Introducing HPC-Enabled Computational Research Training in Africa: A Pilot Study

James Afful
Iowa State University

Titus Nyarko Nde
Boise State University

Onyeka Onyenemezu
Iowa State University

Brian Kyanjo
Georgia Institute of Technology

Benjamin Atribawuni Asaaga
Kwame Nkrumah University of Science and Technology

Joseph Oppong Akowuah
Kwame Nkrumah University of Science and Technology

ABSTRACT

Access to computational training and research-computing mentorship remains uneven across global scientific communities. This paper reports a pilot study of the Center for Computational Research and Education (CCRE), a training initiative designed to introduce students and early-career researchers in Africa to computational research through a staged model of open recruitment, structured preparation, mentored project work, and reflective evaluation. The pilot was advertised through LinkedIn, WhatsApp, and Instagram for more than two months, and 24 applicants registered. Twelve applicants were selected for a mandatory four-module Google Classroom preparatory course; six completed all required modules and advanced to the mentored group-project phase. Evaluation used pre- and post-program survey summaries from the final cohort. The pre-program survey had six responses, and the post-program survey had five responses. Baseline results showed strong motivation to build practical high-performance-computing (HPC) skills, connect with a research-computing community, learn collaborative software practices, and explore computational approaches in participants' domains. They also showed substantial structural barriers: most respondents reported no prior HPC access and no or limited formal training in HPC, parallel computing, or research computing. Post-program respondents gave an average overall rating of 4.20 out of 5, and all five indicated that they would definitely recommend the program. Respondents nevertheless continued to report barriers related to HPC access, local mentorship, and computing cost. The CCRE pilot is presented as an educational-practice model whose principal contribution is a documented and critically evaluated pathway for introducing computational research training under constrained infrastructure and mentorship conditions.

KEYWORDS

Computational Training, HPC Education, Africa, Research Computing, Pilot Study, Project-based Learning, Mentorship, Cyberinfrastructure Workforce Development

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page, along with the DOI information below. To copy otherwise, republish, post on servers or redistribute to lists requires prior specific permission from the first author.

© 2026 Journal of Computational Science Education
https://doi.org/10.22369/issn.2153-4136/17/2/3

1 INTRODUCTION

Computational methods are now central to research in engineering, the physical sciences, the life sciences, climate science, public health, data science, and artificial intelligence. Modern scientific work increasingly requires the ability to use command-line environments, version control, remote computing systems, batch schedulers, parallel workflows, data pipelines, and reproducible computational practices. In many research domains, these skills shape the scale of questions that can be asked and the kinds of evidence that can be produced. Access to computational training, however, remains uneven. High-performance-computing (HPC) and research-computing education require more than hardware. Students need structured learning pathways, mentors, examples of domain-specific computational research, access to computing environments, and peer communities that make computational research feel attainable. Existing work on HPC education identifies persistent challenges in curriculum design, educator preparation, student access, and the integration of HPC with contemporary areas such as artificial intelligence and data science [20]. Reviews of scientific-computing, HPC, and data-science education similarly show growing demand for research-computing instruction and the need for more systematic training pathways [19, 24]. These challenges are intensified where local HPC infrastructure, formal courses, and trained instructors are limited.

The African context requires particular attention. A systematic review of computing-education research in Africa found that the region remains underrepresented in the literature and that published work is unevenly distributed across countries and topics, with programming being the most common topic (43% of 68 studies) [22]. At the same time, African and Africa-oriented HPC initiatives have demonstrated that capacity building is possible when infrastructure, training, mentorship, and community formation are treated together rather than separately [1, 14, 15]. Work on sustainable HPC infrastructure in resource-limited settings further emphasizes that durable computational capacity depends on governance, maintenance, training, local ownership, and long-term support, not simply on acquiring computing resources [12].

This paper reports a pilot study of the Center for Computational Research and Education (CCRE), an initiative designed to introduce students and early-career researchers in African contexts to computational research. The program was intentionally small and was not designed to produce large-scale statistical evidence of learning gains. It was designed to test a practical model for introducing computational training, identify barriers faced by learners, and generate lessons for future cohorts.

The central research and practice questions were:

(1) What kinds of learners were reached by an open, social-media-based recruitment strategy for an Africa-oriented computational training program?
(2) What baseline barriers to HPC and research-computing participation were reported by the final pilot cohort?
(3) How did participants evaluate the program after completing the mentored-project phase?
(4) Which barriers appeared to remain after the pilot?
(5) What design lessons can be drawn for future computational-training programs in African contexts?

The primary contribution of this paper is the documentation and preliminary evaluation of an implementable educational-practice model for introducing African students and early-career researchers to computational research.

Specifically, the study contributes: (i) a staged training pathway connecting open recruitment, application review, structured preparation, and mentored computational projects; (ii) descriptive evidence concerning participant motivations, access barriers, satisfaction, and perceived instructional value; and (iii) practical implementation lessons for future computational training initiatives operating under infrastructure, connectivity, and mentorship constraints.

The study does not claim that the structural barriers identified are newly discovered, nor does it establish that the training intervention produced measurable technical proficiency gains. Rather, it provides an early feasibility and educational-practice account that can inform the design and systematic evaluation of larger future cohorts.

2 RELATED WORK

2.1 HPC and Research-Computing Education

HPC education lies at the intersection of computer science, domain science, systems knowledge, and practical workflow training. Raj et al. describe HPC education as a field facing curricular, institutional, and pedagogical challenges, including the need to define competencies, support educators, and connect HPC learning to artificial intelligence, data science, and contemporary computing education [20]. Ponce et al. review scientific-computing, HPC, and data-science training programs and argue that demand for advanced research-computing instruction has grown substantially, with learners increasingly seeking data-oriented and interdisciplinary computational skills [19]. Van Zon et al. similarly document growth and diversification in scientific-computing training delivered through a supercomputing center, including participation from nontraditional computing disciplines [24].

