April 2022

Volume 13 Issue 1

JoCsE

Journal Of Computational Science Education

Promoting the Use of Computational Science Through Education

ISSN 2153-4136 (online)

Volume 13, Issue 1 **April 2022**

JOCSE

Journal Of Computational Science Education

The Journal Of Computational Science Education (JOCSE), ISSN 2153-4136, published in online form, is a supported publication of the Shodor Education Foundation Incorporated. Materials accepted by JOCSE will be hosted on the JOCSE website and will be catalogued by the Computational Science Education Reference Desk (CSERD) for inclusion in the National Science Digital Library (NSDL).

Subscription: JOCSE is a freely available online peer-reviewed publication which can be accessed at `http://jocse.org`.

Contents

Introduction to Volume 13 Issue 1

Steven I. Gordon
Editor
The Ohio State University
Columbus, OH
gordon.1@osu.edu

FOREWORD

This issue combines two regular submissions to the journal with refereed articles from several conference workshops in 2021. Our thanks to the guest editor Nitin Sukhija from Slippery Rock University of Pennsylvania, for managing the submission and review of the conference papers from HETET'21 at ISC 21, SEHET21 at PEARC21 and BPHTE21 at SC21.

Alm and Bailey provide a summary of a summer research experience for a diverse cross-section of students aimed at developing scientific research skills. Students completed a pre-workshop tutorial and then participated in a ten-week research project. The results show that overall students showed both improved skills and positive scientific career aspirations.

The student paper by Toth et al. summarizes a project where students created a heat map application aimed at visualizing bioinformatic data. Students learned a number of programming techniques and gained in project management and collaboration skills.

The first of the workshop papers by Fortin and O'Cais describe the Magic Castle project to create virtual HPC infrastructure environments for education and training. The article describes the underlying architecture and its application to several training programs.

In their article, He and Hartman-Baker provide an overview of the best practices used at the National Energy Research Supercomputing Center for virtual training through the COVID-19 pandemic. They utilized ten strategies in an effort to provide continued, high quality training opportunities for their users.

Cahill et al. describe a collaborative project Computational and Data Science Curriculum Exchange (C2Exchange) to address the challenges associated with sustained access to computational and data science courses in institutions with high percentage enrollment of students from populations currently under-represented in STEM disciplines. Seven institutions are piloting a program where basic courses in computational and data science are shared across institutions, expanding the opportunities for both students and faculty to integrate those topics into their curricula.

A computing course tailored to the needs of economics majors at Texas A&M University is the subject of the article by Lawrence et al. The course uses a web interface to introduce a number of programming concepts using Python. Exercises are oriented to topics in economics broadening the knowledge of those students in cyberinfrastructure topics.

Sherman et al. describe the work of the Northeast Cyberteam program that addresses the cyberinfrastructure needs for researchers at medium- and small-sized institutions in four northeastern states. Researchers propose projects where their computational and data needs have surpassed the capabilities of their laptops. A team using student facilitators and volunteer mentors work with the project for up to six months to help achieve the research objectives.

He et al. have created four "tech-labs" that introduce researchers to interactive computing environments applied to artificial intelligence and machine learning. Two labs introduce the tools and environment. This is followed by hands-on labs on data exploration and machine learning.

Finally, Lawrence et al. describe the use of a graphical user interface to facilitate adoption and use of HPC resources at Texas A&M University. Using Open OnDemand they have created a system which allows the use of Jupyter notebooks and interactive, containerized environments for instruction and research.

Scientific Skills, Identity, and Career Aspiration Development from Early Research Experiences in Computer Science

Cecilia O. Alm
Rochester Institute of Technology
coagla@rit.edu

Reynold Bailey
Rochester Institute of Technology
rjb@cs.rit.edu

ABSTRACT

The computer science research workforce is characterized by a lack of demographic diversity. To address this, we designed and evaluated an end-to-end mentored undergraduate research intervention to nurture diverse cohorts' skills for research and develop their vision of themselves as scientists. We hypothesized that this intervention would (a) grow scientific skills, (b) increase science identity, and (c) stimulate students to view scientific careers in computer science as future viable options. The evaluation of the hypotheses addressed the limitations in self-evaluation with a multicomponent evaluation framework, comprising five forms of evidence from faculty and students, engaging on team projects, with cohorts additionally participating in professional development programming. Results indicated that students gained in scientific skills and broadened their identity as scientists and, to some degree, strengthened their outlook on research careers. The introduced structured intervention and evaluation framework were part of a US National Science Foundation Research Experiences for Undergraduates (REU) computing-focused summer program at Rochester Institute of Technology and are applicable in other scientific disciplines and institutional settings.

KEYWORDS

Undergraduate computing research; Scientific skills, identity, and career aspirations; Diversity in research

1 INTRODUCTION

Despite being well-documented and extensively debated, underrepresentation among women, AALANA (African American, Latino/a American, Native American), and persons with disabilities in computer science continues to persist, according to US national statistics compiled by the National Science Foundation and the National Center for Science and Engineering Statistics [39]. The level of underrepresentation among these groups is striking in PhD education where the focus is on training graduates to pursue careers as scientists in academia, industry, and government. Table 1 highlights the disparity between the pre-pandemic 2018 US general population estimates, obtained from the the United States Census Bureau [51], and the percentage of doctoral degrees awarded at US public and private Computer Science departments reported by the Computing Research Association [14]. While we report on proportions from 2018 to emphasize that the information was not related to the pandemic, data from 2020 show similar trends. Moreover, while proportions may differ somewhat across data sources, evidence clearly points to a lack of diversity among computer science researchers. Furthermore, data analyzed by the Equity in Graduate Education Resource Center indicates that the proportion of women PhD graduates is lower in computer science compared to other science and engineering disciplines, with even more inequities when considering race and gender together [18].

Responding to the need for increasing diversity and inclusion in the computer science research workforce, we study the impact—for diverse student cohorts—of participating in an end-to-end undergraduate research training program. The theme of the program focused intellectually on sensing humans computationally, using hardware and software. We hypothesize that early research experiences in computing for diverse cohorts will:

(a) grow scientific skills,
(b) increase science identity, and
(c) stimulate students to view scientific careers in computer science as future viable options.

By *scientific skills*, we focus on research skills and knowledge for computer science, thus involving broader scientific skills such as teamwork and science dissemination, and also disciplinary-specific abilities such as computer programming and human subjects research competence in addition to core research process skills such as formulating research questions, finding relevant research literature, and articulating the limitations of different methodological approaches. As discussed by Kim and Sinatra [29], *science identity* is a complex socio-cultural construct, intertwined with self/other recognition as being a researcher and important for retaining people in research, following Carlone and Johnson [9]. We focus here on the perception of belonging, and development into belonging, to the community of practice of computer science research. We also attend to the goal and visions for future *scientific careers in computer science* by examining aspirations for graduate school and progression toward careers as scientists. As detailed below, to examine the above hypotheses, we utilize multi-component evaluation, using data from three years of demographically diverse undergraduate student cohorts and mentors who participated in the early mentored research intervention.

1.1 Early Mentored Scientific Experiences in STEM

It is widely acknowledged that undergraduate research experiences bring positive benefits. Hammack et al., who link mentoring to

https://doi.org/10.22369/issn.2153-4136/13/1/1

Table 1: Comparing population estimates from the 2018 US Census Bureau against the 2018 Taulbee Survey's information on awarded Computer Science doctoral degrees points to a striking underrepresentation of women and AALANA among computer scientists with terminal research degrees. This reaffirms the need for opening up research careers among underrepresented groups early with effective undergraduate research training programs that are evaluated holistically by considering multiple evidence and not just self-assessment.

Demographic	Population estimate	CS PhDs awarded
American Indian or Alaska Native	1.3%	0.1%
Black or African-American	13.4%	1.4%
Hispanic	18.3%	1.8%
Women	50.8%	19.3%
Persons with a disability	8.7%	Not reported

"coaching and apprenticeship" (p. 10), note that mentoring provides multiple positive outcomes for the undergraduate student receiving the mentee experience and cognitive development directly through research [25]. Additionally, analysis from biological/life sciences undergraduate research reveal that mentoring guidance influences students' perceptions of their own research competencies, as observed by Byars-Winston et al. [7]. Wright reported on responses to a survey of almost 800 students by the Computing Research Association's Center for Evaluating the Research Pipeline [56]. The results indicated that the proportion of computing students who applied to graduate school approximately doubled if undergraduates experienced research formally and also roughly doubled decisions to enroll in PhD studies. The National Academies of Sciences, Engineering, and Medicine summarized several considerations and findings about mentoring in undergraduate research in STEM fields such as the many types of mentors there can be, the many guises that mentors can adopt and skill development they can support, the results there can be for faculty and students, existing mentoring-related measurements, as well as the data linking mentoring and academic achievements [37]. Johann and Turbak highlighted the growth provided through a research experience [28] as it "enables undergraduates to make the transition from course-directed consumption of scientific knowledge to participation in the production of knowledge" (p. 280). They reported on a computer science undergraduate research program involving principles of "open-ended closed-endedness" (p. 285)–providing clear objectives for research while ensuring intellectual room for new discoveries–and "structured flexibility" (p. 286)–offering students clear guidance and intellectual freedom concurrently. Similarly, Skorinko emphasized the importance of research for stimulating students, and to engage diverse students, through scholarly experiences at an early stage of their undergraduate careers [46]. Cohoon et al. noted that more undergraduates benefiting from mentoring in research indicated higher likelihood to attend graduate school at highly regarded institutions [12]. In addition, Baker et al. summarized reported benefits to mentors of undergraduate researchers, and observed a positive link between faculty who mentored undergraduate researchers and faculty professional productivity such as being active in publishing research, in pursuing funded research, and in engaging in conferences [3]. Hall et al. observed the importance of mentors' and their institutional environments' recognition of such benefits [24]. In addition, Alvarado et al. reported on findings from analyzing sophomore (second-year) university students who majored in computer science or a related field while participating in a program structured around groups; when comparing to groups of control peers, outcomes pointed to benefits for participant students in terms of grade point average; and there were potential benefits in terms of being confident and growth of research interest [2]. The study presented here recognizes the centrality of mentoring and mentee-mentor interaction in early research experiences, operationalized by integrating both the perspective of students and faculty mentors for examining our hypotheses.

1.2 Diverse Students' Science Identity and Early Scientific Experiences

Prior STEM education literature recognizes mentored research experiences as a mechanism toward broadening participation. Barker noted that students who engage in research as undergraduates are more likely to pursue graduate degrees [5], and Estrada et al. emphasized that it aided students to stay in STEM [19]. Tamer and Stout conducted an analysis with women and men from underrepresented groups and arrived at four factors impacting study participants' intention to pursue academic professorial careers [48]. These included research project collaboration, learning about the graduate student experience and how to apply to graduate school, and gaining insight into how computer science research careers can impact society. Additionally, Tamer and Stout [48] observed that "[b]y promoting interest in the professorate among URMW [defined as "underrepresented men as well as women"] undergraduate students, REUs [research experiences for undergraduates] have the capacity to diversify the types of role models that future generations of women, and men from racial minority groups interact with and aspire to become" (p. 118).

Still, there is much room for improvement and growth of such formative training opportunities, and especially for new empirical study of outcomes from computer science undergraduate research training. Indeed, just two decades ago Johann and Turbak pointed out that computer science lacks the strong tradition of undergraduate research which exists in other science disciplines or in engineering fields [28]. Kim et al. observed that while most of their surveyed undergraduate research programs specifically targeted women, and also mostly successfully recruited a majority of women participants, the men in these programs still felt more confident in their abilities and accomplishments for moving forward in STEM disciplines post-program despite similar performance of the women students [30].

Results produced by Estrada at al. suggested that identifying as a scientist was highly important for downstream making STEM one's career, as opposed to other professions [19]. They noted that "higher education institutions that provide authentic experiences of belonging and inclusion, which are components of science identity, may be more likely to increase their URM [defined as "historically underrepresented minorities"] retention rates" (p. 11).

Follmer et al. noted that, compared to undergraduate research experiences organized by institutions, nationally-spanning programs were more diverse and also able to accommodate individuals without opportunities to engage in research experiences at their respective institutions [20]. The US National Science Foundation offers a program that enables Research Experiences for Undergraduates Sites (cohort-based seasonal programs), recognizing these programs as "an important means for extending high-quality research environments and mentoring to diverse groups of students" and it also aims to reach students "from academic institutions where research programs in STEM are limited" [38]. In general, for inclusive mentoring a diverse mentoring team is important. This includes the involvement of female role models as noted by Doerschuk [16]. Shamir noted the importance of "a broad range of interdisciplinary research topics that can engage and motivate students" (p. 15) [44]. Interdisciplinary projects can help peak computer science undergraduate students' interests. In particular, the program discussed focused on computational sensing research projects linked to motivating applications in impactful societal domains such as education, wellness, smart living, and leisure.

1.3 Science Skills and Identity Development in Mentored Experiences

The notion of "*cognitive apprenticeships*" discussed by Griese et al. [22] and earlier by Collins et al. [13] provided a theoretical lens in this study for examining growth of science skills and science identity in mentored research experiences in computing, and aspirations for pursuing scientific careers. Marra and Pangborn emphasized the value of apprenticeship for engineering skills [34], and Charney et al. reported on how elements of cognitive apprenticeship were adapted in high school-level research-infused experiences involving science, resulting in knowledge gains [10]. The present study connected mentoring to apprenticeship/coaching experiences in computer science research. Childress et al. highlighted the distinction between supervising and mentoring–while providing supervision and support for daily scientific activities is important, mentoring extends beyond supervision into supporting students to enter into and become familiar with the research community and envision themselves as a researcher, which may nurture interests to choose a research career path [11]. Additionally, the Social Cognitive Career Theory (SCCT), introduced by Lent et al. [32], relates to Bandura's *self-efficacy* concept [4] (in the context of the present study: confidence in one's competency to pursue a research career path). It can provide a basis for examining student attitudes about educational career choices as computer scientists as exemplified by Alshahrani et al.'s work [1]. Thiry et al., discussing socio-cultural theoretical perspectives, further remarked that development is influenced by participating in a community where activities are experienced and reflected upon directly [49]. Highlighting the benefits of team science, Johann and Turbak argued that structuring computer science undergraduate research as a collaboration endeavor nurtured students persevering in research activities beyond a time-limited program [28]. Holcomb et al. also observed a need for exposing students to collaboration in a brief programming-focused summer school [26]. Similarly, Sturner et al. noted the value of nurturing teamwork competency in research [47]. These theoretical foci and observations framed the present study.

1.4 Evaluating Science Skills and Science Identity

Shanahan et al. [45] observed that much prior work had been based on self-reporting, covering either perceptions of students, whose ratings of own skills may not correspond well to actual research training achievements, which have been shown to be better determined by faculty, as discussed by Griese et al. [22], or faculty perceptions of undergraduate mentoring. For example, Baker et al. explored faculty perceptions of enabling or limiting factors for undergraduate research mentoring based on focus group data [3]. Thiry et al. [49] noted that self-reporting might be particularly beneficial for understanding advancement in "confidence or interest in a subject" (p. 382), yet it is not the only way to measure educational gains. Instead, as discussed by Linn et al., evaluation of experiential research training will benefit from a comprehensive, holistic approach that relies on multiple, complimentary forms of evidence [33]. In addition, Griese et al. emphasized the importance of considering both roles—mentor and protégé—when studying mentoring in academic contexts [22].

The design and use of a holistic evaluation framework is one of the differentiators that sets the present work apart from prior studies on programs for undergraduate research training, in combination with its assessment of development of scientific skills, identity, and the exploration of seeing oneself as becoming part of a community of research practice in computer science. In contrast, for example Miller et al. focused on student self-report data [36]. We also go beyond the evaluation by Jelen et al. which included student self-report measurements (surveys, interviews), briefly summarized answers to a mentor survey that highlighted time commitments (time spent on mentoring undergraduates, coaching graduate mentors, and perception on usefulness of time investment), and individual "stakeholder insights" (p. 993) narratives [27]. Moreover, this work differs through its examination of an end-to-end early research experience intervention with integrated professional development activities. Considering five forms of evidence, this study assessed the development of research identities, skills, and aspirations for future participation in the computer science research community.

2 MATERIALS AND METHODS

Our mentored undergraduate research intervention and evaluation framework were deployed as part of a US National Science Foundation Research Experiences for Undergraduates summer research program at Rochester Institute of Technology. The intellectual research theme was focused on computational sensing of humans and the program recruited ten undergraduate students annually

from across the United States for a 10-week summer research experience. In addition to engaging in team-based research projects, participating students were also exposed to a suite of programmatic activities designed to grow scientific and career skills and stimulate their view of scientific careers as viable options.

2.1 Participants

Across three cohort years, there were 30 students in total from diverse demographic backgrounds, outlined in Figure 1. In addition, students came from 26 distinct institutions, with most characterized by limited opportunities for computer science research. While attempts were made to engage students from across the USA, more came from the US Northeast given its concentration of universities and colleges. The students participated in 15 team-based computer science research projects, with gender-balanced pairs of students being mentored per project. Of 15 faculty mentors who guided the research experiences, 40% participated all three years and one third in one year. A third of faculty were women. The mentors represented varying ethnic and national backgrounds.

2.2 Early Research Experience Intervention

For ten summer weeks, students experienced engaging in scientific practices. The research projects centered on sensing and analyzing human behaviors and cognitive processes using computer science. Additionally, a preparatory pre phase (e.g., completion of human subjects research certification and an introductory programming course) and a dissemination-focused post phase (with remote meetings with mentors) extended the 10-week experience. The approximate annual timetable of the early research experience in which students participated is in Figure 2. Prior work on undergraduate research training has also highlighted the importance of thoughtful, year-long logistical pre-program planning [55]. Our intellectual theme focused on basic research in the *acquisition*, *fusion*, and *analysis* of multimodal human sensing data. The ten weeks had a two-fold structure: (1) students conducted a team science research project with mentors and (2) they also participated in professional development activities as a cohort (see overview in the Appendix and Figure 3) centered on research and graduate school competencies and knowledge.

Research projects were structured as scaffolded team-science experiences spanning experimental design, data acquisition, analysis and inference with collected data including data visualization, and dissemination with deliverable milestones. Projects provided experience with human subjects experiments including collection and analysis of multimodal human-elicited data, and students participated in the Institutional Review Board (IRB) ethics review application process. The team science structure adopted in the training program involved regular student-student and student-faculty interactions.

Johann and Turbak highlighted the challenge of conducting a research project in a short time span such as just ten weeks [28]. To mitigate this issue, the *pre-program phase* engaged students with preparatory research activities — online human-subjects training and an online computer programming course. Faculty mentor teams also assigned project-relevant pre-readings to bring students up to speed on key relevant research before the program started. The pre-program activities required a reasonable time commitment, recognizing that students were also concluding their regular semester activities. The time-progression of the ten-week *in-program*

	2016-2018 average 3-yr target	2016	2017	2018	Average 3-yr actual
Cohort (+ other participants)	10	**10** (+ 1)	**10** (+ 1)	**10** (+ 2)	**10**
%Women (*43%)	20%	**50%**	**50%**	**50%**	**50%**
% URM (*37%)	20%	**30%**	**30%**	**30%**	**30%**
% Outside home institution	75%	**80%**	**90%**	**100%**	**90%**
% Computing majors	60%	**70%**	**80%**	**90%**	**80%**
% Limited CS research opportunities	60%	**60%**	**60%**	**60%**	**60%**
% Earlier than college junior	25%	**50%**	**30%**	**40%**	**40%**
% With disability	20%	**30%**	10%	**20%**	**20%**
% Other	20%	**30%**	10%	10%	17%

Figure 1: Student demographics. *Other participants* refer to additional undergraduates who participated in mentoring, research, and program activities but not in the evaluation. Comparison averages (*) are aggregates for nationwide undergraduate research training programs from Raicu et al. [41]. Green and bold: Met/exceeded the average 3-year targets set by the program organizers.

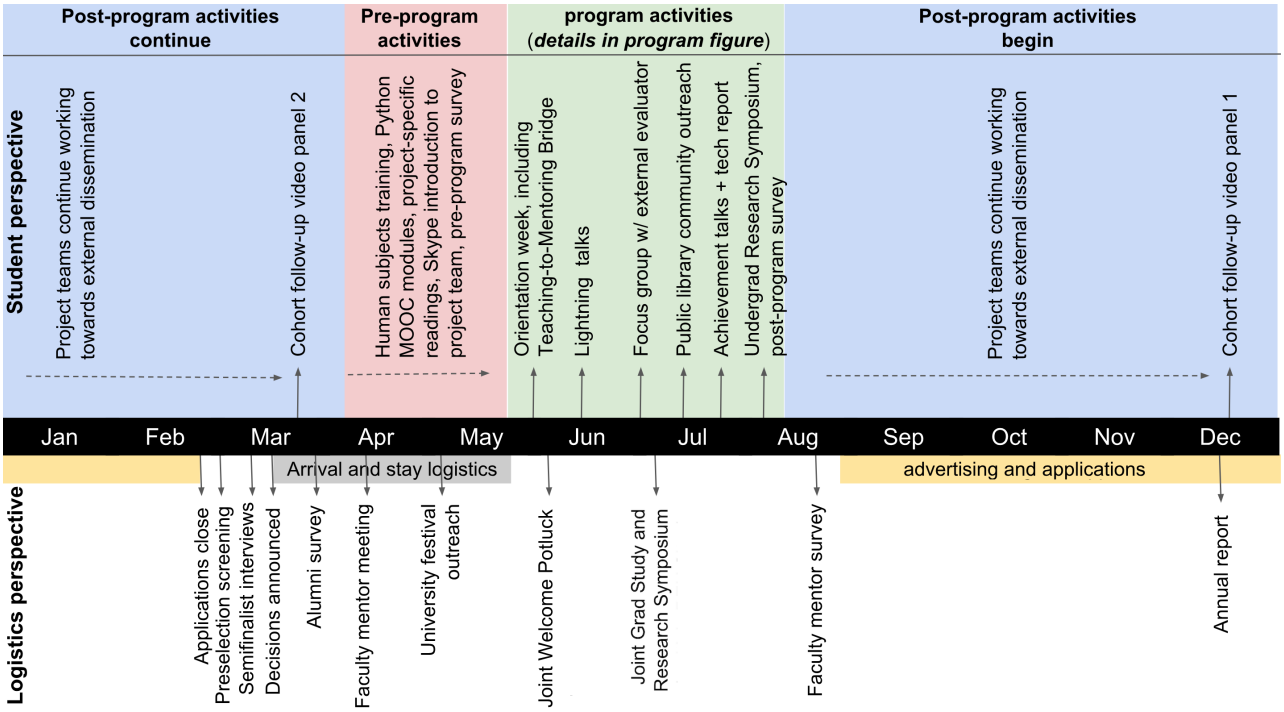


Figure 2: Timetable of annual program logistics. The upper half conveys the student perspective and the lower half organizational logistics. Program details (green interval) are expanded in Figure 4 for the 10-week program.

Figure 3: Professional development activities. A: Workshops with active learning-by-doing. B: Panel discussing grant writing for graduate school with students. C: Public STEM outreach event. D: Coordinated interdisciplinary symposium about graduate school and research. E: Post-program video follow-up. Faces have been blurred for anonymity.

experience is depicted in Figure 4. The leftmost column outlines progression in the research process. A *post-program phase* continued team interaction for external dissemination as well as cohort follow-up with video gatherings and support in the graduate school application process. Students were also encouraged to seek out new research experiences either at their home institutions after the program ended or at other universities. This was motivated by a finding by Estrada et al. [19] using a statistical model that two semesters in undergraduate research "uniquely predict overall science self-efficacy, identity, and values" (p. 10).

Based on a review of numerous publications, Walkington et al. converged on ten beneficial strategies when mentoring undergraduate researchers [52]. Almost all were operationalized structurally in the intervention. For example, we planned ahead for the research process in the pre-program phase; made expectations clear for anticipated outcomes during orientation; provided a challenging as well as emotionally supporting environment; promoted team-building with field trips and joint meals; encouraged gradually increased research independence; enabled networking opportunities in coordinated events and formally discussed disciplinary practices in a journal club and non-credit course; and provided support in scholarly dissemination post-program.

2.3 Multicomponent Evaluation

Inspired by the STEM and computer science mentoring and early research training literature, the evaluation framework integrated both subjective self-reported and objective program-level measures. We considered five forms of evidence (E1–E5) described below. Participating students and mentors consented to completing assessments, and this work was IRB-approved. We focus on measures of proportions and central tendency as well as qualitative responses; Linn et al. noted that experiential reflection was beneficial in undergraduate research experiences [33].

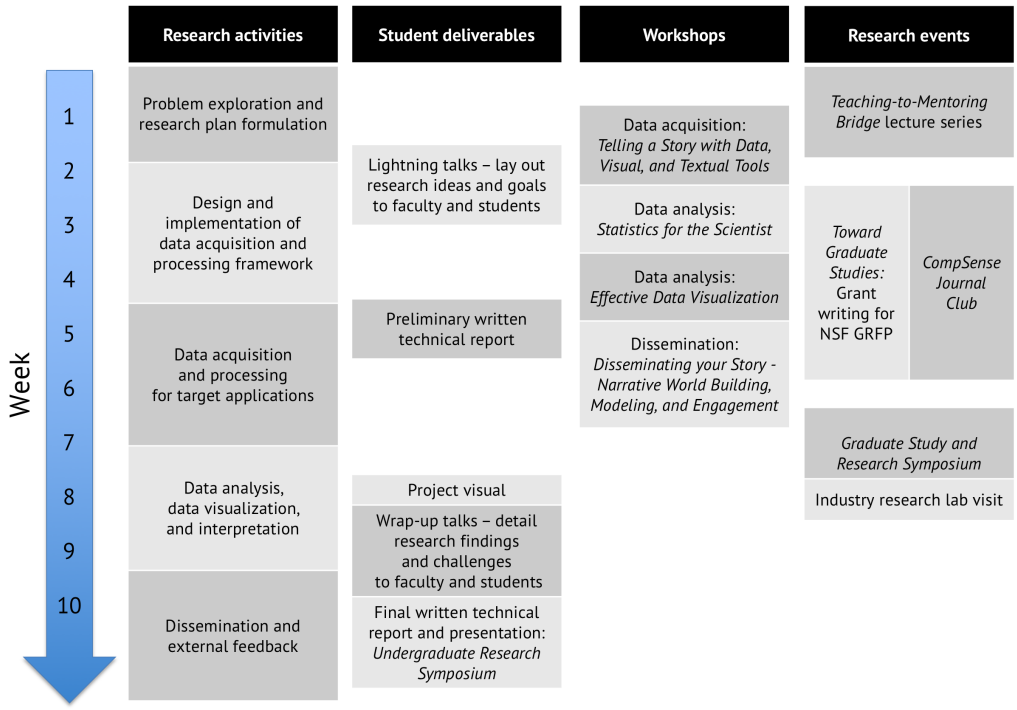


Figure 4: Timeline of the 10-week program, with sequenced steps of the research process. After the initial year, professional development activities were front-loaded to prioritize focus on research later in the program.

2.4 E1: Faculty-assessed Student Skills (Repeated In-program Measures)

Students were individually assessed in three formal *mentor reports* with 33 assessed skills using five-point Likert scale ratings (*strongly disagree* to *strongly agree*) spread out across the summer, administered approximately every three weeks. Skills assessed, in part adapted from a program at Willamette University [54], are listed in Table 2. For example, mentors assessed the student's ability to find, filter, and apply technical scientific information to their project, initiative-taking and time management skills, capability of formulating research questions, ability to deploy sound methodological practices, problem solving and creative skills, ability to link theory with empirical work, presentation and technical writing skills, etc. Additional assessed skills focused on the discipline such as computer software and hardware, software development and computer programming, etc. There were not multiple raters, and for this reason inter-rater reliability measurements are not reported. The analysis focused on whether individual skills, and how many skills in total, were assessed as increasing over the course of the program. (We do not compare the means since in this case, the Likert scales elicit subjective ratings and raters may adopt different calibration strategies.)

2.5 E2: Student Self-assessed Skills and Benefits (Pre and Post Measures)

A program-independent evaluator elicited student self-assessment data in pre and post surveys. As shown in Figure 2, the pre survey occurred right before the start of the program and the post survey in the 10th week of the program. Students assessed their level pre and post (*How would you rank your skill level in the following areas?*) for theoretical or applied scientific skills (research ethics, understanding and critiquing research literature, formulating research questions) and disciplinary-specific skills (programming and system development, human subjects research) on an ordinal 5-point scale. The analysis computed gains in the proportion of students that assessed themselves as being high-skilled, i.e., as *Master, Expert* or *Proficient* vs. low-skilled, i.e., *Familiar* or *Beginner*, given the difference between pre to post self-assessment. In addition, in the post survey, students self-assessed benefits of the early scientific research experience to their development of skills (*Please indicate to what extent the program benefited you for each*) on a 4-point ordinal scale (from *not at all* to *a great extent*). We computed the proportion of students reporting benefits (*somewhat* or *a great extent* ratings), leaving out those who did not (*not at all* or *very little* ratings). These benefits assessed involved scientific skills (such as learning ethical conduct in research, ability to analyze data and interpret results, tolerance for obstacles faced in research), scientific identity (increased academic self-confidence, understanding how researchers think and work on problems), and research career aspirations (readiness for more demanding research, increased confidence in potential for academic career). Lastly, two summative metrics focused on career plans—intent to pursue graduate school in a STEM field or a non-STEM field.

Table 2: Skills mentors assessed three times at approximately three-week intervals (n = 10; 30 annually). Items began with *The student...* except for S20, S22, S23, and S26. The right-most columns indicate items for which average ratings increased across all three assessment opportunities; this applied more to Y2 and Y3. Missing ratings could occur for lack of assessment evidence; these were excluded.

