JOURNAL OF CHEMICAL EDUCATION

pubs.acs.org/jchemeduc

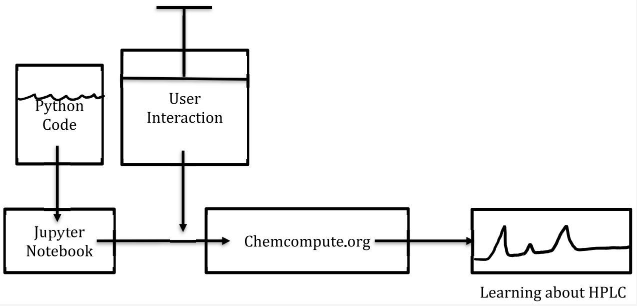
Communication

Online Data Generation in Quantitative Analysis: Excel Spreadsheets and an Online HPLC Simulator Using a Jupyter Notebook on the Chem Compute Web site

M. J. Perri*

Cite This: https://dx.doi.org/10.1021/acs.jchemed.0c00565 Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: Students used an Excel spreadsheet and a Jupyter notebook located at https://chemcompute.org/jupyterhub/ to generate data in lieu of face-to-face laboratory instruction during COVID-19. This communication discusses strengths and weaknesses of both exercises. First, students generated absorbance data using an Excel spreadsheet, but they were overwhelmingly put off by simulated noise in their absorbance values. Second, students simulated running an HPLC using a Jupyter notebook and designed a method to separate three components of Excedrin. Students reported that the simulation helped them to understand HPLC and suggested that the online activity be used as a supplement when face-to-face laboratory instruction resumes. The main weakness of the online activities was that students did not perform any sample preparation. Therefore, they did not experience the extractions and dilutions required to prepare the analytical sample and had a difficult time working backward to calculate starting amounts from simulated sample concentrations. Concerns that students would extract answers from the equations used in the spreadsheet and Jupyter notebook were not realized.

KEYWORDS: *Second-Year Undergraduate, Analytical Chemistry, Distance Learning/Self Instruction, Spectroscopy, Chromatography*

This communication describes an update to the capabilities of Chem Compute (chemcompute.org)[1] that has been implemented to replace in-person experiments with an online simulation. Chemcompute.org is a free site where students can perform computational chemistry using electronic structure and molecular dynamics packages such as GAMESS,[2] Psi4,[3] TINKER,[4] and NAMD.[5] Chem Compute has served over 20 000 users.

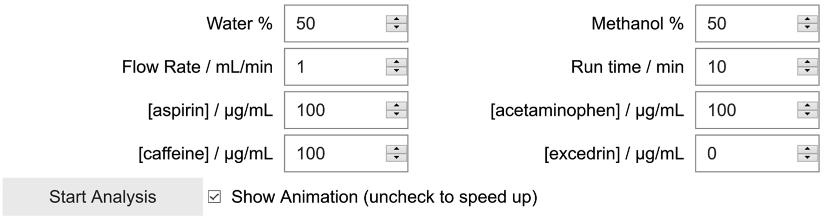
The author was teaching Chemistry 255, Quantitative Analysis, during the Spring 2020 semester with 18 students. The class consists of 2 hours of lecture and 6 hours of lab per week. The first half of the semester focuses on error propagation, equilibrium, and acid/base chemistry. The second half teaches an introduction to instrumental analysis (spectrophotometry, chromatography, and mass spectroscopy). The shelter-in-place orders and switch to online instruction began midsemester just before the instrumental portion of the class. The author used two approaches to virtualize the laboratory component: an Excel spreadsheet to have students generate their own data and an HPLC simulator written in Python in a Jupyter Notebook on Chem Compute. The two approaches had varying successes and failures that will be described below.

DATA GENERATION IN EXCEL: ANALYSIS OF IRON IN VITAMINS BY STANDARD ADDITION

The first approach to engaging students was to have them generate simulated experimental data in Excel to determine the amount of iron in a vitamin by standard addition. This experiment was adapted from a standard textbook experiment for determining iron in vitamins using an external standard.[6] Normally students would extract the iron from a vitamin tablet, react it to form a red iron(II) phenanthroline complex, dilute the reaction mixture, add varying amounts of iron standard (40 μg/mL), and measure the absorbance at 510 nm. With limited time and small children at home, the author created a simple spreadsheet that contained four columns. The first three asked students to enter the volumes of sample and standard to mix and to specify a volumetric flask size. The fourth column calculated the absorbance (A) on the basis of the student-supplied values using the dilution equation and the Beer−Lambert Law (eq 1):