These studies support the premise that computational training should not be restricted to computer-science majors. Scientists and engineers increasingly need practical research-computing fluency. Reviews of HPC foundations for scientists emphasize that domain researchers often require practical HPC knowledge even when their formal training is outside computer science [13]. The CCRE pilot follows this broader direction by targeting learners across engineering, the physical sciences, computational biology, data-oriented fields, and related areas.

2.2 Cyberinfrastructure Workforce Development and Modular Training

Research-computing education is also part of cyberinfrastructure workforce development. The National Science Foundation's cyberinfrastructure learning and workforce-development efforts frame training as essential to broad use of advanced computational resources across the research community [17]. Landscape studies of research computing emphasize that access varies widely across institutions and that less-resourced communities need support to participate in advanced computational work [5]. Microcredential and modular training approaches have been proposed as one mechanism for widening access to computing education, especially for learners who need targeted competencies rather than complete degree programs [6].

The CCRE pilot used a mandatory four-module Google Classroom sequence before the mentored-project phase. In this pilot, the module sequence functioned as both a readiness-building mechanism and a participation filter. This design is consistent with broader movements toward modular, accessible, and scaffolded computational training. Unlike formal microcredential programs, however, the pilot did not issue standardized credentials or conduct externally validated skill assessments. That distinction is important when interpreting the results.

2.3 African Computing Education and HPC Capacity Building

The literature on computing education in Africa is growing but remains uneven. Sanusi and Deriba systematically review general computing education research, rather than HPC-specific instruction, and describe an uneven geographical and topical distribution, including substantial emphasis on programming [22]. Their review is therefore used here to characterize the broader education-research landscape, not as direct evidence of HPC training outcomes or as proof that research-computing education is absent. More directly relevant African examples include HPC ecosystem-development and student-cluster training initiatives [14, 15]. Recent regional work also examines practical ICT-training design and delivery barriers in universities in South Africa, Cameroon, and Nigeria [4]. Although that work is not HPC-specific, it provides relevant evidence about instructional implementation in resource-constrained environments.

Africa-oriented HPC capacity-building efforts provide important precedents. Abiona et al. discuss the ICTP experience in HPC infrastructure setup and training in Africa, emphasizing that capacity building involves developing people and instructor communities, not only deploying machines [1]. The HPC Ecosystems Project provides another model by combining infrastructure deployment, formal training workshops, and community development across multiple countries [15]. The South African Student Cluster Competition further demonstrates how selection, training, mentorship, and authentic technical challenges can support high-level student participation in HPC [14].

These efforts indicate that African HPC capacity building should be understood as an ecosystem problem. CCRE contributes to this literature by documenting a small, project-based, Africa-oriented

pilot that began with open recruitment and proceeded through structured preparation into mentored computational project work.

Recent African research-training initiatives provide additional points of comparison for the CCRE model. BioStruct-Africa describes a community-driven framework integrating computational structural biology, AlphaFold-enabled research, sustained mentorship, and long-term workforce development [2]. Its proposed training targets represent program ambitions rather than independently demonstrated educational outcomes.

A capacity-strengthening initiative involving six East African Community partner states similarly describes practical technical instruction, including introductory HPC exposure, together with training-of-trainers and follow-up support [18].

Although these initiatives address different scientific domains and institutional settings, they illustrate complementary approaches to computational capacity building. Their emphasis on practical research engagement, local instructor development, and sustained institutional support reinforces the need to connect short-term training with longer-term research capacity.

The CCRE pilot adds an implementation-level account of an open-recruitment, staged-preparation, and mentored-project pathway. Its contribution is therefore the documentation of a potentially adaptable educational practice rather than evidence of superior learning outcomes or large-scale program effectiveness.

2.4 Project-Based Learning, Active Learning, and Mentorship

The CCRE pilot included foundational HPC training sessions and a final phase centered on mentored group projects. This design aligns with evidence that active learning improves outcomes in science, technology, engineering, and mathematics compared with lecture-only approaches [11]. Project-based and hands-on computational training are especially important in HPC education because many key competencies are procedural and contextual: learners must understand concepts and learn how to work in real computational environments.

Mentorship is also central. Studies of STEM mentorship show that high-quality mentoring and research experiences can support science identity, efficacy, values, and integration into scientific communities, especially for students from underrepresented groups [3, 9, 21]. For CCRE, mentorship should therefore be interpreted not only as technical troubleshooting but also as a mechanism for supporting research identity, belonging, and persistence in computational research.

2.5 Training Evaluation and Pilot-Study Reporting

Training programs are often evaluated using models that distinguish satisfaction from learning, behavior, and longer-term results. The Kirkpatrick model, for example, separates evaluation into reaction, learning, behavior, and results [16]. This distinction is important for the present pilot. The available CCRE data support claims about reaction, perceived value, baseline barriers, and intended future use.

Pilot studies are best used to examine feasibility, acceptability, process, implementation, and measurement procedures before larger-scale evaluation [8, 23]. The CCRE pilot should therefore be interpreted as an early program evaluation and educational-practice report. Its value lies in documenting the model, identifying barriers, and refining future implementation rather than proving definitive effectiveness.

3 PROGRAM CONTEXT

The Center for Computational Research and Education was developed to provide structured exposure to computational research, HPC, artificial-intelligence-enabled workflows, and collaborative scientific computing. The pilot focused on learners motivated to engage with computational research but who often lacked local access to HPC systems, formal training, mentors, or peers working in the area.

Recruitment relied on open advertisement through LinkedIn, WhatsApp, and Instagram over more than two months. Social-media recruitment was intentional because it allowed the program to reach motivated learners beyond a single classroom or institution. The application review record documented 24 applicants.

The program aimed to select a smaller group for deeper preparation and mentorship. Selection considered motivation, application fit, programming experience, prior HPC exposure, teamwork experience, availability, academic stage, gender representation, and institutional or external balance. Twelve applicants were selected for the mandatory Google Classroom preparatory stage. Six completed all four required modules and advanced to the final pilot cohort. Figure 1 summarizes the pathway.