ID	Skill assessed	Y1	Y2	Y3
S1	is a capable programmer/software developer.		✓	✓
S2	is capable of finding relevant research resources (software, datasets, literature, etc.)		✓	✓
S3	is capable of adapting to unfamiliar software.		✓	✓
S4	is capable of adapting to unfamiliar equipment/hardware.		✓	✓
S5	takes initiative in research tasks beyond what is assigned.			✓
S6	demonstrates eagerness to learn about forms of data she/he has not worked with before.		✓	
S7	understands technical/scientific literature.	✓	✓	
S8	applies insights from research literature to their project.		✓	✓
S9	applies critical thinking in the research process.			✓
S10	demonstrates capability in formulating research questions answerable with data.		✓	✓
S11	demonstrates understanding of sound methodological procedures when setting up experimental work.	✓	✓	✓
S12	demonstrates knowledge or understanding of data analysis procedures.		✓	✓
S13	contributes own creative ideas to the project.		✓	✓
S14	can identify and articulate limitations in research design.		✓	✓
S15	is adept at problem-solving.		✓	✓
S16	is capable of linking theory with empirical work.			✓
S17	prepares effective scholarly presentation materials.	✓	✓	
S18	demonstrates adequate scholarly presentation skills.	✓	✓	
S19	demonstrates adequate scholarly writing skills.	✓		
S20	I can count on the student to meet scheduled commitments.		✓	✓
S21	gives advance notice if unable to keep scheduled tasks or appointments.	✓	✓	
S22	Overall, the student has good time management skills.		✓	
S23	Overall, the student is very dependable.		✓	✓
S24	shows genuine interest in the research project and process.		✓	
S25	takes ownership of the project.		✓	✓
S26	When the student 'gets stuck', she/he seeks paths forward by own initiative.		✓	✓
S27	conducts her/himself in a professional manner in face-to-face interactions.		✓	
S28	conducts her/himself in a professional manner in email correspondence.		✓	
S29	is capable of discussing scholarly concepts with team members.		✓	
S30	is respectful to different points of view.		✓	✓
S31	has the ability to make independent progress between mentor/team meetings.			
S32	contributes to a motivating research team experience.		✓	
S33	demonstrates thoughtfulness in making own decisions or seeking support for evolving directions of the project.		✓	
	Total number of questions for which average ratings increased across mentor assessments	6	28	19

2.6 E3: Faculty-assessed Student Outcomes and Benefits (Post Measure)

Faculty mentors individually completed a survey released immediately after the program ended. Faculty assessed student outcomes (*my students were able to make progress on working independently; my students were able to work on the project we identified for her/him in the way we envisioned*), as well as benefits for students of the early research experience in terms of scientific identity (*program helped students become better researchers; program gave my students a realistic understanding of life as a researcher*) or career aspirations (*program will help my students be more successful in graduate study; program helped my students develop career paths*). We computed the proportion of faculty that *strongly agreed* with corresponding statements.

2.7 E4: Student-led Disseminated Products (Post Measure)

As a loosely relevant objective measure, we tracked research dissemination outcomes: number of accepted refereed team publications at professional venues, number of students as lead author of publications, and number of student presentations about the research. While not directly reflecting perceptions of belonging, research products can be viewed as an objective validator and as overt recognition of belonging to the research community of practice. Presenting research on behalf of their teams can provide students with opportunities to network toward graduate school and research careers; such formative networking was identified as important by Shanahan et al. [45].

2.8 E5: Alumni-assessed Career Aspirations and Progression (Extended Post Measure)

We conducted an alumni survey which gathered facts and anecdotes of post-program activities in both structured and unstructured answer formats. Program alumni (former students) were invited back to complete the survey anonymously. This resulted in 39 individual responses over the three years such that 2016 participants (43.5% of total 39 responses) had three opportunities to complete the alumni survey, 2017 (43.5% of responses) had two opportunities, and 2018 (13% of responses) had one opportunity. Alumni provided information on career progression and engagement in scholarship dissemination. The survey was administered at a point after which graduate school applications would have been submitted and application decisions received. We computed the percent of total responses. Qualitative data from alumni's open-ended comments was also considered.

3 RESULTS

We used the multicomponent framework introduced in the prior section to report on results of the training intervention per three hypotheses about the development of scientific skills, scientific identity, and aspirations for research career. Given the sample size, we focused on descriptive measures and trends as well as qualitative discussion of open-ended answers.

3.1 E1 Results

The right-most three columns of Table 2 indicate per year the skills for which average ratings continuously improved over three assessments. Skills not indicated were assessed on average as having a flat or decreasing rating for at least one of the three assessments.

The research methodology skill (S11) had a continuous increase in all years, highlighting that the intervention supported its steadfast development. Several disciplinary and broader scientific skills also showed a continuous increase in ratings in two of the three years: S1 (computer programming), S2 (resourcefulness), S3 (new software), S4 (new hardware), S7 (technical/scientific literature), S8 (research literature), S10 (formulating research questions), S12 (data analysis), S13 (creative ideas), S14 (research design limitations), S15 (problem-solving), S17 (presentations preparation), S18 (presenting), S20 (accountability), S21 (notifying), S23 (dependability), S25 (taking ownership), S26 (initiative-taking), and S30 (respectfulness). The last two cohort years indicated a more pronounced improvement trend, applicable to 19 or 28 of the assessed skills.

In contrast, skills related to communication (S27–S29) or to other scholarly skills (S5 — initiative taking; S6 — eagerness to explore unknowns; S9 — application of critical thinking; S16 — ability to connect theory and empiricism; S19 — scholarly writing; S22 — time management; S24 — genuine interest; S31 — independence; S32 — motivating a research team; S33 — thoughtful decision-making) did not show as consistent improvement.

3.2 E2 Results

As seen in Figure 5, pre and post comparison of student self-assessed scientific skills showed gain for, especially: *formulating research questions*, *developing a research plan*, *data collection and human subjects considerations*, *data processing*, *understanding and critiquing research literature, research ethics*, and *preparing spoken and written research dissemination*. In contrast, two skills were self-assessed as declining in one year: *grant writing* in Y3 and *programming and application/system development* in Y1. In addition, Figure 6 reveals that students, when completing the post survey, perceived that their abilities generally increased, and especially in Y3. For instance, in Y3, 100% felt that they improved their *ability to critique research literature, ability to analyze data and interpret results*, *skills to effectively disseminate findings*, knowledge of *tools and techniques in the field*, and their *tolerance for obstacles in the research process* either *somewhat* or *to a great extent*. In terms of development of scientific identity, a majority of students reported gains in *increased academic self-confidence* and *understanding how researchers think and work on problems*. In addition, there were gains for *increased confidence in potential for academic career* and *readiness for more demanding research*, relating to research career aspirations and career progression. Additionally, results for summative metrics about career intent are in Table 3. Most participating students intended to pursue graduate study and the majority in STEM fields.

3.3 E3 Results

Figure 7 shows that faculty felt the intervention contributed to student development, although with a lower proportion of *strongly agree* in one year. Faculty also identified benefits for students in terms of developing scientific identity and positive career aspirations toward the scientific profession. For career aspirations, responses affirmed stronger benefit for preparation for graduate school than for careers in research generally. On average, faculty above all indicated they *strongly agreed* that the program *will make students more successful in graduate study* (i.e., research career aspirations and progression), and that the experience helped students *become better researchers* (i.e., scientific identity). The lowest average proportion of *strongly agreed* among faculty respondents was for *gave my students a realistic understanding of life as a researcher* (scientific identity) and *helped my students develop career paths* (research career aspirations and progression).

Table 3: Students' self-reported intent to pursue graduate school (n = 10 annually; 30 total).

Evidence from student evaluation	Y1	Y2	Y3	Mean
1. Intend to pursue grad school in STEM	50%	80%	70%	67%
2. Intend to pursue grad school in non-STEM	30%	10%	10%	17%

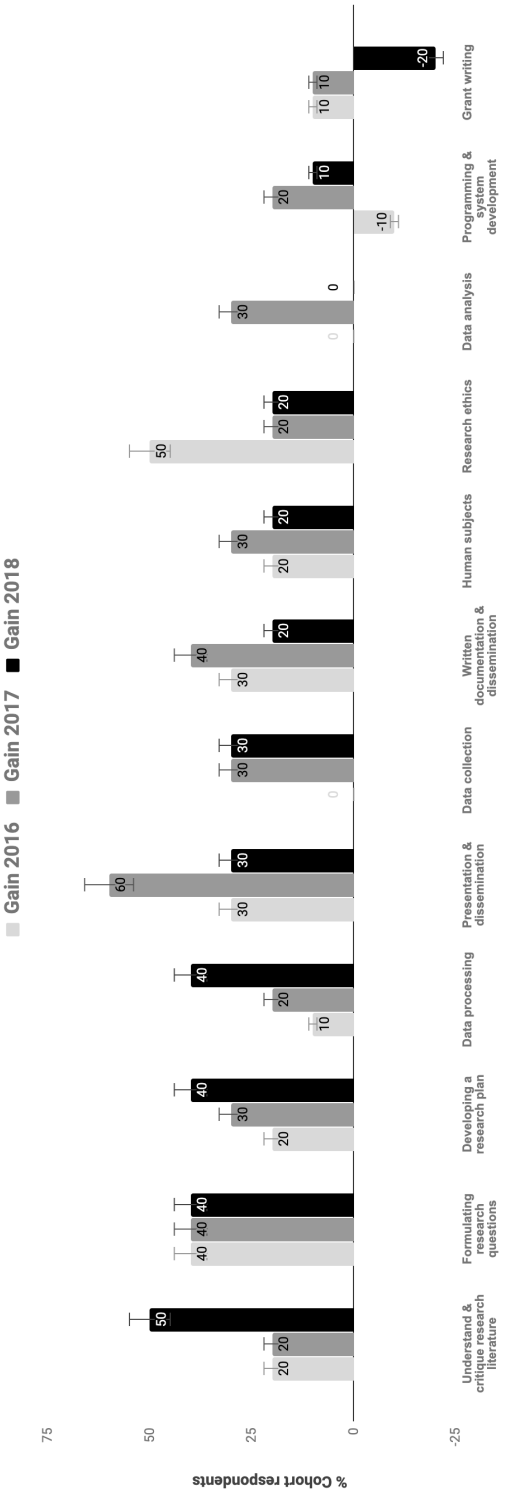


Figure 5: For all three years, students' self-assessment of their skills show cohort gains in most skills in pre and post program comparison. Gain was computed as the difference between the post and pre upper-end skill ratings. (Proportion of n = 10 annually; 30 total; Y2 and Y3 had 9 post responses.)

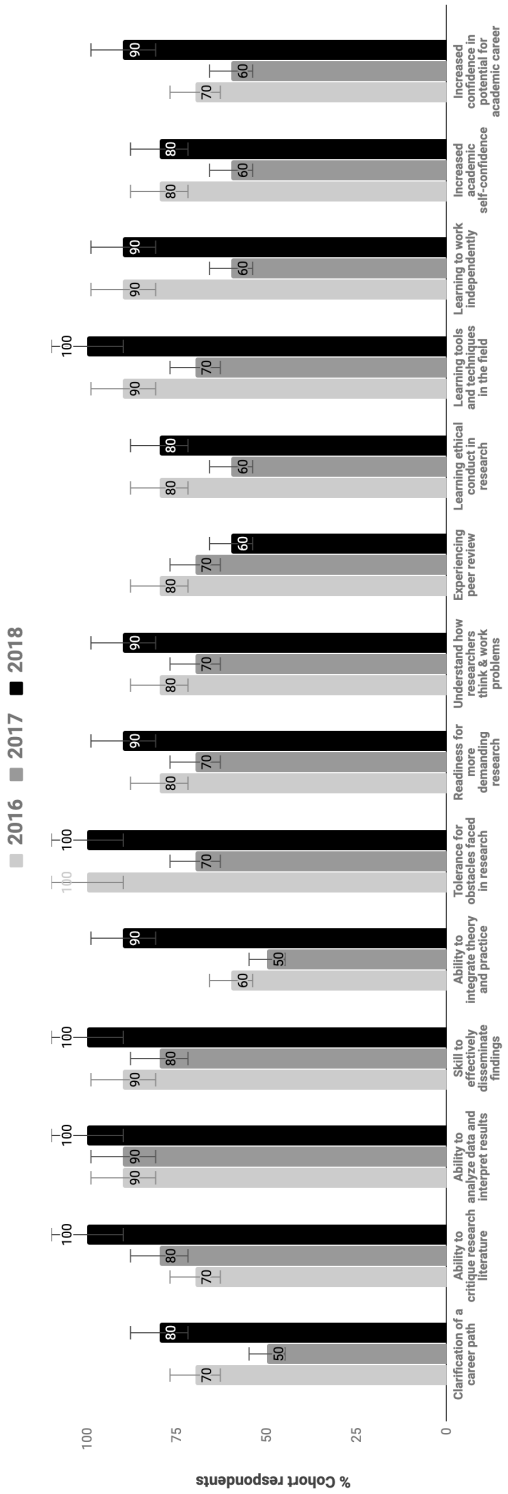


Figure 6: For all three years responses from the student post-program survey shows that the majority of student cohorts perceived gains in skills *somewhat* to *a great extent* in many areas. (Proportion of n = 10 annually; 30 total).

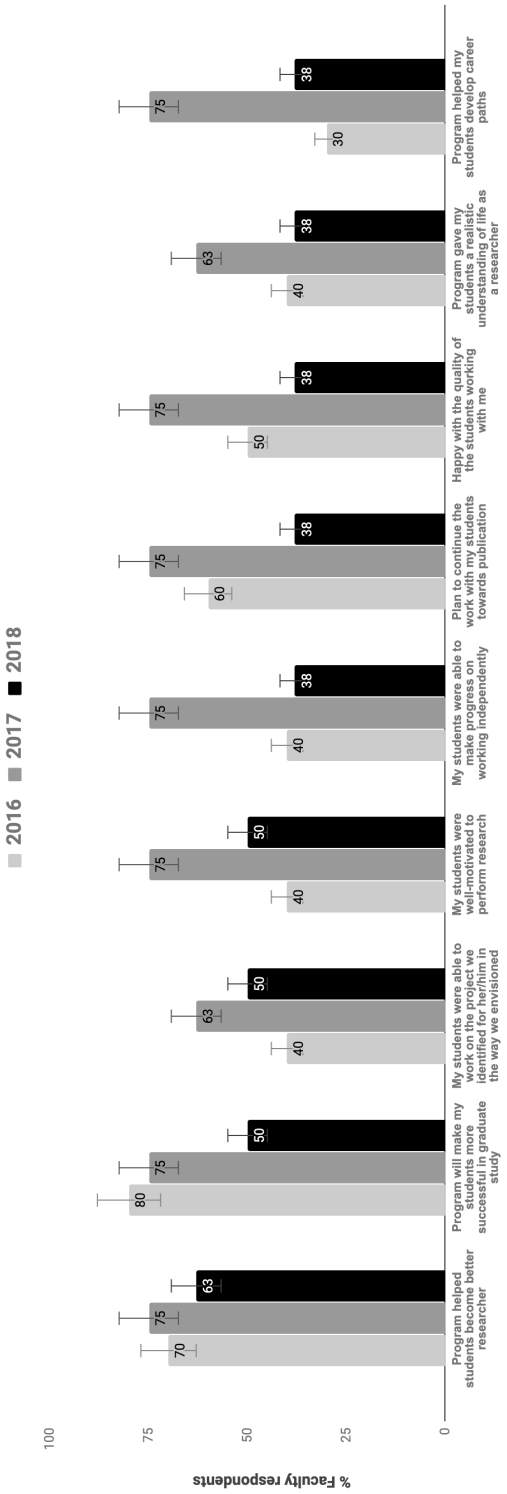


Figure 7: Many faculty *strongly agreed* with statements about student benefits across the three years; others generally *agreed*. (Proportion of n = 10 in Y1 vs. n = 8 in Y2 to Y3.)

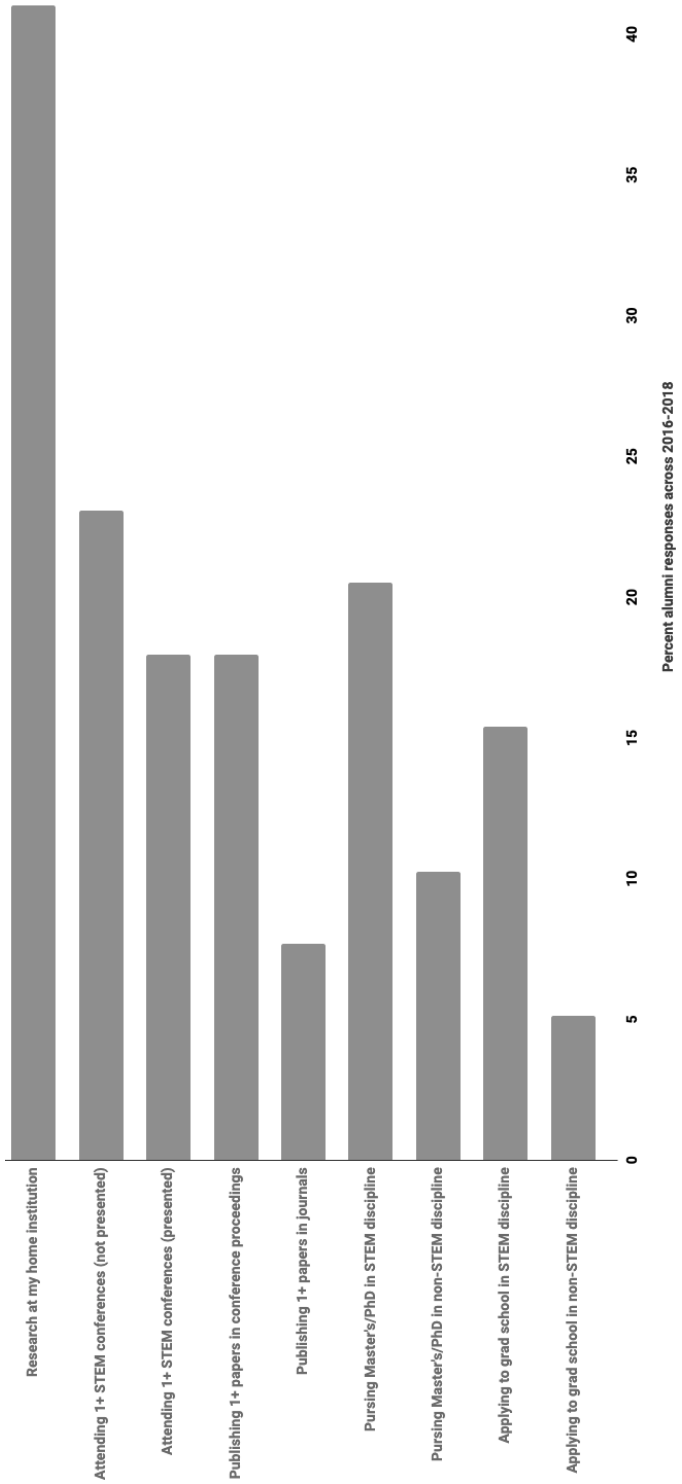


Figure 8: Percent program alumni responses self-reporting career and activities choices for three years (n = 39 total responses). 1+ = one or many.

3.4 E4 Results

The research projects placed computational sensing in the center of a research experience, translating into refereed dissemination of research discoveries, including eleven refereed technical team publications with students as lead authors [8, 17, 21, 23, 31, 35, 40, 42, 43, 50, 53], and two other student-led technical publications [6, 15]. There were additionally 30 pre-publication student-presented talks or posters at a local undergraduate research event, as well as annual pre-publication posters at a national undergraduate research symposium.

3.5 E5 Results

Aggregated results of alumni surveys over the three years are in Figure 8. Students continued to *engage in research after returning to their home institution*. In addition, of the responses, almost a fifth reported they *presented research at a STEM conference* and/or *published a conference proceedings paper*. Over one fifth of the responses also indicated that respondents were *attending graduate school in STEM disciplines*, and/or that respondents *applied or intended to apply to graduate school in a STEM discipline*. Graduate program aspirations included computer science, human-computer interaction or human factors, robotics, computational physics, psychology, game design and development, mathematics, computer engineering or other engineering disciplines.

Additional open-ended comments about recent activities or career plans indicated impact of the research experience for future career visions in computer science and STEM disciplines:

> *[This] program was mentioned in all of my formal and informal interviews for grad school, so I give it a lot of credit for helping me achieve my educational goals.*

> *If it weren't for [this program] and my mentors, I may never have considered applying to get my MS, and my short term career prospects may have been very different.*

> *I was accepted to 2 PhD programs and took a position in [program] at [university]. I started in August, and my research focuses primarily on perception-action and robotics with funding from [organization], technology use and design for people with Autism Spectrum Disorders, and cybersecurity. I've completed one semester, going on two, with research submitted for publication with multiple Human Factors-related conferences and journals.*

> *I intend to earn a PhD in Computer Science and then see where that takes me. I might try to become a professor some day.*

> *I attended [conference], the yearly conference of the [organization], where I attended various career building sessions and got an internship offer from [organization]. In addition, I travelled to IEEE [conference] with my research partner to present our research from our summer [program]. Outside of STEM pursuits, I have resumed a job writing for my school's newspaper ...*

> *Obtain a PhD in Computer or Mechanical Engineering.*

> *I am fairly set on pursuing a PhD after finishing undergrad, although I'm not sure if I want to ultimately go into industry or academia. I am interested in AI for robotics or medical applications and computational physics, and I will probably end up pursuing my PhD research in one of those areas.*

> *... I attended the Grace Hopper [women in computing] conference on a full scholarship, presented research at the [...] Symposium, and completed my computer science degree. Currently I am in a fully-funded machine learning bootcamp program organized by [organization].*

4 DISCUSSION

We reported on the impact of an early research experience for diverse cohorts of undergraduates in computing for growing scientific skills, increasing science identities, and fostering career aspirations. Figure 9 integrates results from all forms of evidence in our evaluation framework. They indicated the benefits of the intervention.

For hypothesis (a), evidence from faculty assessments of students and student self-assessments (E1, E2) indicated that the intervention resulted in gains in scientific skills. That communication and abstract skills were not reported to consistently improve may reflect that extended practice is required to develop them. Moreover, the difference in Table 2 from Y1 compared to Y2–Y3 in the number of consistently improving skills may reflect adjustments made in response to student/faculty feedback in the training schedule, including in front-loading the schedule of professional development activities in the experience which appeared to have better supported skills development.

For hypothesis (b), evidence from students and faculty (E2, E3) indicated that the early scientific experience contributed to scientific identity development. Additionally, the dissemination data (E4) arguably represented a tangible recognition of belonging to the scientific community. That faculty rated students' development of a realistic understanding of research lower may relate to the compressed timeframe of the intervention and that a 10-week research project is not reflective of the typical duration of graduate research projects.

As regards hypothesis (c), evidence from faculty (E3) suggested that the intervention nurtured scientific career progression, especially in terms of preparing students for graduate school, the next step in their careers. Considering the SCCT theoretical framework [32], the majority of students indicated an intent to pursue graduate school in STEM (E2) and also a confidence boost about their capacity for an academic career. These were indicators that the program promoted scientific career aspirations, as were alumni

Five Forms of Evidence				
				
E1	E2	E3	E4	E5
Faculty-assessed student skills	Student self-assessed skills and benefits	Faculty-assessed student outcomes and benefits	Student-led disseminated products	Alumni-assessed career aspirations and progression
repeated in-program measures	*pre and post measures*	*post measure*	*post measure*	*extended post measure*
Key Findings				
There was a rising trend in faculty-assessed student skills over ten weeks, in Y2 and Y3. Skills related to communication or abstract scholarly skills did not show as consistent an improvement.	Most participating students reported plans to pursue graduate study in STEM fields. Students reported gains in scientific skills, science identity, and research career aspirations.	Faculty felt that the experience positively impacted student growth. Faculty confirmed that the experience benefited students' preparation for graduate study.	The intervention resulted in refereed dissemination with most students as lead authors. Students presented research at local, national, and international venues.	Many students continued to engage in research after the program. Close to one-third have commenced or been accepted to PhD programs. Others have continued to Masters.

Figure 9: Overview of evaluation framework components and key findings. A comprehensive multicomponent evaluation framework is adopted to avoid pitfalls associated with self-reporting alone.

responses about planned applications or confirmed acceptances to graduate school. Additional post-contact with alumni completes the picture for hypothesis (c). Approximately a third have already continued on to PhD programs in STEM, and additional students to Master's programs. Alumni have, in particular, placed into graduate programs in computer science or closely related fields, such as degree programs with an interdisciplinary computing focus.

In addition, regarding the need for increasing diversity in computing and in other STEM fields, comments from students in the diverse cohorts acknowledged the importance of diversity and inclusion in the practice of research:

> *Diversity is vital, particularly when the research involves human subjects, because human populations are diverse. The researchers should reflect their work. Additionally, diversity allows for unique experiences and thus ideas and viewpoints.*

> *Diversity can help prevent implicit biases that may be held by a particular group without them even realizing it.*

> *[H]aving a range of perspectives overseeing a project is essential for the best possible outcome.*

> *[Diversity is] very important. It's always nice to see someone who looks like you or have a similar upbringing in the same field of study, especially computer science. Diversity is also important because it allows for different perspectives to be brought to a project or research assignment.*

> *I feel that having a variety of experiences and backgrounds always helps any project, whether in research or otherwise. To this extent, diversity is extremely important as it creates the opportunity for a wealth of ideas and viewpoints to be expressed. Additionally, I believe that people do their best work when they are in an environment they feel welcome in, and having diverse mentors and peers supports this.*

Revisiting the cognitive apprenticeship framework discussed earlier, these comments further emphasize the link between effective apprenticeship and coaching with inclusive mentoring environments, and they highlight the importance of collaborative diversity

in research teams. That students recognize the discipline's challenges and its present limitations in diversity, equity, and inclusion can also be regarded as an indicator of their growing connection and self-identification with the field.

5 CONCLUSION

We reported on and discussed an intervention that provided diverse student cohorts with a computer science-focused early scientific experience. Integrating multiple sources of evidence supported that the intervention helped develop scientific skills and identity, in addition to nurturing research career aspirations and progression. Several participants have already continued on to PhD programs in computer science or STEM. Targeted recruitment outside of the US Northeast may improve geographic diversity of participants. The introduced intervention—see overviews in Figures 2, 3, and 4, and the Appendix—is applicable in other disciplines and institutional settings. Resource needs would primarily include cost and time involved. For example, in the reported study, students received internship stipends, subsistence, and travel support; and the organizers and faculty of the training intervention invested their time before, during, and after the 10-week experience. In future work, we are interested in studying potential long-term implications for undergraduate researchers who publish the results of their research and subsequently pursue a research profession, including topics such as downstream research career productivity and the mentoring practices they adopt as research professionals.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Award No. IIS-1559889. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] Amnah Alshahrani, Isla Ross, and Murray I. Wood. 2018. Using Social Cognitive Career Theory to Understand Why Students Choose to Study Computer Science. In *Proceedings of the 2018 ACM Conference on International Computing Education Research (ICER '18)*. Association for Computing Machinery, New York, NY, USA, 205–214. https://doi.org/10.1145/3230977.3230994

[2] Christine Alvarado, Sergio Villazon, and Burcin Tamer. 2019. Evaluating a Scalable Program for Undergraduate CS Research. In *Proceedings of the 2019 ACM Conference on International Computing Education Research (ICER '19)*. Association for Computing Machinery, New York, NY, USA, 269–277. https://doi.org/10.1145/3291279.3339406

[3] Vicki L. Baker, Meghan J. Pifer, Laura G Lunsford, Jane Greer, and Dijana Ihas. 2015. Faculty as Mentors in Undergraduate Research, Scholarship, and Creative Work: Motivating and Inhibiting Factors. *Mentoring and Tutoring: Partnership in Learning* 23, 5 (2015), 394–410. https://doi.org/10.1080/13611267.2015.1126164

[4] Albert Bandura. 1986. *Social Foundations of Thought and Action: A Social Cognitive Theory*. Prentice-Hall, Englewood Cliffs, N.J.

[5] Lecia Barker. 2009. Student and Faculty Perceptions of Undergraduate Research Experiences in Computing. *ACM Transactions on Computing Education* 9, 1, Article 5 (2009), 28 pages. https://doi.org/10.1145/1513593.1513598

[6] Gustaf Bohlin, Kristoffer Linderman, Cecilia Ovesdotter Alm, and Reynold Bailey. 2019. Considerations for Face-based Data Estimates: Affect Reactions to Videos. In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications - Volume 2: HUCAPP*. INSTICC, SciTePress, Prague, Czech Republic, 188–194. https://doi.org/10.5220/0007687301880194

[7] Angela M. Byars-Winston, Janet Branchaw, Christine Pfund, Patrice Leverett, and Joseph Newton. 2015. Culturally Diverse Undergraduate Researchers' Academic Outcomes and Perceptions of Their Research Mentoring Relationships. *International Journal of Science Education* 37, 15 (2015), 2533–2554. https://doi.org/10.1080/09500693.2015.1085133

[8] Alexander Calderwood, Elizabeth A. Pruett, Raymond Ptucha, Christopher M. Homan, and Cecilia Ovesdotter Alm. 2017. Understanding the semantics of narratives of interpersonal violence through reader annotations and physiological reactions. In *Proceedings of the Workshop on Computational Semantics beyond Events and Roles (at EACL)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 1–9.

[9] Heidi B. Carlone and Angela Johnson. 2007. Understanding the Science Experiences of Successful Women of Color: Science Identity as an Analytic Lens. *Journal of Research in Science Teaching* 44, 8 (2007), 1187–1218. https://doi.org/10.1002/tea.20237

[10] Jeff Charney, Cindy E. Hmelo-Silver, William Sofer, Lenore Neigeborn, Susan Coletta, and Martin Nemeroff. 2007. Cognitive Apprenticeship in Science through Immersion in Laboratory Practices. *International Journal of Science Education* 29, 2 (2007), 195–213. https://doi.org/10.1080/09500690600560985

[11] Herb Childress, Gloria C. Cox, Susan B. Eve, Amy J. Orr, and Julio Rivera. 2009. Mentoring as a Socializing Activity - Supporting Undergraduate Research in the Social Sciences. Mentoring in the Social Sciences. https://www.cur.org/assets/1/7/Socializing.pdf.

[12] Joanne McGrath Cohoon, Margaret Gonsoulin, and James Layman. 2004. Mentoring Computer Science Undergraduates. In *Human Perspectives in the Internet Society: Culture, Psychology and Gender*, K. Morgan, J. Sanchez, C. A. Brebbia, and A. Voiskounsky (Eds.). WIT Press, Southampton, UK, 199–208.

[13] Allan Collins, John Brown, and Ann Holum. 1991. Cognitive Apprenticeship: Making Thinking Visible. *American Educator* 15, 3 (1991), 6–11, 38–46.

[14] Computing Research Association. 2018. 2018 Taulbee Survey. https://cra.org/wp-content/uploads/2019/05/2018_Taulbee_Survey.pdf.

[15] Yancarlos Diaz, Cecilia Ovesdotter Alm, Ifeoma Nwogu, and Reynold Bailey. 2018. Towards an Affective Video Recommendation System. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, Athens, Greece, 137–142. https://doi.org/10.1109/PERCOMW.2018.8480130

[16] Peggy Doerschuk. 2004. A Research and Mentoring Program for Undergraduate Women in Computer Science. In *Proceedings of the 34th ASEE/IEEE Frontiers in Education Conference*. IEEE, Savannah, GA, USA, S2H–7. https://doi.org/10.1109/FIE.2004.1408747

[17] Ashley A. Edwards, Anthony Massicci, Srinivas Sridharan, Joe Geigel, Linwei Wang, Reynold Bailey, and Cecilia Ovesdotter Alm. 2017. Sensor-based Methodological Observations for Studying Online Learning. In *Proceedings of the 2017 ACM Workshop on Intelligent Interfaces for Ubiquitous and Smart Learning (SmartLearn '17)*. ACM, New York, NY, USA, 25–30. https://doi.org/10.1145/3038535.3038536

[18] Equity in Graduate Education Resource Center. Accessed July 2021. Equity in Graduate Education: Numbers.

