Opening the Science Gateway: Lessons from the Materials Project Workshop

John Dagdelen, Joseph Montoya, Shyam Dwaraknath, Eric Sivonxay, Matthew Horton,
Patrick Huck, Shreyas Cholia, Donald Winston, and Kristin Persson,
University of California, Berkeley, Berkeley, CA 94720
Lawrence Berkeley National Lab, 1 Cyclotron Rd, Berkeley, CA 94720
Contact: dwinston@lbl.gov

***Abstract*—The Materials Project (MP) has served as an effective dissemination platform for computational materials science data for nearly 8 years. In its capacity as a Science Gateway, it serves about 60,000 users from around the world with data on over 80,000 crystalline materials. In recent years, MP's growing popularity has been facilitated by educational resources developed by its core team and diligently maintained documentation of the methodology and provenance associated with its data. In this report, we highlight a recent effort by MP to standardize a set of educational materials for its user base in its annual *Materials Project Workshop*, which was conducted in the summers of 2016 and 2017 and will be hosted again in August, 2018. More specifically, we describe our insights on how organization of material, presentation format, formative assessment, and active learning were integrated to produce an effective educational experience for our attendees. We also highlight the lessons we have learned, in the hopes other Science Gateway efforts may draw on our experiences in crafting their own educational and training resources.**

I. Introduction

The Materials Project (MP) [1], [2] is a Science Gateway that aims to accelerate the discovery and design of new materials by computing properties of known and predicted compounds and making these openly available to the research community. Its primary platform is a website hosted at https://materialsproject.org, which features an intelligent search interface, a visual data exploration platform, and a set of "apps." These apps enable various materials analyses such as the construction of phase diagrams, the evaluation of battery chemistries, and the matching of X-ray absorption spectra. In addition, the Materials Project makes all of its data available to users via a RESTful API [13] and actively develops and maintains a suite of open-source software tools for materials analysis [12][14][15].

Since its inception in 2011, MP has grown its database, web infrastructure, and backend automation software considerably. The number of compounds in the MP database has more than doubled since its inception and contains more than 80,000 entries today. Moreover, the Materials Project's user base has grown by an order of magnitude between 2013 and 2018 from roughly 5,000 to about 60,000 users. However, this growth has also brought new challenges in educating users about how to use MP tools and data. To help address these challenges we created the Materials Project Workshop, a live 2-day course on utilizing the Materials Project for materials discovery and design.

In this extended abstract we describe the methods, format, and teaching philosophies that we have found work most effectively for teaching users how to take advantage of our Science Gateway. In addition, we share the lessons we have learned over the past two years from teaching researchers how to use software in a live setting and describe how we were able to dramatically improved the educational experience of our audience through the use of complimentary software tools.

II. Workshop Summary and Teaching Philosophy

A. Topics Covered

Below we describe the topics covered during the 2017 MP Workshop. Each topic was covered in one or two 1¼ hour sessions of live instruction.

- **MP Website** - a walkthrough of the Materials Project website, including the materials explorer, the materials details page, the phase diagram app, the Pourbaix app, and the build-a-battery interface
- **Pymatgen** - a tutorial on the basics of using Pymatgen, an open source python library for working with crystal structures and various materials analyses
- **The Materials API (MAPI)** - a tutorial on how to use the Materials API, a programmatic RESTful interface for accessing the Materials Project database
- **Atomate** - a presentation on how MP automates its electronic structure calculations using the atomate open-source software package[15]
- **MPContribs** - a tutorial on how to contribute data to MP via the MPContribs framework and setting up users as MP contributors [3][4]
- **MongoDB** - a tutorial on using the Pymongo software package to interact with the MongoDB backend infrastructure to build and query a computational database of materials calculations

B. Audience Background

Using a pre- and post-workshop survey of our >50 attendees, we collected background information on our audience related to their prior experience, current positions, and interests. We noted the following:

Presented at Gateways 2018, University of Texas, Austin, TX, September 25-27, 2018. https://gateways2018.figshare.com/

- 35% are from industry, 65% from universities or national laboratories
- 50% described themselves as "experimental scientists"
- Operating system of choice: 68% Windows, 28% macOS, 24% Linux
- 20% of attendees self-identified as beginning programmers and 70% as non-expert programmers
- 65% reported intermediate proficiency or below in "data management"

This information, particularly the pre-workshop survey, was critical to our lesson design and enabled us to tailor material to non-expert programmers, users unfamiliar with our data infrastructure, and users with backgrounds in experimental science rather than simulation.