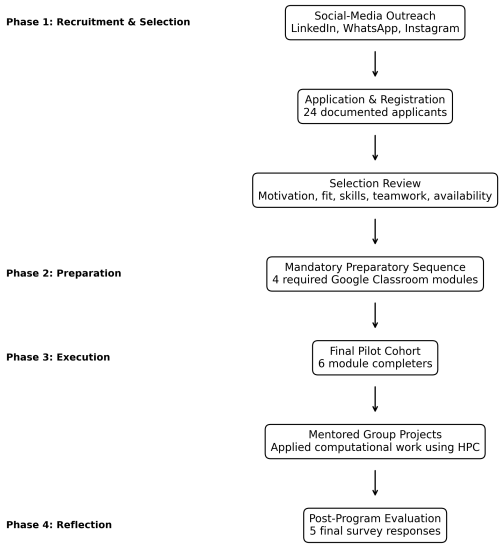


Figure 1: Staged pathway used in the CCRE pilot.

4 PROGRAM DESIGN

The CCRE pilot was organized around four design principles:

(1) **Access:** widen entry into computational research by recruiting beyond traditional local classroom channels.
(2) **Preparation:** provide a structured preparatory sequence before intensive project work.
(3) **Mentored practice:** move participants from exposure to application through group projects and mentor support.
(4) **Community and identity:** help participants see themselves as part of a research-computing community.

These principles are consistent with literature on HPC education, cyberinfrastructure workforce development, active learning, and mentorship [5, 9, 11, 19, 20]. Table 1 maps program components to intended outcomes and available evidence. This mapping connects the evaluation items to program objectives rather than treating the post-program instrument as a generic satisfaction survey.

5 EVALUATION METHODS

The pilot evaluation used pre- and post-program survey summaries. The pre-program survey was completed by six respondents corresponding to the final cohort that advanced into the pilot program. The post-program survey was completed by five respondents. Survey items addressed academic background, disciplinary background, prior HPC access, prior training, motivations, barriers, confidence in computational skills, research identity, satisfaction with program components, perceived impact, internet and connectivity constraints, intended future use of skills, and willingness to recommend the program.

The evaluation was descriptive. Because the sample was small and the available summaries did not confirm paired individual-level responses, inferential statistical tests were not conducted. Counts and percentages are reported where the survey summaries provide sufficient information. Results are interpreted as evidence about feasibility, acceptability, perceived value, and remaining barriers rather than definitive evidence of technical mastery.

The analysis drew on three program documents: a pre-program survey summary with six responses, a post-program survey summary with five responses, and an internal application-evaluation document describing review criteria and potential participants. The internal document contained personally identifiable applicant information. This paper therefore reports only aggregate or procedural information. Applicant names, personal motivations, and identifiable background details are not reported.

The evaluation drew on four complementary sources of evidence, summarized below.

- **Pre-program survey ($n = 6$):** The survey documented participants' academic and disciplinary backgrounds, prior access to HPC resources, previous formal training, motivations for participation, and perceived barriers to computational research. Self-reported technical confidence covered Linux command-line navigation, remote HPC login, batch-job scripting, job scheduling, parallel or distributed computing, and Git version control. Research identity and participants' expectations were also assessed.
- **Post-program survey ($n = 5$):** The survey examined overall satisfaction, perceived usefulness of training components, internet connectivity challenges, technical confidence, research identity, intended future use of computational skills, remaining barriers, overall program rating, and willingness to recommend the program. These responses reflect participants' perceptions and intentions rather than independently verified learning outcomes.
- **Application-evaluation records:** Internal program documentation recorded application-review criteria, participant selection, and progression through the preparatory phase. These records supported description of the recruitment and completion pathway but did not establish the reasons individual applicants failed to complete the required modules.
- **Mentored group-project descriptions:** The program documented two interdisciplinary computational projects: a machine-learning surrogate for reactor thermal-hydraulic analysis and a brain-tumor MRI classification workflow. These descriptions illustrate the nature of project-based computational activities but do not provide independently audited evidence of individual technical proficiency.

The evaluation was descriptive. Because the final cohort comprised only six participants, with five post-program respondents, the findings should not be generalized to the broader population of African students or early-career researchers. Furthermore, individual-level pre- and post-program survey responses were not confirmed as paired. Consequently, differences between the two survey summaries cannot be interpreted as measured within-person learning gains. No inferential statistical tests or causal-effect estimates were performed.

6 RESULTS

6.1 Recruitment and Participation Funnel

The program advertised through LinkedIn, WhatsApp, and Instagram over more than two months. The application-review record documented 24 applicants. Twelve were selected for the Google Classroom preparatory stage, and six completed all four required modules and formed the final pilot cohort.

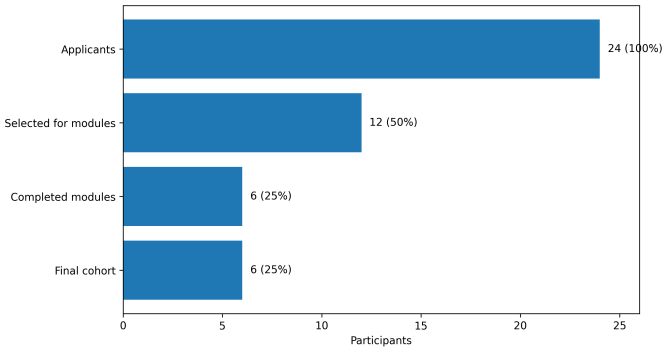


Figure 2: Recruitment, preparation, and participation funnel. Percentages are relative to the 24 documented applicants.

The program generated broader interest than it could intensively support, and progression into the final cohort required completion of preparatory work. As shown in Figure 2, 24 applicants were recorded, 12 were selected for the mandatory preparatory stage,

Table 1: CCRE training components, intended outcomes, and available evidence.