[19] Mica Estrada, Paul R. Hernandez, and P. Wesley Schultz. 2018. A Longitudinal Study of How Quality Mentorship and Research Experience Integrate Underrepresented Minorities into STEM Careers. *CBE Life Sciences Education* 17, 1 (2018), ar9.

[20] D. Jake Follmer, Sarah Zappe, Esther Gomez, and Manish Kumar. 2019. Student Outcomes from Undergraduate Research Programs: Comparing Models of Research Experiences for Undergraduates (REUs). *Scholarship and Practice of Undergraduate Research* 1, 1 (2019), 20–27.

[21] Aliya Gangji, Trevor Walden, Preethi Vaidyanathan, Emily Prud'hommeaux, Reynold Bailey, and Cecilia Ovesdotter Alm. 2017. Using Co-captured Face, Gaze and Verbal Reactions to Images of Varying Emotional Content for Analysis and Semantic Alignment. In *Proceedings of AAAI Workshop on Human-Aware Artificial Intelligence*. The AAAI Press, Palo Alto, California, 621–627.

[22] Emily R. Griese, Tracy R. McMahon, and DenYelle B. Kenyon. 2017. A Research Experience for American Indian Undergraduates: Utilizing an Actor-partner Interdependence Model to Examine the Student-mentor Dyad. *Journal of Diversity in Higher Education* 10, 1 (2017), 39–51.

[23] Nikita Haduong, David Nester, Preethi Vaidyanathan, Emily Prud'hommeaux, Reynold Bailey, and Cecilia Ovesdotter Alm. 2018. Multimodal Alignment for Affective Content. In *Proceedings of AAAI Workshop on Affective Content Analysis*. The AAAI Press, Palo Alto, California, 21–28.

[24] Eric E. Hall, Helen Walkington, Jenny Olin Shanahan, Elizabeth Ackley, and Kearsley A. Stewart. 2018. Mentor Perspectives on the Place of Undergraduate Research Mentoring in Academic Identity and Career Development: An Analysis of Award Winning Mentors. *International Journal for Academic Development* 23, 1 (2018), 15–27. https://doi.org/10.1080/1360144X.2017.1412972

[25] Jennifer Hammack, Robin Lewis, Rebecca McMullen, Caitlin Powell, Rosalie Richards, Doreen E. Sams, and Jeanetta Sims. 2017. *Mentoring Undergraduate Research Handbook, 2nd Edition*. Undergraduate Research and Creative Endeavors (URACE) Georgia College Knowledge Box, Georgia College & State University.

[26] Katherine Holcomb, Jacalyn Huband, and Tsengdar Lee. 2020. Lessons Learned from the NASA-UVA Summer School and Internship Program. *The Journal of*

Computational Science Education 11 (2020), 3–7. Issue 1. https://doi.org/10.22369/issn.2153-4136/11/1/1

[27] Ben Jelen, Julia Dunbar, Susan Monsey, Olivia K. Richards, and Katie A. Siek. 2019. Utilizing the Affinity Research Group Model in a Summer Research Experience for Undergraduates Program. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19)*. Association for Computing Machinery, New York, NY, USA, 990–996. https://doi.org/10.1145/3287324.3287501

[28] Patricia Johann and Franklyn A. Turbak. 2001. Lumberjack Summer Camp: A Cross-Institutional Undergraduate Research Experience in Computer Science. *Computer Science Education* 11, 4 (2001), 279–304. https://doi.org/10.1076/csed.11.4.279.3830

[29] Ann Y. Kim and Gale M. Sinatra. 2018. Science Identity Development: An Interactionist Approach. *International Journal of STEM Education* 5 (2018), ar51. https://doi.org/10.1186/s40594-018-0149-9

[30] Karen A. Kim, Amy J. Fann, and Kimberly O. Misa-Escalante. 2011. Engaging Women in Computer Science and Engineering: Promising Practices for Promoting Gender Equity in Undergraduate Research Experiences. *ACM Transactions on Computing Education* 11, 2, Article 8 (2011), 19 pages. https://doi.org/10.1145/1993069.1993072

[31] Victoria J. Kraj, Thomas Maranzatto, Joe Geigel, Reynold Bailey, and Cecillia Ovesdotter Alm. 2020. Evaluating Audience Engagement of an Immersive Performance on a Virtual Stage. *Frameless* 2, Article 4 (2020), 15 pages. Issue 1.

[32] Robert Lent, Steven Brown, and Gail Hackett. 1994. Toward a Unifying Social Cognitive Theory of Career and Academic Interest, Choice, and Performance. *Journal of Vocational Behavior* 45 (1994), 79–122. https://doi.org/10.1006/jvbe.1994.1027

[33] Marcia C. Linn, Erin Palmer, Anne Baranger, Elizabeth Gerard, and Elisa Stone. 2015. Undergraduate Research Experiences: Impacts and Opportunities. *Science* 347, 6222 (2015), ar1261757. https://doi.org/10.1126/science.1261757

[34] Rose M. Marra and Robert Pangborn. 2001. Mentoring in the Technical Disciplines: Fostering a Broader View of Education, Career, and Culture In and Beyond the Workplace. *New Directions for Teaching and Learning* 2001, 85 (2001), 35–42. https://doi.org/10.1002/tl.4

[35] Rebecca Medina, Daniel Carpenter, Joe Geigel, Reynold Bailey, Linwei Wang, and Cecilia Ovesdotter Alm. 2018. Sensing Behaviors of Students in Online vs. Face-to-Face Lecturing Contexts. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, Athens, Greece, 77–82. https://doi.org/10.1109/PERCOMW.2018.8480398

[36] Tyler Miller, Jung-Han Kimn, and Stephen P. Gent. 2019. Holistic Summer Undergraduate Research Program in High Performance Computing Research. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning) (PEARC '19)*. Association for Computing Machinery, New York, NY, USA, Article Article 82, 7 pages. https://doi.org/10.1145/3332186.3332215

[37] National Academies of Sciences, Engineering, and Medicine. 2017. *Undergraduate Research Experiences for STEM Students: Successes, Challenges, and Opportunities*. The National Academies Press, Washington, DC. https://doi.org/10.17226/24622

[38] National Science Foundation. 2019. Research Experiences for Undergraduates (REU): Sites and Supplements. Program solicitation NSF 19-582. https://www.nsf.gov/pubs/2019/nsf19582/nsf19582.htm.

[39] National Science Foundation and the National Center for Science and Engineering Statistics. 2019. Women, Minorities, and Persons with Disabilities in Science and Engineering: 2019. Special Report NSF 19-304. https://www.nsf.gov/statistics/wmpd.

[40] Nse Obot, Laura O'Malley, Ifeoma Nwogu, Qi Yu, Wei Shi Shi, and Xuan Guo. 2018. From Novice to Expert Narratives of Dermatological Disease. In *2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, Athens, Greece, 131–136. https://doi.org/10.1109/PERCOMW.2018.8480162

[41] Daniela Raicu, Audrey Rorrer, and Jamie Payton. 2018. Common Application Trends from Recruitment, Evaluation, Tracking. Presentation at 2018 CISE REU PIs meeting.

[42] Kelsey Rook, Brendan Witt, Reynold Bailey, Joe Geigel, Peizhao Hu, and Ammina Kothari. 2019. A Study of User Intent in Immersive Smart Spaces. In *2019 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, Kyoto, Japan, 227–232. https://doi.org/10.1109/PERCOMW.2019.8730692

[43] Monali Saraf, Tyrell Roberts, Raymond Ptucha, Christopher Homan, and Cecilia Ovesdotter Alm. 2019. Multimodal Anticipated versus Actual Perceptual Reactions. In *Adjunct of the 2019 International Conference on Multimodal Interaction (ICMI '19)*. Association for Computing Machinery, New York, NY, USA, Article 2, 5 pages. https://doi.org/10.1145/3351529.3360663

[44] Lior Shamir. 2017. Authentic Computer Science Undergraduate Research Experience Through Computational Science and Research Ownership. *Journal of Computational Science Education* 8, 2 (2017), 10–16.

[45] Jenny Olin Shanahan, Elizabeth Ackley-Holbrook, Eric E Hall, Kearsley A Stewart, and Helen Walkington. 2015. Ten Salient Practices of Undergraduate Research Mentors: A Review of the Literature. *Mentoring & Tutoring: Partnership in Learning* 23, 5 (2015), 359–376. https://doi.org/10.1080/13611267.2015.1126162

[46] Jeanine L. M. Skorinko. 2019. Looking Back at Undergraduate Research Experiences to Promote the Engagement of Undergraduates in Publishable Research at an R2 Institution. *Frontiers in Psychology* 10 (2019), 1316. https://doi.org/10.3389/fpsyg.2019.01316

[47] Kelly K. Sturner, Pamela Bishop, and Suzanne M. Lenhart. 2017. Developing Collaboration Skills in Team Undergraduate Research Experiences. *PRIMUS* 27, 3 (2017), 370–388. https://doi.org/10.1080/10511970.2016.1188432

[48] Burçin Tamer and Jane G. Stout. 2016. Understanding How Research Experiences for Undergraduate Students May Foster Diversity in the Professorate. In *Proceedings of the 47th ACM Technical Symposium on Computing Science Education (SIGCSE '16)*. Association for Computing Machinery, New York, NY, USA, 114–119. https://doi.org/10.1145/2839509.2844573

[49] Heather Thiry, Sandra L. Laursen, and Anne-Barrie Hunter. 2011. What Experiences Help Students Become Scientists? A Comparative Study of Research and other Sources of Personal and Professional Gains for STEM Undergraduates. *The Journal of Higher Education* 82, 4 (2011), 357–388. https://doi.org/10.1080/00221546.2011.11777209

[50] McKenna Tornblad, Luke Lapresi, Christopher Homan, Raymond Ptucha, and Cecilia Ovesdotter Alm. 2018. Sensing and Learning Human Annotators Engaged in Narrative Sensemaking. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*. Association for Computational Linguistics, New Orleans, Louisiana, USA, 136–143. https://doi.org/10.18653/v1/N18-4019

[51] United States Census Bureau. 2018. Population Estimates, July 1, 2018, (V2018).

[52] Helen Walkington, Kearsley A. Stewart, Eric E. Hall, Elizabeth Ackley, and Jenny Olin Shanahan. 2019. Salient Practices of Award-winning Undergraduate Research Mentors–Balancing Freedom and Control to Achieve Excellence. *Studies in Higher Education* 45, 7 (2019), 1519–1532. https://doi.org/10.1080/03075079.2019.1637838

[53] Regina Wang, Bradley Olson, Preethi Vaidyanathan, Reynold Bailey, and Cecilia Ovesdotter Alm. 2019. Fusing Dialogue and Gaze From Discussions of 2D and 3D Scenes. In *Adjunct of the 2019 International Conference on Multimodal Interaction (ICMI '19)*. Association for Computing Machinery, New York, NY, USA, Article 1, 6 pages. https://doi.org/10.1145/3351529.3360661

[54] Willamette University. 2014. Department of Psychology: Research Internship Evaluation Form.

[55] Kwai Wong, Stanimire Tomov, and Jack Dongarra. 2020. Project-Based Research and Training in High-Performance Data Sciences, Data Analytics, and Machine Learning. *The Journal of Computational Science Education* 11 (2020), 36–44. Issue 1. https://doi.org/10.22369/issn.2153-4136/11/1/7

[56] Heather Wright. 2020. One Year Later, CERP Data Still Indicate REU Participation Relates to Graduate School Enrollment. *Computing Research News* 32, 2 (February 2020).

A APPENDIX: OVERVIEW OF PROFESSIONAL DEVELOPMENT PROGRAMMING

- *Workshops:* Students attended workshops on research-relevant topics such as human subject research, statistics, data visualization and research storytelling. (Figure 3: A).
- *Toward Graduate Studies:* Students developed understanding about graduate school or initial grant writing skills for graduate school, a new topic to most participants. We focused on a national program for graduate fellows and students also met a panel of prior recipients (Figure 3: B).
- *Public STEM Outreach:* Students presented a hands-on exhibit of various sensing technologies at a local public library, interacting with library patrons of all ages (Figure 3: C).
- *Coordinated Interdisciplinary Events:* The organizers coordinated interdisciplinary joint activities with nearby programs that were both social and academic in nature, including a graduate study and research symposium featuring sessions with research talks by doctoral candidates for insight into PhD-level STEM research, and panels about graduate school (Figure 3: D).
- *Journal Club:* Students discussed papers within the scope of the program's intellectual theme. Written and oral reflection

exercises coached critical scholarly reading and technical writing.

- *Industry Research Lab Visit:* Students visited an industrial research lab as part of having them consider a range of research career paths. At this visit, students interacted with professionals and received feedback on their project-in-progress.
- *Observing a PhD Defense:* Students were encouraged to attend a PhD dissertation defense.
- *Teaching-to-Mentoring Bridge:* In talks, mentors shared about their research journey and discussed topics of expertise of relevance to the program's intellectual focus such as: *Facial expressions in VR and affective computing*, *Visual perception and what we learn from eye tracking*, *Linguistic sensing and computers making linguistic sense*, and *Intelligent systems that learn deeply*.
- *Post-program Sessions:* In video get-togethers, organizers reconnected with the cohort and let alumni share about their progress and activities at home institutions (Figure 3: E).

Creating a Graphical Tool for Non-Programmers to Use to Make Heatmaps

Nicholas Alicea
Centre College
Danville, KY
nicholas.alicea@centre.edu

Akenpaul Chani
Centre College
Danville, KY
aken.chani@centre.edu

Lam Le
Centre College
Danville, KY
lam.le@centre.edu

Hayata Suenaga
Centre College
Danville, KY
hayata.suenaga@centre.edu

David Toth
Centre College
Danville, KY
david.toth@centre.edu

Selam Van Voorhis
Centre College
Danville, KY
selam.vanvoorhis@centre.edu

Jessica Wooten
Piedmont University
Demorest, GA
jwooten@piedmont.edu

ABSTRACT
Heatmaps are used to visualize data to enable people to quickly understand it. While there are libraries that enable programmers to create heatmaps with their data, scientists who do not typically write programs need a way to quickly create heatmaps to understand their data and use those figures in their publications. One of the authors is not a programmer but needed a way to generate heatmaps for their research. For a summer undergraduate research experience, we created a program with a graphical user interface to allow non-programmers, including that author, to create heatmaps to visualize their data with just a few mouse clicks. The program allows the user to easily customize their heatmaps and export them as PNG or PDF files to use in their publications.

Categories and Subject Descriptors
J3 [Life and Medical Sciences] — Biology and genetics

General Terms
Design, human factors

Keywords
Heatmap, Bioinformatics, Genetics, Software development, Student experience, Student research

1. INTRODUCTION
One of the authors is a biologist who needed a tool to enable the creation of a visual representation of bioinformatics research data for research projects on which they were working. That author has minimal experience in computer programming. They reached out to one of the other authors, a computer scientist, to see if some of the computer scientist's students could do a project that could produce the necessary tool, a program with a simple graphical user interface (GUI) that would allow the creation of heatmaps from data sets stored in comma-separated value (CSV) files. The GUI would allow the user to create heatmaps without needing to write computer programs. The project would allow the students to gain experience in software development and user interface design while building a tool to advance science.

2. BACKGROUND
A heatmap is a representation of data in the form of a map or diagram in which data values are represented as colors depending on the magnitude of the value [1]. "They are often used to visualize high-frequency data or when seeing general patterns is more important than exact values" [2]. A general heatmap takes a two-dimensional set of values as its input and maps each of the values to a color. Heatmaps can be used to visualize or represent raw data in an easily digestible format that can enable users to understand their data. Figure 1 shows a sample heatmap with made-up data for levels of contamination in a small region of a city. The heatmap makes it easy to spot the extremely high levels of contamination in the regions represented by Row 5 Column 4 and Row 8 Column 10, in contrast to areas with high, moderate, and low levels of contamination. Our project was to create a GUI to allow scientists who are inexperienced with coding to easily create a heatmap to display their data.

DOI: https://doi.org/10.22369/issn.2153-4136/13/1/2

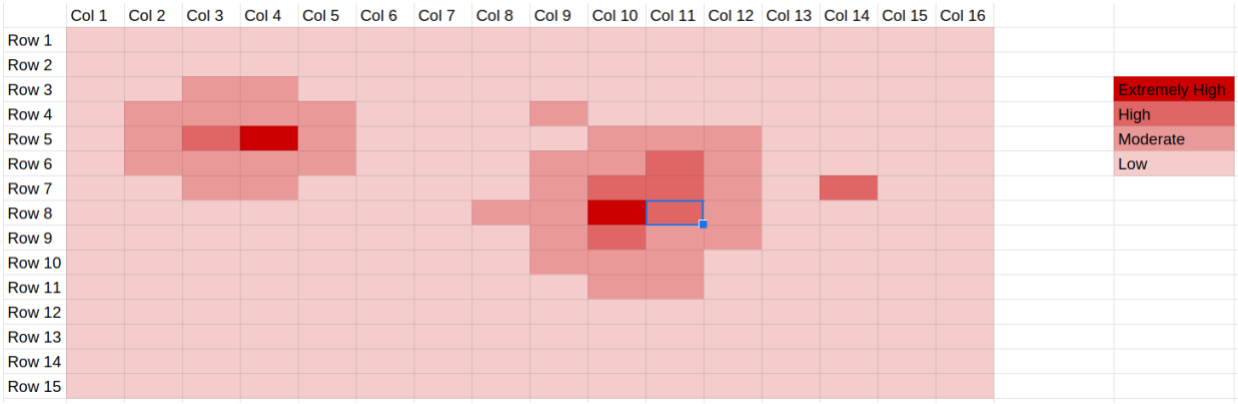


Figure 1. A sample heatmap showing contamination levels in a city.

The biologist author's primary focus was identifying taxonomies of bacteria that are particularly prevalent among similar microbiome samples. A microbiome is the total collection of microorganisms present in a particular environment. A bioinformatics lab took the author's data and analyzed them for occurrence of operational taxonomic units (OTUs), resulting in a CSV file composed of raw data representing the occurrence of each OTU in each sample. An OTU can be defined as a categorical classification based on genetic similarity, providing a method of understanding what microbes are present in a sample through genetic analysis. A heatmap is the accepted way of displaying the data for such a project.

3. METHODS

Several different factors impacted the development of the program. The data input file format, the importance of ease of use of the program, and the existence of any libraries that would aid in the development were all important.

3.1 Data

The data sets that would be used as input to the program were stored in CSV files. They contained the information about bacteria found on two different populations studied by one of the authors. The first set consisted of ninety samples taken from either salamanders or environmental controls, and the second set consisted of forty samples taken from either dead bodies or environmental controls.

3.2 Interface Design

When designing the user interface, we felt it was critical that a person with no programming experience could easily use it without investing more than a couple minutes to learn to use the tool. We also wanted to make the tool prevent the user from being able to make mistakes, to make it even easier to use. To ensure we accomplished these goals, we spent a considerable amount of time discussing the ways the user interface could be implemented and making sure it prevented the user from making errors. We made mockups of different possible interface designs and discussed the benefits and drawbacks of them. Ultimately, we chose to implement the graphical user interface as the common user interface pattern known as "task wizard." In a task wizard interface, the user answers a small number of questions on each screen, clicking a "next" button or "back" button to navigate between screens. On the final screen, the user clicks a "finish" button to complete the task. Since most computer users have installed software using one of these task wizard interfaces, and since we could prevent errors using this kind of interface, we designed the program's user interface as a task wizard. Figures 2, 3, and 4 show the screens of our user interface.

3.3 Implementation Details

To create the GUI, we used Tkinter, a module in the Python standard library which serves as a GUI toolkit. Tkinter is centered around event driven programming, which means the program responds to actions the user takes such as clicking on buttons or selecting an option from a dropdown menu [3]. We used a Python plotting library, Matplotlib, and its extension NumPy [4,5]. Both provide an object-oriented API for embedding plots into applications with Tkinter. To create heatmaps, we used seaborn, a Python data visualization library based on Matplotlib [6]. We also used another software library, pandas, which is used for data manipulation and analysis and provides data structures and operations for manipulating numerical tables of data such as the OTU occurrence data sets [7].

4. RESULTS

The GUI has two screens, a setup screen shown in Figure 2 and a heatmap screen shown in Figures 3 and 4.

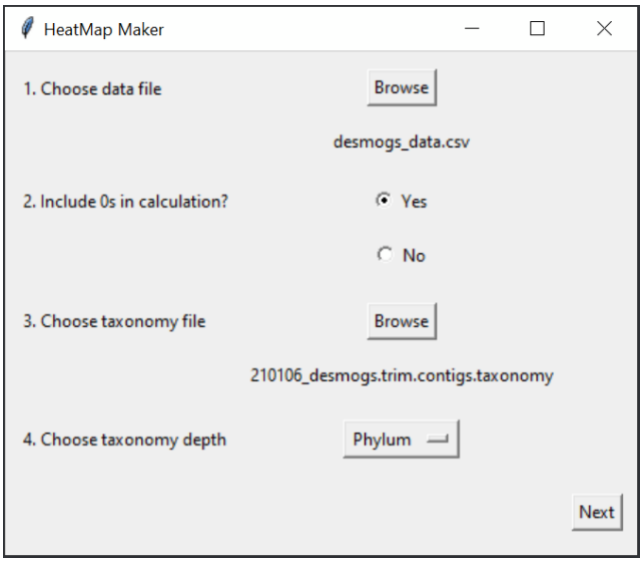


Figure 2. The setup screen.

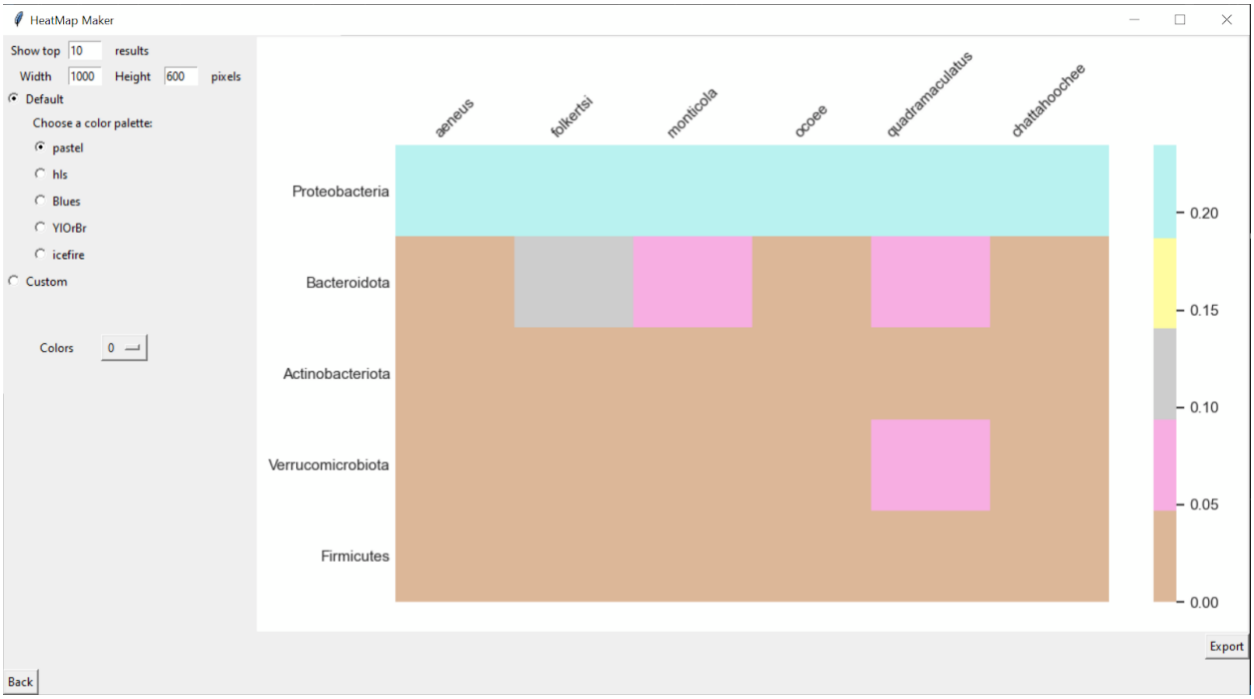


Figure 3. The heatmap screen with the default color scheme selected.

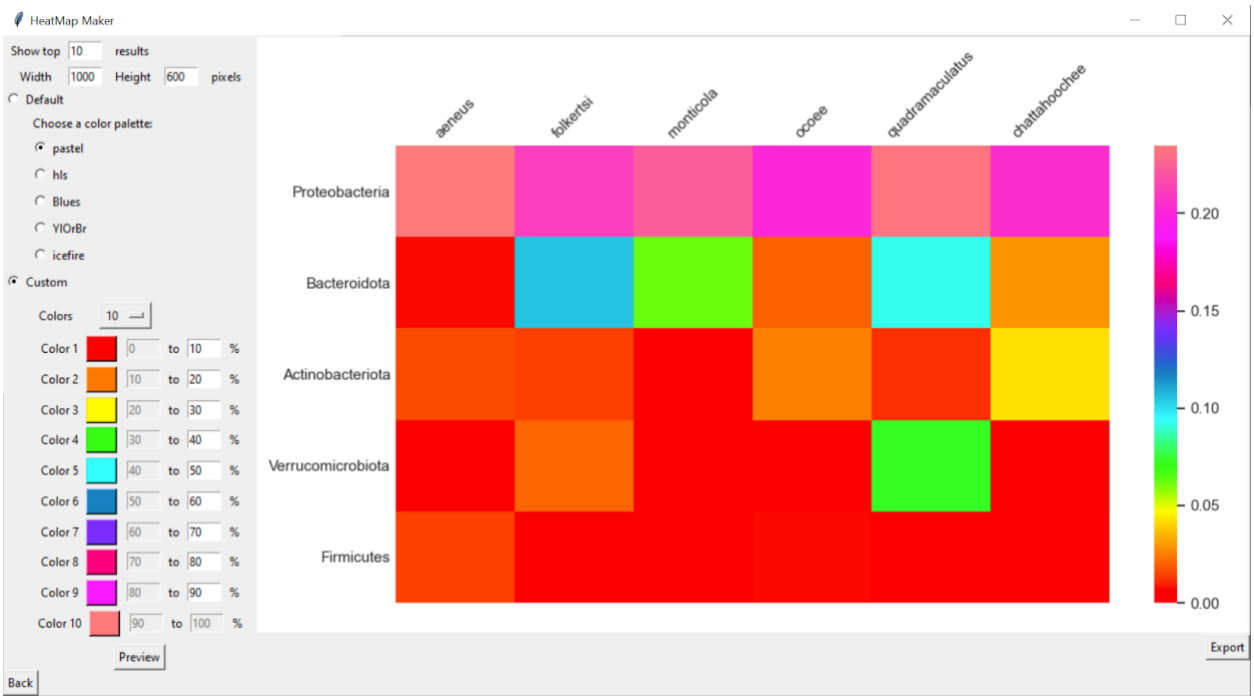


Figure 4. The heatmap screen with the custom color scheme selected.

4.1 Setup Screen

On the setup screen, the user selects two input files, a data file, and a taxonomy file. The program forces the user to select the two required files before allowing them to continue to the Heatmap screen. The user may choose whether to include zeros in the category averages as well as choose the taxonomy depth represented in the heatmap. If an unavailable taxonomy depth is chosen, an error message will appear, and the user must select a new depth before continuing.

4.2 Heatmap Screen

The heatmap screen is displayed after the user clicks the Next button on the setup screen and the calculations have finished. The user may return to the setup screen by hitting the Back button at any point. On the heatmap screen, the user can choose how many results (rows) they want to be displayed on the heatmap. If the maximum number of results is less than what the user chooses, no extra rows will be added. The rows are sorted in descending order by an average of the row's values. The user can also determine the width and height of the heatmap in pixels. There are two color options, default and custom. With the default option, the user can choose one of the default color schemes provided by seaborn [6]. With the custom option, the user can choose from four to ten colors to create their own color scheme as well as designate each color a percentage range. The user must choose all colors and enter continuously ascending range values before clicking the Preview button, or an error message will appear. Once all input fields are filled, the percentage of each color will be scaled correspondingly to the data, and the custom heatmap will be drawn. Clicking the Preview button will automatically change the plot option from Default to Custom, and selecting any of the default color schemes will change the plot option from Custom to Default. When the user is satisfied with the colors they have chosen, they can click the Export button to export the heatmap in either PNG or PDF format. To quit the program, the user can click the X button in the window's title bar and a prompt will ask them to confirm if they want to quit the program.

5. FUTURE WORK

There are several possible opportunities for further development of this GUI. Currently, the program requires the input files to be in a specific format. The first thing we could work on is to generalize the program to accept more input file formats, enabling the GUI to a wider scope of use. We could also allow for greater customization of what in the data file is being analyzed, such as letting the user choose the column in the input file by which the heatmap will be categorized. Since we made this GUI to make biological heatmaps, we could also include the option to italicize labels that are species names.

6. REFLECTIONS

This is the first time we got to do a real-world project that is both enriching and helpful. We had to learn how to facilitate cross-disciplinary communication and collaboration between biologists and computer scientists. We are fortunate that one of the authors on our team had prior experience working with bioinformatics to help bridge the knowledge gap. We had former experience with Python in class, but this project provided us a chance to apply what we learned, further familiarizing us with the language and opening up our programming possibilities. We also had to learn tools for collaboration and version control such as Git and GitHub. To create a GUI, we had to plan ahead to really get a handle on our code. We got to experience the software development process like real software developers. We first had a mock-up for the GUI then proceeded to prototyping and programming it. We had several meetings for feedback with our "client" along the way. Since we carefully planned our design before getting into coding, adjusting the software was not too challenging. As we made the GUI, it was important to think from a user perspective, because something that may make sense to implement as a programmer may not make sense to a user. All in-class assignments have rubrics and well-defined goals, so this is also our first time working with something where the goals for this project were fluid and open-ended. The team experience we had was invaluable since we could have diverse approaches to solve the problem and work together to achieve a product.

7. REFERENCES

[1] HEAT MAP English Definition and Meaning | Lexico.com. Retrieved from https://www.lexico.com/en/definition/heat_map

[2] Schwabish, J. 2021. *Better Data Visualizations*. Columbia University Press, New York.