Special Issue: Insights Gained While Teaching Chemistry in the Time of COVID-19

Received: June 3, 2020
Revised: July 29, 2020



Water %	50	Methanol %	50
Flow Rate / mL/min	1	Run time / min	10
[aspirin] / μg/mL	100	[acetaminophen] / μg/mL	100
[caffeine] / μg/mL	100	[excedrin] / μg/mL	0

Figure 1. HPLC simulation controls in the Jupyter notebook. Students were asked to vary the mobile phase composition, flow rate, and run time to optimize the separation to achieve baseline separation in a minimum run time.

$$A = 0.0234\left(\frac{85V_{\text{sample}}}{V_{\text{total}}} + \frac{40V_{\text{standard}}}{V_{\text{total}}}\right) + [\text{RAND}() \times 0.02 - 0.01] \quad (1)$$

Excel's RAND() function was used to add random scatter to the data. The students would record the "reading" and analyze their data to find the concentration of iron in the sample and in the theoretical vitamin tablet. The spreadsheet was set up to mimic our in-lab environment, where all of the solutions are made up in volumetric flasks for simplicity and consistency.

Problems Are Not Problems, but Features Induce Frustration

The author was worried about one problem: students could inspect eq 1 by clicking in the fourth column. A quick glimpse would identify the sample concentration of 85 μg/mL and a fictional extinction coefficient of 0.0234 $(\mu\text{g/mL})^{-1}$. A few minutes of time was spent attempting to hide the formula, but the author was unsuccessful. In the end there was no evidence that students shortcut the experiment this way. All of the students generated data points and plotted absorbance versus concentration of the standard added to find the analyte concentration as the x intercept. Furthermore, if a student did extract the solution concentration from the Excel equation, that could be seen as proof that the student understood the equations involved. The author's worry was unfounded, and the spreadsheet was used in the intended fashion by students. During review it was noted that there is a Protection tab under "Format Cells" that can be used to hide equations. This will be incorporated into next year's version of this spreadsheet.

A random offset was added to the absorbance, so that students' data would appear more realistic. The RAND() function in Excel returns a value between 0 and 1, so RAND() × 0.02 − 0.01 generates an offset of ±0.01 AU. This allows students to find the uncertainty in the slope using Excel's LINEST function and to propagate their uncertainty. The author thought that this feature would be a nice addition to the experiment and that students would be happy that their data would seem more authentic rather than a perfect fit to a line.

As usual, there was a disconnect between what faculty see as features and what students see as problems. The RAND() function recalculates its output whenever any cell is changed. Students attempted to use the same spreadsheet to generate and record their data, but entering their data into a separate cell caused the original data to recalculate. The author observed this behavior when creating the spreadsheet but decided that this was a nice feature, as it mimics normal instrument fluctuations. If the students left their cuvettes in the UV−vis spectrometer, the reading would fluctuate as well. The RAND() function caused a change of only a couple percent in the readings, and this should have given the students an idea of the uncertainty in the measurement. Multiple students complained about this behavior. They became frustrated that a value of 0.204 changed to 0.205 after it was recorded and saw this as an invalidation of their data. This same frustration occurs in the lab as electronic balances often fluctuate. Modern UV−vis instruments do not fluctuate, as students usually click a button to take one measurement instead of monitoring a continuous readout. In the future, part of the experiment will focus on having students explore natural fluctuations in measurements. An alternative is to set all calculations to manual (see the Supporting Information).

A final issue that arose involved the sample preparation part of the experiment. Despite a lab manual and video-recorded lab lecture explaining the preparation procedure, students did not understand how the sample was prepared. Understanding the sample preparation is required to calculate the original amount of iron (17 mg) in the vitamin tablet from the concentration of a diluted sample extract (85 μg/mL). They also lost out on the concept of derivatization with *o*-phenanthroline to make the extracted iron absorb at 510 nm. Understandably, students read just the information they need, and often background information is lost. The author will add more emphasis on the sample preparation portion of the lab manual and possibly a lab quiz or other activity to reinforce the concepts.

One benefit of this spreadsheet is that it is trivial to collect additional data points. Students will usually ask what the minimum number of data points is and whether they have "enough". Students did not ask that question this time, and one student calculated 20 data points for their standard addition plot! Because of this, the activity could be extended to include replicate analyses and a discussion of instrument precision, which is determined by the RAND() function above.