Component	Intended outcome	Available evidence
Social-media recruitment	Reach motivated learners beyond a single institution or formal course	Recruitment record; pre-survey source-of-awareness item
Application review	Select applicants with strong motivation, availability, and readiness for project-based training	Internal selection document
Google Classroom modules	Establish baseline preparation and identify participants able to complete required asynchronous work	Module completion; final cohort size
Foundational technical sessions	Build confidence in Linux, remote computing, job scheduling, scripting, and version control	Pre- and post-program confidence items
Mentored group projects	Move from passive exposure to applied computational work	Post-survey impact item; reactor-safety Streamlit tool; MRI classification and Grad-CAM tool
Office hours and troubleshooting	Reduce technical barriers and support persistence	Post-survey satisfaction item
Guest research sessions	Expose participants to computational-research pathways and community	Post-survey satisfaction item
Final presentation and closeout	Develop research communication and accountability	Post-survey satisfaction item; project presentations

and six completed all four modules and advanced to the mentored-project phase. Thus, one quarter of documented applicants entered the final pilot cohort.

The available records do not establish why some selected applicants did not complete the required modules. Future cohorts should track module-level completion, time burden, connectivity constraints, and reasons for withdrawal to distinguish barriers to participation from other causes of attrition.

6.2 Application Review and Baseline Characteristics

Table 2 summarizes the dimensions considered during the application-review process. The internal evaluation document indicates that review criteria included motivation, application fit, programming experience, prior HPC exposure, teamwork experience, availability, gender representation, and academic or career stage. These criteria highlight a design tension for future computational capacity-building programs. Intensive mentorship requires selecting participants with the readiness and availability to persist, whereas the broader capacity-building objective also requires expanding opportunities for learners with limited prior exposure to HPC.

Table 3 summarizes academic level, disciplinary background, prior HPC access, and prior formal training among the six pre-program respondents.

The final cohort was interdisciplinary and mostly lacked prior access to HPC systems. Five of six respondents reported no previous access to an HPC cluster or supercomputing resource, while one reported access through an external program or grant. Three reported no formal training, two reported partial exposure, and one reported an online course or workshop. No respondent reported completing a full university course in HPC, parallel computing, or research computing.

Participants reported strong motivation across technical, community, and professional dimensions. Table 4 summarizes their selected goals.

These motivations show that participants sought technical instruction, community, collaborative software practice, and pathways into computational research. This supports the decision to design CCRE as a mentored cohort rather than a one-time workshop. Figure 3 summarizes the barriers reported by respondents before and after the program. The pre-program survey showed substantial barriers to HPC learning. All six respondents reported a lack of local instructors or mentors with HPC expertise. Five reported lack of access to HPC systems and the cost of cloud or commercial computing resources. Four reported a lack of a structured curriculum and a lack of nearby peers working in the area.

These findings are consistent with literature emphasizing that computational capacity requires training, infrastructure, mentorship, and communities of practice [1, 5, 12]. Because the pre-program survey included six respondents and the post-program survey included five respondents, the two distributions should be interpreted as separate descriptive summaries rather than paired measurements of individual change.

6.3 Post-Program Satisfaction and Perceived Value

The post-program survey received five responses. As summarized in Table 5, respondents gave the program an average overall rating of 4.20 out of 5. Four respondents assigned a rating of 4, while one assigned a rating of 5. All five respondents indicated that they would definitely recommend the program to a colleague or peer.

These findings suggest favorable participant reactions within the small responding cohort. However, satisfaction and recommendation responses should not be interpreted as objective evidence of technical skill acquisition or long-term research capacity development.

These results support a claim of high acceptability and perceived value but do not establish technical mastery. In the Kirkpatrick framework, the data primarily represent reaction and perceived usefulness, not independently measured learning, behavior change, or long-term institutional impact [16]. Table 6 presents respondents' assessments of the program components that contributed most strongly to their perceived learning experience.

Two of the five respondents selected working on the team project using real HPC resources as the component with the greatest impact. Foundational technical sessions, peer collaboration and team

Table 2: Application review criteria used in the CCRE pilot.

Criterion	Role in selection
Motivation and fit	Primary criterion for identifying applicants aligned with computational-research training
Programming experience	Readiness indicator for technical participation
Prior HPC exposure	Background indicator; lack of exposure did not automatically exclude applicants
Teamwork experience	Indicator of readiness for mentored group projects
Availability	Feasibility criterion for participation in a time-bound pilot
Gender representation	Broadening-participation consideration during review
Academic or career stage	Tie-breaker, with attention to undergraduate and master's-level applicants
Institutional or external balance	Consideration of institution-linked and external applicants

Table 3: Baseline characteristics of pre-program survey respondents ($n = 6$).

Domain	Category	Count
Academic level	Undergraduate student	3
	Master's student	1
	PhD student	0
	Early-career researcher or postdoctoral scholar	2
	Other	0
Disciplinary background	Engineering	3
	Physical sciences	1
	Computational biology or bioinformatics	1
	Other	1
	Computer science or software engineering	0
	Mathematics or statistics	0
	Data science, artificial intelligence, or machine learning	0
Prior HPC access	No prior access	5
	External program or grant	1
	University-provided access	0
	Cloud-platform access	0
Prior formal training	No formal training	3
	Partial exposure	2
	Online course or workshop	1
	Full university course	0

Table 4: Goals for joining the CCRE pilot ($n = 6$).

Goal	Count
Build practical HPC skills for research	6
Connect with a research-computing community	6
Learn collaborative software practices	6
Explore computational approaches in the participant's domain	6
Strengthen a CV or graduate-school application	5
Other	0

Table 5: Post-program satisfaction and recommendation results ($n = 5$).