[3] Introduction to GUI Programming with tkinter — Object-Oriented Programming in Python 1 Documentation. Retrieved from https://python-textbok.readthedocs.io/en/1.0/Introduction_to_GUI_Programming.html

[4] Matplotlib - Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Matplotlib

[5] NumPy. Retrieved from https://numpy.org/

[6] seaborn.heatmap — seaborn 0.11.1 documentation. Retrieved from https://seaborn.pydata.org/generated/seaborn.heatmap.htm

[7] Pandas (software) - Wikipedia. Retrieved from https://en.wikipedia.org/wiki/Pandas_(software)

Magic Castle — Enabling Scalable HPC Training through Scalable Supporting Infrastructures

Félix-Antoine Fortin
Université Laval
Québec, Canada
felix@calculquebec.ca

Alan Ó Cais
Jülich Supercomputing Centre
Jülich, Germany
a.ocais@fz-juelich.de

ABSTRACT

The potential HPC community grows ever wider as methodologies such as AI and big data analytics push the computational needs of more and more researchers into the HPC space. As a result, requirements for training are exploding as HPC adoption continues to gather pace. However, the number of topics that can be thoroughly addressed without providing access to actual HPC resources is very limited, even at the introductory level. In cases where access to production HPC resources is available, security concerns and the typical overhead of arranging for account provision and training reservations make the scalability of this approach challenging.

Magic Castle aims to recreate the supercomputer user experience in public or private clouds. To define the virtual machines, volumes, and networks that are required in a cloud-provider agnostic way, it uses the open-source software Terraform and HashiCorp Language (HCL). These resources are then configured using the configuration management and deployment tool Puppet to replicate a virtual HPC infrastructure with a full scientific software stack, and including a feature-rich JupyterHub environment. The final resource is accessible both through a web browser and via SSH, making it trivially OS-agnostic for the trainees.

Through the use of Magic Castle, we demonstrate that it is possible to dynamically provision virtual HPC system(s) in public or private cloud infrastructure easily, quickly, and cheaply. We also show that such infrastructures can support accelerators and fast interconnects, meaning that they can still be considered "true" HPC resources.

KEYWORDS

Education, Training, HPC, Cloud-computing

1 INTRODUCTION

Compute Canada [2] provides HPC infrastructure and support to every academic research institution in Canada. It uses CVMFS [5], a software distribution system developed at CERN, to make the Compute Canada research software stack available on its HPC clusters and anywhere else with internet access [1]. This enables replication of the Compute Canada experience outside of its physical infrastructure.

https://doi.org/10.22369/issn.2153-4136/13/1/3

Building upon this capability, an open-source software project named Magic Castle [8] emerged that aims to recreate the Compute Canada user experience in public or private clouds. Once Magic Castle is configured and deployed, the user is provided with a complete HPC cluster software environment including a Slurm scheduler, a Globus Endpoint, JupyterHub, LDAP, DNS, and thousands of research software applications compiled by experts with EasyBuild [10].

Magic Castle is compatible with AWS, Microsoft Azure, Google Cloud, OpenStack, and OVH. It can also be easily extended to support other cloud providers. While there are quite a few other cloud HPC open source projects (such as [6, 11, 13, 16]), Magic Castle has extensive provider support and ships with a complete production-ready scientific software stack. This makes it an exciting pedagogical platform for *scalable* HPC training. It allows for the possibility of quickly and easily creating event-specific HPC training clusters at minimal cost and side stepping thorny issues such as local site security or resource configuration policies.

2 DESIGN

The Magic Castle project is defined by an infrastructure-as-code component that is responsible for generating a cluster architecture in a public or private cloud infrastructure. Magic Castle does this in a cloud-provider-agnostic way using Terraform and HashiCorp Language (HCL) [9], which defines the virtual machines, volumes, and networks that are required to replicate a virtual HPC infrastructure. The infrastructure definition is packaged as a Terraform module that users can customize as they require.

A Puppet [14] environment component configures the cluster instances based on their role. This includes the configuration of the scientific software stack. Magic Castle has recently been extended to include support for the *European Environment for Scientific Software Installations* (EESSI) software stack [7], which also uses the Compute Canada software distribution system as a reference design.

In Figure 1, the final architecture of the configured infrastructure is shown. Starting from the Terraform module, it takes about 20 minutes to fully provision the system (including configuration of support for GPUs and/or special interconnects).

3 CURRENT STATUS

Compute Canada delivers about 150 training workshops per year, and Magic Castle is used extensively by many of these workshops since 2018.

The LearnHPC project [12] has also adopted Magic Castle with the goal of creating an EESSI HPC user experience for training purposes, primarily on the Fenix Research Infrastructure [15]. LearnHPC

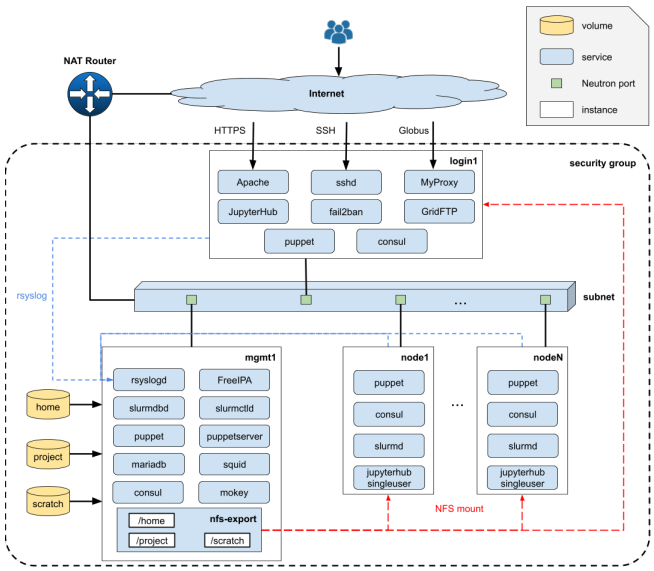


Figure 1: Overview of the architecture of Magic Castle.

is also collaborating with HPC Carpentry to ensure that the full set of lessons connected to HPC Carpentry [3] are supported on the training infrastructure.

OS support is available for RedHat and the RedHat-based variants CentOS, Rocky Linux, and AlmaLinux. As regards hardware capabilities, GPU support on Azure, AWS, Google Cloud, and OpenStack deployments has been tested with Magic Castle, as have support for the Infiniband interconnect provided by Azure and the EFA interconnect provided by AWS. Support for additional hardware capabilities with specific providers are driven by user requests and contributions.

Developer documentation for how to add support for an additional cloud provider is available [4] with a specific example for Alibaba Cloud given.

3.1 Cost Optimisation

"Spot" instances are allocated from the spare compute capacity of the cloud provider, usually at heavily discounted rates. However, these may be withdrawn/replaced by the provider at any time. On AWS, for example, typical eviction rates for high-end instances are below 5% and are therefore well-suited to compute nodes when coupled with the resilience features of Slurm. Spot instances can lead to savings of more than 70% of the cost of provision and are currently supported by Magic Castle on AWS, Microsoft Azure, and Google Cloud.

4 FUTURE WORK

Due to their typical high-availability nature, training clusters are likely to remain idle for a significant portion of their lifetime. Dynamic scalability of the provided resources has been a much-requested feature for Magic Castle and would likely greatly reduce the cost of provisioning. However, dynamic scalability would typically require Slurm to have access to the cloud provider API, meaning there would be a risk of exposing the cloud-provider credentials of the organisation if the cluster was compromised. This was seen to be an unacceptable prospect and would also require a custom implementation per provider. An implementation of dynamic provisioning is currently under development that is provider-agnostic and does not have this flaw.

ACKNOWLEDGMENTS

Alan Ó Cais acknowledges support from the European Union's Horizon 2020 research and innovation program, under grant agreement No. 676531 (project E-CAM) and grant agreement No. 823964 (project FocusCoE).

REFERENCES

[1] Maxime Boissonneault, Bart E. Oldeman, and Ryan P. Taylor. 2019. Providing a Unified Software Environment for Canada's National Advanced Computing Centers. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning) (PEARC '19)*. Association for Computing Machinery, New York, NY, USA, Article 51, 6 pages. https://doi.org/10.1145/3332186.3332210
[2] Compute Canada. 2021. Main website. (2021). https://www.computecanada.ca/
[3] HPC Carpentry. 2021. Main website. (2021). https://www.hpc-carpentry.org/ https://www.hpc-carpentry.org/.
[4] Magic Castle. 2021. Extending Magic Castle to support a new cloud provider. (2021). https://github.com/ComputeCanada/magic_castle/blob/main/docs/design.md#using-reference-design-to-extend-for-a-new-cloud-provider
[5] C. Condurache and I. Collier. 2014. CernVM-FS – beyond LHC computing. *Journal of Physics: Conference Series* 513, 3 (2014), 032020.
[6] elasticluster. 2021. elasticluster/elasticluster: Create clusters of VMs on the cloud and configure them with Ansible. (2021). https://github.com/elasticluster/elasticluster
[7] European Environment for Scientific Software Installations (EESSI). 2021. Documentation. (2021). https://eessi.github.io/docs/
[8] FÃĪlix-Antoine Fortin et al. 2021. ComputeCanada/magic_castle: Magic Castle 11.3. (July 2021). https://doi.org/10.5281/zenodo.5121948
[9] HashiCorp. 2021. Terraform. (2021). https://www.terraform.io/ https://www.terraform.io/.
[10] Kenneth Hoste, Jens Timmerman, Andy Georges, and Stijn De Weirdt. 2012. EasyBuild: Building Software With Ease. In *High Performance Computing, Networking, Storage and Analysis, Proceedings*. IEEE, 572–582. http://dx.doi.org/10.1109/SC.Companion.2012.81
[11] Cluster in the Cloud. 2021. Cluster in the Cloud documentation. (2021). https://cluster-in-the-cloud.readthedocs.io/en/latest/
[12] LearnHPC. 2021. LearnHPC - Scalable HPC Training. (2021). https://www.learnhpc.eu/
[13] AWS ParallelCluster. 2021. aws/aws-parallelcluster: AWS ParallelCluster is an AWS supported Open Source cluster management tool to deploy and manage HPC clusters in the AWS cloud. (2021). https://github.com/aws/aws-parallelcluster
[14] Puppet. 2021. Powerful infrastructure automation and delivery. (2021). https://puppet.com/ https://puppet.com/.
[15] Fenix RI. 2021. FENIX Research Infrastructure. (2021). https://fenix-ri.eu/
[16] stackhpc.openhpc. 2021. stackhpc/ansible-role-openhpc: Ansible role for OpenHPC. (2021). https://github.com/stackhpc/ansible-role-openhpc

A ARTIFACT DESCRIPTION: MAGIC CASTLE — ENABLING SCALABLE HPC TRAINING THROUGH SCALABLE SUPPORTING INFRASTRUCTURES

A.1 Abstract

This paper does not contain computational results. The reference release of Magic Castle for this publication is version 11.3 [8].

Best Practices for NERSC Training

Yun (Helen) He
NERSC, Lawrence Berkeley National Laboratory
Berkeley, CA
yhe@lbl.gov

Rebecca Hartman-Baker
NERSC, Lawrence Berkeley National Laboratory
Berkeley, CA
rjhartmanbaker@lbl.gov

ABSTRACT

The National Energy Research Supercomputing Center (NERSC) at Lawrence Berkeley National Laboratory (LBNL) organizes approximately 20 training events per year for its 8,000 users from 800 projects, who have varying levels of High Performance Computing (HPC) knowledge and familiarity with NERSC's HPC resources. Due to the novel circumstances of the pandemic, NERSC began transforming our traditional smaller-scale, on-site training events to larger-scale, fully virtual sessions in March 2020. We treated this as an opportunity to try new approaches and improve our training best practices. This paper describes the key practices we have developed since the start of this transformation, including:

- Considerations for organizing events;
- Collaboration with other HPC centers and the DOE ECP Program to increase reach and impact of events;
- Targeted emails to users to increase attendance;
- Efficient management of user accounts for computational resource access;
- Strategies for preventing Zoombombing;
- Streamlining the publication of professional-quality, closed-captioned videos on the NERSC YouTube channel for accessibility;
- Effective communication channels for Q&A;
- Tailoring training contents to NERSC user needs via close collaboration with vendors and presenters;
- Standardized training procedures and publishing of training materials; and
- Considerations for planning HPC training topics.

Most of these practices will be continued after the pandemic as effective norms for training.

KEYWORDS

HPC training, Best practices, Virtual training, Remote training, NERSC, GPU, COVID-19, Zoom, Closed captions

1 INTRODUCTION

The National Energy Research Scientific Computing Center (NERSC) [6] is the primary High Performance Computing (HPC) facility for the Office of Science in the U.S. Department of Energy. NERSC deploys advanced HPC and data systems for more than 8,000 scientists in 800 projects across a wide range of scientific and computational disciplines, including climate modeling, material sciences, fusion, high-energy physics, nuclear physics, biological research, and a host of other scientific endeavors.

The flagship production system at NERSC is the Cray XC Cori system [2], with Intel Xeon Haswell and Intel Xeon Phi KNL compute nodes. The upcoming HPE EX Perlmutter system [10] will contain both AMD CPU and NVidia GPU compute nodes. In preparation for Permultter, we launched the Cori GPU [3] testbed system, a small 18-node cluster of NVidia GPUs, to facilitate porting, benchmarking, and testing efforts for NERSC Exascale Science Applications Program (NESAP) [8] application codes.

NERSC users, including over 1,000 new users annually, possess a wide spectrum of HPC knowledge and familiarity with the usage of HPE/Cray systems. A successful training program for NERSC users is therefore one of the key components of our user support portfolio to enable effective usage of the system resources with efficient programming models. NERSC organizes approximately 20 training events [7] per year.

As a result of the COVID-19 pandemic situation, all of our ongoing training events transitioned into virtual sessions. This gave us an opportunity to try a number of new approaches and iterate on improvements. In this paper, we describe the key practices we have developed during this time.

2 BEST PRACTICES

2.1 Considerations for Organizing Events

Since 2020, NERSC has focused on multi-part training series. This was partly motivated by the constraints of the COVID pandemic; with all events being remote and digital, a format of multiple, shorter sessions was found more effective than full-day or full-week events. The shorter hours per day are more friendly for attendees across different time zones and allow them to have time for their other responsibilities outside the training hours.

For complex topics such as CUDA, OpenACC, or OpenMP, the multi-day format allows users to interleave the consumption of knowledge with the completion of homework. For example, the Deep Learning for Science School in 2020 transitioned from a four-day event to a series of weekly webinars over three months. Similarly, the NESAP hackathons transformed from intensive, single-week events to the same total number of hours spread across a couple of months.

2.2 Collaborating with other HPC Centers and DOE ECP Training Program

Often, NERSC, the Oak Ridge Leadership Facility (OLCF) [9], the Argonne Leadership Computing Facility (ALCF) [1], and the broader Department of Energy (DOE) Exascale Computing Project (ECP) Training Program [5] have common training needs on topics such

https://doi.org/10.22369/issn.2153-4136/13/1/4

as GPU programming and software tools. Collaborating with other training efforts helps increase the reach and impact of events and reduces staff workload for organizing these events individually.

There have been many successful collaborations among these organizations in the past year. A few examples include NERSC users being invited to participate in CUDA and OpenACC training series provided by NVidia and driven by OLCF staff. Meanwhile, OLCF users were invited to join the NVidia HPCSDK OpenMP Offload training and the HPCToolkit training driven by NERSC. NERSC and ECP also collaborated to provide CMake and E4S software stack trainings and invited OLCF and ALCF users. ALCF also included NERSC on some GPU architecture and GPU profiling trainings.

For all these events, staff from the HPC centers and programs worked closely together with the presenters on training logistics, event web pages, compute-system access, and hands-on exercises. Attendees listened to the same presentations and worked on hands-on exercises on various systems to which they had access and on which they already had familiarity with the user environment, with help from HPC center staff and vendors. For participants without a home system or wanting to try something new, we provided NERSC training accounts.

2.3 Targeted Emails to Users

Our training events are announced in the NERSC weekly emails along with many other NERSC center updates. We had found that there was often a tendency for these announcements to be overlooked. Targeted emails to users who may be interested in certain trainings can substantially boost registrations and attendance. We often see a 5–10X registration increase within 30 minutes of sending a targeted email.

For example, new users are contacted for our "Introduction to NERSC Resources" training. Users who have run only single-node jobs according to a Slurm job-accounting analysis are encouraged to take the one-day "Crash Course in Supercomputing" (to learn parallel programming).

2.4 Efficient Management of User Accounts

Management of user accounts for access to special computational resources is often necessary. For example, training attendees may need temporary access to NERSC resources for an event. We have automated the setup of training accounts so a staff member can simply request an event by providing the number of accounts needed and the dates and duration of the event. Then a 4-letter code will be generated for a user to apply for an account associated with the event. The user account will be approved instantly with the login credentials provided. The training accounts will be cleared automatically as well.

Oftentimes, we create compute node reservations for trainings with a hands-on component. It is more convenient for the Slurm batch scheduler for the users eligible for the reservation to be in a small number of projects. In order to satisfy this need, our practice is to add existing NERSC users to the project we use for training, which is allocated from NERSC overhead resources. There are also times when users need to be enabled for a special queue; this is the means of access to our GPU testbed on Cori. At first, the only way to add new users was by hand within the NERSC account management platform IRIS [13], which was quite cumbersome. We have now significantly simplified this process by developing more features such as batch adding or removing users to or from a project or special queue via IRIS APIs.

One major issue with the above approach was that when a user registers for two events that overlap in time, they would have been batch removed after the end of the first event while the user still needs to remain in the training project for the second event. Initially we had to guard these situations manually. After extensive discussions, we have implemented a mechanism that separates individual training events and adds users to separate training projects to prevent any interferences.

2.5 Strategies for Preventing Zoombombing

One of our online events was Zoombombed. After the event, we brainstormed mechanisms to prevent future bombings and incorporated them into our standard practices so that we are prepared for the worst. We recommend the following:

- Use a password on the meeting,
- Do not advertise the URL on the web,
- Enable a waiting room (if possible),
- Turn off participant screen sharing and annotation,
- Mute upon entry,
- Create an unadvertised meeting as backup, and
- Enlist a co-host to help manage attendees.

We also learned that in real time, upon the occasion that Zoombombing happens, there is a "Suspend Participant Activities" button in the "Security" menu that will lock the meeting, disable screen sharing & chat, turn off all audio & video, and give the host a chance to report the incident to Zoom. An excellent resource on this topic, from which we drew many helpful tips, is [12].

2.6 Publishing Recordings with Professional Closed Captions

The Department of Energy is committed to making its information and communications technology accessible to individuals with disabilities. The virtual format for training offers opportunities for improved accessibility, and in 2020 we made it a high priority to make use of these. NERSC made its training materials and recordings public and invested in creating closed captions for the recordings.

By working with NERSC administrative assistants, Berkeley Lab IT and procurement teams, and the support teams for Zoom and Rev (the supplier of captioning services) [11], we have reached a streamlined process to publish professional-quality closed captions for training videos on the NERSC YouTube channel[1]. Fully captioned videos for current training events are already online. We continue to work on retro-captioning our past training videos for a larger impact to the wide user community. These training videos have long-lasting impacts on users and for the public HPC education.

During Zoom meetings, we also enable live subtitles and full transcripts so attendees can toggle them on/off and are also able

[1]The NERSC Training YouTube channel is accessible at https://www.youtube.com/c/NERSCTraining-HPC

to save full transcripts. The real-time captions are helpful; however, they are not as accurate as professional services, and certain custom keywords (such as NERSC) and technical terms (such as OpenMP, Slurm, parallelize, etc.) are often misspelled. These errors are eliminated in the professional-quality captions afterwards with a list of keywords provided by NERSC for the manual captioners.

2.7 Effective Communications for Q&A

A fully virtual event makes Q&A particularly challenging. We adapted to the situation with technology solutions. Slack workspaces were created for most user events and used for Q&A, for sharing presentations, and for continued discussions during and after events. Events not using Slack used a Google Doc in which users could pose questions. When working with external organizations such as HPE and OpenACC, we also used Slack connect to allow people from each workspace to be in the same Slack channel for conversations.

We would like to especially emphasize the multiple benefits of using Slack. We usually create a private #organizers channel for the organizers from different institutions/vendors to use while planning the events. We discuss every detail in this channel before and during the event, planning dates, agenda, system access, presentation slides, hands-on codes, surveys, and more. Using a Slack channel is much lighter and quicker than email exchanges.

Combining Zoom with Slack or Google Docs is more effective for participant communications than Zoom alone. While Zoom does have a chat and Q&A feature, we found these other tools more effective for discussions. As a result, we used Slack or Google Docs for participant interactions in addition to Zoom during the training events. Both are effective for offline discussions and managing Q&A threads. We often had co-hosts monitoring chat and Google Docs and technical experts standing by to answer questions.

Additional practices that improved the user experience included allowing participants to unmute themselves to ask questions in smaller events and during hands-on sessions. We have also used Zoom breakout rooms with screen share and Zoom polls in some events.

At an ECP panel on virtual training best practices [4] held in September 2021, we learned two additional tips. First, in addition to Q&A, Google Docs can also be used for shared note-taking and for quickly gathering attendees feedback such as individual progress for hands-on exercises. Second, Zoom annotation is handy for visualizing feedback; for example, we can ask people to put up a given choice of stamp (green check, red heart, question mark, etc.) that has an agreed-upon meaning.

2.8 Tailoring Training Content for NERSC Users

NERSC put a lot of effort into collaborating with presenters to tailor training content to NERSC user needs (from the largest scope down to small details of materials covered), trying out exercises ahead of time, and providing constructive suggestions. These efforts helped increase the quality of trainings for NERSC users. We also received positive feedback and appreciation from the presenters on developing improved and productive trainings.

For example, the Parallelware training in October 2020 was the culmination of a 10-month collaboration with Appentra staff. The resulting training materials were highly customized to NERSC user needs. We held multiple planning meetings to prepare for this training, studied crucial NESAP user codes and other ECP applications for motifs, provided detailed feedback on training slides and exercises, and performed a survey.

Another example is working with HPE to schedule the standard HPE EX programming environment training at a time close to when users would gain access to the new Perlmutter machine and to collaborate on adjusting the contents of the training to NERSC needs. We established a Slack Connect channel and held multiple long discussions to plan the training, which proved highly beneficial. We agreed to begin with a training for staff focused on concepts that are new with the Perlmutter system (benefiting our system configuration efforts) followed by a short introduction to Perlmutter for users and then an expanded user training with hands-on exercises after users are enabled on the system.

2.9 Standardized Training Procedures

For NERSC HPC consultants and staff from other groups who may provide trainings, we have internally published a guide to hosting a training event that details all the steps and offers guidelines and best practices. It includes detailed information and standard procedures for creating event web pages, registration forms, announcements, and training accounts; details on how to batch add/remove users to/from a project and requesting compute node reservations; and logistics for Zoom, Slack, Q&A, training materials, reminder emails and calendar invitations, welcome and logistics slide templates, Zoombombing prevention, slide publication, production of video recordings with closed captions, post-processing videos, and post-event surveys.

2.10 Considerations for Training Topics

NERSC continues to provide frequent and wide-ranging training opportunities for users. NERSC staff held seventeen distinct training events in 2020 and eighteen events in 2021 as of September. Some were standalone events; others were presented as a series. The courses targeted a broad range of audiences and topics, including getting started with NERSC, GPU architecture, profiling, machine learning/deep learning, tools, running jobs, science applications, programming models (OpenMP, CUDA, OpenACC, etc.), and services. Attendee counts ranged from 10–15 to more than 200 people per event. Feedback from attendees was uniformly positive.

Two courses traditionally offered to Berkeley Lab summer students ("Introduction to NERSC Resources" and "Crash Course for Supercomputing") were opened to all NERSC users in 2021 for the first time, in part because we were no longer constrained by the in-person classroom size limit. Other recent training topics by NERSC included LMOD, CI/CD, checkpointing/restarting, SpinUp, CMake, and OpenMP Offload.

We continue to come up with additional topics on which we may want to offer future training. One recent new training topic was HPCToolkit, and we are beginning conversations with NVidia about adapting for NERSC and offering their Bootcamps, Deep Learning Institute training materials, and Compilers Premier Support. Other topics under consideration are the Perlmutter user environment and

application optimization, Parallelware tools, GPU programming, AI for Science, Kokkos, SYCL, debuggers, etc.

3 CONCLUSION

Remote training has worked better than expected. The virtual training format has proved effective despite the lack of in-person guidance for hands-on exercises and nonverbal feedback from the audience. We were not aware of a good real-time whiteboard sharing tool initially but later found Google Jamboard to be quite useful.

The added advantages of virtual trainings include:

- Larger capacity for training;
- No added administrative overhead of physical meeting planning, site access approvals, and food ordering;
- No user travel needed;
- Ease of arranging multi-session training series;
- Ability to schedule sessions a few days apart without impacting travel
- Shorter training days, so attendees can do some other work during the day;
- Easy to do 1-hour webinars; and
- Increased flexibility for experts, especially for hackathons; some experts can drop in for a partial event, some can present a talk, and some can offer offline Slack help.

After in-person gatherings become safe and commonplace, NERSC will likely still offer more remote training events than in-person ones. We will have some remote-only events and some hybrid events, and can arrange in-person rooms for local people or anyone interested in coming on site, especially for those events with a large hands-on component. We will give remote attendees the same level of attention, if not more than what is given to in-person attendees.

Our recommendation is to continue with most of the practices described in this paper after the pandemic as effective norms of training at NERSC, such as continuing collaborations, continuing shorter and multi-day events, and continuing webinar events.

ACKNOWLEDGMENTS

This research used resources of the National Energy Research Scientific Computing Center (NERSC), a U.S. Department of Energy Office of Science User Facility located at Lawrence Berkeley National Laboratory, operated under Contract No. DE-AC02-05CH11231.

REFERENCES

[1] 2021. ALCF Training. Retrieved September 29, 2021 from https://www.alcf.anl.gov/support-center/training-overview
[2] 2021. Cori. Retrieved September 2021 from https://docs.nersc.gov/systems/cori/
[3] 2021. Cori-GPU. Retrieved September 2021 from https://docs-dev.nersc.gov/cgpu/
[4] 2021. ECP Panel on Training Virtualization. Retrieved September 29, 2021 from https://www.exascaleproject.org/event/strategies-for-working-remotely-panel-series-training-virtualization/
[5] 2021. ECP Training Events. Retrieved September 29, 2021 from https://www.exascaleproject.org/training-events/
[6] 2021. NERSC. Retrieved September 29, 2021 from http://www.nersc.gov
[7] 2021. NERSC Training Events. Retrieved September 29, 2021 from https://www.nersc.gov/users/training/events/
[8] 2021. NESAP. Retrieved September 29, 2021 from https://www.nersc.gov/research-and-development/nesap/
[9] 2021. OLCF Training. Retrieved September 29, 2021 from https://www.olcf.ornl.gov/for-users/training/
[10] 2021. Perlmutter. Retrieved September 2021 from https://www.nersc.gov/systems/perlmutter/
[11] 2021. Rev: Convert Audio & Video To Text. Retrieved September 29, 2021 from https://www.rev.com
[12] Katie Pratt and Lou Woodley. 2021. CSCCE Tech Tip Sheet - Zoom bombing: How to deal with bad actors during Zoom events. https://doi.org/10.5281/zenodo.4645429
[13] G. Torok, M. R. Day, R. J. Hartman–Baker, and C. Snavely. 2020. Iris: Allocation Banking and Identity and Access Management for the Exascale Era. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–11. https://doi.org/10.1109/SC41405.2020.00046

Building a Computational and Data Science Workforce

Katharine Cahill
Ohio Supercomputer Center
Columbus, OH
kcahill@osc.edu

Linda Akli
Information Technology Initiatives
Southeastern Universities Research Association
Washington, DC
akli@sura.org

Tandabany Dinadayalane
Department of Chemistry
Clark Atlanta University
Atlanta, GA
dtandabany@cau.edu

Ana Gonzalez
Department of Mathematics
University of Puerto Rico Mayagüez Campus
Mayagüez, PR
anacarmen.gonzalez@upr.edu

Raphael D. Isokpehi
College of Science, Engineering and Mathematics
Bethune-Cookman University
Daytona Beach, FL
isokpehir@cookman.edu

Asamoah Nkwanta
Department of Mathematics
Morgan State University
Baltimore, MD
asamoah.nkwanta@morgan.edu

Rachel Vincent-Finley
Department of Mathematics and Physics
Southern University and A&M College
Baton Rouge, LA
rachel_finley@subr.edu

Lorna Rivera
Lorna Rivera Consulting LLC
Lorna@lornarivera.com

Ahlam Tannouri
Department of Mathematics
Morgan State University
Baltimore, MD
ahlam.tannouri@morgan.edu

ABSTRACT

Under-representation of minorities and women in the STEM workforce, especially in computing, is a contributing factor to the Computational and Data Science (CDS) workforce shortage. In 2019, 12 percent of the workforce was African American, while only 7 percent of STEM workers were African American with a bachelor's degree or higher. Hispanic share of the workforce increased to 18 percent by 2019; Hispanics with a bachelor's degree or higher are only 8 percent of the STEM workforce [1].

Although some strides have been made in integrating CDS competencies into the university curriculum, the pace of change has been slow resulting in a critical shortage of sufficiently qualified students at both the baccalaureate and graduate levels. The NSF Working Group on Realizing the Potential of Data Science final report recommends "strengthening curriculum at EPSCoR and Minority Serving Institutions (MSI) so students are prepared and competitive for employment opportunities in industry and academia" [2]. However, the resource constraints and large teaching loads can impede the ability of MSIs and smaller institutions to quickly respond and make the necessary curriculum changes.

Ohio Supercomputer Center (OSC) in collaboration with Bethune Cookman University (B-CU), Clark Atlanta University (CAU), Morgan State University (Morgan), Southeastern Universities Research Association (SURA), Southern University and A&M College (SUBR), and the University of Puerto Rico at Mayagüez (UPRM) are piloting a Computational and Data Science Curriculum Exchange (C^2Exchange) to address the challenges associated with sustained access to computational and data science courses in institutions with high percentage enrollment of students from populations currently under-represented in STEM disciplines.

The goal of the C^2Exchange pilot is to create a network for resource constrained institutions to share CDS courses and increase their capacity to offer CDS minors and certificate programs. Over the past three years we have found that the exchange model facilitates the sharing of curriculum and expertise across institutions for immediate implementation of some courses and long-term capacity building for new Computational and Data Science programs and minors.

KEYWORDS

Computational and data science education, Computational and data science minors, Curriculum consortia, Broadening participation in computational and data science, Computational thinking, Computing education programs

1. CDS WORKFORCE SHORTFALL

There is a well-established need for a STEM workforce with a working knowledge of computational and data science (CDS) [3]. Although some strides have been made in integrating CDS competencies into the university curriculum, the pace of change has been slow, resulting in a critical shortage of sufficiently qualified students at both the baccalaureate and graduate levels. There are significant resource constraints contributing to the slow pace of implementation of undergraduate CDS curriculum that are limiting the production of CDS literate CI-Users from STEM domain sciences. There has been growth in the number of graduate programs in computational science and related data science fields. Though CDS at the undergraduate level would provide a pipeline for graduate programs and prepare students for the workforce, implementation is slower than expected due to several curriculum

DOI: https://doi.org/10.22369/issn.2153-4136/13/1/5

associated challenges. These curriculum challenges include lack of expertise to teach courses, limited access to advanced computing resources to conduct large scale data analytics, and institutional climate for collaborative instruction.