SIMULATION IN A PYTHON JUPYTER NOTEBOOK: ANALYSIS OF ANALGESICS BY HPLC

The second approach at engaging students was to have them simulate the use of an HPLC for the "Analysis of Analgesics by High Performance Liquid Chromatography" experiment.[6,7] Normally students would dilute standards and then crush and dissolve an Excedrin tablet and dilute the solution to create a sample. Analysis is by HPLC (isocratic) with UV detection. The author had more time available and desired to give students a more authentic experience. The idea and theory of creating an HPLC simulator was found in two manuscripts about Excel-based HPLC simulation.[8,9] Other HPLC simulators exist online,[10,11] but they show only the detector output (not inside the column), do not generate peak integrals for quantification of unknowns, and in some cases rely on outdated Java applications. Designing a new web-based HPLC simulator is an involved process: doing math in JavaScript, displaying graphs and videos on a webpage, and combining all of the parts into a functional user experience. The author looked for a unified platform that embraced a pleasant programming language. Creating the

HPLC simulator inside of a Python Jupyter notebook[12] was the answer. Jupyter notebooks provide an online platform for Python scripts. They can combine user controls to setup the HPLC method using ipywidgets,[12] math capabilities with the NumPy library,[13,14] and graphing and animations using Matplotlib.[15] The Jupyter notebook was deployed on Chem Compute at https://chemcompute.org/jupyterhub/.

Emulating an HPLC Method Editor with ipywidgets

Instead of asking students to vary the values of variables and rerun cells, ipywidgets allow one to create text box inputs that are mapped to variables in the notebook. Buttons allow students to run functions by clicking. These widgets were used to recreate a minimalistic HPLC method editor (Figure 1). Students were asked to vary the mobile phase composition, flow rate, and run time in order to optimize a separation of aspirin, caffeine, and acetaminophen in Excedrin. The students then quantified the components in Excedrin with their optimized settings.

Simulating and Visualizing the Separation

The simulation allows students to analyze the standards (aspirin, acetaminophen, and caffeine) at various concentrations. For example, if a student selects 100.0 μg/mL aspirin, the script will calculate a peak area of 100,554 mAU min (using a molar absorptivity that roughly recreates past students' data). A student uses the simulator to determine the peak areas at various concentrations (0, 5, 10, 50, and 100 μg/mL) and generates a standard curve for each of the three components. Students then analyze the sample (Excedrin) at various concentrations (corresponding to masses of an Excedrin tablet dissolved in the solvent). When students choose to add Excedrin, the Python script calculates the amount of each of the three components present in the Excedrin and adds that to any standards entered. Students from previous years reported a typical tablet mass of 810 mg (each tablet contains 250 mg of aspirin, 250 mg of acetaminophen, and 65 mg of caffeine according to the label). Students are asked to begin by dissolving 500 mg of the 810 mg tablet in a 100 mL volumetric flask and then to dilute 1:10 in a 10 mL volumetric flask. The student then enters this concentration (500.0 μg/mL Excedrin) into the simulator. The script internally converts that to a mixture of 154.3 μg/mL aspirin, 154.3 μg/mL acetaminophen, and 40.1 μg/mL caffeine and reports peak areas based on those values. Students are asked to reflect on whether the peak areas fall within their calibrated range (aspirin and acetaminophen are too high) and to dilute their Excedrin sample if needed. Then students use the simulated peak areas and convert back to concentrations using their standard curves. Students can combine sample and standard as well. A student could select 100.0 μg/mL aspirin and 100.0 μg/mL Excedrin. This would correspond to a spike and can help confirm the retention time of peak assignments.

The simulator was written in Python and follows Kadjo's algorithm.[9] One-dimensional arrays were used to represent the mobile phase and stationary phase for each compound. Each plate is represented by one column in each array. At each time step, the total amount of a compound is partitioned between the stationary and mobile phases in each plate by a predefined partition constant (K value). The K value was empirically set to roughly recreate separations from previous years (using an HP 1050 with a C-18 column/reversed-phase). K values were modified on the basis of the mobile phase composition using the relative polarities of water and methanol[6] and adjusted to reproduce past years' retention times. The mobile phase is propagated to the right by one plate per time step, and any compounds that eluted off the column are accumulated in the detector variable. Changes in flow rate are emulated by adjusting the length of the column (number of plates). The mobile phase array (nominally 100 plates × 100 time steps) is plotted as a graph and as a video animation (the array was copied many times to give the appearance of a 2D column cross section). This simple mathematical model gives a reasonably useful look at the bands in the column (Figure 2). Students can see the bands separate and spread as they run through the column and can see the detector graph register as they elute. The simulation completes in seconds, but generating the animation showing the bands in the column takes a couple of minutes. Finally, the detector output is graphed, and an integrated peak table is displayed (Figure 3). The main goal was to visualize the separation and allow students to quantify their unknown Excedrin tablet. A full treatment of the effect of K and the retention time was beyond the scope of this simulation, but other online simulators are available for this purpose (e.g., http://hplc-simulator.herokuapp.com).[10]