Measure	Result
Post-program survey respondents	5
Average overall program rating	4.20/5
Respondents giving a rating of 4	4
Respondents giving a rating of 5	1
Respondents who would definitely recommend the program	5/5

dynamics, and mentor feedback and guidance were each selected by one respondent. Guest-researcher talks were not selected as the single most impactful component by any respondent. These responses provide preliminary insight into which activities participants valued most, but the very small sample prevents robust comparisons among instructional components.

This result supports one of the pilot's central design claims: computational training should move beyond passive exposure into mentored practice. The finding aligns with active-learning evidence in STEM education and HPC-education literature emphasizing hands-on practice [7, 10, 11].

Internet connectivity remained a relevant participation constraint. As shown in Table 7, three of the five post-program respondents reported moderate internet issues that caused them to

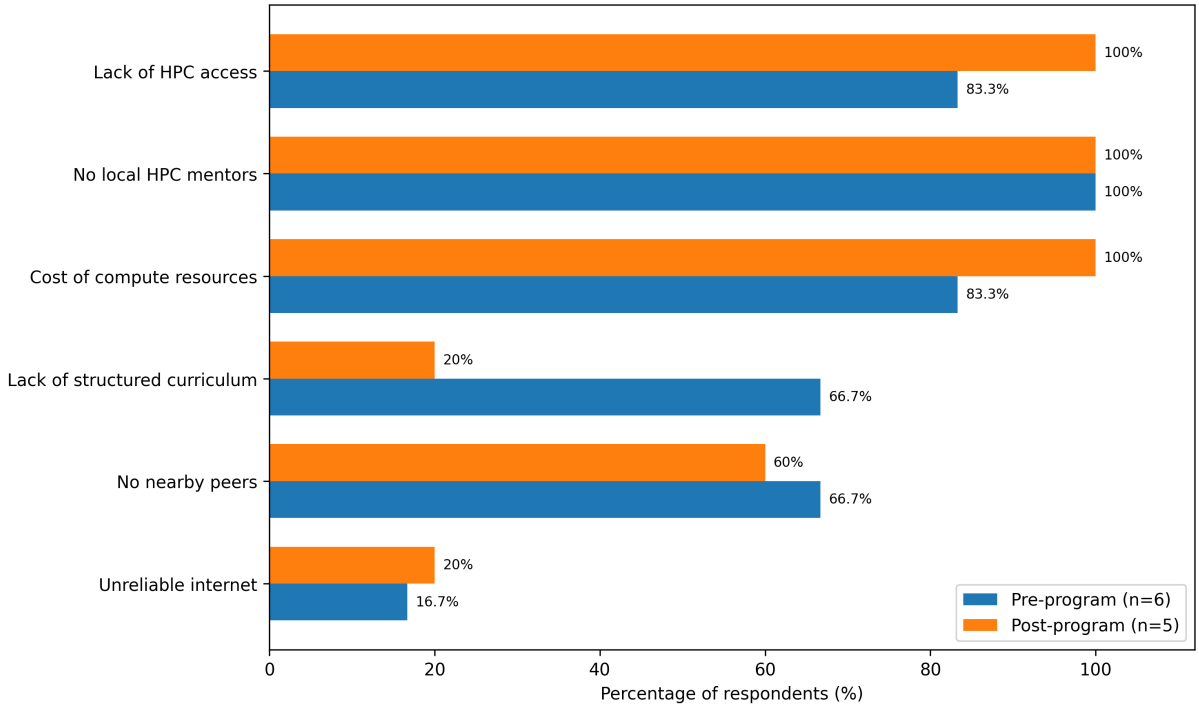


Figure 3: Reported barriers to HPC learning in the pre- and post-program surveys. Because the surveys had different denominators and responses were not confirmed as paired, the values should be interpreted descriptively rather than as individual-level change.

Table 6: Program aspect reported as having the greatest impact on learning ($n = 5$).

Aspect	Count
Team project using real HPC resources	2
Foundational technical sessions	1
Peer collaboration and team dynamics	1
Mentor feedback and guidance	1
Guest-researcher talks	0

Table 7: Reported effect of internet connectivity on participation ($n = 5$).

Connectivity effect	Count
Moderate issues; missed some sessions or content	3
Minor issues; participation not significantly affected	1
No issues	1
Significant issues; participation substantially limited	0

Table 8: Intended future use of CCRE skills ($n = 5$).

Intended use	Count
Include the experience in graduate-school applications	5
Apply to future HPC programs or internships	5
Apply HPC methods to current research or coursework	4
Share knowledge with peers or teach others	4
Contribute to HPC capacity at the participant's institution	2
Pursue a role in research computing or data science	1

miss some sessions or content. One respondent reported minor connectivity issues without substantial participation effects, while one reported no connectivity problems. No respondent reported severe connectivity difficulties that substantially prevented participation.

This connectivity-impact measure is distinct from the general barriers checklist reported in Figure 3. The two questions therefore capture related but different aspects of participants' experiences.

Connectivity did not prevent completion among the post-survey respondents, but it affected participation for a majority. Future cohorts should therefore provide asynchronous backups, downloadable materials, low-bandwidth options, and structured catch-up mechanisms.

Table 8 summarizes how respondents expected to use what they learned.

These results indicate strong intended transfer, but they remain statements of intent. A stronger future evaluation should track whether participants subsequently use HPC methods in coursework or research, apply to programs, mentor peers, contribute to local capacity, or produce computational-research outputs.

Table 9 compares pre- and post-program barrier reports. Because the pre-survey had six respondents and the post-survey had five,

both counts and percentages are shown. Because responses were not confirmed as paired, differences should not be interpreted as measured individual-level change.

The proportion reporting a lack of structured curriculum was lower in the post-program summary, while access, mentorship, and cost remained highly prevalent. These descriptive comparisons suggest that the program may have helped address the need for a structured learning path, but the data do not establish causality or individual-level change. The findings clarify the scale of the problem: training programs can create exposure and structure, but durable computational capacity requires sustained infrastructure access, local mentor development, and affordable computing pathways.

