Access to Computing Education Using Micro-credentials for Cyberinfrastructure

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

Richard Lawrence
rarensu@tamu.edu
HPRC[1]

Zhenhua He
happidence1@tamu.edu
HPRC[1]

Wesley Brashear
wbrashear@tamu.edu
HPRC[1]

Honggao Liu
honggao@tamu.edu
HPRC[1]

Andrew J. Palughi
ajp2795@tamu.edu
HPRC[1]

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

Xin Yang
karen89@tamu.edu
Medical College of Wisconsin
Milwaukee, WI, USA

Jacob Pavelka
jacobpavelka98@tamu.edu
Texas A&M University
College Station, TX, USA

Ritika Mendjoge
ritika.mendjoge@tamu.edu
HPRC[1]

Marinus Pennings
pennings@tamu.edu
HPRC[1]

Randy McDonald
randymcdonald@tamu.edu
HPRC[2]

Gerry Pedraza
gpedraza@tamu.edu
HPRC[2]

Sunay V. Palsole
sunay.palsole@tamu.edu
HPRC[2]

ABSTRACT

In response to an increasing demand for digital skills in industry and academia, a series of credentialed short courses that cover a variety of topics related to high performance computing were designed and implemented to enable university students and researchers to effectively utilize research computing resources and bridge the gap for users with educational backgrounds that do not include computational training. The courses cover a diverse array of topics, including subjects in programming, cybersecurity, artificial intelligence/machine learning, bioinformatics, and cloud computing. The courses are designed to enable the students to apply the skills they learn to their own research that incorporates use of large-scale computing systems. These courses offer advantages to generic online courses in that they teach computing skills relevant to academic research programs. Finally, the micro-credentials obtained from these courses are transcriptable, may be stacked with existing degree programs and credit-bearing courses to create a larger degree plan, and offer a meaningful mechanism of adding to a student's resume.

KEYWORDS

Micro-credentials, Computing Education, Python, R, Artificial Intelligence, Machine Learning, Bioinformatics, Cybersecurity, Linux

https://doi.org/10.22369/issn.2153-4136/14/2/5

1 INTRODUCTION

Growing federal and state-funded programs have contributed to increase the availability of cyberinfrastructure (CI) resources to researchers at institutions of all sizes [1][2][3]. Job-skills data collected by National Science Foundation CORD (for Texas) and similar studies by other Federal (Department of Energy COVID-19 preparedness report etc.) and private agencies (Deloitte etc.) show an increasing demand in industry and academia for digital skills in the areas of coding, artificial intelligence (AI/ML), bioinformatics, and cybersecurity. Now, more than ever, the efficacy of a research team is limited by their ability to effectively use CI resources. Simultaneously, research workflows integrate interdisciplinary approaches and rely on researchers having expertise in several domains of science. As such, researchers find them simultaneously requiring assistance in (i) learning how to use technologies on CI resources, and (ii) finding support in CI-enabled fields of science that go beyond their local fields of expertise. Micro-credentials offer an interesting opportunity toward alleviating this urgent need to train researchers in computing technologies. They could provide an accelerated introduction for new students and faculty who are interested in computing for research. By design, these courses should be of short durations, ensuring that researchers can readily take them without committing to an entire semester of study, and should teach computing skills relevant for pursuing further academic coursework.

The High Performance Research Computing group has previously developed and offered short courses using a model of continuous improvement. Materials for courses presently in use can

[1]Texas A&M University High Performance Research Computing, College Station, TX
[2]Texas A&M University Engineering Studio for Advances Instruction & Learning, College Station, TX

be dated to early iterations in 2017 [4]. Methods of teaching and delivery have been explored and reported in previous publications [10][7][5][6][9]. It was shown that users are interested in remote learning options and that these can be effective [7]. It was argued that researchers prefer to learn code using interactive graphical interfaces and therefore computing educators should leverage appropriate platforms when teaching computing skills [10]. The platforms that were identified for this purpose were OnDemand, which is a browser interface for HPC systems, and Google Colab, which is a freely available graphical IDE for coding practice.