The Ohio Supercomputer Center Education staff track computational and data science (CDS) programs and maintain a list at HPC University [4]. The listing currently shows that there are only twenty-three US institutions with undergraduate computational science programs or minors, seventy-five institutions with graduate level programs, thirty-three institutions with undergraduate data science degree or minor programs, and seventeen with graduate level programs. Only five minority serving institutions are represented in these numbers: Chaminade University, Fisk University, Jackson State University, New Mexico State University, North Carolina A&T University, University of Hawaii at Hilo, and University of San Francisco.

Computational science requires the integration of expertise across several disciplines: mathematics, computer science, and the domain science and engineering disciplines. The organization of universities into discipline-oriented departments with budget models tied to those departments is a major obstacle to the development of interdisciplinary courses in computational science. Upper level courses and even some introductory courses in each discipline are typically oriented toward majors and require a number of prerequisites. Departments are heavily oriented toward providing the courses for their majors, making it difficult to provide courses for non-majors desiring CDS concentrations or minors without committing to a large number of classes. For example, computer science curricula are focused on meeting extraordinary demands for classes in that major, making the education of science majors in the basic computer programming and database management techniques a very low priority.

At smaller institutions and those with smaller STEM programs, departmental workload of faculty reduces their availability for the development of interdisciplinary curriculum. New courses and specialty courses are often not offered due to insufficient number of students to meet minimum enrollment for the course set by the institutions/university. Class exercises require access to advanced computing resources or tools not available locally.

Institutions starting new programs face challenges in recruiting students for the new course offerings given the schedule constraints, pre-requisites, and competing well-established course options. Even the existing programs can be challenged with garnering sufficient numbers of students to justify the continued use of limited faculty resources to offer them with sufficient frequency. These challenges are generally more pronounced at MSIs where budgets are tight, teaching loads are high, faculty with the expertise and experience to teach computational science is either unavailable or insufficient, and the access to appropriate software and hardware environments is limited.

2. CURRICULUM EXCHANGE OVERVIEW

The C^{2}Exchange project is made up of 7 collaborators including four Historically Black Colleges and Universities (HBCU) and one Hispanic Serving Institution (HSI). These partners are collaborating to address the challenges experienced by resource constrained institutions and, in particular, those faced by MSIs.

Computational science expertise at individual institutions is often limited to one or two areas, making it impossible to offer a curriculum aimed at a broad range of STEM majors. The goal of the C^{2}Exchange pilot is to create a network for resource constrained institutions to share CDS courses/curriculum and increase their capacity to offer CDS minors and certificate programs for STEM majors. The proposed exchange model allows the sharing of that expertise across institutions for immediate implementation of some courses and long-term capacity building for the implementation of CDS minors.

The C^{2}Exchange implementation is guided by CDS competencies developed by OSC in collaboration with academic faculty and industry, data collected via campus visits conducted by the XSEDE Education and Broader Engagement programs, outcomes of the SURA-led Advancing Computational Science at MSIs workshop, and the requirements articulated by the participating institutions through a series of surveys, conference calls, and site visits.

3. COLLECTIVE IMPACT

The practical implementation of a computational science program for undergraduates often requires the addition of new courses as well as changes in existing courses so that disruption to the entire curriculum is minimized and the most efficient use is made of faculty and support personnel. The C^{2}Exchange takes advantage of national efforts to integrate CDS into the curriculum and applies distance learning technologies to form a consortium with sufficient critical mass to enable stable CDS offerings at the academic institution partners.

Implementation of computational and data science curriculum is a strategic priority for all of the participating institutions. The C^{2}Exchange builds on the academic partners' prior participation in CDS training and education events and their experience offering hybrid courses. Additionally, the participating institutions completed surveys to identify technology support, experience, current course offerings, and the envisioned institutional benefits of participating in the C^{2}Exchange pilot. Figure 1 presents the goals of the partner academic institutions.

Figure 1. C^{2}Exchange academic institutions' goals.

The courses each institution receives fills a gap in their current offerings because of one or more of the following scenarios: (a) the course does not exist at their institution; (b) the course exists but no instructor is available to teach the course; (c) there is not sufficient student enrollment to offer the course.

4. C^{2}EXCHANGE FOUNDATION

4.1 CDS Curriculum and Competencies

While developing an interdisciplinary undergraduate minor program in computational science at a group of Ohio institutions, a set of competencies were identified to guide the creation of computational science courses and course materials. The competency-based approach allowed institutions to design their curriculum in a flexible way by integrating portions of the

computational science materials into existing courses, creating new courses focused on computational science, or doing a combination of the two.

Maintained by OSC on the HPC University website, the competencies were created by the participating faculty and then reviewed by a business advisory committee that offered some advice on topic emphasis and breadth. Since that time, a number of courses and instructional modules have been developed and tested in a variety of instructional formats. The top-level competencies resulting from this work are shown in Figure 2. Each top-level competency area has recommended content and learning objectives. Figure 3 presents the Area 1: Simulation and Modeling content [5].

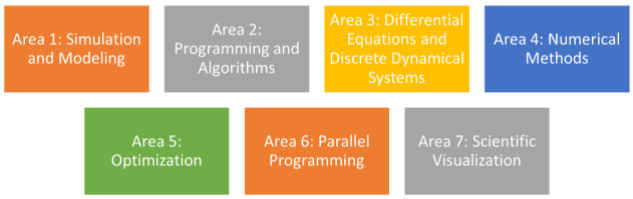


Figure 2. Top level computational science competencies.

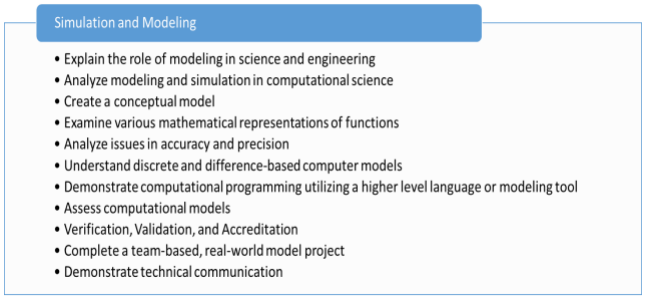


Figure 3. Computational science competencies — simulation and modeling subareas.

4.2 Blended Education

An online course on parallel computing offered through a partnership between XSEDE and UC Berkeley initially was offered as a Massive Online Open Course (MOOC). XSEDE's evaluation of this initial offering was consistent with the research indicating that MOOCs were ineffective and resulted in very high dropout rates and poor learning outcomes [6]. XSEDE's course was subsequently offered as a Small Private Online Course (SPOC) with collaborating faculty from a variety of higher education institutions that offered the course for credit.

This subsequent offering using a blended learning approach was found to be an effective educational model for teaching computational science courses at a variety of institutions. The lecture materials are delivered as online videos followed by quizzes to gauge student comprehension. Computing exercises are completed by all students to apply the knowledge gained in the lectures. Students may also complete a final project where they use the parallel computing techniques reviewed in the class to an application of their own choice. Local faculty members serve as advisors to these students and were responsible for grading as well as altering exercises as they saw fit for their students.

The advantages of this model are the course materials are readily available and the local instructors can offer the course with a minimum of preparation time. Students gain by being offered a course that was not previously available at their institution. The face-to-face component enabled students with a weaker computing background to gain the help they needed to complete the assignments. Moreover, keeping the classroom environment gives the students the focus they needed to complete the course. This offering had 91% completion rate compared to the MOOC with only 5% completion rate.

5. COURSE DEVELOPMENT AND DELIVERY

In the C^{2}Exchange pilot project, the participating institutions agreed to offer one course and participate as a recipient of at least one of the courses offered by the other consortium members. Each institution will take the lead in creating the first version of each course and agree to have their faculty be the local instructor for courses led by other institutions. Each institution will work with their administration to institute a formal minor or certificate program in computational science that uses both their local courses and the shared, online courses as part of the curriculum.

The courses offered during the pilot are drawn from the undergraduate competencies identified in Figure 2 and available at HPC University [5]. Initially only four courses will be offered to provide enough time for adapting courses to be delivered as a Small Private Online Course and to offer each course more than once for an effective evaluation.

Introduction to Modeling & Simulation is a 4-credit course that introduces the principles of modeling and simulation combined with an introduction to programming principles and skills using Python. It covers the construction, development, and study of mathematical representations of different classes of models; basic examples; techniques for fitting a function to an experimental data set; and selected case studies are included. This course may be offered to students well into their STEM major who are interested in developing computational skills.

The primary goal of the course is to introduce basic concepts of computational science to a diverse student body. The aim is not to produce experts in computational science but to provide the tools and skills that can benefit the personal and professional lives of individual and allow then to better collaborate with computational scientists. The course is targeted to students of all majors.

Computational Linear Algebra topics include Review of MATLAB and Basic concepts from Linear Algebra I; Cholesky Decomposition; Singular Value Decomposition; Principal Component Analysis; Matrix Approximation; Maximum Likelihood (Linear and Logistic regression); Support Vector Machines; Clustering algorithms, Gaussian Mixture Models; Dimensionality reduction techniques; and as time permits, applications. Most STEM majors are required to take a linear algebra course for their degree programs. Infusing more computation into their background through topics involving computational linear algebra will broaden their training and help prepare them for graduate and professional schools as well as the workforce.

The course was renamed Matrix Methods for Data Science and Machine Learning and will be offered for Mathematics, Computer Science, Physics, and Engineering majors at Morgan State University; this will result in students being better prepared to confront emerging challenges of a new data-driven world and to

respond to the workforce needs; the course will open doors for graduate work in mathematical and computational sciences graduate programs and job opportunity in Data Science, Machine Learning, and AI fields.

Introduction to Computational Chemistry and Molecular Modeling is comprised of lectures and labs to introduce the concepts of computational chemistry and molecular modeling and their applications in chemistry and biochemistry. This course is mainly for upper level undergraduate students of chemistry and biology majors, but students in any STEM discipline can take this course. The students in this course will learn how to use computers in chemistry as well as related fields like molecular sciences, drug design, biomedical, and materials science.

The Transdisciplinary Data Visualization learning experiences are designed to provide students with the procedures and principles for the design and deployment of interactive visual representations of large professionally collected datasets of societal relevance. The pre-requisite is a senior academic status with prior statistics or computing course at 200 level or above. The course is to be optimized for 4-week, 8-week and normal semester durations as well as face-to-face and online learning environments.

The courses reflect a blend of core competencies for an undergraduate computational science certificate program as well as the expertise of our partners and are based on currently offered courses to reduce development time. Introduction to Modeling and Simulation, Data Visualization, and Computational Linear Algebra are foundational courses for any computational science program, while Computational Chemistry and Molecular Modeling is a discipline specific course.

Through course development discussions, the syllabi and materials for the courses are reviewed collectively prior to implementation. Thus, all participating institutions are providing input on content and identifying how the material will fit into their curriculum.

Table 1. Courses received each year.

	Fall 2019	Fall 2020	Fall 2021
Modeling & Simulation	x	x	x
Matrix Methods Machine Learning		x	x
Computational Chemistry	x	x	x
Data Visualization (Winter term)			x

6. GOVERNANCE — MANAGEMENT AND OWNERSHIP

Providers such as XSEDE, Virtual School of Computational Science and Engineering, and the Great Lakes Consortium Virtual School successfully offer multi-site training and blended online courses in Parallel Programming and other High Performance Computing topics. However, their course offerings don't fully support the goals of C^{2}Exchange.

1. They offer a just a few courses and don't cover all the topics needed to develop a CDS minor for STEM majors.
2. The sites or receiving institutions do not influence what or when courses are offered, thus making it difficult for local or receiving sites to plan out their curriculum offerings.
3. There isn't knowledge exchange between the providers and the participating sites, thus lacking intentional capacity building at the receiving sites.

The C^{2}Exchange is designed to create a scalable network of institutions that can collectively offer CDS minors, concentrations, or certificates with minimal investment. The academic institution partners collaboratively create and maintain a multi-year plan identifying what courses will be offered, when the courses will be offered, and who will be the lead for each course. The active management role by the academic institution partners provides the potential for stable CDS curriculum offerings at their institutions and a pathway to the implementation of minor and certificate programs.

Without receiving credit, it will be difficult to recruit students to enroll in the course, and there would be little incentive to complete the course. Each of the participating institutions has mechanisms for setting up courses for credit as long as there is a faculty member responsible at the receiving institutions. How the courses are listed in the catalog vary depending on whether there is an existing course or there are options to offer it as a seminar or special topics course during the initial pilot. To date, no impediments at the lead or other institutions offering the courses have been encountered. The development of a C^{2}Exchange Memorandum of Agreement or similar agreement is being explored to ensure courses continue to be offered.

7. EVALUATION

C^{2}Exchange is employing a robust evaluation designed to provide formative information to guide program improvement as well as a summative assessment of program effectiveness and impact. The ultimate goal of the evaluation is to validate and document the effectiveness of the model exchange for enabling CDS minor and certificate programs and disseminate findings through publication and presentation. The evaluation will utilize a Values-Engaged Educative Approach (VEE) [9]. The VEE approach, developed with NSF-EHR support, defines high quality STEM educational programming as that which effectively incorporates cutting edge scientific content, strong instructional pedagogy, and sensitivity to diversity and equity issues. In the VEE approach, a key role of the evaluator is to work closely with program implementers to promote their understanding of program theory, implementation and impact.

The evaluation is designed to answer four questions:

1. Implementation: Is the C^{2}Exchange project being implemented on schedule and as planned?
2. Effectiveness: Are key components of the C^{2}Exchange model (e.g. enrollment, retention, curriculum development, course exchange, Annual Project Meeting, etc.) operating effectively and for whom? How might they be improved?
3. Impact: What outcomes (e.g. scientific impact, gains in scientific knowledge, improved technical skills, certification, student employment outcomes, increased institutional capacity, etc.) are associated with participation in the C^{2}Exchange programs? How does impact vary across groups? What is the value-added of participation in the C^{2}Exchange program?
4. Institutionalization: How and to what extent are elements of the C^{2}Exchange programs becoming institutionalized to ensure sustainability of program components? What opportunities and barriers exist?

Two rounds of evaluation have been conducted with the exchange partners to cover the fall semester offerings of 2019 and 2020. The survey found that faculty view C²Exchange participation as a valuable professional development activity with added benefits for local students and respondents generally report higher levels of experience in advanced computing areas compared to 2019.

A survey respondent commented, "The C²Exchange has provided excellent professional learning experience for me. The Exchange Model is being expanded at my institution in a way to promote knowledge sharing, trust, reciprocity, and collaboration."

We hope that with expansion of exchange we are able to measure student impact directly in our future work.

8. CONCLUSION

The pilot so far has demonstrated that the C²Exchange facilitates the exchange of strengths in the computational and data science curriculum available at the partnering institutions. Requirements for governance and inter-institutional agreements are being explored, and the initial implementation structures will be determined by the end of the pilot in summer 2022.

9. ACKNOWLEDGMENTS

This work is funded by the National Science Foundation Grant Cybertraining:CIU: Computational and data science literacy exchange (CSE-1829717).

10. REFERENCES

[1] National Science Board, National Science Foundation. 2021. The STEM Labor Force of Today: Scientists, Engineers and Skilled Technical Workers. Science and Engineering Indicators 2022. NSB-2021-2. Alexandria, VA. Retrieved from https://ncses.nsf.gov/pubs/nsb20212

[2] 2016. Realizing the Potential of Data Science: Final Report from the National Science Foundation Computer and Information Science and Engineering Advisory Committee Data Science Working Group. (Dec. 2016).

[3] National Academies of Sciences, Engineering, Medicine. 2016. Future Directions for NSF Advanced Computing Infrastructure to Support U.S. Science and Engineering in 2017-2020. Retrieved from http://sites.nationalacademies.org/CSTB/CompletedProjects/CSTB_087924

[4] HPC University: Graduate and Undergraduate Programs. Retrieved from http://hpcuniversity.org/students/undergraduatePrograms/

[5] HPC University: Competencies. Retrieved from http://hpcuniversity.org/educators/competencies/

[6] Gordon, S. I., Carey, K. and Vakalis, L. 2008. A Shared, Interinstitutional Undergraduate Minor Program in Computational Science. *Computing in Science and Engineering* 10, 5 (Sept./Oct. 2008), 12–16, DOI: https://doi.org/10.1109/MCSE.2008.127

[7] Gordon, S. I., Demmel, J., Destefano, L. and Rivera, L. 2016. Implementing a Collaborative Online Course to Extend Access to HPC Skills. *Computing in Science & Engineering* 18, 1 (Jan./Feb. 2016). DOI: https://doi.ieeecomputersociety.org/10.1109/MCSE.2016.6

[8] Hughes Darden, C., Akli, L., Bapna, S., Bhattacharya, P., Emdad, A., Hargett, S., Ellington, R., Nkwanta, A. and Zaveri, J. (2019). Interventions Addressing Recruitment and Retention of Underrepresented Minority Groups in Undergraduate STEM Disciplines. *Culturally Responsive Strategies for Reforming STEM Higher Education: Turning the TIDES on Inequity*. (Jan. 19, 2019). DOI: https://doi.org/10.1108/978-1-78743-405-920191014

[9] Greene, J., DeStefano, L., Burgon, H. and Hall, J. 2006. An Educative, Values-engaged Approach to Evaluating STEM Educational Programs. *New Directions for Evaluation* 109 (Mar. 2006), 53–72. DOI: https://doi.org/10.1002/ev.178

Tailored Computing Instruction for Economics Majors

Richard Lawrence*
HPRC†
Texas A&M University
College Station, TX
rarensu@tamu.edu

Zhenhua He*
HPRC†
Texas A&M University
College Station, TX
happidence1@tamu.edu

Wesley Brashear
HPRC†
Texas A&M University
College Station, TX
wbrashear@tamu.edu

Ridham Patoliya
HPRC†
Texas A&M University
College Station, TX
ridhampatoliya@hprc.tamu.edu

Honggao Liu
HPRC†
Texas A&M University
College Station, TX
honggao@tamu.edu

Dhruva K. Chakravorty
HPRC†
Texas A&M University
College Station, TX
chakravorty@tamu.edu

ABSTRACT

Responding to the growing need for discipline-specific computing curricula in academic programs, we offer a template to help bridge the gap between informal and formal curricular support. Here, we report on a twenty-contact-hour computing course developed for economics majors at Texas A&M University. The course is built around thematic laboratories that each include learning objectives, learning outcomes, assignments, and assessments and is geared toward students with a high-school level knowledge of mathematics and statistics. Offered in an informal format, the course leverages the wide applicability of the Python programming language and scaffolding offered by discipline-specific, hands-on activities to introduce a curriculum that covers introductory topics in programming while prioritizing approaches that are more relevant to the discipline. The design leverages technology to offer classes in an interactive, Web-based format for both in-person and remote learners, ensuring easy access and scalability to other institutions as needed. To ensure easier adoption among faculty and offer differentiated learning opportunities for students, lectures are modularized to 10-minute segments that are mapped to other concepts covered during the entire course. Class notes, lectures, and exercises are pre-staged and leverage aspects of flipped classroom methods. The course concludes with a group project and follow-on engagements with instructors. In future iterations, curriculum can be extended with a capstone in a Web-based asynchronous certification process.

Keywords

Python, Cybertraining, Google Colab, Economics, Cyber infrastructure (CI)

1. INTRODUCTION

The success of broadening participation in computing efforts relies on effectively developing a continuum of informal and formal learning environments for computing instruction. Simultaneously, we have to offer programming education that should be tailored to the needs of the students based on their discipline and level of commitment, rather than apply a one-size-fits-all approach. Indeed, recruitment and engagement can be improved by offering computing training in which examples and exercises are tailored to the academic and professional interests of the student body. Scaffolding can be achieved by adopting a tailored approach that focuses on a subset of topics that lead to a discipline-relevant final project that offers a feeling of accomplishment. The programming language, learning platform, and technologies should be coupled with continuing activities to encourage the interested learner to continue the process well beyond the duration of the activity. Simultaneously, we have to teach in incremental modules that let students solve small problems that are tied to real-world examples. To benefit the larger community, we should incorporate Web-based interactive computing avenues that ensure scalability and reproducibility at the core. Finally, the approach should be grounded in best practices in cyberinfrastructure technologies and reviewed pedagogical approaches. Such an approach should ideally not be limited by the computing technologies that are available to the students, but rather focus on making the current generation of technologies accessible to the students.

Texas A&M High Performance Research Computing has a legacy of offering informal courses geared toward adoption of CI practices in the regional researcher community [1, 4, 6, 7]. These have extended from our "short courses" that cover several cyberinfrastructure topics using hands-on exercises. In previous works, we have studied means to promote programming at the early undergraduate level [2], relied on visualization to study cybersecurity, and have explored opportunities to expand computing to the middle and high school levels [3]. With a view toward improving student learning in remote learning environments, we have explored pedagogical approaches such as our peer-moderated and peer-taught "Primers" [5]. We have built software platforms to facilitate the use of CI technologies to improve reproducibility in the sciences and coupled them with technology enabling "Tech labs" to improve CI adoption in research [7]. These efforts have been tied into asynchronous approaches that leverage interactive computing and social media to further CI technology adoption at Texas A&M University.

* Both authors contributed equally

† High Performance Research Computing

DOI: https://doi.org/10.22369/issn.2153-4136/13/1/6

As computing becomes more prevalent, programming is increasingly viewed as a critical career skill. Academic programs have moved to offer their graduates opportunities to develop skills in languages such as Python, R, MATLAB, or C++. Academic programs that do not have courses that directly tie into computing have relied on informal or certificate-based programs to help their students gain these skills. Here, we use our experience in offering CI-based training to develop a structured approach to teach "Python Programming" to graduate students in the Economics program at Texas A&M University. A template for adoption at institutions at other sizes is included here.

2. METHODS

The course was designed to be taught live in a mixed in-person and virtual hybrid modality and was structured to use active-learning methods. Curricular materials could be accessed via a Google Classroom (via Google Drive) for easy sharing. A course template was developed with unique branding and thematic elements to ensure that the course has a unique identity and to develop a sense of familiarity among students as we revisited old concepts. To ensure continuity during the classes, slide decks for the course were structured with learning objectives, concepts visited theory, hands-on exercises, and take-home assignments. To facilitate hands-on exercises while ensuring portability and equitable access to all students, Python examples and exercises were delivered in the form of Jupyter Notebooks as shown in Table 1, hosted in a Google Drive (students got a copy), and edited and run in Google Colab. The Notebooks contained a mix of informative lecture elements, interactive examples, and exercises. With a view toward student engagement, exercises used current real-world examples and relied on visualization approaches. Scaffolding was achieved by including a detailed description of topics covered in each coding block along with pointers to previously covered concepts. Assessments and assignments were offered using the notebooks. Curricular materials including presentation slides were offered before the class, allowing students to work on the exercises asynchronously. To facilitate a review of the previous thematic section, a video covering the major topics was offered before the lesson. Google Classroom allowed for seamless integration of all these technologies, with automatic distribution of the Jupyter Notebook files containing the course materials as well as recording student progress. From the students' perspective, this solution allowed the use of only a few clicks to both navigate the course and launch the Colab editor to do their coursework.

An initial list of course topics were identified via polling, informed discussions, and the general format used in Python education. Topics were developed in consultation with the Economics program at Texas A&M. Learning objectives, learning outcomes, and assignments were developed for each section. At its core, the format was modularized to enable easy adoption in teaching scenarios. Toward achieving this goal, the identified course topics were divided into thematic sections as described in Table 1. Each thematic section included three 50-minute lectures with hands-on exercises. Each lecture was divided into ten-minute modules that were mapped to other modules in the course, helping revisit topics later in the course. The last two thematic sections covered several topics introduced during the course. A detailed registry of each module's dependency on previous modules was developed. As such, a future instructor could mix and match these modules to create a new course or grab all the modules that lead to an advanced topic. The course included mandatory and optional assignments for each minute lecture. Assignments gradually built up in difficulty level, offering opportunities for differentiated learning. Students could choose to work on the take-home exercises or see the solutions by double-clicking on a cell. Participation was tracked at multiple points throughout each day, which determined course pass/fail for students. To facilitate student retention and participation, office hours were offered by the instructors during the week. To facilitate continued engagement on the conclusion of the course, HPRC leveraged its "Bring Your Own Science," a one-on-one researcher engagement service to work with student groups on their group projects. Several measures were considered to ensure that students participating using the remote option had an enriching experience. Adopting the best practices developed in our "Primers" and "Technology Laboratories" approaches, we maintained live chat via a peer moderator, online help offered via breakout rooms on Zoom, and the option to participate in remote office hours.

An important aspect of the development of this course was that it was not the sole work of any individual, but rather a collaborative effort of several instructors. This allowed for a diverse offering of teaching styles and helpd to ensure that relevant examples would be included at all stages of the course.

Table 1a. Concepts covered during the course. Each thematic section includes three 50-minute lectures that have accompanying, in-class, hands-on exercises and take-home assignments.

Thematic Section	Topics Covered
Introduction	Google Colaboratory, Variables
	Files, Data Types
	Dates and Times, how to use Functions, User Input
Algorithms	Operations
	Blocks, Control Structures
	Control Structures, Errors
Data Structures	Lists and Strings
	Lists, Loops, Dictionaries, Classes
	Arrays
Data Tools	Python libraries, Scatter plot, Line plot, Subplot, Candlestick plot
	Series, Index, Values, DataFrame Creation
	DataFrame Entry Retrieval, Filtering, Sorting
Data Analysis	DataFrame histogram, Missing and duplicate data handling
	Merge DataFrame (left, right, outer, inner)
	Linear regression, Train data, Test data, Predict, Accuracy

Table 1b (continuation of Table 1a).

Thematic Section	Topics Covered
Data Scavenging	Web-scraping, HTML, Tags, Browser Inspect
	Requests library, FRED API (short for Application Programming Interface)
	Beautiful Soup

3. RESULTS

The course was offered in Fall 2021 to the first-year graduate students enrolled in the Economics program at Texas A&M University. Enrollment was limited to 70, with the majority of students preferring to attend the sessions in-class. 69 students attended the first day of classes, with 47 students completing the course. The course was evenly structured in two learning components. In the first half of the course, we offered three thematic sections that covered the Python programming basics over ten contact hours. The second half covered different applications of Python for students of Economics over another ten hours. This format lent the course to two continued education credits. Details of each section are described in Table 1, and the course syllabus is included as supporting information. The course culminated with capstone exercises that used the Beautiful Soup library to "Web scrape" a website with economic data and used finance and plotting libraries to generate candlestick charts. These exercises offered an opportunity to reinforce concepts that the participating students had learned during the thematic sections.

3.1 Learning Outcomes

3.1.1 Orientation and Introduction

- Motivate the use of Python for Economics
- Familiarity with Jupyter IDE via Google Colab
- Understand general programming concepts
- Know what Python is, where it comes from, how to use

3.1.2 Programming Skills

- Import from external libraries (e.g. numpy)
- Use the assignment operator
- Inspect variables with print()
- Read and write a simple file
- Handle common types of data
- Inspect variables with type()
- Using functions with arguments
- Handle dates and times with numpy.datetime
- Get user input
- Apply mathematical rules (order of operations)
- Apply logical rules (comparisons)
- Define control structures with whitespace
- Use functions, loops, and conditionals to implement algorithms
- Handle errors
- Organize data into simple data structures (string, list, array)
- Interact with data structures (index, slice, mask)
- Integrate data structures into program control (loops, array operations)
- Organize data into advanced data structures (dictionary, class)
- Read HTML to locate data in web page code

3.1.3 Data Skills

- Visualize data with Matplotlib
- Create scatter plot, color map, best-fit-line
- Organize data with Pandas data structures
- Manipulate data with Pandas data methods
- Handle missing data
- Analyze data with Pandas data methods
- Create linear regression models with Scikit-Learn
- Retrieve data from the Web
- Parse HTML format to extract data
- Organize Web-scavenged data into data structures
- Make observations about data and adapt algorithms to match

Table 2a. In-class exercises and take-home assignments for additional learning. * indicates a take-home exercise.

Topics and exercises covered	Topics and exercises covered
Example Assignment	National Economics Data*
Hello World	Classes demo*
Your First Variables	Talking Cats*
Variables Quiz*	Array Basics
Text Files (preview)	Array Operations*
Common Variable Types	Scatter Plot
How to use Functions	Line Plot
Datetime	Subplots
The Droid*	Color Plots
Data Type Quiz*	Series
User Input*	Creating a DataFrame method
Story Generator*	Retrieve and Drop Rows
User Input Quiz*	Select, Filter, and Sort Rows
Arithmetic and Comparisons	Read/Write files
Units of Time*	Group data
Operations Quiz*	DataFrame plots and histogram
Functions	Missing and duplicate data*
Conditionals	Merge data*
More Conditionals	Pandas DataFrame*
Compute Pi*	Matplotlib Pandas*
Control Structures Quiz*	Candlestick Plot*
Errors and Files*	Linear Regression
Calculator	HTML
String Index	Pandas HTML
List Properties	Requests
List Logic	FRED API
List Loops	Regular Expression
Capstone for Lists and Strings	Beautiful Soup and Pandas for web-scraping

Table 2b. (Continuation of Table 2a).

Topics and exercises covered	Topics and exercises covered
Capstone with Dictionaries and Libraries	Candlestickplots

3.2 Description of Course Content by Thematic Section

3.2.1 Introduction

Students were first introduced to Python using the Jupyter Notebook provided by Google Colaboratory. These exercises built up their understanding of programming concepts and Python language syntax. The most important topics were covered in class, while a few were provided as take-home assignments. All of the exercises were directly related to future assignments that depended on these fundamentals. Particular emphasis was placed on the Numpy datetime64 data type as an example to support the future lessons on time series data. The use of files was introduced early because they would be critical for data analysis exercises later. Modules were introduced early despite not being traditionally considered a fundamental topic because of their massive importance in future lessons. Students were shown how modules could be imported (i.e., import <module_name>) and incorporated within their own code. In this course, the Numpy (the module/library introduced earlier) arrays were extensively used to generate data in Pandas and the Matplotlib exercises.

3.2.2 Algorithms

Students practiced with several topics that share a dependence on the Python language syntax element called indentation. These are the control structures: functions, conditionals, and loops. In practice, control structures are primarily used for the implementation of algorithms, which is not a primary focus of Economics research. Thus, these topics were less connected to future exercises. However, we still made use of them, when possible, to reinforce that learning. The take-home exercises for this session reinforced the use of files and built upon one another to form a sort of mini-lesson in themselves.