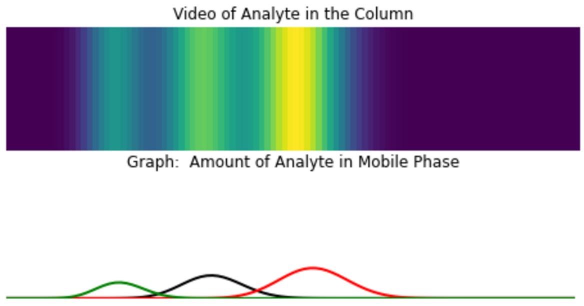
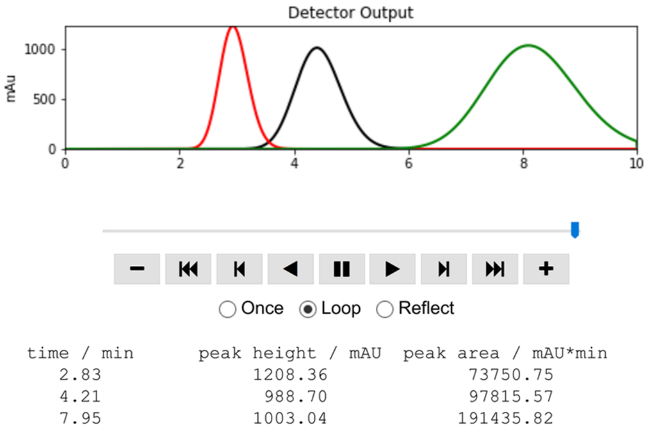


Figure 2. Simulated 2D cross section of the bands as they separate in an HPLC column (top) and a graph of the amount of each compound vs column plate (bottom). The graphs are animated. From left to right: caffeine, aspirin, acetaminophen.



```
time / min      peak height / mAU  peak area / mAU*min
  2.83               1208.36            73750.75
  4.21                988.70            97815.57
  7.95               1003.04           191435.82
```

Figure 3. Simulated detector output (top) and peak integral table (bottom). Peaks were calculated per compound, not retention time (due to limited development time). Peak order (from left to right): 2.83 min, acetaminophen; 4.21 min, aspirin; 7.95 min, caffeine. Peak times, heights, and areas include a random variance of ±1%.

Student Feedback (Survey Instrument)

Anecdotally, students reported that they enjoyed the exercise. Students ran the simulation multiple times to optimize the separation. Some viewed it as a challenge and wanted to beat the simulator. In a face-to-face lab situation, students get to touch the HPLC only once, so they do not have the ability to iterate and optimize. The simulation takes only a couple of minutes to simulate a run, which allows students to iterate faster than our instrument, which takes 10−20 min per run.

Students were surveyed on their perception of the simulator using a seven-point Likert scale (strongly disagree−strongly agree). The author was worried that the Python environment would intimidate the students, but only two out of 11 students responded that the simulator was intimidating because of the Python. Of the 11 respondents, six responded that the HPLC simulator made them want to learn how to program in Python or how to use computers in chemistry (three agree, three somewhat agree). One student continued to access the site as the author was writing the manuscript for this communication (2 weeks after the class final) to browse the other Jupyter notebooks available.

All of the respondents thought that the online HPLC simulation helped them to understand how compounds are separated in an HPLC (nine strongly agree, two agree). They also felt that the video animation of the inside of the column helped their understanding of compound separation in HPLC (10 strongly agree, one agree). Similarly, they all appreciated that the method control text input boxes used the same format as an actual HPLC (eight strongly agree, three agree). Respondents liked that they could access the online simulation at any time rather than only during lab hours. Several students reported difficulties in caring for others or limited access to technology, so online laboratories that were available 24/7 made it easier for the students to complete assignments. Finally, all of the students recommended including the HPLC simulator as a supplement to an in-person HPLC experiment.