6.4 Applied Project Outcomes

The six participants were divided into two groups, each tasked with executing an end-to-end computational project using HPC resources and machine-learning workflows.

The first group developed a machine-learning surrogate model for steady-state thermal-hydraulic safety analysis of a 10 MW Material Testing Reactor. Using an HPC-generated database of 1,000 samples from the PLTEMP/ANL v4.3 simulation code, the team evaluated Random Forest, Gradient Boosting, and feedforward neural-network models. The group identified Gradient Boosting as the most accurate of the evaluated surrogate approaches and deployed the model through an interactive Streamlit graphical interface intended to support rapid safety assessments without rerunning the computationally expensive source simulations.

The second group developed an automated brain-tumour classification tool using magnetic-resonance images. The group used two-phase transfer learning to fine-tune EfficientNetB3, VGG16, and DenseNet121 convolutional-neural-network architectures. The resulting ensemble was deployed through a web application with Grad-CAM heatmaps that visualized image regions influencing the model's classifications.

These group artifacts show that, with mentor support, the cohort participated in end-to-end computational work extending from foundational workflow preparation to model training, comparison, and web deployment. The evidence does not independently establish each participant's technical mastery or individual contribution.

7 DISCUSSION

7.1 A Staged Model for Computational Training

The CCRE pilot provides preliminary evidence of the feasibility and acceptability of a staged model for introducing computational training in African contexts. The model begins with open outreach, proceeds through application review, uses mandatory asynchronous preparation to establish baseline readiness, and advances participants who complete the preparation into mentored group projects. This structure differs from a one-time workshop.

The recruitment funnel may reflect time constraints, module difficulty, internet access, competing obligations, or other factors, but the available data do not identify the reasons. Future programs should collect process data on module engagement, missed sessions, reasons for noncompletion, and participant time burden.

The post-program survey suggests that applied project work was an important component of the experience. Working on a team project using real HPC resources was the most frequently selected high-impact component, although it was chosen by only two of the five respondents. This finding is consistent with evidence that active learning can improve STEM outcomes and that computational skills are best learned through practice [11]. In HPC education, learners need to interact with computing environments, scripts, errors, queues, dependencies, filesystems, and collaboration tools that cannot be learned fully through lectures alone.

Future cohorts should preserve and strengthen the group-project phase. The completed artifacts show that the cohort, with mentor support, participated in domain-specific computational work involving a thermal-hydraulic surrogate model with Streamlit and an MRI-classification ensemble with Grad-CAM visualization. Future evaluations should assess such artifacts through a structured rubric and should distinguish team-level products from individual learning outcomes.

The survey summaries also show that mentorship is not peripheral. All six pre-program respondents reported a lack of local instructors or mentors with HPC expertise, and all five post-program respondents identified the same barrier. This finding is consistent with mentorship literature showing that mentoring can support research identity, belonging, and integration into scientific communities [3, 9, 21].

For CCRE, mentorship should be treated as infrastructure. A program that depends on a small number of expert mentors will be difficult to scale. Future cohorts could use a tiered mentorship model in which alumni serve as peer mentors for subsequent cohorts. Over time, this could help transform CCRE from a training program into a distributed community of practice.

The most important structural finding is that barriers related to HPC access, local mentorship, and computing cost remained highly prevalent after the program. The pilot created perceived value while revealing limits that a short training intervention cannot resolve. Sustainable HPC capacity in resource-limited settings depends on access, maintenance, governance, training, and local ownership [12]. Future CCRE work should therefore combine training with partnerships for compute access, institutional support, and mentor development.

7.2 A Practical Framework for Future Cohorts

Based on the pilot results, four pillars are proposed for future computational-training efforts in African contexts:

(1) **Access:** provide learners with exposure to real computational environments and workflows.
(2) **Preparation:** use structured modules to establish baseline readiness before intensive project work.
(3) **Mentored practice:** organize learners into projects with technical and research mentorship.
(4) **Sustainability:** build alumni mentorship, institutional partnerships, and longer-term compute access.

This framework is grounded in both the pilot data and literature on HPC education, cyberinfrastructure workforce development, active learning, and mentorship [5, 9, 11, 20].

Table 9: Reported barriers before and after the CCRE pilot. Percentages use the relevant survey denominator.

Barrier	Pre count	Pre %	Post count	Post %
Lack of access to HPC systems	5/6	83.3%	5/5	100.0%
No local instructors or mentors with HPC expertise	6/6	100.0%	5/5	100.0%
Cost of cloud or commercial computing resources	5/6	83.3%	5/5	100.0%
Lack of a structured curriculum or learning path	4/6	66.7%	1/5	20.0%
No peers working in this area nearby	4/6	66.7%	3/5	60.0%
Unreliable internet connection	1/6	16.7%	1/5	20.0%
No significant barriers	0/6	0.0%	0/5	0.0%

8 LIMITATIONS

This pilot has several important limitations.

First, the sample size was small: the pre-program survey had six responses, and the post-program survey had five. Results should therefore be interpreted descriptively rather than inferentially.

Second, the available survey summaries do not confirm paired individual-level responses. Without anonymous identifiers linking pre- and post-program responses, individual-level change cannot be measured reliably.

Third, the evaluation relies heavily on self-report. Participants reported satisfaction, perceived value, confidence, and intended future use, but the pilot did not include an independent assessment of technical skill.

Fourth, the survey summaries do not provide raw item-level data for all Likert-scale constructs. Exact means and changes for each confidence or identity item should be calculated only from verified raw survey exports.

Fifth, attrition occurred between registration, selection, module completion, and final participation. The current data document the funnel but do not explain why selected participants did or did not complete the preparatory modules.

Sixth, the final cohort was selected rather than randomly sampled. This was appropriate for a pilot program but limits generalizability.

Seventh, long-term outcomes are unknown. The post-program survey captures intended future use, not whether participants subsequently used HPC in research, applied to programs, published work, mentored peers, or contributed to institutional capacity.