2 METHODS

Funded by the Texas Higher Education Coordinating Board (THECB) program for accelerated credentials, these credentialed courses serve multiple audiences. Our surveys and interviews of industry employers and academic programs, coupled with participation trends in workforce programs indicate a strong need for a program offering quick stackable credentials in digital skills to graduate students, postdoctoral associates and professionals. Students emphasize that they should have opportunities to develop these skills at their own pace. To ensure that learners with different learning needs are accommodated, the credential courses are offered through both in-person workshops, and online asynchronous options. Each course incorporates a combination of brief lectures with hands-on exercises to optimize student engagement with the material. The course offerings cover a variety of topics, grouped categorically as follows: Coding, Cybersecurity, Artificial Intelligence/Machine Learning, Bioinformatics, and Cloud Computing. Next, our credentialed short courses provide a bridge to computing for senior undergraduate students and junior graduate students in non-traditionally computational programs; e.g. Economics.

For asynchronous courses, the presentation material was generated in Microsoft PowerPoint, Google Colab, and the learnR package. The PowerPoint slides were useful in explaining broad concepts, such as the block structure of control flow in Python programs, and Colab was useful in explaining the mechanics of the broad concepts, such as illustrating the required arguments for a Python package function. Additionally, Colab could be used to facilitate demonstrations and practice assignments for the students. Google Colab gave students access to the Python programming language as well as a Linux kernel for practicing system administration.

As part of our design, we utilized best-practices in curricular design such as building modules to ensure easier adoption in credit bearing courses, matching assignments to learning objectives and outcomes, and adopting principles of interactive lecturing and adult learning principles to ensure student engagement. Each course was developed in a modular manner. Courses were organized into modules and topics. Modules served as distinct groupings of materials and topics were used to break modules into 10-15 minute digestible lectures in which students could take breaks after and assess their previous course in its entirety. The presentation material followed this organizational structure.

Upon preparing the presentation material, recordings were made using the facilities in the Engineering Studio for Advanced Instruction and Learning (eSAIL). An MP4 format was used for these recordings to allow additional data to be associated with them, such as closed captioning. The Descript software suite was used to implement closed captioning and edit the video recordings. This suite included an AI to predict the captioning from the audio file. This accelerated the process of closed captioning significantly.

Multiple-choice quizzes were chosen for these courses' assessment. While not ideal for assessing programming comprehension, this type of quiz allowed for automatic assessment, where a short answer quiz would require manual assessment. Quiz questions covered broad programming concepts, course specific concepts (Machine learning courses required understanding of gradient descent), and programming syntax and usage concepts.

Finally, Canvas was chosen as the delivery medium for the asynchronous courses. Canvas is a learning management system that is commonly used by universities for official coursework. On Canvas, video files and coding assignments can be given to the students together, and quizzes can be given to the students with randomized quiz questions such that students will be less likely to take the same quiz.

The courses, and credentials are delivered with the assistance of Texas A&M Experiment Station Educating Generations (TEES EDGE), a group with a long history of working with industry and State agencies to deliver workforce development training and education at the post baccalaureate, graduate and continuing education levels. The TEES EDGE website provides a full set of services that allows registration, tracking and awarding of credentials to include record maintenance for participants of the program. The courses utilize a variety of different resources to train students. Several courses require students to run various software programs with example data directly on our campus computing clusters. Other courses use platforms such as Google Colab and the learnR package in R to engage students.

Several parameters are identified for each course. These included the CIP CODE, the designated major fields of study (Data Science, CyberSecurity etc.), intended audience (undergraduate students, graduate students, postdoctoral associates, professionals, teachers etc.), the duration of the course (days/ hours / weeks), equivalent professional development units, the number of contact hours that a student will spend with the instructor or learning materials, the mode of delivery (digital, face-to-face, or hybrid), and the possible linking programs. The Classification of Instructional Programs (CIP) code is the federal government classification system that standardizes fields of study across the U. S. The codes in Table 1 whose first two digits are 11 fall under the category of "Computer and Information Sciences and Support Services." The codes beginning with 16 are "Biological and Biomedical Sciences," and the codes under 30 are "Multi/Interdisciplinary Studies." [8]. Additionally, for each course, quizzes and feed-back surveys were developed. To ensure rapid deployment of credential courses, we leveraged existing curricular materials and worked with experienced instructors. Many of the credentialed courses were built on previously offered workshops or other courses [9]. This has allowed us to leverage the feedback we have received from previous students to make improvements and ensure that the credentialed courses are beneficial to the learners. Instructional support and course curricula were delivered by experienced instructors at the High Performance Research Computing (HPRC) group at Texas A&M. HPRC's decade-long informal training programs have taught students computing digital and analytics skills using methods compatible with best practices for computing, so the students who complete our programs will be ready to use computing resources responsibly and effectively. The Engineering Studio for Advanced Instruction & Learning (eSAIL) at Texas A&M led the instructional design of the project. The eSAIL team worked with the subject-matter experts at HPRC to build interactive and flexible online, blended, and technology enabled face-to-face courses. The group has state of the art facilities and includes instructional and learning designers, multimedia specialists, and a learning architect. eSAIL's instructional designers ensure that learning outcomes of the course are measured, and all feedback is folded into a continuous improvement cycle.