3.2.3 Data Structures

The most relevant topic for Economics research is the Data Structure, which is a strategy for keeping large amounts of data organized for effective processing. In Python, these structures are the List, Dictionary, and Array (with NumPy). These exercises depended on existing knowledge of Python fundamentals, most especially data typing and interacting with files. The exercises here were in the next session to build a useful network of tools that can handle time series data.

3.2.4 Data Tools

Students were introduced to two commonly used data handling libraries: Matplotlib [10] for visualization and Pandas [9] for data structure. Matplotlib was introduced first, leaning heavily on the Arrays lesson previously covered in order to handle the data that was to be visualized. The example data to be visualized was selected from publicly available Housing market data for relevance to Economics. This same data would be revisited in a future session featuring linear regression tools. Pandas includes two data structures, the simpler of which is the Series, a one-dimensional labeled array, and multiple Series together form a DataFrame, which is the most used structure. These data structures directly combined the elements taught in the previous session, especially the lessons on Dictionaries and Arrays. The primary kind of data in Economics research is a time series, so an emphasis was placed on those by including them as examples in many lessons. This built upon the Numpy [8] datetime64 introduced in the first session. This data type integrates with the Pandas [9] dataframe custom index feature, which is a staple method of handling time series data. The time series concept was revisited in an advanced exercise, the candlestick plot, a time series finance data visualization from the popular mplfinance [11, 12] library that builds upon the Pandas time series data structure.

3.2.5 Data Analysis

In addition to the data structure provided by the Pandas library, students were taught how to perform several basic data analysis operations using the same library. This included manipulation of a dataset to collect data into groups, add and remove elements, and filter to search for elements meeting certain logical criteria. This built upon concepts introduced in the second session where the operations that were previously used with conditional control structures were recycled for this new purpose. To tie in with economic research, the example data for manipulation practice in this session were chosen to be a sample of National Economic Data, which reprised the data set previously introduced in a Dictionary-focused take-home assignment.

Linear regression is one of the primary techniques economists use to determine correlations between different variables. We introduced students to linear regression using visualizations and a practical example. Students were also introduced to a machine learning package (Scikit-learn) and its functions. Apart from Scikit-learn, the rest of the code was built upon the modules that students learned during preceding sessions (Pandas, NumPy, and Matplotlib). A Housing Price dataset example was used to give participants a feel for visualizing, cleaning, and manipulating real-world economic data using Pandas.

3.2.6 Data Scavenging

Nowadays, most data, for example economics data, live online. Web scraping is an efficient method to collect data for research, sales, and marketing, popularity comparison, etc. Students were introduced to common Web data concepts including HTML basics, HTTP requests, and the JSON data structure. These naturally built upon previously taught skills, especially functions and data structures, including Pandas.

Federal Reserve Economic Data (FRED) [13] is a public resource hosted by the St. Louis Federal Reserve bank. This is commonly used by economics researchers as a free source of economics data. This specific API was also requested by faculty in the department. FRED API is a traditional Web API which can be handled easily in Python. Data is downloaded in the JSON format, which integrates easily into the Python dictionary data structure. This example was used as an advanced exercise building upon previous Python concepts while also priming students for their future studies using the same data source.

While APIs offer a straightforward means to retrieve data, we note that most websites do not have APIs for data extraction. For more general Web-data extraction, we covered the Beautiful Soup [11] library. Beautiful Soup is a Python library to pull data out of HTML files. The example use case for this lesson was salary statistics of a given job position in one or more cities.

3.3 Feedback from Economics Graduate Students

Surveys released each week collected student impressions of the material covered that day. Student impressions were generally positive about the relevance of the materials, interest in the topics, and opportunities offered by the course. There was a significant interest in converting this learning opportunity into a certificate that could be digitally displayed on professional social media sites such as LinkedIn. The students were able to follow along and use the Google Classroom platform effectively. From the survey responses, the most requested change was to slow down, because the syllabus was too ambitious in its pacing. Incorporating this request into the course plan resulted in omission of the capstone project and a few other exercises scattered throughout. We were pleasantly surprised to receive no comments criticizing the relatively advanced nature of the API data retrieval and Web-scraping exercises.

For the learning outcomes, 100% of the students who participated in the survey agreed that they were able to learn what they expected. Take-home assignments for continued learning were read but not graded. We observed that assignments were largely attempted the night before the previous session. Given the opportunity, we learned that offering solutions on the Web platform may dissuade students from working on the problems. To encourage more discussion, this feature was removed in later weeks. Students were now asked to attend office hours to get the solutions to the advanced problems. As such, student activity with regard to take-home assignments for continued learning was found to be unenthusiastic. Many students did not attempt the at-home exercises, citing that their other courses kept them too busy. This is an unfortunate but predictable result of the choice to do participation-only grading. This in turn had a negative impact on attendance at our hours, too, since the harder take-home assignments were designed to foster a discussion during office hours. In the future, homework should not represent a significant part of the curriculum unless true grading can be offered.

4. CONCLUSION

During this course, students were introduced to fundamental competencies and subject-specific applications in Python programming. This 20-contact-hour course built on an easy-to-use Web-based platform and represented a scalable opportunity for programming language instruction targeted at students of specific disciplines. Here, we showed that as we work to broaden participation in computing, tailoring examples and exercises to the interests of the student body increases student-engagement and facilitates student learning. The format gave us opportunities to include modules that increased awareness of cyberinfrastructure practices such as code optimization and parallel programming that are common in research computing. This will dovetail into our current series of HPRC courses that cover topics in Artificial Intelligence and Machine Learning. A tailored approach can focus on a subset of topics that lead to a discipline-relevant final project, which offers a feeling of accomplishment while serving as a capstone exercise in a certification-styled effort. An asynchronous version of the course is under development, which will include videos that accompany class slides and the Google Colab notebooks. This version will be made available to the community for wider adoption. A version with mandatory assignments and a capstone will be offered for certification.

5. SUPPORTING INFORMATION

The slide decks and some of the training materials used in this study are available to the community via the Texas A&M HPRC website [14]. The course syllabus is included as supporting information. Please send us feedback about your adoption experience via an email to help@hprc.tamu.edu.

6. ACKNOWLEDGEMENTS

This work was supported by the Department of Economics at Texas A&M University, the National Science Foundation (NSF) award number 1925764, “CC* Cyberteam SWEETER”, NSF award number 2019129, “MRI:FASTER”, NSF award number 1730695, “CyberTraining: CIP: CiSE-ProS: Cyberinfrastructure Security Education for Professionals and Students,” NSF award number 1818253, “Frontera: Computing for the Endless Frontier,” and NSF award number 2019136, “CC* BRICCs: Building Research Innovation at Community Colleges.”

7. REFERENCES

[1] Chakravorty, D. K., Pennings, M., Liu, H., Rodriguez, D. M., Jordan, L. T., Ghaffari, N., and Le, S. D. 2019. Effectively Extending Computational Training Using Informal Means at Larger Institutions. *Journal of Computational Science Education* 10, 7 (Jan. 2019), 40–47. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/7

[2] Chakravorty, D. K., Pennings, M., Liu, H., Wei, Z., Rodriguez, D. M., Jordan, L. T., McMullen D.F., Ghaffari, N., and Le, S. D. 2019. Evaluating Active Learning Approaches for Teaching Intermediate Programming at an Early Undergraduate Level. *Journal of Computational Science Education* 10, 7 (Jan. 2019), 61–66. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/10

[3] Chakravorty, D. K., Pennings, M., Liu, H., Thomas, X., Rodriguez, and Perez, L. M. 2020. Incorporating Complexity in Computing Camps for High School Students - A Report on the Summer Computing Academy Program at Texas A&M University. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 12–20. DOI: https://doi.org/10.22369/issn.2153-4136/11/1/3

[4] Chakravorty, D. K., and Pham, M.T. 2020. Evaluating the Effectiveness of an Online Learning Platform in Transitioning Users from a High Performance Computing to a Commercial Cloud Computing Environment. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 26–28. DOI: https://doi.org/10.22369/issn.2153-4136/11/1/5

[5] Chakravorty, D. K., Perez, L. M., Liu, H., Yosko, B., Jackson, K., Rodriguez, D.M., Trivedi, S.H., Jordan, L.T., and Le, S.D. 2021. Exploring Remote Learning Methods for User Training in Research Computing. *Journal of Computational Science Education* 12, 2 (Feb. 2021), 11–17. DOI: https://doi.org/10.22369/issn.2153-4136/12/2/2

[6] Lawrence, R.M., Pham, T. M., Au, P. T., Yang, X., Hsu, K., Trivedi, S.H., Perez, L.M., and Chakravorty, D. K. In press. Expanding Interactive Computing to Facilitate Informal Instruction in Research Computing. *Journal of Computational Science Education*.

[7] He, Z., Tao, J., Perez, L.M., and Chakravorty, D. K. In press. Technology Laboratories: Facilitating Informal Instruction for Cyberinfrastructure infused Data Sciences in Virtual Settings. *Journal of Computational Science Education*.

[8] Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., … and Oliphant, T. E. 2020. Array programming with NumPy. *Nature* 585, (Sept. 2020), 357–362. https://doi.org/10.1038/s41586-020-2649-2

[9] McKinney, W. 2010. Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference* 445, 51–56. DOI: https://doi.org/10.25080/Majora-92bf1922-00a

[10] Hunter, J. D. 2007. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering* 9, 3 (May–June 2007), 90–95. DOI: https://doi.org/10.1109/MCSE.2007.55

[11] Richardson, L. 2007. Beautiful Soup Documentation.

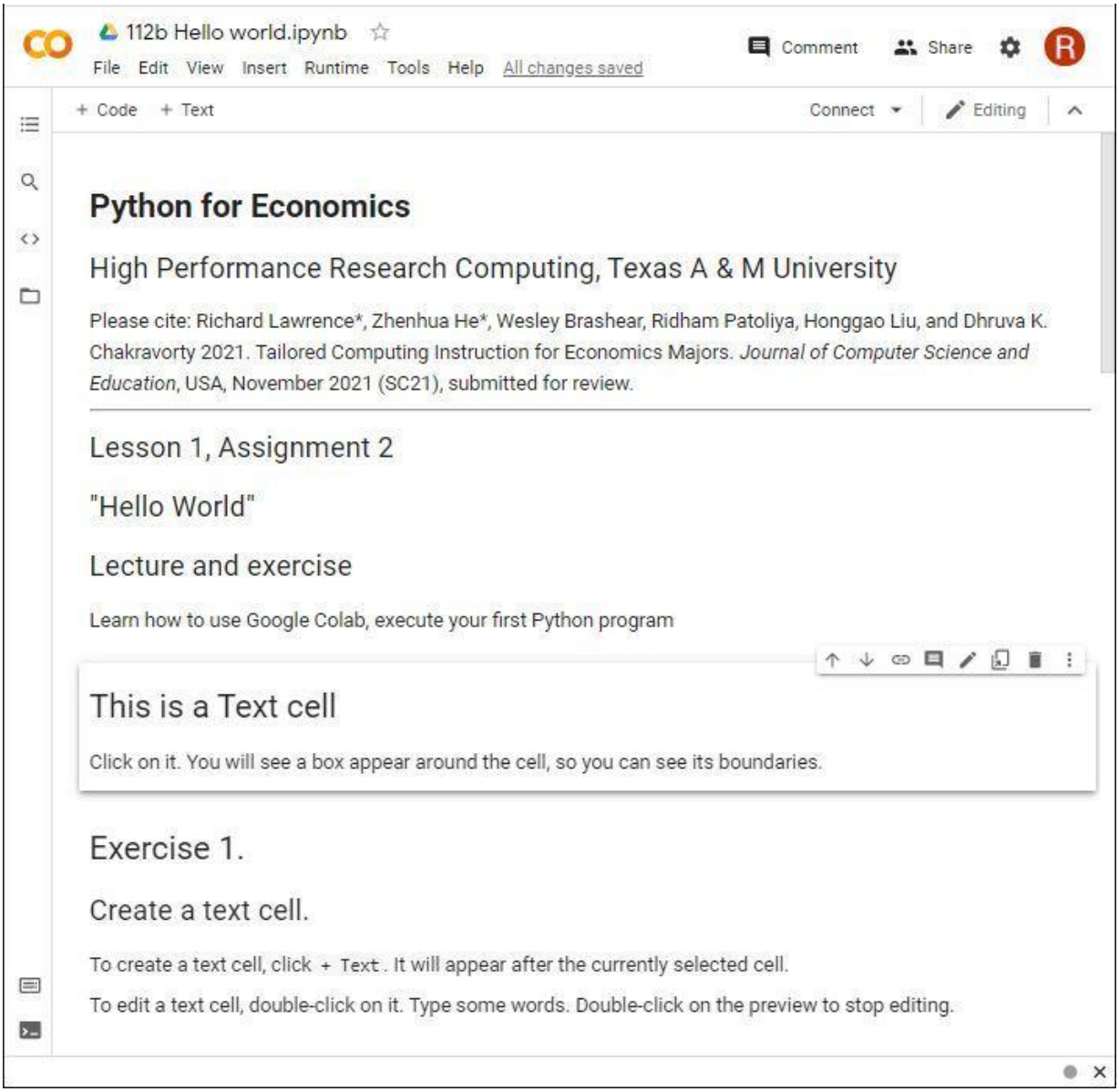
[12] Goldfarb, D. 2019. Mplfinance Documentation.

[13] 1997. FRED, Federal Reserve Economic Data. St. Louis, MO: Federal Reserve Bank of St. Louis. Software, E-Resource. Retrieved from the Library of Congress, https://lccn.loc.gov/98802805.

[14] Texas A&M High Performance Research Computing. Python for Economics Graduate Students. Retrieved from https://hprc.tamu.edu/events/workshops/2021-09-10-PyEcon.html

APPENDIX: REPRODUCIBILITY

Figure 1. "Hello world" notebook in Google Colab, first few cells.



Leveraging Northeast Cyberteam Successes to Build the CAREERS Cyberteam Program: Initial Lessons Learned

Andrew Sherman
Yale University
New Haven, CT
andrew.sherman@yale.edu

John Goodhue
Massachusetts Green High Performance Computer Center
Holyoke, MA
jtgoodhue@mghpcc.org

Julie Ma
Massachusetts Green High Performance Computer Center
Holyoke, MA
jma@mghpcc.org

Kaylea Nelson
Yale University
New Haven, CT

Eric Brown
University of New Hampshire
Durham, NH

Christopher Carothers
Rensselaer Polytechnic Institute
Troy, NY

Galen Collier
Rutgers, The State University of New Jersey
New Brunswick, NJ

Adrian Del Maestro
University of Tennessee, Knoxville
Knoxville, TN

Andrea Elledge
University of Vermont
Burlington, VT

Wayne Figurelle
Penn State University
University Park, PA

John Huffman
University of Delaware
Newark, DE

Gaurav Khanna
University of Rhode Island
Kingston, RI

Neil McGlohon
Rensselaer Polytechnic Institute
Troy, NY

Sia Najafi
Worcester Polytechnic Institute
Worcester, MA

Jeff Nucciarone
Penn State University
University Park, PA

Anita Schwartz
University of Delaware
Newark, DE

Bruce Segee
Univ. of Maine
Orono, ME

Scott Valcourt
Roux Institute at Northeastern University
Portland, ME

Ralph Zottola
University of Alabama at Birmingham
Birmingham, AL

ABSTRACT

Given the pivotal role of data and cyberinfrastructure (CI) in teaching and scientific discovery, it is essential that researchers at small and mid-sized institutions be empowered to fully exploit them. While access to physical infrastructure is essential, it is equally important to have access to people known as Research Computing Facilitators (RCFs) who possess a mix of technical knowledge and interpersonal skills that enables faculty to make the best use of available computing resources. Meeting this need is a significant challenge for small and mid-sized institutions that do not have the critical mass to build teams of RCFs on site.

Launched in 2017, the National Science Foundation (NSF) funded Northeast Cyberteam (NECT) built a program to address these challenges for researchers/educators at small and mid-sized institutions in four states — Maine, Massachusetts, New Hampshire, and Vermont — while simultaneously developing self-service tools that support management and execution of RCF engagements. These tools are housed in a Portal called Connect.cyberinfrastructure and have enabled adoption of program methods by the broader research computing community. Initiated in 2020, the NSF-funded Cyberteam to Advance Research and Education in Eastern Regional Schools (CAREERS) has leveraged the NECT methods and tools to jumpstart a program that supports researchers at small and mid-sized institutions in six states and lays the groundwork for an additional level of support via a distributed network of experts directly accessible by the researchers in the

DOI: https://doi.org/10.22369/issn.2153-4136/13/1/7

region. This paper discusses findings from the first four years of NECT and the first year of CAREERS.

Keywords

Workforce development, Research computing facilitator, Project portal, Ask.CI, CNCT.CI, Connect.Cyberinfrastructure, Ask.Cyberinfrastructure, Northeast Cyberteam, CAREERS Cyberteam

1. INTRODUCTION

Given the pivotal role of data and cyberinfrastructure (CI) in teaching and scientific discovery, it is essential that researchers at small and mid-sized institutions be empowered to fully exploit them. Access to physical infrastructure is essential, and equally important is access to people known as Research Computing Facilitators (RCFs), who possess a mix of technical knowledge and interpersonal skills that enables them to help researchers and educators use CI resources efficiently [1].

Providing access to RCFs can be challenging for small and mid-sized institutions for several reasons: (1) RCFs are in short supply and are difficult to recruit and retain, especially for institutions where budgeting for even one position is challenging to motivate; (2) it is impossible for a single RCF to have sufficiently broad expertise to cover all disciplines needing support; (3) if a lone RCF on campus changes jobs, research and education projects dependent on that RCF may slow or grind to a halt.

Launched in July 2020, the Cyberteam to Advance Research and Education in Eastern Regional Schools (CAREERS) aims to develop, implement, test, and refine a distributed approach to making research computing more accessible to researchers at small and mid-sized institutions via a regional collaboration. The CAREERS program region covers six eastern states: Connecticut, Delaware, New Jersey, New York, Pennsylvania, and Rhode Island. To do so, CAREERS builds on ideas developed by prior Cyberteam initiatives — the NSF-sponsored Northeast Cyberteam. Initiated in 2017, the Northeast Cyberteam Program (NECT) [2, 4, 5] is a collaborative effort across Maine, New Hampshire, Vermont, and Massachusetts that helps researchers at small and mid-sized institutions who are using advanced computing resources, while simultaneously training a new generation of RCFs. The program combines direct assistance to computationally intensive research projects; experiential learning opportunities that pair experienced mentors with students interested in research computing facilitation; sharing of resources and knowledge across large and smaller institutions; and development of tools that enable efficient oversight and facilitate replication of these ideas.

One of the most fundamental skills of successful facilitators is their ability to quickly learn enough about new domains and applications to draw parallels with their existing knowledge and to help solve the problem at hand. Recognizing this, a key concept in training facilitators through experiential learning is providing tools to enable self-service learning. The Connect.cyberinfrastructure (Cnct.CI) Portal is used to access the self-service learning resources that provide just-in-time information delivery to participants as they embark on projects in unfamiliar domains [6]. It also serves as a repository for best practices, tools, and techniques developed during a project. The NECT program places intentional emphasis on capturing and disseminating best practices to leverage and build on existing solutions whenever practical.

CAREERS leveraged the NECT tools and methods to jumpstart a similar program in six additional states. Simultaneously, CAREERS will lay the groundwork to provide a new layer of support through which professional RCF domain expertise can be shared with researchers across the region. Significant learning has taken place during the adaptation of the NECT methods and tools and is reported here.

2. NORTHEAST CYBERTEAM — METHODS AND TOOLS

The collaborators in the Northeast Cyberteam [3, 4] recognized a diversity of specialized knowledge at various institutions in its four-state region, as well as an abundance of researchers at small and mid-sized institutions who could benefit from increased use of large-scale cyberinfrastructure. However, to do so, these researchers would need to overcome barriers such as not knowing how to start, whom to talk to, or how to address problems when they arise. A typical engagement, which we call a “project,” is proposed by a researcher whose computational or data needs have surpassed the processing power of their laptop, or who desires to apply a new technique (e.g., machine learning) in their work.

Each project involves a researcher seeking to better utilize cyberinfrastructure, a student facilitator, and a mentor with relevant domain expertise. This team works together over a period of three to six months to help the researcher move beyond the daunting inflection point where their needs exceed current computation and/or storage resources to a solution that makes effective use of new cyberinfrastructure. Student facilitators are recruited from institutions in the region and are paid for their participation at a rate that is based on experience level and project duration. The projects have involved students at a range of skill levels and usually represent their first experience with research computing facilitation. The exposure of this student cohort to facilitation work, and ensuring a positive and empowering interaction, is crucial to our goal of expanding the workforce pipeline. Mentors are volunteers and are usually professional RCFs from research computing groups at larger institutions in the region who have subject matter expertise relevant to the project. Mentors are paired with student facilitators assigned to a project and work with them to develop a strategy to provide direct assistance to the researcher in making use of appropriate cyberinfrastructure resources. Through this model, students receive training in and exposure to facilitation that otherwise would not be available, research projects move forward to use advanced cyberinfrastructure, and the effectiveness of the mentor is multiplied. The individuals involved may all be at the same institution, but in many cases, they are at different institutions throughout the region.

Program direction is set by a Steering Committee that meets weekly and is composed of leaders from each of the larger institutions that serve as anchors for the program, a Program Manager who coordinates day-to-day activity, and key advisors from a few other institutions. The Steering Committee approves all projects undertaken. For a selection of projects, the Steering Committee relies less on soliciting competitive applications from researchers (though intellectual merit does naturally play a role in the selections) and more on outreach to faculty at smaller institutions who can benefit from access to cyberinfrastructure but are either unaware of available resources or have given up after a poor experience. Care has been taken in sourcing and monitoring projects to ensure that they (1) lead to results that might not otherwise have been achieved and (2) establish a model for engagement that others at the participating institution can follow [4].

While most costs go to student support, the NECT also invested in active management, including a Program Manager, who keeps the flow of projects running smoothly, and steering committee members who recruit, qualify, and oversee projects in their home states. While the value of program management is often overlooked, this investment has been critical to success. It has enabled several important outcomes including (1) efficient recruiting of projects, students, and mentors; (2) development of process, tools, and strategy; (3) effective communication across the anchor institutions; and (4) ability to explain the purpose and benefits of the program to grant administrators who have sometimes expressed initial skepticism about supporting this kind of collaboration across institutions. The Program Manager also coordinates across the geographically dispersed team of participants, who have varying amounts of time to invest in the program, and provides leadership to ensure forward progress.

2.1 Monthly Meetings

To create a convivial team environment despite the geographic distribution of participants, the NECT hosts a monthly Zoom meeting in which student facilitators are expected to share their experiences with each other and learn about other projects currently underway. Since the program is often their first exposure to research computing, this gives student facilitators a window into the diversity of domains, techniques, and subject matter that research computing encompasses, and it provides mentoring and networking opportunities with professional RCFs in different regions and at differently resourced institutions. It also gives them an opportunity to learn how other student facilitators are approaching their projects, and it provides support, especially during the initial stages of projects, where the learning curves can be quite steep.

At the beginning of a project, after the student facilitator has had an opportunity to meet with their mentor and researcher and has a solid understanding of the task at hand, they prepare and present a simple "launch presentation" that includes brief descriptions of the project, goals, timeline, and what they hope to learn. An important consideration in the launch presentation is the use of a scaffolded template that builds confidence and lowers barriers and anxiety around presenting about a topic to which the student facilitator might have just been introduced.

At subsequent meetings, student facilitators give brief verbal status updates until, at the end of the project, they present a "wrap presentation" summarizing what was accomplished, what was learned, and what contributions were made to the self-service learning tools and Git repository housed on the Portal. The monthly meetings, presentations, and project status updates are intended to build camaraderie while giving students additional exposure to the world of research computing and facilitation. The results have been very positive, with students and mentors offering insights and advice to each other both at the meeting and out of band.

2.2 The Portal

With so many simultaneous moving parts, NECT needed to develop a new tool to manage and organize the Cyberteam projects and participants as well as collect and disseminate the expertise and knowledge resources required to efficiently complete the projects. To serve this need, NECT developed the Cnct.CI Portal [2, 4, 6].

The Steering Committee relies heavily on the Portal for management of project workflows and capturing project outcomes. The Portal advertises projects and assists in recruiting mentors and student facilitators. The Portal is also used to access and aggregate self-service learning resources that provide just-in-time information delivery to participants as they embark on projects in unfamiliar domains [6]. There is usually not enough time to enroll in a traditional training course or attend a seminar when a new domain or application is encountered. Therefore, the goal of these learning resources is to reduce the need for direct assistance; reduce duplication of effort by adapting and building awareness of available documentation, training, application software and software utilities; and supplement these resources where there are high-impact opportunities.

A uniform underlying infrastructure is provided by a common tagging infrastructure and voting capabilities modeled after crowd-sourced repositories such as Stack Exchange [3]. This infrastructure allows a user to click on a tag from any part of the Portal and obtain a listing of all related Portal content, including mentor profiles, project descriptions, frequently asked questions, and training resources. With a continuously growing and evolving list of tags, this creates the opportunity to search for content in a granular yet curated manner. This underlying tagging mechanism also facilitates mentor matching and identification of targeted learning resources. For example, a Cyberteam member seeking advice on a particular topic can search by individual tags to obtain a listing of all Portal users that have identified that topic as a skill in their profile. The voting capabilities are a crowd-sourced mechanism to ensure that content stays up to date in the rapidly evolving world of research computing where new tools and methods are constantly emerging. Participants in the portal can vote on certain content to confirm its relevance. Curation by moderators ensures removal of obsolete information, often with guidance from the voting.

Many Portal functions have utility beyond the NECT, and the tag-based skills-matching functions in the Portal benefit from broad participation because deep knowledge is widely dispersed throughout the research computing community. Therefore, the Portal was developed with an eye toward making it possible for other communities to adopt it while maintaining their own branding and project workflows. This has already happened, with several pending proposals planning to use the portal for management of their own projects.

2.3 Expansion

The NECT enables researchers at small and mid-sized institutions in the region to take advantage of cyberinfrastructure as their work requires it. Simultaneously, NECT exposes a new generation of potential facilitators to the exciting and dynamic field of research computing earlier in their careers than most professional research computing facilitators came into the practice. As of March 2021, the NECT has launched 48 projects at 23 institutions, pairing a diverse population of student RCFs with knowledgeable mentors to assist researchers and educators in the region.

As noted previously, the strength of Portal features such as mentor matching and aggregation of learning resources grows with the size of the population participating. As a result, an active effort to find collaborators willing to bring their communities to the Portal has been underway for several years [4, 6]. Explorations to expand the Portal have yielded opportunities to collaborate with other programs focused on workforce development for the research computing community, including six Cyberteams and the XSEDE Campus Champions program. Representatives from each program meet monthly to exchange experiences, prioritize feature requests, and curate the tag taxonomy. Opportunities to partner with other communities of practice are constantly sought after and welcome.

3. LAUNCHING CAREERS

As noted in the introduction, the CAREERS program was initiated in July 2020. It leverages the NECT tools and methods to expand the RCF workforce pipeline, train student facilitators, and make advanced computing resources more readily available to researchers at small and mid-sized institutions in six additional states. The CAREERS program is led by one "anchor" institution from each state: Yale University (CT), University of Delaware (DE), Rutgers University (NJ), Rensselaer Polytechnic Institute (NY), Penn State (PA) and University of Rhode Island (RI). Yale serves as the PI institute for the grant. The goal for the student facilitator program is to complete 72 projects over the three-year duration of the grant (on average, four projects per state per year). Simultaneously, CAREERS is laying the groundwork for a new layer of support which shares domain and professional RCF expertise throughout the region.

The CAREERS Cyberteam adopted the NECT experiential learning model and engaged the NECT Program Manager to serve as Co-Program Manager, ensuring efficient transfer of knowledge. This has been an effective strategy for building a process that brings value to researchers at small and mid-sized institutions in a new region. It has also provided valuable insights that have led to adjustments that will improve the scalability of the NECT tools and methods beyond the 10 states (4 in NECT and 6 in CAREERS) where they are currently deployed. Some of these adjustments were made during the initial design of the CAREERS program, and others have been motivated by lessons learned during deployment.

3.1 Initial Adjustments to the Northeast Model

3.1.1 Program Management

As noted above, the CAREERS program engaged the NECT Program Manager to serve as Co-Program Manager on the CAREERS team, partnered with a Co-Program Manager new to the process. This approach has yielded several positive outcomes, outlined below in the Lessons Learned section.

3.1.2 Flexible Financial Setup

The NECT program distributed funding for students equally across three states — Maine, New Hampshire, and Vermont, with a slightly heavier burden placed on Massachusetts, where there is an abundance of small and mid-sized institutions as well as major research-oriented institutions. However, with the diversity of geography, institution, and population density as well as characteristics of institutions in the six states covered by the CAREERS program, it is necessary to have a more flexible arrangement so that each state can proceed at a rate that is natural for the conditions in that state (# of researchers/projects) and anchors (capacity to host projects).

A key decision was to have Yale serve as a "central banker" for the payments to student facilitators. We pre-allocated funds for just one project per year to each of the other anchor institutions and included the corresponding participant support costs in their subaward budgets. We then assigned the remaining project participant support funding to Yale's budget, where it can be disbursed flexibly and appropriately as projects are selected and funded.

As expected, this has allowed us to put money into the projects that need it, wherever they happen to be anchored and whenever they happen to start. We have also been able to adjust the funding for individual projects to reflect the level of the students and the duration of the projects without getting mired in whether a particular subaward budget has the right amount of remaining funds. Finally, it has allowed us to adapt to a significant amount of administrative and policy diversity among the various institutional offices (e.g., finance, sponsored projects, and international student offices) that we partner with at the institutions involved in the program. To cite one example encountered so far, some of our anchor institutions may not be able to pay participant support costs to their own students — even for training projects on their own campuses. However, they can pay the costs for students from other schools who participate in those projects, and, it turns out, their own students are also permitted to participate, so long as a different institution (Yale, in our case) officially oversees the relationships with those students and pays them.

4. LESSONS LEARNED

Porting an idea from one realm to another always presents challenges and requires adjustments. In this section, we highlight a few of the lessons that we have learned from the first year of the CAREERS program.

4.1 Portability of the Northeast Cyberteam Methods and Tools

One important takeaway has been how well the Cyberteam program developed by the NECT has worked for CAREERS "out of the box." When porting the Cyberteam procedures to CAREERS, it was unknown how the differences (e.g., new institutions, new steering committee members, new and larger region) between NECT and CAREERS would affect the efficiency of the process. However, we have been reassured to find that the NECT program design adapted quite smoothly. Our new institutions have different business processes, internal cultures, and resources; and our steering committee is composed of new faces who bring fresh perspectives to the idea of a Cyberteam. Yet, far more aspects of the Cyberteam process have worked seamlessly in CAREERS than have needed adjustment, and we have launched 19 projects in the first eight months of the program.