Eighth, the small cohort creates privacy risks. Reporting should avoid identifiable combinations of country, institution, academic stage, field, and personal motivation unless explicit consent permits their disclosure.

9 RECOMMENDATIONS FOR FUTURE IMPLEMENTATION

The pilot suggests five concrete improvements:

(1) **Use anonymous participant identifiers.** Link future pre- and post-program surveys through anonymous identifiers so that individual-level change can be analyzed without compromising privacy.
(2) **Track module analytics.** Record module starts, completion, quiz scores, time to completion, and reasons for noncompletion where possible.
(3) **Provide low-bandwidth alternatives.** Record sessions, provide downloadable slides and materials, and design assignments that can be completed under intermittent connectivity.
(4) **Develop alumni pathways.** Invite participants who complete the program to mentor future cohorts, lead focused sessions, or maintain project repositories.
(5) **Secure institutional and compute partnerships.** Pursue partnerships with universities, national laboratories, cloud-credit programs, HPC centers, and research networks.

10 CONCLUSION

This paper presented a pilot study of the Center for Computational Research and Education, an initiative designed to introduce students and early-career researchers in African contexts to computational research, HPC, artificial-intelligence-enabled workflows, and collaborative scientific computing. The pilot used a staged model consisting of open social-media recruitment, application review, mandatory Google Classroom preparation, selective progression into a final cohort, mentored group projects, and post-program evaluation.

Final-cohort respondents were highly motivated but had limited prior HPC access and formal training. Post-program respondents rated the program highly and unanimously indicated that they would recommend it. The team-project component using real HPC resources was the most frequently selected high-impact aspect. At the same time, respondents continued to report barriers related to HPC access, local mentorship, and cost.

The main lesson is that computational training in African contexts should be designed as an ecosystem rather than a one-time workshop. Training can provide structure, exposure, perceived value, and community, but durable capacity requires sustained computing access, mentor development, institutional partnerships, and pathways for alumni to support future learners. The CCRE pilot is therefore best understood as a practical and critically evaluated model for developing a future phase of computational-research training.

DATA AVAILABILITY

The analysis is based on program-level survey summaries and an internal application-evaluation document. Because the underlying materials may contain personally identifiable information, raw survey and application materials are not publicly released. De-identified aggregate data used in the tables can be made available upon reasonable request, subject to ethical and privacy constraints.

ACKNOWLEDGMENTS

The authors thank the participants of the CCRE pilot program for their engagement, persistence, and feedback. The authors acknowledge the mentors, speakers, and collaborators who supported the program. The authors thank Doreen Anyeley Afful for serving as the administrative lead. The authors also thank the Boise State University HPC team for providing computing resources and office-hour support used by the trainee groups. The authors acknowledge the use of the open-source Brain Tumor MRI Dataset by Masoud Nickparvar, which supported the biomedical-classification project.

CONFLICT OF INTEREST STATEMENT

The authors declare no competing interests.

REFERENCES

[1] Olatunde Abiona, Clement Onime, Stefano Cozzini, and Sebsibe Hailemariam. 2011. Capacity Building for HPC Infrastructure Setup in Africa: The ICTP Experience. In *2011 IST-Africa Conference Proceedings*. IEEE, 1–8. https://doi.org/10.1109/ISTAFRICA.2011.6107383

[2] Justice O. Amofa, Juergen George, Noella A. Okumu, et al. 2026. BioStruct-Africa's scalable framework for AlphaFold-enabled research training and sustainable workforce development in Africa. *Communications Biology* 9 (2026), 321. https://doi.org/10.1038/s42003-026-09711-x

[3] Kaitlyn Atkins, Bryan M. Dougan, Michelle S. Dromgold-Sermen, Hannah Potter, Viji Sathy, and A. T. Panter. 2020. "Looking at Myself in the Future": How Mentoring Shapes Scientific Identity for STEM Students from Underrepresented Groups. *International Journal of STEM Education* 7, 1 (2020), 42. https://doi.org/10.1186/s40594-020-00242-3

[4] Maria Assunta Cappelli and Abdeljalil Akkari. 2025. Academic ICTs training in South Africa, Cameroon, and Nigeria. Strategies for ICT training course design. *Education and Information Technologies* 30 (2025), 10463–10510. https://doi.org/10.1007/s10639-024-13200-1

[5] Dhruva K. Chakravorty, Sarah K. Janes, James V. Howell, Lisa M. Perez, Amy Schultz, Marie Goldie, Austin L. Gamble, Rajiv Malkan, Honggao Liu, Daniel Mireles, Yuanqi Jing, Zhenhua He, and Tim Cockerill. 2022. Expanding the Reach of Research Computing: A Landscape Study: Pathways Bringing Research Computing to Smaller Universities and Community Colleges. In *Practice and Experience in Advanced Research Computing (PEARC '22)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3491418.3535169

[6] Dhruva K. Chakravorty, Richard Lawrence, Zhenhua He, Wesley Brashear, Honggao Liu, Andrew J. Palughi, Lisa M. Perez, Xin Yang, Jacob Pavelka, Ritika Mendjoge, Marinus Pennings, Randy McDonald, Gerry Pedraza, and Sunay V. Palsole. 2023. Access to Computing Education Using Micro-credentials for Cyberinfrastructure. *The Journal of Computational Science Education* 14, 2 (2023), 23–27. https://doi.org/10.22369/issn.2153-4136/14/2/5

[7] Paweł Czarnul, Mariusz Matuszek, and Adam Krzywaniak. 2024. Teaching High-Performance Computing Systems: A Case Study with Parallel Programming APIs: MPI, OpenMP and CUDA. In *Computational Science – ICCS 2024*, Leonardo Franco, Clélia de Mulatier, Maciej Paszynski, Valeria V. Krzhizhanovskaya, Jack J. Dongarra, and Peter M. A. Sloot (Eds.). Springer Nature Switzerland, Cham, 398–412. https://doi.org/10.1007/978-3-031-63783-4_29