C. *Workshop Format*

The MP Workshop takes place over the course of two days with an optional half-day that precedes the workshop covering basic Python programming. Each day is broken up into 1¼ hour long sessions that each cover a different topic of interest. Other than an introductory overview presentation on the Materials Project, which kicks off the workshop, all of the sessions are interactive and include hands-on exercises which students complete as they follow along with the session instructor.

For each of these sessions, we provide tutorial Jupyter notebooks with worked examples and hands-on exercises. Students follow along during the tutorials and interact with code examples in real-time to complete practical tasks that are representative of how the software is used for real research tasks. In Figure 1, we show an excerpt from a Jupyter notebook corresponding to the Materials API section in which the user has retrieved an electronic bandstructure and plotted it using pymatgen.

All of the workshop materials are openly available as a Github repository [5].

D. *Teaching Philosophy and Formative Assessment*

Our teaching methodology draws heavily from the model used by Software Carpentry (SC) [6], which is participant-focused rather than material-focused, emphasizes regular assessment, and encourages collaborative lesson planning under version control. Furthermore, a few of us have been trained as SC instructors and led SC workshops in topics relevant to the development of science gateways [7][8][9]. As in many SC workshops, attendees are given sticky-notes (one red, one green) which they use to indicate whether they have successfully completed an interactive task or need guidance/assistance from one of 5-6 instructors monitoring the audience. At the end of each session, learners are asked to write both positive feedback and constructive criticism on the green and red sticky-notes, respectively, which are collected by helpers. As such, workshop instructors receive regular feedback on both their personal pedagogical approaches and on the general impressions of workshop material in real time. We have found this practice extremely helpful for tailoring our teaching for our particular audiences and refining our approach towards future workshops.

III. Technical Infrastructure

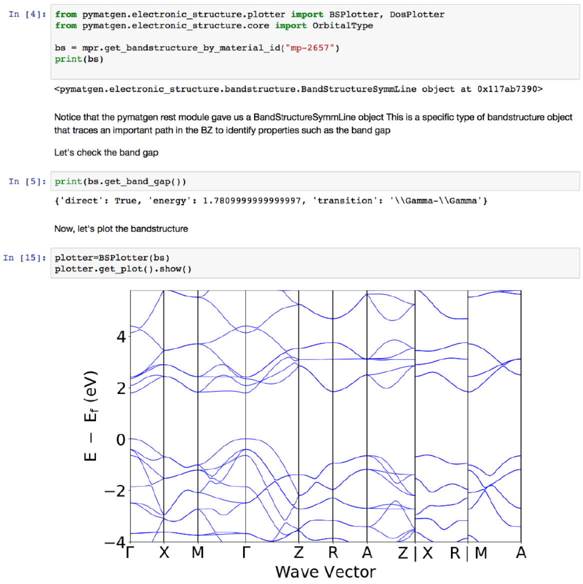
A. *Jupyter Notebooks*

We have found Jupyter notebooks to be an invaluable resource for teaching users how to leverage our software tools. Jupyter notebooks are interactive notebook environments that allow users to combine code with visualizations, documentation, and programmable widgets. They are particularly useful as teaching tools in our instruction sessions in which we guide participants through tutorial Python notebooks containing working examples and interactive exercises. Our own observations and the feedback collected from the audience at the end of each day indicate that using Jupyter notebooks in this setting has significantly helped with user engagement and information retention.

B. *JupyterHub and Docker*

In the past, we requested that MP workshop attendees install the required software (Python, Jupyter, required libraries, etc) on their personal computers prior to the start of the workshop. As one might expect, a subset of users had installation issues, the exact nature of which could not be anticipated due to the wide diversity of operating systems and system configurations, causing them to spend time troubleshooting with workshop helpers instead of following along with the session instructor.