4.2 Active Program Management Facilitates Rapid Adoption of Process

One of the major takeaways from the NECT was the value of the Program Manager (PM). Program management by steering committee members at the institutions anchoring the program in each state was instrumental in recruiting projects, students, and mentors and in advocating for the program within the anchor institution and the surrounding schools. The NECT Program Manager organized the development of process, tools, and strategy as well as communications among program participants and in the research computing community.

CAREERS incorporated this knowledge into its program in two ways. First, since the Cyberteam concept and infrastructure had already been established by the NECT, CAREERS was able to function efficiently with 25% of the program management time spent in the first year of the NECT. Secondly, we elected to divide the program management time over two Program Managers instead of one. With a larger region and a goal of almost twice as many projects, it was necessary to have more than one set of eyes overseeing the program management. For example, early in CAREERS, through our marketing efforts, we had an unexpectedly robust but very welcome influx of student facilitators and found ourselves in the enviable position of needing to quickly match them, based on interests and skills, with nearly two dozen projects. Considering this, dividing the work between a team of Program Managers ensured CAREERS got off the ground quickly and operated smoothly. Onboarding the second co-PM also efficiently

exposed areas on the Portal where clarification/documentation was necessary. Having co-PMs also created a sense of camaraderie, and the Co-PMs were able to thoughtfully consider situations and adjust the process and tools as necessary. On the flip side, the other steering committee members were less involved with overall program management than their counterparts on NECT because there was more delineation between the anchor institution lead role and program manager role on the steering committee.

4.3 Shift in Steering Committee Role

When adapting the NECT methods to the new CAREERS program, we experienced a natural shift in the relationship between the steering committee and the NECT methodology. The NECT steering committee, having invented the methods, knows the procedures intimately and possesses an inherent sense of ownership over the process of running it. On the other hand, the CAREERS steering committee was tasked with learning the NECT process and how it works. As the NECT procedures and concept were ported into the new program, this led to a teacher-learner paradigm between the Program Managers and the steering committee that was not present for the NECT program. One way this difference was apparent was in the process of onboarding the new CAREERS steering committee. Having not had a direct hand in designing the Cyberteam project procedures, the new steering committee members identified several opportunities to improve the documentation of the procedures as well as areas where the methodology itself could be made more explicit and, therefore, easier to learn and follow.

4.4 Systematic Outreach Yields Robust Results

With established marketing materials, process, methodology, and proven results (i.e., NECT's successful experience) at the outset, CAREERS was able to tap into large numbers of students at two of its anchor schools: Rutgers University and the Rensselaer Polytechnic Institute. At Rensselaer, the co-PI introduced CAREERS to the Rensselaer Center for Open Source (RCOS), a community of open-source student developers at RPI that cultivates an inclusive, creative, and entrepreneurial community that seeks to empower students to develop open-source solutions to real-world problems. The ability to approach so many students early on allowed CAREERS to rapidly build a deep pipeline of students that could then be matched with projects as they were identified.

4.5 Portal Facilitates Adoption of Process, Creates Opportunities for Enhancement

A major resource that NECT made available to CAREERS was the Connect.cyberinfrastructure Portal. The Portal is a valuable tool for advertising the Cyberteam, maintaining active projects, and connecting the students and mentors with new projects. In addition to the public-facing portion of the Portal, the backend of the Portal has detailed forms for project descriptions, progress tracking, and outcomes. This backend has been instrumental in standardizing and reinforcing the process for launching and managing a successful Cyberteam project. When seeking projects, the steering committee members are empowered with a predetermined set of questions to present to a potential project leader. At project creation, it guides the project leaders on requirements for the project, such as an appropriate description, milestones, and deliverables that are consistent with a research facilitation engagement. As the project progresses, we use it to track progress towards the predetermined timeline and milestones. And at the end of the project, the project outcomes are captured for future reference. Across all these steps, the structure provided by the Portal ensures adherence to a standard process, leading to decreased friction from the very first CAREERS project.

When using the Portal, the CAREERS Cyberteam identified several improvements to be made to accommodate the expansion to both a new region and new users. We added a User Guide to enable more self-directed onboarding. The new User Guide details the different participant roles in a Cyberteam project and how each member can interact with the Portal to submit or apply for a project, respectively. In response to the enthusiastic influx of potential student facilitators, we added an "I'm Interested" button to every project page that students could use to indicate that they would like to be considered for that project. Student interest is collected in a report that the Program Managers use to assess potential student-project matches to present as candidates to the appropriate steering committee member. Lastly, a few small features were added to the Portal to enable the Program Managers to handle the expansion of both the user base and the projects. These include a sorted project view to find projects easily in various states and a report for tracking which students are engaged on which projects and which students are still seeking a position.

4.6 Working with Varying OSP and HR Policies Requires Precision and Adaptability

The Office of Sponsored Projects (OSP) at each anchor institution is a valuable partner for the program as they handle the disbursement of payments to our student facilitators for participating in their projects. We knew embarking on the CAREERS program that working with six different OSPs (one for each anchor institution) would present some logistical hurdles as we oriented everyone onto the same page. We found certain terminology was more effective in describing the student role in the program, which resulted in a smoother interaction. For example, the NSF grant that supports the CAREERS project is funding student facilitators via participant support costs. We found that it was important to use terms that reflected this category of expense (which is oriented toward educational experiences) and avoid terms such as "number of work hours," which connote student employment. Our efforts to "speak the language" of our OSP and HR partners have improved interactions. We found that the same approach was helpful when engaging with International Student Offices for F-1 visa student facilitators, as their situations have strict guidelines around how they can work or earn money.

5. FUTURE DIRECTIONS

The CAREERS program has been operating for approximately one year out of the three-year grant period. Therefore, we still have several objectives to tackle as well as an eye on goals for the future after the end of the grant in 2023. Some key areas of focus for the upcoming year are outlined below.

5.1 Develop a Distributed Expertise Network

In addition to expanding the NECT process to a new, larger region, CAREERS has a second novel objective to establish a distributed expertise network of professional RCFs directly accessible by researchers and educators in the region. As noted earlier, many small and mid-sized institutions cannot financially justify a full-time RCF, and providing domain expertise across all parts of the research computing community is challenging, even for institutions with multiple RCFs [5]. The CAREERS program plans to pilot a lightweight and cost-effective system to share RCF expertise across institutions of all sizes across the CAREERS region, leveraging the mentor network that we are building in the Portal. Engagement with

this expertise network will provide and create opportunities to multiply available expertise throughout the region while providing direct support to researchers at small/mid-sized institutions for shorter "help desk-like" engagements.

5.2 Provide Organized Access to Local, Regional and National Compute Resources

In addition to a lack of access to RCFs, researchers at small and mid-sized institutions may not have reliable channels for access to high-performance computing resources. The formation of the CAREERS Cyberteam has provided a route for researchers at these target institutions to acquire the cycles and storage necessary to effectively conduct their proposed projects from regional resources. The CAREERS and Northeast Cyberteams have offered access to supercomputers and clusters from several of their participating anchor institutions. For example, on one of its earliest projects, CAREERS provided computing resources allocated from Rensselaer Polytechnic Institute's AiMOS supercomputer, ranked 29th in the November 2020 TOP500 list of the world's most powerful computing systems. Building pipelines of access to a spectrum of local, regional, and national computing resources, as appropriate, plays a pivotal role in establishing an effective CI strategy for the region. We seek to build on this as CAREERS develops.

5.3 Enhance Portal and Further Engage with Research Cyberinfrastructure Community

Throughout the CAREERS program, we will continue to add enhancements and improvements to the Cnct.CI Portal as the needs arise. Along with the expansion of the Portal to CAREERS, several other Cyberteams and organizations have adopted the Portal to facilitate their programs, and any improvements made are accessible to every program. We have already seen benefits from the CAREERS enhancements in Northeast Cyberteam operations. Likewise, the expansion of the Portal to the additional Cyberteams creates opportunities to attract CAREERS participants, especially mentors and distributed RCFs. We will continue to engage with the community to identify opportunities to promote collaboration across the research cyberinfrastructure community through the methods and tools encapsulated within the Portal.

6. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. OAC-1659377, "CC* Cyber Team: Improving Access to Regional and National Cyberinfrastructure for Small and Mid-Sized Institutions" and Grant No. OAC-2018873, "CC* Team: CAREERS: Cyberteam to Advance Research and Education in Eastern Regional Schools."

We thank the many Cyberteam student facilitators for their participation in this effort and the Cyberteam mentors, researchers, and educators who have created experiential learning experiences for them; the Ask.CI site and locale moderators, and the hundreds of contributors to Ask.CI; to the leaders of the CAREERS, Great Plains Network, Kentucky, RMACC, SWEETER, TRECIS Cyberteams; and Research Computing leaders at the Colorado School of Mines Research Computing that enabled the participation of their respective constituents in the Connect.CI Portal Expansion pilot.

7. REFERENCES

[1] Gregory E. Monaco, Donald F. McMullen et al. 2016. The Role of Regional Organizations in Improving Access to the National Computational Infrastructure: A Report to the National Science Foundation. June 2016. DOI: https://doi.org/10.13140/RG.2.1.3171.9925

[2] John Goodhue, Julie Ma, Adrian Del Masetro, Sia Najafi, Bruce Segee, Scott Valcourt and Ralph Zottola. 2020. Northeast Cyberteam — Building an Environment for Sharing Best Practices and Solutions for Research Computing. In *2020 IEEE High Performance Extreme Computing Conference (HPEC)*. (Sept. 2020), 1–5. DOI: https://doi.org/10.1109/HPEC43674.2020.9286254

[3] Julie Ma, Torey Battelle, Katia Bulekova, Aaron Culich, John Goodhue, Jacob Pessin, Vanessa Sochat, Dana Brunson, Tom Cheatham, Sia Najafi, Chris Hill, Adrian Del Maestro, Bruce Segee, Ralph Zottola, Scott Valcourt, Zoe Braiterman, Raminder Singh, Robert Thoelen and Jack Smith. 2021. Ask.Cyberinfrastructure.org: Creating a Platform for Self-Service Learning and Collaboration in the Rapidly Changing Environment of Research Computing. *Journal of Computational Science Education* 12, 2, (Feb. 2021), 37–40. DOI: https://doi.org/10.22369/issn.2153-4136/12/2/9

[4] John Goodhue, Julie Ma, Adrian Del Maestro, Sia Najafi, Bruce Segee, Scott Valcourt and Ralph Zottola. 2020. Northeast Cyberteam: Workforce Development for Research Computing at Small and Mid-sized Institutions. In *PEARC '20: Practice and Experience in Advanced Research Computing*, July 26–30, 2020. Portland, OR. Association for Computing Machinery, New York, NY, 402–406 DOI: https://doi.org/10.1145/3311790.3396662

[5] John Goodhue, Julie Ma, Adrian Del Maestro, Sia Najafi, Bruce Segee, Scott Valcourt and Ralph Zottola. 2020. Northeast Cyberteam Program — A Workforce Development Strategy for Research Computing. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 8–11, DOI: https://doi.org/10.22369/issn.2153-4136/11/1/2

[6] John Goodhue, Julie Ma, Adrian Del Maestro, Sia Najafi, Bruce Segee, Scott Valcourt and Ralph Zottola. 2020. Extended Abstract: Northeast Cyberteam: Methods, Results and Expansion via the Cyberteam Portal. (Nov. 2020). In *SC '20: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. Retrieved from https://sc20.supercomputing.org/proceedings/sotp/sotp_pages/sotp129.html

Technology Laboratories: Facilitating Instruction for Cyberinfrastructure Infused Data Sciences

Zhenhua He[1]
happidence@tamu.edu

Jian Tao[1]
jtao@tamu.edu

Lisa M. Perez[1]
perez@tamu.edu

Dhruva K. Chakravorty[1]
chakravorty@tamu.edu

ABSTRACT

While artificial intelligence and machine learning (AI/ML) frameworks gain prominence in science and engineering, most researchers face significant challenges in adopting complex AI/ML workflows to campus and national cyberinfrastructure (CI) environments. Data from the Texas A&M High Performance Computing (HPRC) researcher training program indicate that researchers increasingly want to learn how to migrate and work with their pre-existing AI/ML frameworks on large scale computing environments. Building on the continuing success of our work in developing innovative pedagogical approaches for CI-training approaches, we expand CI-infused pedagogical approaches to teach technology-based AI and data sciences. We revisit the pedagogical approaches used in the decades-old tradition of laboratories in the Physical Sciences that taught concepts via experiential learning. Here, we structure a series of exercises on interactive computing environments that give researchers immediate hands-on experience in AI/ML and data science technologies that they will use as they work on larger CI resources. These exercises, called "tech-labs," assume that participating researchers are familiar with AI/ML approaches and focus on hands-on exercises that teach researchers how to use these approaches on large-scale CI. The tech-labs offer four consecutive sessions, each introducing a learner to specific technologies offered in CI environments for AI/ML and data workflows. We report on our tech-lab offered for Python-based AI/ML approaches during which learners are introduced to Jupyter Notebooks followed by exercises using Pandas, Matplotlib, Scikit-learn, and Keras. The program includes a series of enhancements such as container support and easy launch of virtual environments in our Web-based computing interface. The approach is scalable to programs using a command line interface (CLI) as well. In all, the program offers a shift in focus from teaching AI/ML toward increasing adoption of AI/ML in large-scale CI.

[1] High Performance Research Computing, Texas A&M University, College Station, TX

DOI: https://doi.org/10.22369/issn.2153-4136/13/1/8

CCS CONCEPTS

•**CS**→Computer Science; •**Cybertraining**→training on using cyberinfrastructure; •**HPC**→high performance computing • interactive computing • training • containers

Keywords

Artificial intelligence, Machine learning, Cyberinfrastructure, Portal, Jupyter notebooks, Keras, Training, Scikit-learn, High performance computing, Pedagogy, HPRC[1]

1. INTRODUCTION

AI/ML and data science frameworks have been rapidly adopted in various fields of science and engineering. We find that most researchers, however, first interact with these technologies on their personal computing devices. As we continue to work through the impact of the COVID-19 pandemic and remote working scenarios, we see that researchers use these devices to simultaneously create reports and publications, attend workshops and conferences, and use teleconferencing techniques. These devices continue their routine roles of serving as a means of communication and entertainment as well. As we move toward larger community-shared data sets, researchers struggle to cope past the barriers of computing and managed storage on their devices. In a similar vein, the scientific community is moving toward workflows like Federated learning approaches in AI/ML and data science that allow for data to be kept local to a site while the model is trained and shared among the collaborating sites. With the data residing on-site, federated learning approaches meet the privacy and compliance needs of data but are currently not viable on the personal devices used by researchers. Taken together, we note that researchers' personal computing machines, regardless of the hardware specifications, are unable to accommodate AI and machine learning workloads at scale.

In this emerging scenario, researchers find their work limited by their choice of technology. To get more access to computing and storage, they are increasingly looking toward campus, regional, and national large-scale computing options. Over the last decade, and in particular in the last couple of years, the CI landscape has grown increasingly complex. Campus CI resources, commonly referred to as clusters, have evolved from operating traditional high-performance computing (HPC) environments to now simultaneously operating mixed environments that support batch schedulers, containers, virtual machines, cloud-bursting, composability via software (Kubernetes) and hardware (Liqid composability). Accelerators such as graphical processing units (GPUs) now run in half-, single -, mixed-, and double-precision modes. Data is stored in different formats and may be generated or streamed into systems. In such an environment, researchers new to

CI environments rapidly encounter an intimidating cyberinfrastructure (CI) landscape. It is perhaps ironic that these seemingly daunting technologies were developed to support AI/ML frameworks.

While the development of scientific computing applications and analysis of scientific data continue to be done on the command line, a broad swath of researchers by and large prefer graphical interfaces where they can interactively develop their applications and visualize their data at the same time. To facilitate the transition of researchers to this seemingly daunting environment, we have adopted interactive graphical interfaces like Jupyter Notebooks [1] and Google Colab where researchers can develop their applications and visualize their data at the same time. Such interactive development environments can be paired with web-based portals such as the Design Safe portal and Open OnDemand that offer researchers the ability to interact with CI in application- and systems-driven approaches, respectively [2]. Adoption challenges around compatibility and usability remain in these approaches as well. Furthermore, "quality of life" features that are common in AI/ML toolkits on public cloud-based environments are often not available on these resources. We have previously reported on our work in advancing the state of CI-training at Texas A&M. Texas A&M High Performance Research Computing (HPRC) offers over sixty training seminars, courses, workshops, and week-long computing-camps that support CI aspirations ranging from middle school students to CI professionals [3–10]. During the COVID-19 pandemic-imposed workplace changes, we have learned that pedagogical approaches commonly adopted in in-person environments do not translate to a virtual learning setting. During late spring 2020, we adopted "peer-mentored" and "peer-led" learning approaches that were coupled to persistent chat (SWEETER Slack workspace, 970+ researchers) and class videos offered in short, intermediate, and longer formats [9]. Continuing in our quest to offer researchers scaffolded techniques in computing environments, we recently reported how support for features like containers, virtual environments, quota allocations, and easy buttons on a Web-based computing interface led to a new approach toward introducing containerization [10]. We anticipate that these practices will help improve the adoption of FAIR (Findability, Accessibility, Interoperability, and Reusability) [14] and FEAT (Fairness, Ethics, Accountability, and Transparency) [15] standards in research computing.

Data from the HPRC ticketing systems, short course participation trends, and user feedback show that researchers need a new form of AI/ML training that focuses on adoption of AI/ML practices on our CI environments rather than courses that merely focus on teaching the AI/ML technologies themselves. Engaging, accessible, and interactive computing environments offer new opportunities for CI-infused teaching while simultaneously improving user adoption of new technologies. These environments allow researchers to move away from the CLI and explore other avenues to learn and interact with popular AI/ML frameworks such as Keras, TensorFlow, and Torch. These avenues are explored in popular industry-sponsored courses like the NVIDIA Deep Learning Institute, Intel's offering on AI/ML, or campus CI offerings like our courses that introduce TensorFlow and Pytorch. In this study, we report on our approach toward advancing AI/ML training on CI resources in an approach named the Technology Laboratories or tech-labs (Figure 1).

2. METHODS

We take a leaf from the pedagogical approach used in the tradition of Physical Science laboratory classes, during which exercises were stacked and techniques simultaneously taught while elucidating the concepts covered during classroom lectures. In these programs, the study material was divided into two distinct groups. Students first learned a foundational approach, typically how to use a scientific instrument. Then they used that instrument to conduct a series of experiments, each geared toward understanding and exploring scientific concepts. The tech-labs teach learners in a similar vein. They first introduce researchers to the CI technologies and then show them how to effectively work with their existing research workflow in a large-scale CI environment. Much like physics laboratories, we structured a series of exercises that first helped the researcher gain familiarity with the "instrument" or mode of computing. This could be the command line, a graphical user interface (GUI) for a scientific application, or a Jupyter Notebook. Here, we explore the use of Jupyter Notebooks.

The tech-labs are geared toward improving adoption of better CI practices in research environments. As such, they assume that a researcher is proficient in the AI/ML approaches that will be used. During these labs researchers learn how to use these techniques on clusters. Pre-requisites and learning objectives were identified to ensure that researchers do not misunderstand the purpose of these courses and manage participant expectations. Interested learners who were new to AI/ML techniques were directed toward community learning resources such as our short courses that covered these topics at an introductory level [16]. Toward facilitating a flipped classroom approach, curricular materials are pre-staged on the HPRC website along with relevant Git

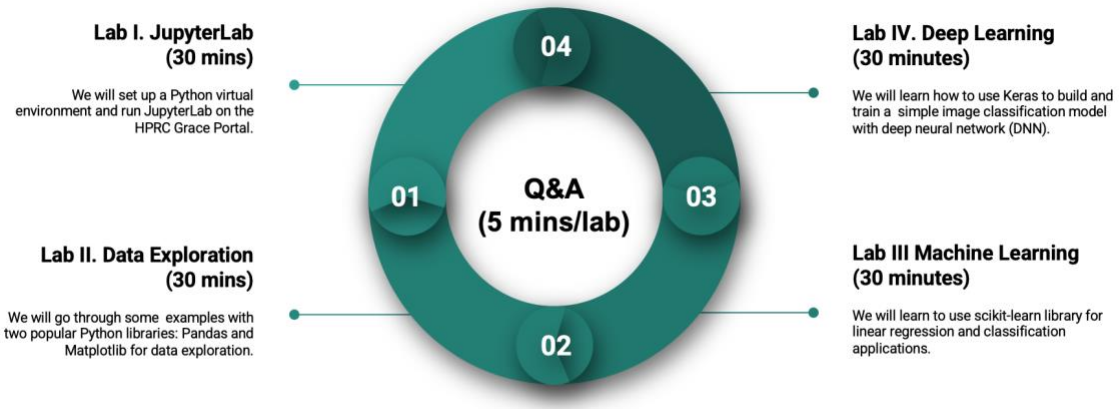


Figure 1. Structure of the AI/ML and Data Science Technology Laboratories.

repositories. Delivery is performed entirely using hands-on practices on CI resources, with no time spent exclusively on lectures introducing the AI/ML topics.

The class size is typically limited to 20 researchers to ensure that the instructor can assist the learners as they work through the exercises. Employing teaching assistants offers the option to build out the classroom. The tech-labs are divided into four components. This is described in the schematic presented in Figure 1. In the first session, researchers learn about the platform that will be used to access or interact with the CI resources. This session lasts for 15 minutes and is followed by three sessions of equal length that cover the AI/ML areas of interest. Each session is separated with a five-minute session during which participants can take a break or interact with the instructor.

2.1 Using the HPRC Open OnDemand (OOD) Web Portal

Noting the appeal of interactive computing approaches, especially to researchers who are not familiar with large-scale CI practices, we have developed a special version of the Open OnDemand platform for researchers at Texas A&M HPRC [10]. The adoption of this portal by researchers and its use in CI-training have been reported elsewhere. During the tech-labs, we first teach researchers how to use CI resources via the portal. Figure 2 shows the various applications available on the Texas A&M portal on the Terra supercomputer. Researchers learn that the home and scratch directories can be accessed, and files can be uploaded and downloaded easily. Researchers can view the status of their active jobs from the Jobs tab and access the shell from the Clusters tab, offering convenient access to the command line interface. Under the Interactive Apps tab, there is ready access to interactive applications such as Bio apps, GUI apps, JupyterLab, Jupyter Notebook, etc., for users to use.

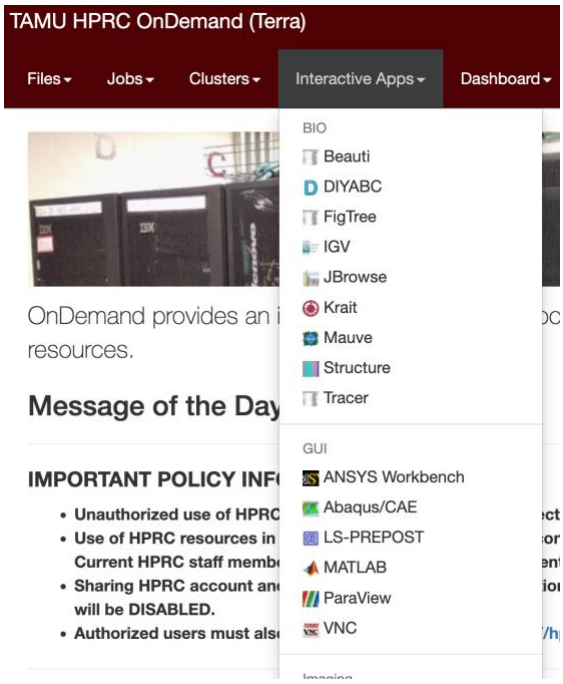


Figure 2. Some software applications built into the Texas A&M HPRC Open OnDemand (OOD) Portal for the Terra computing cluster.

2.2 Setting up Virtual Environments

Virtual environments offer a convenient mechanism for each project to have its own isolated environment on CI resources where its required dependencies can be installed regardless of what dependencies other projects require. We next teach the researchers how to use Anaconda commands to create virtual environments. These instructions are shown in Figure 3. For this project, we install scikit-learn and tensorflow packages into the virtual environment.

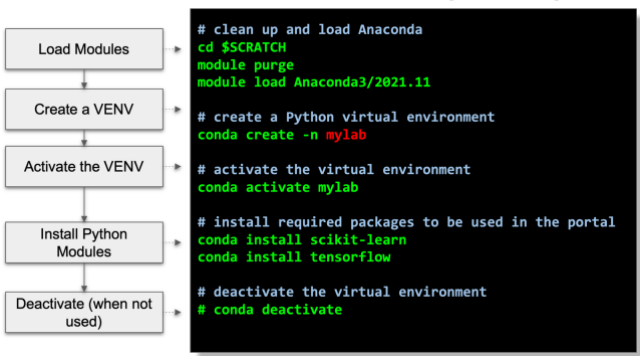


Figure 3. Creating a virtual environment with Anaconda commands.

2.3 Data Exploration Laboratory

In this section, we introduce some data science problems and two popular python libraries: Pandas and Matplotlib. Pandas is a Python library for data manipulation and analysis while Matplotlib is used for data visualization. These are taught using the JupyterLab interface as shown in Figure 4. During this session, researchers learn data manipulation skills. They are introduced to Pandas via examples of the two Pandas data structures — series and dataframe — and operations are provided such as retrieving and dropping entries, indexing, selecting, filtering, sorting, and ranking (based on the positions after sorting). The skills learned will be used in the exercises in the next session. Also, examples of using the Matplotlib object-oriented API to create figures and plots are taught. The advantage of using an object-oriented API becomes apparent when more than one figure is created or when a figure contains more than one subplot. Colormap, contour figures, surface plots, wire-frame plot, and contour plots with projections are also introduced. Colormaps and contour figures can be used to plot functions with two variables with the third dimension encoded. Passing a projection='3d' keyword argument to the add_axes or add_subplot methods can enable plotting 3D figures for better visualization.

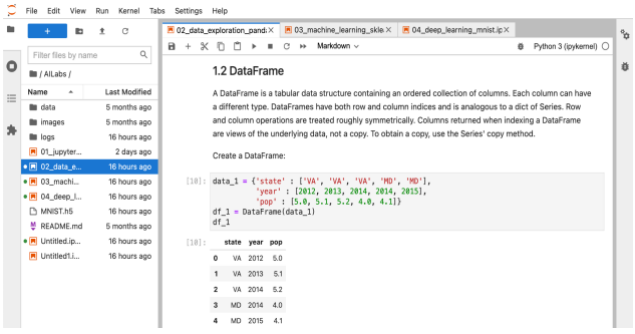


Figure 4. JupyterLab interface for the Data Exploration Lab.

2.4 Machine Learning Laboratory

In the machine learning session, we show the relationship between machine learning and artificial intelligence. [12]. During this lab, researchers learn how to use a machine learning library named Scikit-learn to work on regression, classification, and clustering problems with different algorithms. Linear regression is used to estimate the relationship among variables and predict a continuous-valued attribute of an object. It fits a linear model to minimize the sum of squares between the observations and predictions. In these exercises, researchers import the required libraries and models, generate an x-dataset with the numpy.linspace() function and a y-dataset, create an instance of the LinearRegression() model, and fit the model with x- and y-datasets. Researchers next check the coefficients for the linear regression model and the determination coefficient R2 with score() function, and visualize the data points and the best fit line. They finally work on a polynomial fitting exercise with linear regression by modifying the x-dataset with more polynomial terms with numpy.hstack().

Researchers are next shown how classification is used to identify the category an object belongs to based on a training dataset in which the membership of the objects is known. They are introduced to three concepts here: (i) Support vector machine (SVM) aims to find the hyperplane that separates binary sets with maximum margin to both classes. (ii) K-Nearest Neighbors (KNN) works based on the assumption that every data point belongs to the same class with the majority of its surrounding data points. In other words, it classifies a new data point based on similarity. (iii) Clustering is the task of separating the data points into groups such that the data points in the same groups are more similar than the data points in other groups. SVM and KNN classifiers are introduced in the lab's exercises. Students worked on a K-Means clustering exercise as well. In this exercise, the K-Means algorithm starts with a set of randomly selected centroids that are the beginning centers for every cluster and performs iterations to optimize the centroids' locations.

2.5 Deep Learning Laboratory

Deep Learning (DL) is included in the tech-labs since it finds applications in research. Because of its capability to handle high-dimensional data, DL is good for automatic feature extraction as well. During this session, researchers learn that deep learning is a subset of machine learning methods that are based on neural networks to improve algorithms by data and the relationship among AI, ML, and DL. This relationship is described in Figure 5. Three different DL methods including supervised learning, unsupervised learning, and reinforcement learning are explained. In supervised learning, the models are trained with labeled datasets. Regression and classification problems belong to this learning type. In unsupervised learning, the models are trained with unlabeled datasets. Clustering problems are in this category. In reinforcement learning, there are no training datasets, and it is about how agents take actions in an environment to maximize the reward.

During this session, researchers are taught to distinguish between traditional modeling that utilizes different numerical methods to solve the governing equations, and ML modeling that trains models with datasets and predicts unknown data with the trained models.

In the final session, researchers are introduced to Keras [17], the popular open-source neural network library. They perform an exercise during which they build a handwritten digits classifier [18]. The components include how to import the required libraries; load, split, and normalize the MNIST (Modified National Institute of Standards and Technology) [13] dataset; build a multi-layered neural network model with Sequential class; compile the model with an optimizer and a loss function; train the model with fit function on the train datasets; evaluate the model on test dataset; and predict. Finally, they study the images that were not correctly predicted and understand the potential reasons leading to these erroneous results.

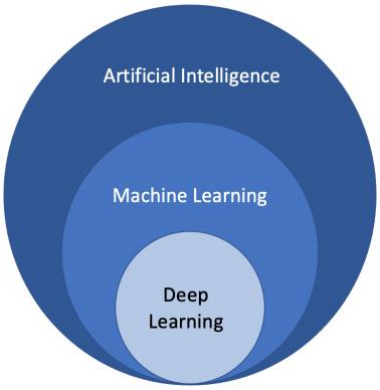


Figure 5. Relationship between AI, ML and DL.

The topics and in-class exercises covered during the tech-labs are summarized in Table 1. We have offered these technology laboratories every semester since 2020. These have been offered in hybrid (in-person and remote) and in remote (virtual, Zoom) formats.

Table 1a. Covered topics, in-class examples and exercises.

Topics and exercises covered	Topics and exercises covered
Create a virtual environment*	Write a dataframe to a file
Launch JupyterLab on OOD portal*	Matplotlib — line plot
Create a Pandas series	Matplotlib — subplots
Get the index and values of a series	Matplotlib — color map
Series indexing	Matplotlib — contour figures
Series filtering	Matplotlib — 3D figures
Series sorting	Case study — house market data*
Series mathematical operations	Linear regression
Create a Pandas dataframe	Polynomial fitting with linear regression
Specify the sequence of dataframe columns	SVM classification
Add a column to a dataframe	KNN classification
Retrieve a row from a dataframe	K-Means clustering
Retrieve a column from a dataframe	Principal component analysis

Table 1b (continuation of Table 1a).