3 RESULTS

3.1 Overview of Courses

The coding courses include incremental levels of Python (Fundamentals, Intermediate, and Advanced) and R (Introductory and Intermediate). These courses stack with each other and build a foundation for courses in the other categories. In these courses, the students learn foundational knowledge of programming, such as flow control, data structures, and object-oriented programming. They also learn skills in data science, such as analysis and visualization. In addition to the CI-based courses, we have also created a 3-contact hour asynchronous course titled "Fundamentals of Cybersecurity" that introduces professionals to basic topics in Cybersecurity. This introductory course teaches students the important concepts and terminology of cybersecurity. This covers fundamental cybersecurity principles such as the types of cyberattacks and how to defend against them.

The Artificial Intelligence/Machine Learning (AI/ML) section includes "Fundamentals of AI/ML", which introduces some fundamentals of AI and ML including their relationship, different types of data, training and testing, common types of learning techniques (supervised and unsupervised learning) and applications (regression, classification, and clustering). The short course "Introduction to Deep Learning with TensorFlow" gives a brief introduction to deep learning with TensorFlow (an open-source software library for machine intelligence) and covers basic concepts of deep learning methods. "Introduction to Deep Learning with PyTorch" covers the basic concepts of deep learning and PyTorch with examples. PyTorch is based on the Torch library and can be used for various applications such as computer vision and it provides tensor computing that could be accelerated with GPUs. Finally, the short course "Using SciKit-Learn for AI and ML" introduces some fundamentals of AI and ML and machine learning algorithms including linear regression, logistic regression, Support Vector Machine (SVM), K Nearest Neighbors (KNN), and K-Means clustering with guided practices.

A number of topics are covered in the Bioinformatics section, including "RNA-seq and Differential Expression". In this course, students learn the basic steps that need to be completed for differential expression analyses, including library quality control and trimming, read alignment to a reference genome, generating count files, and differential expression analysis and data visualization in R. Another short course in this section is "Introduction to Metagenomics". This course covers fundamental concepts of conducting metagenomic experiments with next-generation sequencing data, including working with whole genome sequencing data, targeted amplicon sequencing, metagenomic assembly methods, and using the Qiime2 software suite. The course "Introduction to ChIP-seq" teaches students the basic workflow to analyze data generated when combining chromatin-immunoprecipitation with massively parallel sequencing. The Bioinformatics section also includes "Short Variant Discovery", where students use example data to work through a typical short variant discovery pipeline, from library QC and trimming, to mapping reads to a reference genome, and calling variants with the GATK software suite.

The final section on Cloud Computing contains several in-person courses, including "Fundamentals of Linux", "Linux for Administrators", "Job Scheduling SLURM", "Containers and Orchestration", and "Introduction to Cloud and Cluster Computing". The asynchronous offerings include "Parallel Computing Using OpenMP", which introduces students to parallelizing their code using OpenMP and covers topics including OpenMP concepts, OpenMP program layout, work-sharing constructs, synchronization pragmas, and OpenMP tasks. Lastly, the course "Parallel Computing Using MPI" covers the Message Passage Interface (MPI), a standard library to create parallel codes for distributed systems. Topics covered in this course include MPI terminology, Communicators, Point to Point communications, and collective communications.

3.2 Course Details

Learning Objectives by course

- Fundamentals of Cybersecurity
  - Define computer security and approaches to implementing it.
  - Provide general definitions for computer security concepts
  - Describe methods of Social Engineering.
  - Describe Malware and the software vulnerabilities it preys on.
- Fundamentals of Artificial Intelligence and Machine Learning
  - Understand the fundamentals of AI/ML including AI and ML relationship, Training and Testing, Supervised and unsupervised learning, Regression, classification, and clustering.
  - Solve some simple regression, classification, and clustering problems with a machine learning library.
- Introduction to Deep Learning with TensorFlow
  - Understand the fundamentals of deep learning, including what deep learning is and why we need it, its learning principle, convolution, pooling operations and neural networks.
  - Use TensorFlow Keras API to build and train an image classification neural network
- Introduction to Deep Learning with PyTorch
  - Understand the fundamentals of deep learning
  - Use PyTorch framework to build and train an image classification neural network
- Using SciKit-learn for Artificial Intelligence and Machine Learning
  - Understand the fundamentals of AI/ML

