy/)
Linear regression	Implement calibration curve statistics with the SciPy library (https://www.scipy.org/citing.html), protein concentration calculations
Creating plots in Jupyter Notebooks	Create scatter plots with best-fit lines with the Matplotlib library (https://matplotlib.org/stable/citing.html), annotate plots, create plots with confidence intervals using the Seaborn library (https://seaborn.pydata.org/citing.html)
Nonlinear regression I	Inspect data, calculate slopes and initial velocities for enzyme kinetic data
Nonlinear regression II	Create functions, perform nonlinear curve fitting with scipy.optimize, print f-strings, annotate and smooth best fit curves

1. What is your experience with computing and coding?
2. What would you like to compute that you cannot compute right now?

The workshop is a live coding exercise influenced in large part by the principles of the *Software Carpentries Curriculum Development Handbook*,[17] which means no copying and pasting - you will actually write the code yourself, make your own mistakes and learn from those mistakes. In addition to the host, there are workshop co-hosts present who assist people when challenges arise. The workshop includes four 3-h sessions and starts with the assumption of no prior experience with Python or Jupyter Notebooks. The focus is on practice, giving you tools that you can apply immediately when you return home. In each module, we attempt to answer "What?", "Why?" and "How?" questions, in that order. Each session includes exercises and homework. The workshop consists of eight modules that are described in Table 1, along with brief descriptions and lists of Python libraries that are utilized in each session. Each module begins with questions, learning objectives and key points that will be addressed.

5 | A CLOSER LOOK AT THE FILE PARSING MODULE

The goal of this module is to learn how to use Python to extract information from text files using a PDB file as an example. On opening each module, the user will find a question, specific objectives and key points that refer to skills the user will gain on completing the module. For example, in the File Parsing Module, you will find the following questions and objectives:

1. Question

a. How do I sort through all the information in a text file and extract particular pieces of information?

2. Objectives

a. Open a file and read its contents line by line.
b. Search for a particular string in a file.
c. Manipulate strings and change data types.
d. Print results to a new file.

3. Key points

a. You should use the os.path module to work with file paths.
b. One of the most flexible ways to read in the lines of a file is the readlines() function.
c. An if statement can be used to find a particular string within a file.
d. The split() function can be used to separate the elements of a string.
e. You will often need to recast data into a different data type when it is read in as a string.

The session begins with learning to use the os library to work with file paths in Jupyter Notebooks, followed by reading the lines from the PDB file into a file that you will use in the notebook. You then search for a specific

text pattern ("string") in the file and print that to the screen. Finally, you learn to search for a specific line number in the imported file. The other modules follow a similar pattern: question, objectives, key points, live-coding instruction, exercises and homework.

A total of 27 participants from the Teaching Science with Big Data conference attended the first Python Scripting for BMB workshop (four 3-h virtual sessions between July 9 and 16). A group of faculty from that workshop are developing additional Jupyter Notebooks for BMB education. Please contact Paul Craig (paul.craig@rit.edu) if you would like to join our Slack channe). Future workshops will be offered under the direction of the Molecular Science Software Institute. Please contact Paul Craig or Jessica A. Nash (janash@vt.edu) if you are interested in participating in a workshop or in hosting a workshop at your institute or in your region.

6 | LINKS TO RESOURCES

Here are a few helpful links. The first two are directly related to the workshop.

1. Github home for Python Scripting for BMB (https://github.com/MolSSI-Education/python-scripting-biochemistry).
2. Python Scripting for BMB Jupyter Book (https://education.molssi.org/python-scripting-biochemistry/chapters/setup.html).
3. Molecular Sciences Software Institute (https://molssi.org/).
4. MolSSI Education on Github (https://github.com/molssi-education).

ORCID

Paul A. Craig https://orcid.org/0000-0002-2085-7816
Jessica A. Nash https://orcid.org/0000-0003-1967-5094
T. Daniel Crawford https://orcid.org/0000-0002-7961-7016

REFERENCES

1. Kluyver T, Ragan-Kelley B, Pérez F, Granger B, Bussonnier M, Frederic J, et al. Jupyter Notebooks – a publishing format for reproducible computational workflows. Positioning and power in academic publishing: players, agents and agendas; Amsterdam, The Netherlands: IOS Press; 2016. p. 87–90.
2. MolSSI – The Molecular Sciences Software Institute. Accessed October 6, 2021. https://molssi.org/.
3. Krylov A, Windus TL, Barnes T, Marin-Rimoldi E, Nash JA, Pritchard B, et al. Perspective: computational chemistry software and its advancement as illustrated through three grand challenge cases for molecular science. J Chem Phys. 2018;149:180901.
4. Wilkins-Diehr N, Crawford TD. NSF's inaugural software institutes: the science gateways community institute and the molecular sciences software institute. Comput Sci Eng. 2018;20:26–38.
5. McDonald AR, Nash JA, Nerenberg PS, Ball KA, Sode O, Foley JJ IV, et al. Building capacity for undergraduate education and training in computational molecular science: a collaboration between the MERCURY consortium and the molecular sciences software institute. Int J Quant Chem. 2020;120:e26359.
6. Ringer McDonald A. Teaching programming across the chemistry curriculum: A revolution or a revival? Washington, DC: American Chemical Society; 2021. p. 1–11.
7. Nash JA, Pritchard BP. Coding, software engineering, and molecular science - teaching a multidisciplinary course to chemistry graduate students. Washington, DC: American Chemical Society; 2021. p. 159–71.
8. Anaconda | The world's most popular data science platform. Accessed October 6, 2021. https://xddebuganaconda.xdlab.co/.
9. Madhavan K, Zentner L, Farnsworth V, Shivarajapura S, Zentner M, Denny N, et al. nanoHUB.Org: cloud-based services for nanoscale modeling, simulation, and education. Nanotechnol Rev. 2013;2:107–17.
10. Altschul SF, Madden TL, Schäffer AA, Zhang J, Zhang Z, Miller W, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucl Acids Res. 1997;25:3389–402.
11. Mistry J, Chuguransky S, Williams L, Qureshi M, Salazar GA, Sonnhammer ELL, et al. Pfam: the protein families database in 2021. Nucl Acids Res. 2021;49:D412–9.
12. Sievers F, Higgins DG. In: Katoh K, editor. Multiple sequence alignment: methods and protocols. New York, NY: Springer US; 2021. p. 3–16.
13. Welcome to Python.org. Accessed October 6, 2021. https://www.python.org/.
14. Moodle - Open-source learning platform | Moodle.org. Accessed October 6, 2021. https://moodle.org/.
15. D2L Brightspace home page. Accessed October 6, 2021. https://www.d2l.com/.
16. Python scripting for computational molecular science. Accessed October 6, 2021. https://education.molssi.org/python_scripting_cms/.
17. E. Becker and F. Michonneau (2021) The carpentries curriculum development handbook. https://github.com/carpentries/curriculum-development

How to cite this article: Craig PA, Nash JA, Crawford TD. Python scripting for biochemistry and molecular biology in Jupyter Notebooks. Biochem Mol Biol Educ. 2022;50(5):479–82. https://doi.org/10.1002/bmb.21676