Academic Integrity

The jupyterhub server creates a home directory for users when they log in, and this provided a quick check on the number of students who accessed the simulation. This number was less than the number of students who submitted lab reports. No investigations/actions were pursued. Many students reported problems during the pandemic: loss of internet access, difficulty finding a permanent residence, and mental health issues. Face-to-face laboratories were performed in pairs earlier in the semester, so students may have done the exercise together. Next semester the author will specify whether online laboratories should be performed individually or if group work is acceptable.

Shortcomings of the Simulation

The simulation permitted compound concentrations, mobile phase composition, flow rate, and run time to be set. Students reported that they were happy they could adjust the method and rerun the simulation. However, there was no time to program a mobile-phase gradient, so that topic was absent but could be added in time for next year's classes. Because of limited development time, the detector output was reported per compound. This means that a poor separation where all compounds overlap will give the same results as a proper separation. This is more difficult to implement and would involve summing all compounds to give a total chromatogram and then peak picking/integrating the chromatogram. Because students did not physically make solutions, they missed practicing the dilution technique. Some lab reports had trouble with calculating the amounts of aspirin, acetaminophen, and caffeine in the original tablet. This could be attributed to the loss of the sample preparation experience, but students in face-to-face laboratories consistently have this difficulty as well.

SUMMARY

Two approaches for generating data for online experiments during Spring 2020 were attempted in Sonoma State University's Quantitative Analysis class. The first generated absorbance data in Excel using the Beer−Lambert Law. The author was surprised that students reacted negatively to an added random offset in the calculated absorbance data. The second approach used a Jupyter notebook to simulate an HPLC separation. The Jupyter notebook provided the opportunity to visualize the separation of the compounds in the column, which gave students insight that they usually do not get in a face-to-face HPLC experiment.

HAZARDS

There are no hazards associated with these online activities.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available at https://pubs.acs.org/doi/10.1021/acs.jchemed.0c00565.

Standard addition Excel spreadsheet (XLSX)
Column separation animation (MP4)
Helpful notes from a reviewer to solve issues with hiding cells and enabling manual recalculation (PDF, DOCX)
HPLC student instructions (PDF, DOCX)

AUTHOR INFORMATION

Corresponding Author

M. J. Perri − *Department of Chemistry, Sonoma State University, Rohnert Park, California 94928, United States;* orcid.org/0000-0002-5007-1187; Email: mark.perri@sonoma.edu

Complete contact information is available at:
https://pubs.acs.org/10.1021/acs.jchemed.0c00565

Notes

The author declares no competing financial interest.
The simulation can be accessed online at https://chemcompute.org/jupyterhub/ (by launching a notebook server and accessing experiments/HPLC Simulator.ipynb). All students can run the simulator on the site for free.

ACKNOWLEDGMENTS

This work used the Extreme Science and Engineering Discovery Environment (XSEDE),[16] which is supported by the National Science Foundation (Grant ACI-1548562). This work used XSEDE Jetstream[17] at Indiana University through allocation TG-CDA170003. The author acknowledges invaluable help on the setup and usage of Jetstream by J. Fischer and B. Beck. A security review of Chem Compute was provided by the Science Gateways Community Institute Incubator Consulting Services.[18] This work approved by the SSU IRB (exempt category 2 application no. 3121). The author thanks a helpful reviewer for notes on how to hide an equation in Excel and how to set calculations to manual (to prevent automatic recalculation with RAND()).

REFERENCES

(1) Perri, M. J.; Weber, S. H. Web-Based Job Submission Interface for the GAMESS Computational Chemistry Program. *J. Chem. Educ.* **2014**, *91* (12), 2206−2208.