We made preventing these kinds of issues a priority when preparing our most recent workshops (2017 and 2018) and addressed the problem through the use of two tools: JupyterHub and Docker. JupyterHub was developed by Project Jupyter



```
In [4]: from pymatgen.electronic_structure.plotter import BSPlotter, DosPlotter
        from pymatgen.electronic_structure.core import OrbitalType

        bs = mpr.get_bandstructure_by_material_id("mp-2657")
        print(bs)

        <pymatgen.electronic_structure.bandstructure.BandStructureSymmLine object at 0x117ab7390>
```

Notice that the pymatgen rest module gave us a BandStructureSymmLine object This is a specific type of bandstructure object that traces an important path in the BZ to identify properties such as the band gap

Let's check the band gap

```
In [5]: print(bs.get_band_gap())
        {'direct': True, 'energy': 1.7809999999999997, 'transition': '\\Gamma-\\Gamma'}
```

Now, let's plot the bandstructure

```
In [15]: plotter=BSPlotter(bs)
         plotter.get_plot().show()
```

Fig. 1. An excerpt from a Jupyter notebook in which the bandstructure corresponding to TiO_2 is retrieved using the MAPI and plotted inline.

for creating multi-user hubs which spawn, manage, and proxy multiple instances of the single-user Jupyter notebook server [10], while Docker is a program that performs operating-system-level virtualization (containerization). The practical result of using these tools together was that our students could access their own personal versions of the tutorial notebooks via their web browser without installing anything on their own systems prior to the workshop. This setup also allowed us to implement a ”live notebook” tool whereby participants who fell behind in a lesson could load a periodically updated snapshot of the instructor’s live Jupyter notebook. This was achieved by (1) having the instructor use a so-called ”notebook magic” command to autosave their notebook every five seconds, and (2) using a simple bash loop script to securely copy the notebook living inside the instructor’s Docker container to a remote directory outside the container, where it was served by an nginx daemon via a publicly accessible URI.

In order to achieve this seamless user experience, we created a Docker image [11] with the entire Materials Project software stack pre-installed along with a MongoDB service and a port-mapping scheme that allowed users to locally host and inspect relevant web services. The environment also mocked SLURM batch job management capabilities so that we could walk participants through the mechanics of queue submission to supercomputing resources. We then configured JupyterHub on a set of Debian/Ubuntu Linux nodes, each featuring 64GB RAM and 16 CPU cores, to provision notebook servers as copies of this Docker container so that each notebook had all of the required software pre-installed and properly configured with no extra effort on the students’ part. Because we were using a Docker Swarm, we could simply add new nodes to the swarm whenever we needed to initialize additional users.

Though this workshop used resources of the US Dept. of Energy available to the Materials Project, there is no significant barrier to replicating the setup on a popular cloud resources provider such as Google Compute Platform (GCP) or Amazon Web Services (AWS). We highly recommend similar workshops adopt this strategy as it greatly streamlined the set-up process for the event and saved time that would have otherwise been wasted on debugging. Moreover, it helped create a better overall learning experience for the workshop attendees.

C. Authentication

In previous workshops, we set up authentication via GitHub, which is a standard configuration option for JupyterHub. However, this required (1) collecting GitHub usernames of workshop attendees in order to approve access, and (2) walking participants through retrieving and installing their Materials Project API keys in their JupyterHub environment. This year, we set up the Materials Project website as a Central Authentication Service (CAS) through a third-party Django package. In this way, users were able to navigate directly to our JupyterHub instance, automatically authenticate as an MP user, and find that their API keys were already registered with the system. After making this change, we observed a significant reduction in the number of authentication issues that needed to be addressed by workshop staff compared to previous years.

D. Container management interface

We used *portainer* [16] - a simple management user interface for Docker - to identify and remedy malfunctioning user environments quickly. It enabled us to monitor the resources used on the host system and to inspect container logs on demand. There were several times that user container processes were able to be killed and restarted to rapidly resolve issues, bringing participants back and up to speed with the current lesson. Portainer has also made it significantly easier for us to gain shell access to running containers or deploy updated Docker images across swarm nodes when immediate updates to the installed software stack were necessary.

IV. Teaching Software in a Live Setting

A. Prioritizing Clear Communication

The exchange of information between the instructor and the audience should be made as robust as possible. In addition to providing amplification of the instructor’s voice and ensuring the sound quality prior to the start of the workshop, organizers should make every effort to choose a space that allows for the audience to sit close enough to the viewing screen to read code easily. This is often closer than what is required for most presentations. In addition, when drafting the tutorial notebooks, instructors should keep in mind that they will have to use higher magnification and font sizes as well as full screen mode during their tutorials for visual clarity. If not planned for, they may find themselves scrolling between notebook cells to show text/code, which is hard to follow as an audience member.