Table 1: The credential courses provided by the program

Theme	Course Title	CIP CODE(s)	Duration	Delivery Mode
Cybersecurity	Fundamentals of Cybersecurity	11.1003	0.3 PDU	Asynchronous
Coding	Fundamentals in Python Programming	11.0201	1 PDU	Both
	Intermediate Python Programming for Data Science	11.0202, 30.71	1 PDU	Both
	Advanced Python Programming with Xarray and Dask	11.0202, 30.71	0.5 PDU	Both
	Fundamentals R Programming	11.0201	1 PDU	Both
	Intermediate R Programming	11.0202	1 PDU	Both
	GPU Programming with CUDA	30.3001	0.3 PDU	Live
AI/ML	Fundamentals of Artificial Intelligence and Machine Learning	11.0102	0.3 PDU	Both
	Introduction to Deep Learning with TensorFlow	11.0804, 11.0202	0.3 PDU	Both
	Introduction to Deep Learning with PyTorch	11.0804, 11.0202	0.3 PDU	Both
	Using Scikit-Learn for Artificial Intelligence and Machine Learning	11.0804, 11.0104, 11.0202	0.3 PDU	Both
Bioinformatics	RNA-seq and Differential Expression	11.0104, 11.0401, 26.1103	0.3 PDU	Both
	Short Variant Discovery	11.0104, 11.0401, 26.1103	0.3 PDU	Both
	Introduction to Metagenomics	11.0104, 11.0401, 26.1103	0.3 PDU	Both
	Introduction to ChIP-seq	11.0104, 11.0401, 26.1103	0.3 PDU	Both
Linux	Fundamentals of Linux	11.0201	0.3 PDU	Both
	Linux for Administrators	11.1006, 11.1001	0.3 PDU	Live
Cloud Computing	Job Scheduling with SLURM	11.0103	0.3 PDU	Live
	Containers and Orchestration	11.0103	0.3 PDU	Live
	Introduction to Cloud and Cluster Computing	11.0103	0.3 PDU	Live
	Parallel Computing Using OpenMP	11.0201	0.3 PDU	Asynchronous
	Parallel Computing Using MPI	11.0201	0.3 PDU	Asynchronous

  - Understand some commonly used machine learning algorithms including Linear Regression,
  - Logistic Regression, Support Vector Machine (SVM) and K-Means Clustering.
  - Use Scikit-learn machine learning library to solve regression, classification and clustering problems.
- RNA-seq and Differential Expression
  - Learn about the different techniques used to generate RNA-seq libraries
  - Learn how to properly design a differential expression study
  - Understand the basic steps that need to be completed for differential expression analyses
- Short Variant Discovery
  - Learn how to work through a typical short variant discovery pipeline from library QC and trimming, to mapping reads to a reference genome, and calling variants with the GATK software suite.
- Metagenomics
  - Learn how to use the Qimme2 bioinformatics platform with example data.
- ChIP-seq
  - Learn about next generation sequencing library QC and trimming
  - Align reads to a reference genome
  - Filtering and sorting alignment files
  - Calling ChIP-seq peaks
- Fundamentals of Linux
  - Utilize some commonly used Linux commands for management of files and directories, I/O redirection, customizing environment, and text processing.
- Linux for Administrators
  - Master some Linux administration skills including account management, packages installation, monitoring disk usage, and process control.

Assessment

The course quizzes are in the format of multiple choice questions as shown in Figure 1, which are used to evaluate how well the students understand what they learned and how well they mastered the material.

3.3 Promotion and Initial Reception

The short courses offering micro-credentials were announced to a broad, Texas A&M affiliated audience during a presentation open

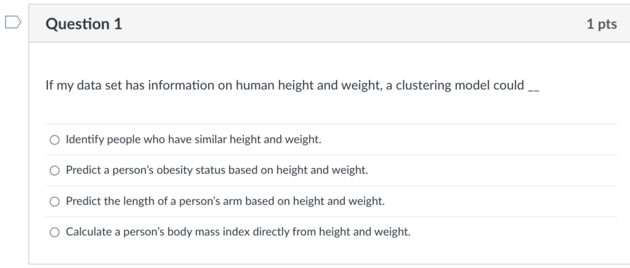

Figure 1: An example of multiple choice quiz questions.

to the university. During the presentation, we discussed the micro-credentialing program, the basic structure of the courses (i.e. modular, stackable design), and detailed the courses that were being offered. The first in-person micro-credentialing course scheduled following this presentation (an 8-week series covering Python programming) resulted in 79 individual registrations before being closed due to physical space constraints. Subsequent requests for this in-person short course were directed to the online asynchronous offering.