Topics and exercises covered	Topics and exercises covered
Drop a row from a dataframe	Linear regression with a neural network library Keras
Drop a column from a dataframe	Build a handwritten digit classification model
Dataframe filtering	Train the model
Dataframe sorting	Evaluate the model performance
Load a file to a dataframe	Make predictions with the trained model

3. RESULTS

The tech-labs were offered by Texas A&M HPRC from fall 2020. Each tech-lab session ran for a duration of two-and-a-half hours. The labs assumed that researchers have prerequisite knowledge of the AI/ML and data frameworks. This marked a departure from our regular course of instruction where all materials were covered at an introductory level. The tech-labs were offered in hybrid (in-person and virtual) and virtual-only settings. Virtual classrooms were offered on the Zoom video-conferencing program, with dedicated support facilitated by the breakout room functionality. In spring 2022, the classes returned to a hybrid format. Table 2 lists the teaching modality and the number of registered students.

We have previously found that researchers are comfortable with this duration in sessions that include hands-on exercises. During the sessions, students created teams to complete the in-class exercises. Creating such teams offered the researchers opportunities for peer-learning as well as continued discussions after the classroom. To accommodate the taxing demands of learning via a "live" virtual session, we included several breaks. The Zoom "class" included a main session with several breakout rooms for teams' projects. With a view toward supporting researchers at different levels of learning, a competition-based approach was not adopted.

Table 2. Technology laboratories offered at Texas A&M.

Date offered	Modality	Registered attendees
2022-03-11	In-person	20*
2021-10-29	Hybrid	44
2021-06-02	Virtual	17
2020-10-30	Virtual	55

* registration for hybrid modality but course was held in an in-person modality.

During these tech-labs, learners were introduced to Jupyter Notebooks. This was followed by exercises in data exploration using Pandas and Matplotlib, machine learning using Scikit-learn for linear regression and classification applications, and Deep Learning using Keras to create and train a simple image classification model with a deep neural network (DNN). This is made possible by introducing a series of enhancements such as container support and easy launch of virtual environments in our Web-based computing interface. The approach can be readily expanded to support CI-adoption of Python-based AI/ML frameworks on the command line, AI/ML in Matlab, and other data science approaches. In all, the program offers a shift in focus from teaching AI/ML toward increasing adoption of AI/ML in large-scale CI.

4. CHALLENGES FACED AND LESSONS LEARNED

Owing to its format, the tech-lab encourages discussions between the instructor and participating researchers. It is not surprising that the tech-labs are a demanding teaching experience during which participant researchers ask several questions. This is particularly problematic during virtual sessions when using breakout rooms to have students communicate with each other. The instructor can join the breakout rooms to help answer their 'big' questions that they cannot solve together in a breakout room. It is, however, challenging for a single instructor to handle several breakout rooms. More teaching assistants should be trained for the short course to answer questions if the breakout room feature is used. Learning from our challenges in supporting all participating researchers, in summer 2021, we moved from a single instructor-supported instruction model to one that included two teaching assistants.

5. SUPPORTING INFORMATION

All training materials used in this study are available to the community via the Texas A&M HPRC website [16]. Videos and course recordings are available at the Texas A&M HPRC channel on YouTube. The community is invited to join the SWEETER slack workspace at https://hprc.tamu.edu/sweeter. Surveys and review exercises that will be developed as part of this longitudinal study may be requested from the author. Please send us feedback about your adoption experience via an email to help@hprc.tamu.edu.

6. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation (NSF) award number 1925764, "CC* Cyberteam SWEETER," and NSF award number 2019129, "MRI:FASTER," NSF award number 1730695, "CyberTraining: CIP: CiSE-ProS: Cyberinfrastructure Security Education for Professionals and Students", NSF award number 2019136, "CC* BRICCs: Building Research Innovation at Community Colleges," and NSF award number 1829799, "Cybertraining: CMS3."

7. REFERENCES

[1] 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. and Jupyter development team. 2016. Jupyter Notebooks — a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. F. Loizides & B. Schmidt (Eds.), 87–90. DOI: https://doi.org/10.3233/978-1-61499-649-1-87

[2] Hudak, D., Johnson, D., Chalker, A., Nicklas, J., Franz, E., Dockendorf, T. and McMichael, B. L. 2018. Open OnDemand: A web-based client portal for HPC centers. *Journal of Open Source Software* 3, 25 (May 2018), 622. DOI: https://doi.org/10.21105/joss.00622

[3] Texas A&M High Performance Research Computing Website. Retrieved from https://hprc.tamu.edu

[4] Chakravorty, D. K., Pennings, M., Liu, H., Wei, Z., Rodriguez, D. M., Jordan, L. T., McMullen, D., Ghaffari, N. and Le, S. D. 2019. Effectively Extending Computational Training Using Informal Means at Larger Institutions. *Journal of Computational Science Education* 10, 1 (Jan. 2019), 40–47. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/7

[5] Chakravorty, D. K., Pennings, M., Liu, H., Wei, Z., Rodriguez, D. M., Jordan, L. T., McMullen, D., Ghaffari, N., Le, S. D., Rodriquez, D. and Buchanan, C. 2019. Evaluating Active Learning Approaches for Teaching Intermediate Programming at an Early Undergraduate Level. *Journal of Computational Science Education* 10, 1 (Jan. 2019), 61–66. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/10

[6] Seo, J. H., Bruner, M., Payne, A., Gober, N., McMullen, D. and Chakravorty, D. K. 2019. Using Virtual Reality to Enforce Principles of Cybersecurity. *Journal of Computational Science Education* 10, 1 (Jan. 2019), 81–87. DOI: https://doi.org/10.22369/issn.2153-4136/10/1/13

[7] Chakravorty, D. K., Pennings, M., Liu, H., Thomas, X., Rodriguez, D. and Perez, M. 2020. Incorporating Complexity in Computing Camps for High School Students — A Report on the Summer Computing Academy Program at Texas A&M University. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 12–20. DOI: https://doi.org/10.22369/issn.2153-4136/11/1/3

[8] Chakravorty, D. K. and Pham, M. T. 2020. Evaluating the Effectiveness of an Online Learning Platform in Transitioning from High Performance Computing to a Commercial Cloud Computing Environment. *Journal of Computational Science Education* 11, 1 (Jan. 2020), 93–99. DOI: https://doi.org/10.22369/issn.2153-4136/11/1/15

[9] Chakravorty, D. K., Perez, L. M., Liu, H., Yosko, B., Jackson, K., Rodriguez, D., Trivedi, S. H., Jordan, L. and Le, S. 2021. Exploring Remote Learning Methods for User Training in Research Computing. *Journal of Computational Science Education* 12, 2 (Feb. 2021), 11–17. DOI: https://doi.org/10.22369/issn.2153-4136/12/2/2

[10] Lawrence, R., Pham, T. M., Au, P. T., Yang, X., Hsu, K., Trivedi, S. H., Perez, L. M. and Chakravorty, D. M. In press. Expanding Interactive Computing to Facilitate Informal Instruction in Research Computing. *Journal of Computational Science Education*.

[11] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal* 2014, 239 (Mar. 2014), 2. Retrieved from https://dl.acm.org/doi/10.5555/2600239.2600241

[12] Arthur L. Samuel. 1959. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3, 3 (Jul. 1959), 210–229. DOI: https://doi.org/10.1147/rd.33.0210

[13] Yann LeCun, Corinna Cortes and Christopher J. C. Burges. MNIST handwritten digit database. Retrieved from http://yann.lecun.com/exdb/mnist/

[14] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, et al. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3 (Mar. 2019), 160018. DOI: https://doi.org/10.1038/sdata.2016.18

[15] Ayanna Howard, Jason Borenstein and Kinnis Gosha. 2019. NSF-funded Fairness, Ethics, Accountability, and Transparency (FEAT) Workshop Report. In *NSF Workshop Reports* (Oct. 2019). Retrieved from https://par.nsf.gov/servlets/purl/10139705

[16] Texas A&M High Performance Research Computing. Short Courses. Retrieved from https://hprc.tamu.edu/training/

[17] Keras: the Python deep learning API. Retrieved from https://keras.io/

[18] GitHub - happidence1/AILabs. Retrieved from https://github.com/happidence1/AILabs

Expanding Interactive Computing to Facilitate Informal Instruction in Research Computing

Richard Lawrence[1]
rarensu@tamu.edu

Tri M. Pham[1]
phamminhtris@tamu.edu

Phi T. Au[1]
thau@tamu.edu

Xin Yang[1]
karen890@tamu.edu

Kyle Hsu[1]
hsuk1001@tamu.edu

Stuti H. Trivedi[1]
stutitrivedi1373@tamu.edu

Lisa M. Perez[1]
perez@tamu.edu

Dhruva K. Chakravorty[1]
chakravorty@tamu.edu

ABSTRACT

Successful outreach to computational researchers for informing about the benefits of switching to a different computing environment depends on the educator's ability to showcase practical research and development workflows in the new computing environment. Interactive, graphical computing environments are crucial to engage learners in computing education and offer researchers easier ways to adopt new technologies. Interactive, graphical computing allows learners to see the results of their work in real time, which provides the needed feedback for learning and enables chunking of complex tasks. Moreover, there is a natural synergy between computing education and computing research; researchers who are exposed to new computing skills within the context of an interactive and engaging environment are more likely to retain the new skills and adopt the new computing environment in their research and development workflows. Support for interactive, graphical workflows with modern computing tools in containerized computing environments has to be incorporated on high performance computing systems. To begin to address this deficiency, here we discuss our approach to teach containerization technologies in the popular integrated development environment of the Jupyter Notebook. We report on our scheme for implementing containerized software environments for interactive, graphical computing within the Open OnDemand (OOD) framework for research computing workflows, providing an accessible on-ramp for researchers transitioning to containerized technologies. In addition, we introduce several quality-of-life improvements for researchers and educators that will encourage them to continue to use the platform.

[1] High Performance Research Computing, Texas A&M University, College Station, TX

DOI: https://doi.org/10.22369/issn.2153-4136/13/1/9

CCS CONCEPTS

•**CS**→Computer Science; •**Cybertraining**→training on using cyberinfrastructure; •**HPC**→high performance computing • interactive computing • training • containers

Keywords

Open on demand, Cyberinfrastructure, Portal, Jupyter notebooks, Singularity, Containers, Training

1. INTRODUCTION

Cyber Infrastructure (CI) is challenging; a complex computing environment is a barrier to learning for beginning researchers. There is an urgent need to offer scaffolded techniques in computing environments to reduce the barriers toward adoption of computing approaches. Making the computing environment engaging and more accessible provides opportunities for teaching in that environment while simultaneously improving user adoption of new technologies.

While the development of scientific computing applications and analysis of scientific data continues to be done on the command line, a broad swath of researchers by and large prefer interactive, graphical interfaces where they can develop their applications and visualize their data at the same time. An extremely popular example is the Jupyter Notebook [1] integrated development environment for Python users. Many important scientific research tasks are done in Python due to its ease of use in rapid development. For example, the popular artificial intelligence and machine learning (AI/ML) frameworks Keras, TensorFlow, and Torch are all Python language compatible frameworks. Although the workload for machine learning would most likely be distributed across a cluster, development is done interactively. This enables researchers to quickly debug their code by visualizing its output. The application in its final form would be exported to a text file for use in a batch script. If the researchers wished to deploy their application in a specific computing environment, then it makes sense that they would want to be developing and debugging applications in that same environment.

Powerful AI/ML tools such as the above are being widely adopted by the research community, as are the standards for research and development, FAIR (Findability, Accessibility, Interoperability, and Reusability) and FEAT (Fairness, Ethics, Accountability and Transparency). However, adoption of CI technologies and data frameworks that are needed to effectively support those standards

are lagging behind the adoption curve. As a result, educators lack the tools to properly teach AI/ML skills. This has hindered the adoption of FEAT and FAIR community standards for research. We need to find ways to make it easier to teach researchers how to use these technologies.

Education for research computation is well-served by interactive, graphical environments, which follows from the same reasoning. Researchers learning to use Python or any of the scientific Python libraries are more likely to retain knowledge and continue using the platform if it is the platform that will be used for research and development workflows. The Jupyter Notebook is well-suited for both Python education and development. The ability to visualize code, inputs, and outputs at the same time makes the integrated development environment a must for Python education. Cluster administrators are well-motivated to support the Jupyter notebook and other interactive and graphical computing interfaces to lower the barrier of entry for future researchers.

The Open OnDemand (OOD) platform provides a framework for supporting interactive and graphical environments [4]. It provides a Web interface to the computing cluster, which is advantageous because Web interfaces are generally platform-independent, which is good news for users working remotely from their personal machines. OOD interfaces on its back end with the computing cluster's batch system. Users can submit and monitor their batch jobs using an interactive graphical interface. The template for the job management interface is customized for each cluster. OOD also provides a framework for creating additional Web servers in the form of batch jobs assigned to compute nodes in the cluster. OOD allows the user to conveniently connect through a Web-based interface to this personal Web server. This is useful for a wide variety of graphical applications that can be served to Web browsers, from the humble Jupyter Notebook to as much as an entire virtual desktop.

The OOD platform installed out-of-the-box can be enhanced with features for educators and researchers. Its primary function is to support specific tasks but not necessarily to provide a comprehensive suite of tools to support a more general workflow. This is disadvantageous for educators who wish to quickly and effectively on-ramp researchers to the cluster for two reasons. First, if the workflow task the educators wish to teach isn't supported by the OOD platform, then they may need to use unrelated platforms for their teaching, which confuses researchers and leaves them with no clear transition to the development phase. Second, if researchers need to continue using command-line interfaces for some aspects of their workflow, then the benefit of adopting the OOD platform is marginal, which limits their openness to future education delivered through the platform.

One of the most important technologies for compliance with these principles is the container, a technique for bundling a software environment for maximum portability. A container allows the bundled software to run on any machine with the same basic architecture. The most popular container technology is Docker [5], which provides a file format for bundling software environments and a public repository named DockerHub for sharing those files. Many important scientific research frameworks, such Bioconda, LAMMPS, and Tensorflow, have already been containerized in the Docker format, allowing researchers working at home quick access to scientific software. However, for security reasons, Docker is not suitable for use on high performance clusters where many users share a filesystem. A solution to this problem is Singularity [6], which is an alternative to Docker that can both read the Docker file format and safely operate in cluster environments. Containers are clearly an important technology for the future, yet support for containers and especially support for container education lags behind the curve.

Adding container support to existing interactive, graphical interfaces will help educators better demonstrate container usage, showcase the utility of the approach, elevate the level of discourse to more advanced topics like reproducibility in computational sciences, and ultimately make it easier for researchers to navigate the transition to containerized computing environments. Alongside this, cluster administrators can also support sharing locally developed containers with external researcher collaborators. In the long run, these strategies will lead to a greater rate of container adoption and adoption of the FAIR and FEAT principles for scientific research, especially in the fields making use of biologically-inclined and AI/ML tools.

2. METHODS

In order to facilitate computational skills training for researchers, an interactive mechanism was needed. Open OnDemand offers that, but it is limited in its general scope to Jupyter notebooks and apps that represent the desktop. These were used for some educational courses. In addition, quality of life improvements were developed and support for containers was implemented. These features were also used for some educational courses.

2.1 Experimental Design

Open OnDemand has been used as an educational platform in several Short courses and Primers [2] since 2019. Students were polled for feedback after each to determine the achievement rate for the educational goals of each course. Results from this program have been reported previously [3].

2.2 System Design

At Texas A&M, several quality-of-life improvements and add-ons were implemented in the Open OnDemand framework, seen in Figure 1. These generally fall under the scope of User Experience, which is to say, how easily the user can get what they need from the platform. This is especially important for researchers who aren't necessarily trained in cluster computing technologies but still want to take advantage of those resources. These features help to draw researchers to the platform, which makes them more receptive to education delivered through the platform.

The helpdesk ticket submission form is a tool that allows researchers who encounter problems to request help from the cluster administrators by populating a form conveniently located in the OOD web interface. This makes it easier for researchers to get help since they don't have to leave the website to find instructions for how to do so. It also helps researchers get quick responses from cluster administrators because the form fields encourage the user to provide the administrators with the information needed to understand the problem.

The job composer form is a tool that allows researchers to edit batch scripts and submit them to the cluster by populating a form conveniently located in the OOD web interface. Parameters are selected in fields, while a preview shows the script that results from those parameter choices. Templates are provided for common scientific programming applications such as R and MATLAB. This makes it easier for educators to showcase the basics of batch scripting without the burden of also teaching command line syntax. It also enables researchers to get started quickly with their research tasks.

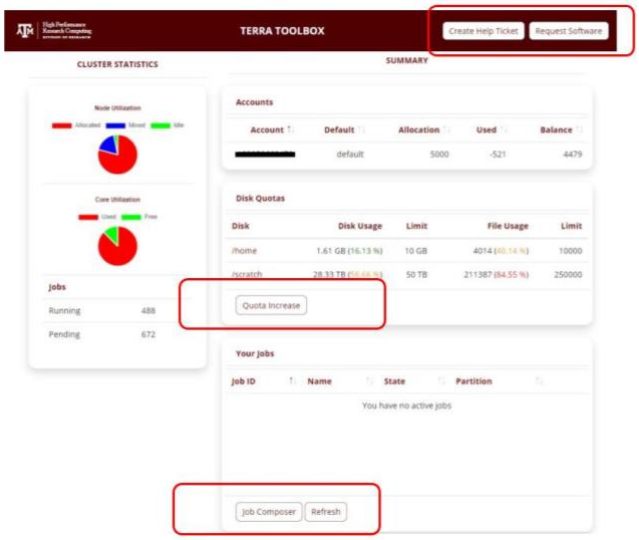


Figure 1. Screenshot of Dashboard page from Terra cluster maintained by Texas A&M High Performance Research Computing (HPRC). Tables shown: Disk Quota, Allocation, and Job monitoring. Highlighted: links to Create Help Ticket, Request Software, Quota Increase, and Job Composer forms.

The quota request form is a tool that allows researchers to ask for changes to their disk space allocations by populating a form conveniently located in the OOD web interface. This is a simple task, but researchers need to do it so rarely that when it comes time for it, they are often confused. Streamlining this process helps save time for researchers and cluster administrators.

The view-only link feature for the VNC interactive app is a tool for screen sharing implemented in the OOD framework. This is used by educators to provide instant feedback while teaching interactive computing skills. Each interactive job in the OOD framework creates a web server on a compute node and connects that web server to the student's browser. The student uses that web interface to perform activities on the compute node. In the case of VNC, the web server can also provide a second copy of the web interface that is read only. A link to this read-only web server can be distributed to allow teachers and collaborators to watch live as the job owner interacts with the compute node. In the case of VNC, this is useful for education of scientific computing skills for graphical software applications that display in a window, which is the majority.

A scheme for supporting containerized environments within OOD was developed. Support for containers can be added to any OOD interactive app simply by inserting the correct container runtime syntax in the template job script. In the case of the Terra cluster at Texas A&M University, the container runtime is Singularity [6]. For example, if the command to launch a Jupyter Notebook application that appears in the job template takes the form "jupyter notebook [arguments]," then the new syntax would be "singularity exec [image] jupyter notebook [args]." This substitution is relatively trivial, which is exactly why containers are such an important technology for clusters to support.

The major hurdle for implementing containerized app support in OOD is not the container syntax itself but the integration of that syntax into a larger structure that is flexible enough to support multiple computing environments. Although containers are fantastic, they are not for everyone. Cluster administrators would wish to continue maintaining their local computing environments to meet their users' needs. OOD does not provide a significant example or tutorial on how to go about supporting multiple computing environments. However, the underlying technologies are powerful enough to enable the necessary developments.

The OOD user interface for each app is assembled from a minimal list of specifications, which is convenient for simple apps, but the available options for customizing that user interface through the specifications are limited. The most significant limitation is that every field and widget with which the user may or may not need to interact must be displayed. In the case of multiple computing environments, where the information that is needed to select an environment varies greatly depending on which environment is to be used, this quickly grows out of control, hindering usability. The solution to this problem is to implement the OOD optional feature to support JavaScript in the user interface. This allows for the addition of scripts for toggling the visibility and values of the fields based on the content of the fields as shown in Figure 2.

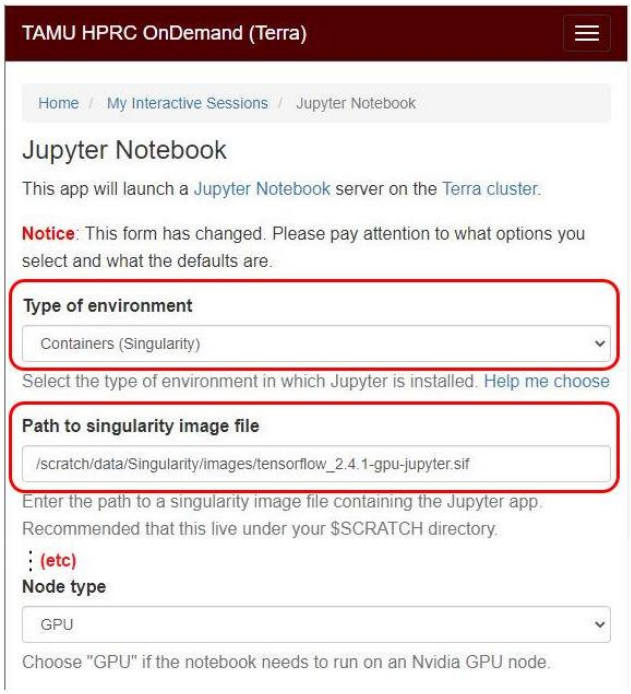


Figure 2. Screenshot of the Jupyter Interactive App form from the Terra cluster at HPRC. Highlighted: Drop-down menu for selecting environment type, Field for selecting container file.

3. RESULTS

Open OnDemand has been used by Texas A&M High Performance Research Computing (HPRC), and Laboratory for Molecular Simulation (LMS) for research computing since 2018. Results for using OOD in our introductory courses during 2020 have already been reported [3]. With over 1600 active users on the platform, several courses now use the Texas A&M OOD platform as the preferred teaching method. This includes a variety of advanced scientific computing skills in addition to the basic topic previously studied, as shown in Table 1. An important feature that accompanies this growth is the view-only link for VNC sessions, which enables a huge number of scientific applications that are compatible with VNC. However, the real winners are the many scientific applications based on the Python language, all of which

can now be demonstrated using the same hardware and software environment the researchers will ultimately use. The educational benefits of an interactive computing Web interface have made themselves evident through rapid adoption by instructors at Texas A&M HPRC as shown in Table 1. The view-only link VNC feature of OOD was also used during the instruction of CHEM 119: Fundamentals of Chemistry I, a formal laboratory course taught in Fall 2020 at Texas A&M. Ninety-five students practiced their computational skills as part of the course.

Table 1. List of informal HPRC short courses and Primers in Spring 2021. New OOD features that were used in the course and the number of participants attending. Courses continue to be offered on these topics.

Course Title	OOD Features	Partici-pants
Linux Primer	Shell access	110
Schedulers Primer	Job composer	292
Jupyter Notebook Primer	Jupyter Notebook	126
Cluster Usage Primer	Quota request form	203
Introduction to Python	Jupyter Notebook	235
Scientific Python	Jupyter Notebook	28
Introduction to Containers	Container support in Jupyter Notebook	4
Molecular Dynamics — NAMD	View-only link VNC	6
Introduction to Quantum Chemistry Simulations with ORCA	View-only link VNC	17
Drug Design — COVID19 for the cure	View-only link VNC	12
Intro to Xarray and Dask	JupyterLab Notebook	13
Intro to Deep Learning with Pytorch	JupyterLab Notebook	11
Intro to Deep Learning with Tensorflow	JupyterLab Notebook	19

3.1 Case Study Teaching Containerization on Clusters

The newest educational offering, Introduction to Containers [7], featuring OOD support for Containerized Jupyter Notebooks, demonstrates the importance of this ongoing effort. Introduction to Containers was a short course of two-and-a-half hours offered on April 30, 2021, October 15, 2021, and March 11, 2022. It was developed for the purpose of educating researchers about the use of containers. Due to COVID-19 conditions, the course was developed to be delivered virtual-only via Zoom.

The intended learning objectives were to:

1. Provide the researchers the information they need to decide if it is worth investing their time into making the switch to using containers in their research computing environment.
2. Familiarize the researchers with basic container workflow tasks, including use of the Singularity runtime, utilization of Docker-format container repositories, and support for containerized interactive graphical applications.
3. Demonstrate advanced applications of containers across multiple scientific disciplines to appeal to a wide audience.

To accomplish this, the course began with an overview of container technologies, leading into interactive exercises with which the students could follow along, and finally showcasing a selection of advanced topics. Learners were offered the materials from the course (for future reference) via the course website and our wiki. Wiki entries were updated to reflect the contents of the course.

The course had a number of descriptive examples for students to observe. These were followed with hands-on examples that allowed the participating researchers to practice their newly acquired knowledge. The Open OnDemand platform helped to make this course a success. Easy access to the cluster compute nodes via the OOD web interface removed the need for students to install SSH software on their local machines. Although command-line usage was used for some exercises, the equivalent batch script variations of those workflows were also demonstrated, which are entirely compatible with the OOD Job Composer feature. In principle, the researcher need not use the command line at all for those workflows. Finally, container support for interactive applications such as the Jupyter Notebook gave researchers the comfort of knowing that they could continue using their favorite graphical workflows should they choose to make the switch to containers.

The course was evaluated by polling the students for feedback at the end of the course. The poll was delivered through Zoom during the course and also by Google Form after the end of the course. Although there were few responses (due to the low attendance), the responses were overwhelmingly positive, indicating a high rate of success for the first two learning objectives. Students indicated that the course did give them the information they needed to decide whether to make the switch to containers and that they had gained familiarity with the basic workflows. The feedback indicated that the advanced topics were not of interest to most students, which leads us to the conclusion that the course can also be shortened by removing the advanced topics and offering it as a 1-hour primer. This could potentially increase interest in the course by decreasing the time commitment and by allowing the course to be offered at different times. We further surmise that offering the course during the last week of the semester had limited the number of participants in the course.

The most important improvement that is needed for the introduction to containers course is advertisement to counter the low attendance rate of the first course offering. Now that the course has been shown to be successful, researchers should be made aware that container education is an option for them. One way to deliver this advertisement is to subtly (or not-so-subtly) place visible evidence of container support throughout the OOD web interface as the support is expanded. This will help with recognition among the large audience of researchers who have been using the OOD platform exclusively since the success of the basic Primers and Short Courses of 2020.

A major benefit of the containers course is the cross-topic learning gain. Researchers learning how to use containers are by definition exposed to the principles of sharing and portability that led to the creation of the container technology. This helps to drive long-term adoption of FAIR and FEAT practices in research computing. Knowledge of containers in general is relevant for scientific research, especially in the fields making use of AI/ML tools.

4. CONCLUSIONS AND FUTURE DIRECTIONS

Since 2018, Texas A&M HPRC and LMS have successfully used the OOD platform to offer informal instruction at week-long summer schools, online Primers, and our short course program. Here, we reported on our efforts while highlighting a containerization course and Primer that benefitted from expanded capabilities added to the Texas A&M OOD framework. The use of OOD in the course, coupled with enhanced usability features, help increase learner engagement, improve adoption of the technology, and encourage continuous use of the technology in future research. Several of these features are developed and maintained by undergraduate and graduate students in the Texas A&M HPRC student fellows program. While singularity was the chosen containerization technology, the course offered a mechanism for researchers to use a Docker alternative. Recognizing the challenges faced in evaluating virtual courses, this course has been offered in a hybrid face-to-face and virtual setting with the goal of collecting more statistics on researcher usage and learning. We hope to further the learning gains by improving the rich feature set on our local Texas A&M OOD by including access to our knowledge base and ensuring access to a locally hosted container registry.

5. SUPPORTING INFORMATION

All training materials used in this study are available to the community via the Texas A&M HPRC website [2]. Videos and course recordings may be accessed via the Texas A&M HPRC channel on YouTube. The community is invited to join the SWEETER Slack workspace at https://hprc.tamu.edu/sweeter. Surveys and review exercises that will be developed as part of this longitudinal study may be requested from the author. Please send us feedback about your adoption experience via an email to help@hprc.tamu.edu.

6. ACKNOWLEDGEMENTS

This work was supported by the National Science Foundation (NSF) award number 1925764, “CC* Cyberteam SWEETER,” and NSF award number 2019129, “MRI:FASTER”, NSF award number 1730695, “CyberTraining: CIP: CiSE-ProS: Cyberinfrastructure Security Education for Professionals and Students”, NSF award number 2019136, “CC* BRICCs: Building Research Innovation at Community Colleges,” and NSF award number 1829799, “Cybertrainining: CMS3”.

7. REFERENCES

[1] Thomas Kluyver, Benjamin Ragan-Kelley, Fernando Pérez, Brian Granger, Matthias Bussonnier, Jonathan Frederic, Kyle Kelley, Jessica Hamrick, Jason Grout, Sylvain Corlay, Paul Ivanov, Damián Avila, Safia Abdalla, Carol Willing and Jupyter development team. 2016. Jupyter Notebooks — a publishing format for reproducible computational workflows. In *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. Fernando Loizides and Birgit Scmidt (Eds.), 87–90. DOI: https://doi.org/10.3233/978-1-61499-649-1-87

[2] Texas A&M High Performance Research Computing Website. Retrieved from https://hprc.tamu.edu

[3] Dhruva K. Chakravorty, Lisa M. Perez, Honggao Liu, Braden Yosko, Keith Jackson, Dylan Rodriguez, Stuti H. Trivedi, Levi Jordan and Shaina Le. 2021. Exploring Remote Learning Methods for User Training in Research Computing. *Journal of Computational Science Education* 12, 2 (Feb. 2021), 11–17. DOI: https://doi.org/10.22369/issn.2153-4136/12/2/2

[4] Dave Hudak, Doug Johnson, Alan Chalker, Jeremy Nicklas, Eric Franz, Trey Dockendorf and Brian L. McMichael. 2018. Open OnDemand: A web-based client portal for HPC centers. *Journal of Open Source Software* 3, 25 (May 2018), 622. DOI: https://doi.org/10.21105/joss.00622

[5] Dirk Merkel. 2014. Docker: lightweight linux containers for consistent development and deployment. *Linux Journal* 2014, 239 (Mar. 2014), 2. Retrieved from https://dl.acm.org/doi/10.5555/2600239.2600241

[6] Texas A&M HPRC wiki page for Singularity. Retrieved from https://hprc.tamu.edu/wiki/SW:Singularity

[7] Texas A&M HPRC Containers Short Course Website. Retrieved from https://hprc.tamu.edu/training/intro_containers.html

April 2022

Volume 13 Issue 1

ISSN 2153-4136 (online)