Rather than running pre-written code in notebook cells, we type code live. When teaching this way, mistakes are inevitable. However, this is a feature of our teaching methodology rather than a bug. Explaining and recovering from error messages in a live setting provides a valuable opportunity for teaching; not just how the code works in an ideal setting, but also the skills necessary to use it in the real world. Care should be taken to explain code as it is being typed as it serves the dual purpose of setting a pace that the audience can follow and reinforcing more fundamental concepts for those with less programming experience.

B. User Feedback

We have found that collecting user feedback is one of the best ways to assess the effectiveness of our materials and teaching strategies. Our feedback consists of a pre-workshop survey, post-session sticky comments (one positive and one constructive criticism), and a post-workshop survey. A large portion of the positive feedback that we collected expressed appreciation for the interactive tutorials and the utility of the software that they learned in these guided sessions. In our first workshop much of the constructive criticism communicated frustration at technical issues causing audience members to fall behind in the interactive sessions whereas after transitioning to the JupyterHub/Docker setup most of the constructive criticism concerned the pacing of the tutorials or the amount of material covered. Of these new comments, about 3/4 expressed that the pacing was too fast or that too much material was covered and the other 1/4 held the opposite opinion. This ratio seems to be generally correlated with the distribution of programming experience within the audience. By collecting this user feedback we were able to qualitatively and quantitatively measure the impact of our teaching strategies and set actionable priorities.

Conclusion

In hosting the annual Materials Project workshop for two consecutive years, we have learned many lessons that we use to augment the delivery of the workshop and interactivity of the Materials Project and its core software packages. We emphasize the importance of providing supplemental information, such as workshops, discussion boards, and wikis alongside Science Gateways to remove barriers to adoption and use. We have found workshops to be very effective resources when audience members participate in guided hands-on tutorials and have access to individual help from workshop staff as they run into any difficulties during the session. The essence of the workshop revolves around interaction with students both during lecture sessions and in gathering feedback to tailor material and delivery in the present workshop sessions, subsequent sessions, and future workshops. Guided by this feedback, we took proactive measures to improve the learning experience for our audience including reducing the time lost for software installation by using JupyterHub and Docker to provide workshop attendees with the necessary tools to engage in each session.

Acknowledgment

We would like to thank Murathan Aykol, Kiran Matthew, Shyue Ping Ong, and Anubhav Jain for their help developing the Materials Project Workshop. We acknowledge support from the the U.S. Department of Energy (DOE), Office of Science, Basic Energy Sciences (BES), under The Materials Project (BES DOE Grant No. KC23MP).

References

[1] A. Jain, S. P. Ong, G. Hautier, W. Chen, W. D. Richards, S. Dacek, S. Cholia, D. Gunter, D. Skinner, G. Ceder, and K. A. Persson, Commentary: The Materials Project: A materials genome approach to accelerating materials innovation, APL Materials, vol. 1, no. 1, p. 011002, 2013.

[2] https://materialsproject.org

[3] https://materialsproject.org/mpcontribs

[4] https://github.com/materialsproject/MPContribs

[5] https://github.com/materialsproject/workshop-2018

[6] https://software-carpentry.org

[7] https://dwinston.github.io/2016-03-02-lbnl/

[8] https://dwinston.github.io/2017-08-30-lbl-pyseclang/

[9] https://dwinston.github.io/pymongo-novice-materials/

[10] https://github.com/jupyterhub/jupyterhub

[11] https://github.com/materialsproject/mp-jupyter-docker

[12] https://github.com/materialsproject

[13] Ong, S. P., Cholia, S., Jain, A., Brafman, M., Gunter, D., Ceder, G., Persson, K. A. (2015). The Materials Application Programming Interface (API): A simple, flexible and efficient API for materials data based on REpresentational State Transfer (REST) principles. Computational Materials Science, 97, 209215. https://doi.org/10.1016/j.commatsci.2014.10.037

[14] Ong, S. P., Richards, W. D., Jain, A., Hautier, G., Kocher, M., Cholia, S., Ceder, G. (2013). Python Materials Genomics (pymatgen): A robust, open-source python library for materials analysis. Computational Materials Science, 68, 314319. https://doi.org/10.1016/j.commatsci.2012.10.028

[15] Mathew, K., et al. Atomate : A high-level interface to generate, execute, and analyze computational materials science workflows. Computational Materials Science, 139, 140152. https://doi.org/10.1016/j.commatsci.2017.07.030

[16] https://portainer.io