[8] Sandra M. Eldridge, Claire L. Chan, Michael J. Campbell, Christine M. Bond, Sally Hopewell, Lehana Thabane, and Gillian A. Lancaster. 2016. CONSORT 2010 Statement: Extension to Randomised Pilot and Feasibility Trials. *BMJ* 355 (2016), i5239. https://doi.org/10.1136/bmj.i5239

[9] 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. https://doi.org/10.1187/cbe.17-04-0066

[10] Kristen Finch, Ryan Beck, Xiaosong Li, Nam Pho, and Xiao Zhu. 2025. Computational Skills Training for Undergraduate Researchers in Molecular Engineering. *The Journal of Computational Science Education* 16, 1 (2025), 50–56. https://doi.org/10.22369/issn.2153-4136/16/1/10

[11] Scott Freeman, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active Learning Increases Student Performance in Science, Engineering, and Mathematics. *Proceedings of the National Academy of Sciences* 111, 23 (2014), 8410–8415. https://doi.org/10.1073/pnas.1319030111

[12] Ronald Galiwango, Christopher J. Whalen, Grace Kebirungi, Mugume T. Atwine, Rodgers Kimera, Alfred Ssekagiri, Timothy W. Kimbowa, Edward Lukyamuzi, Mike Nsubuga, and eLwazi ODSP. 2025. Ten Simple Rules for Building and Maintaining Sustainable High-Performance Computing Infrastructure for Research in Resource-Limited Settings. *PLOS Computational Biology* 21, 9 (2025), e1013481. https://doi.org/10.1371/journal.pcbi.1013481

[13] Pablo García-Risueño and Pablo E. Ibáñez. 2012. A Review of High Performance Computing Foundations for Scientists. arXiv:cs.DC/1205.5177

[14] Bryan Johnston, Nick Thorne, Matthew Cawood, Eugene de Beste, David Macleod, and John Poole. 2026. A Retrospective on South Africa's Student Cluster Competition and Its Model for Inclusive HPC Outreach and Training (2012–2020). *The Journal of Computational Science Education* 17, 1 (2026), 42–49. https://doi.org/10.22369/issn.2153-4136/17/1/6

[15] Bryan Johnston, Lara Timm, David MacLeod, and John Poole. 2024. Ten Years of the HPC Ecosystems Project: Transforming HPC in Africa for the Past Decade. In *Practice and Experience in Advanced Research Computing (PEARC '24)*. Association for Computing Machinery, New York, NY, USA, Article 35, 8 pages. https://doi.org/10.1145/3626203.3670537

[16] Donald L. Kirkpatrick. 1994. *Evaluating Training Programs: The Four Levels*. Berrett-Koehler Publishers, San Francisco, CA.

[17] National Science Foundation. 2017. *Blueprint for Cyberinfrastructure Learning and Workforce Development*. Technical Report. National Science Foundation. Cyberinfrastructure Learning and Workforce Development Blueprint.

[18] Julien A. Nguinkal, Florian Gehre, Hakim I. Lagu, Emmanuel Achol, Eric Nzeyimana, John N. Kiiru, Joseph Nyandwi, Gregory W. Dumo, Nyambura Moremi, Susan N. Nabadda, Isabelle Mukagatare, Andrea Molina, Jürgen May, and Muna Affara. 2025. A capacity strengthening model toward self-reliant and sustainable one-health workforce in six East African Community Partner States. *Frontiers in Public Health* 13 (2025), 1636817. https://doi.org/10.3389/fpubh.2025.1636817

[19] Marcelo Ponce, Erik Spence, Ramses van Zon, and Daniel Gruner. 2019. Scientific Computing, High-Performance Computing and Data Science in Higher Education. *The Journal of Computational Science Education* 10, 1 (2019), 24–31. https://doi.org/10.22369/issn.2153-4136/10/1/5

[20] Rajendra K. Raj, Carol J. Romanowski, John Impagliazzo, Sherif G. Aly, Brett A. Becker, Juan Chen, Sheikh Ghafoor, Nasser Giacaman, Steven I. Gordon, Cruz Izu, Shahram Rahimi, Michael P. Robson, and Neena Thota. 2020. High Performance Computing Education: Current Challenges and Future Directions. In *Proceedings of the Working Group Reports on Innovation and Technology in Computer Science Education (ITiCSE-WGR '20)*. Association for Computing Machinery, New York, NY, USA, 51–74. https://doi.org/10.1145/3437800.3439203

[21] Rachael D. Robnett, Peter A. Nelson, Eileen L. Zurbriggen, Faye J. Crosby, and Martin M. Chemers. 2018. Research Mentoring and Scientist Identity: Insights from Undergraduates and Their Mentors. *International Journal of STEM Education* 5, 1 (2018), 41. https://doi.org/10.1186/s40594-018-0139-y

[22] Ismaila Temitayo Sanusi and Fitsum Gizachew Deriba. 2024. What Do We Know About Computing Education in Africa? A Systematic Review of Computing Education Research Literature. *SSRN Electronic Journal* (2024). https://doi.org/10.2139/ssrn.4826205 Also available as arXiv:2406.11849.

[23] Lehana Thabane, Jinhui Ma, Rong Chu, Ji Cheng, Afisi Ismaila, Lorena P. Rios, Reid Robson, Marroon Thabane, Lora Giangregorio, and Charles H. Goldsmith. 2010. A Tutorial on Pilot Studies: The What, Why and How. *BMC Medical Research Methodology* 10 (2010), 1. https://doi.org/10.1186/1471-2288-10-1

[24] Ramses van Zon, Marcelo Ponce, Erik Spence, and Daniel Gruner. 2019. Trends in Demand, Growth, and Breadth in Scientific Computing Training Delivered by a High-Performance Computing Center. *The Journal of Computational Science Education* 10, 1 (2019), 53–60. https://doi.org/10.22369/issn.2153-4136/10/1/9