(2) Schmidt, M. W.; Baldridge, K. K.; Boatz, J. A.; Elbert, S. T.; Gordon, M. S.; Jensen, J. H.; Koseki, S.; Matsunaga, N.; Nguyen, K. A.; Su, S.; Windus, T. L.; Dupuis, M.; Montgomery, J. A. General Atomic and Molecular Electronic Structure System. *J. Comput. Chem.* **1993**, *14* (11), 1347−1363.
(3) Parrish, R. M.; Burns, L. A.; Smith, D. G. A.; Simmonett, A. C.; DePrince, A. E.; Hohenstein, E. G.; Bozkaya, U.; Sokolov, A. Y.; Di Remigio, R.; Richard, R. M.; Gonthier, J. F.; James, A. M.; McAlexander, H. R.; Kumar, A.; Saitow, M.; Wang, X.; Pritchard, B. P.; Verma, P.; Schaefer, H. F.; Patkowski, K.; King, R. A.; Valeev, E. F.; Evangelista, F. A.; Turney, J. M.; Crawford, T. D.; Sherrill, C. D. Psi4 1.1: An Open-Source Electronic Structure Program Emphasizing Automation, Advanced Libraries, and Interoperability. *J. Chem. Theory Comput.* **2017**, *13* (7), 3185−3197.
(4) Ponder, J.; Richards, F. Tinker Molecular Modeling Package. *J. Comput. Chem.* **1987**, *8*, 1016−1024.
(5) Phillips, J. C.; Braun, R.; Wang, W.; Gumbart, J.; Tajkhorshid, E.; Villa, E.; Chipot, C.; Skeel, R. D.; Kale, L.; Schulten, K. Scalable Molecular Dynamics with NAMD. *J. Comput. Chem.* **2005**, *26* (16), 1781−1802.
(6) Harris, D. C. *Quantitative Chemical Analysis*; Macmillan, 2010.
(7) Ferguson, G. K. Quantitative HPLC Analysis of an Analgesic/Caffeine Formulation: Determination of Caffeine. *J. Chem. Educ.* **1998**, *75* (4), 467.
(8) Fasoula, S.; Nikitas, P.; Pappa-Louisi, A. Teaching Simulation and Computer-Aided Separation Optimization in Liquid Chromatography by Means of Illustrative Microsoft Excel Spreadsheets. *J. Chem. Educ.* **2017**, *94* (8), 1167−1173.
(9) Kadjo, A.; Dasgupta, P. K. Tutorial: Simulating Chromatography with Microsoft Excel Macros. *Anal. Chim. Acta* **2013**, *773*, 1−8.
(10) Boswell, P. G.; Stoll, D. R.; Carr, P. W.; Nagel, M. L.; Vitha, M. F.; Mabbott, G. A. An Advanced, Interactive, High-Performance Liquid Chromatography Simulator and Instructor Resources. *J. Chem. Educ.* **2013**, *90* (2), 198−202.
(11) Boswell, P. G.; Stoll, D. R. http://hplcsimulator.org.
(12) Kluyver, T.; Ragan-Kelley, B.; Pérez, F.; Granger, B. E.; Bussonnier, M.; Frederic, J.; Kelley, K.; Hamrick, J. B.; Grout, J.; Corlay, S.; Ivanov, P.; Avila, D.; Abdalla, S.; Willing, C. Jupyter Notebooks—A Publishing Format for Reproducible Computational Workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*; Loizides, F., Schmidt, B., Eds.; IOS Press, 2016; pp 87−90.
(13) Oliphant, T. E. *A Guide to NumPy*; Trelgol Publishing USA, 2006; Vol. *1*.
(14) van der Walt, S.; Colbert, S. C.; Varoquaux, G. The NumPy Array: A Structure for Efficient Numerical Computation. *Comput. Sci. Eng.* **2011**, *13* (2), 22−30.
(15) Hunter, J. D. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.* **2007**, *9* (3), 90−95.
(16) Towns, J.; Cockerill, T.; Dahan, M.; Foster, I.; Gaither, K.; Grimshaw, A.; Hazlewood, V.; Lathrop, S.; Lifka, D.; Peterson, G. D.; Roskies, R.; Scott, J. R.; Wilkins-Diehr, N. XSEDE: Accelerating Scientific Discovery. *Comput. Sci. Eng.* **2014**, *16* (5), 62−74.
(17) Stewart, C. A.; Cockerill, T. M.; Foster, I.; Hancock, D.; Merchant, N.; Skidmore, E.; Stanzione, D.; Taylor, J.; Tuecke, S.; Turner, G.; Vaughn, M.; Gaffney, N. I. Jetstream: A Self-Provisioned, Scalable Science and Engineering Cloud Environment. In *XSEDE '15: Proceedings of the 2015 XSEDE Conference: Scientific Advancements Enabled by Enhanced Cyberinfrastructure*, St. Louis, MO, July 26−30, 2015; ACM: New York, 2015; article 29.
(18) Lawrence, K. A.; Zentner, M.; Wilkins-Diehr, N.; Wernert, J. A.; Pierce, M.; Marru, S.; Michael, S. Science Gateways Today and Tomorrow: Positive Perspectives of Nearly 5000 Members of the Research Community. *Concurrency Comput.: Pract. Exper.* **2015**, *27* (16), 4252−4268.