4 CONCLUSIONS

The micro-credential-bearing short-courses offer hands-on and project-based learning experiences to students in a choice of live-training and asynchronous educational scenarios. These courses will be developed and iteratively refined in consultation with our academic, workforce, and industry partners to ensure they meet the current and expected needs of academia and the labor market. We ensured rapid deployment of these courses by building on top of successful training models that have been used in the past for other upskilling efforts. The program is offered to students residing in Texas, via a web-based credential system that tracks student participation and offers easy integration for providing upskilling, enabling students state-wide to display their digital prowess, and stacking formal, informal, professional education programs.

The presented credentialed short courses have two major benefits that make them a superior choice to a generic online programming course. First, exercises and topics are chosen that teach skills relevant for scientific research. Second, the credentials provided offer a meaningful reward for completion that can be recognized by academic programs and industry professionals. To promote sustainability, these training materials will transition from informal efforts into curricular products. The collaboration leverages expertise in facets of computational sciences and large scale computing to address a number of long-standing issues encountered in accessing computing resources. From this work, a new, shareable model is provided that increases the accessibility of available resources in areas of multi-disciplinary appeal for researchers. The goal is that CI facilitators at institutions can utilize these resources in training and research workflow support activities.

ACKNOWLEDGMENTS

This work was supported by the Texas Higher Education Coordinating Board (THECB), 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", NSF award number 2019136, "CC BRICCs: Building Research Innovation at Community Colleges", and NSF award number 2112356 "ACES, Accelerating Computing for Emerging Sciences".

REFERENCES

[1] 2020. NSF Major Research Instrumentation Program (MRI). Retrieved August 28, 2023 from https://beta.nsf.gov/funding/opportunities/major-research-instrumentation-program-mri

[2] 2021. NSF Campus Cyberinfrastructure (CC*). Retrieved August 28, 2023 from https://beta.nsf.gov/funding/opportunities/campus-cyberinfrastructure-cc

[3] 2022. Advanced Cyberinfrastructure Coordination Ecosystem: Services and Support (ACCESS). Retrieved August 28, 2023 from https://access-ci.org/

[4] 2023. High Performance Research Computing Past Short Courses. Retrieved August 28, 2023 from https://hprc.tamu.edu/training/previous.html

[5] Dhruva Chakravorty and Minh Tri Pham. 2020. Evaluating the Effectiveness of an Online Learning Platform in Transitioning Users from a High Performance Computing to a Commercial Cloud Computing Environment. *The Journal of Computational Science Education* 11 (Jan. 2020), 93–99. Issue 1. https://doi.org/10.22369/issn.2153-4136/11/1/15

[6] Dhruva K. Chakravorty, Marinus "Maikel" Pennings, Honggao Liu, Zengyu "Sheldon" Wei, Dylan M. Rodriguez, Levi T. Jordan, Donald "Rick" McMullen, Noushin Ghaffari, and Shaina D. Le. 2019. Effectively Extending Computational Training Using Informal Means at Larger Institutions. *The Journal of Computational Science Education* 10 (Jan. 2019), 40–47. Issue 1. https://doi.org/10.22369/issn.2153-4136/10/1/7

[7] 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. *The Journal of Computational Science Education* 12 (Feb. 2021), 11–17. Issue 2. https://doi.org/10.22369/issn.2153-4136/12/2/2

[8] Institute for Educational Sciences. 2020. The Classification of Instructions Programs. Retrieved September 9, 2022 from https://nces.ed.gov/ipeds/cipcode

[9] Richard Lawrence, Zhenhua He, Wesley Brashear, Ridham Patoliya, Honggao Liu, and Dhruva K. Chakravorty. 2022. Tailored Computing Instruction for Economics Majors. *The Journal of Computational Science Education* 13 (April 2022), 32–37. Issue 1. https://doi.org/10.22369/issn.2153-4136/13/1/6

[10] Richard Lawrence, Tri M. Pham, Phi T. Au, Xin Yang, Kyle Hsu, Stuti H. Trivedi, Lisa M. Perez, and Dhruva K. Chakravorty. 2022. Expanding Interactive Computing to Facilitate Informal Instruction in Research Computing. *The Journal of Computational Science Education* 13 (April 2022), 50–54. Issue 1. https://doi.org/10.22369/issn.2153-4136/13/1/9