Project Description

1 Background, Motivation, and Goals

The overarching goal of this CI Users (CIU) interdisciplinary project is to develop an innovative, scalable, and personalizable Advanced Cyberinfrastructure (AC) training for engineering and computer science freshman college students using Data Science as the medium. This project named AdCyL – Advanced Cyberinfrastructure Learning – provides training in foundational knowledge, methods, and skills required for effectively exploiting AC capabilities in solving Computational and Data-Enabled Science and Engineering (CDS&E) problems in the biomedical domain. We expect this AC training to inspire engineers and computer scientists to engage in biomedical research as well as contribute new AC methods and tools.

The proposed AC training will be offered in the form of a freshman-level course, which will be suitable for all engineering and computer science majors. This course will be part of the First Year Program (FYP) repertoire of courses in the College of Engineering and Technology (CET) at East Carolina University (ECU). The course delivery mechanism will enable anytime and anywhere learning in both formal and informal settings. In the subsequent three years and beyond, we expect the students to apply AC methods and tools to solving CDS&E problems in general and biomedical problems in particular.

1.1 Big Data and Data Science

Recent advances in storage technologies, high-performance computing, and pervasive sensing are driving the production of unprecedented volumes of data [1]. Big data is often characterized by five Vs: volume, velocity, variety, veracity, and value. Most big data is *unstructured* and requires *information extraction* methods to glean information. Furthermore, big data volume and velocity render conventional computational approaches to its processing infeasible. Big data processing requires cluster-based, distributed system architectures and AC methods and tools [2].

Dhar [3] defines Data Science as *the systematic study of the extraction of generalizable knowledge from data.* Data Science provides a framework for processing, modeling, analyzing, visualizing, interpreting, and reasoning about data. Data Science workflows typically involve the following processes: data sensing; data filtering and sampling; data collection, cleansing, processing, and integration; data storage and retrieval; modeling and analytics; and visualization, interpretation, and communication.

Big Data is essential for Data Science and vice versa. Theory and experimentation are viewed as the two traditional pillars of the scientific method [4]. Theoretical science creates new ideas and models, and scientists experimentally validate these ideas by collecting data and comparing it to the models. According to the U.S. Presidential Information Technology Advisory Committee's 2005 report titled *Computational Science: Ensuring America's Competitiveness*, computational science is considered the third pillar of the scientific method. Computational Science enables building and testing models of complex phenomenon such as nuclear weapon simulations, and formation of galaxies. The advent of Big Data ushered in the fourth pillar or paradigm of science – Data Science [5, 6]. Data Science approaches offer a new paradigm for solving ill-posed and difficult problems — managing the complexity of the problem domain by building simple but high quality models by harnessing the power of data.

Given the sheer volume of data and its heterogeneity, specially configured cluster computing systems are required to provide *performance at scale* for Data Science applications [7]. Knowledge and skills needed to analyze and interpret terabyte or larger datasets are quite different from those that are needed for small-scale datasets. This is where AC meets Data Science. Big Data and Data

Science are the two sides of the same coin with AC as the underlying foundation. Cyberinfrastructure (CI) refers to high-performance computing (HPC) environments which encompass sensors and instruments for data acquisition, devices for data storage, and software for a range of activities – data filtering, sampling, compression, cleaning and integration, search and retrieval, pattern discovery, and visualization. AC refers to CI environments that use extreme massive parallel computation and distributed computing to provide *performance at scale* on terabyte/petabyte-scale Big Data.

1.2 Project Goals

Given the advances in Big Data, AC, and emerging opportunities, the overarching goal of this **interdisciplinary project**, termed AdCyL, is to provide AC training to first-year college students in engineering and computer science disciplines, using the Data Science as an attractive medium. Specific goals of AdCyL are: **Goal 1:** Design and develop an innovative, scalable, and personalizable AC-enabled Data Science course; **Goal 2:** Integrate innovative and inclusive pedagogy into course teaching and learning content to achieve broader impact; **Goal 3:** Foster interdisciplinary collaborative research in data-science and AC; and **Goal 4:** Significantly increase the participation of underrepresented groups in both Data Science/AC research and workforce. The project team will also perform formative and summative assessment of project work products and their impact. Lastly, the project team will disseminate widely the project work products and maximize their usage and impact across the country.

2 Intellectual Merit

The AdCyL project uses Darwin Information Typing Architecture (DITA) framework to develop teaching and learning contents that are reusable, personalizable, scalable, sustainable, and accessible from anywhere using a range of computing devices. Our approach enables a single-source, multi-channel delivery model, which eliminates redundancy and inconsistency. It also minimizes the effort required to keep the training materials current and relevant. Instructors and learners can assemble training materials quickly to suit different purposes without the need for programming. The biomedical domain chosen for the project offers numerous opportunities to apply AC methods and tools, and Data Science approaches for analyzing, model building, and visualizing biomedical text and image Big Data. In addition to data volume and velocity, the domain offers data heterogeneity and data integration challenges, which in turn help to illustrate the critical role of AC and Data Science in the biomedical domain.

3 Broader Impacts

The training materials will be released in formats suitable for multiple deployment options. First, they can be deployed on cloud infrastructure such as Amazon Web Services. Second, the materials will also be available as *docker* containers for easy desktop installation. This will contribute to increasing the number of computer scientists and engineers trained in AC for computational problem solving. Furthermore, the project will **provide AC training to a substantial number of students from underrepresented groups**. We are currently promoting Computer Science (CS) education in three local high schools in Greenville, North Carolina through a Google igniteCS grant. We will also train CS teachers from these and other high schools in the region so that they can incorporate select AdCyL materials into their courses. Furthermore, we will leverage our ongoing collaboration with Pitt Community College, and Pitt County Early College High School and train their CS instructors as well. We will extend this training to other community colleges and early college high schools in the region. ECU is developing two new degree programs – B.S. in Software Engineering and M.S. in Data Science. These programs are expected to debut in Fall 2018. These efforts are mutually complementary to the AdCyL project.

4 Advisory Board

The AdCyL project Advisory Board is comprised of experts from different areas of engineering, and computer science. The advisory board will provide strategic guidance to the project team to help them achieve the project's goals. This board is different from the existing advisory board for the ECU Computer Science department's academic programs. The advisory board members will meet with the project team twice a year. The board members may travel to East Carolina University (ECU) or participate through a teleconference. The board will be briefed about the last six-month progress of the project. The board members will then provide feedback and suggest any corrective actions. The board members are: (1) Dr. Wendy C. Newstetter, Assistant Dean for Educational Research and Innovation, Georgia Institute of Technology, Atlanta, Georgia. (2) Dr. Srini Ramaswamy, Software Technology Manager, ABB Inc., Cleveland/Akron, Ohio area (3) Dr. David Chen, High-Performance Computing solution architect, Raleigh-Durham, North Carolina area, (4) Kurt Schmidt, Business Development Manager and GPU-based Deep Learning expert, Raleigh-Durham, North Carolina area, and (5) Prof. Harry Ploehn, Dean of College of Engineering and Technology, East Carolina University, Greenville, North Carolina.

5 Advanced Cyberinfrastructure Environment

A high-performance computational platform is integral to the AdCyL project. As discussed in the *Facilities, Equipment, and Other Resources* document, we have a cluster-based computing environment to host AC and Data Science tools. We also have two standalone servers – IBM S822LC power server and Nvidia DGX-1 station – which are specifically built for AI and deep learning applications. We will use the Anaconda distribution [8] as the software environment. The Anaconda distribution is open-source and has over six million users worldwide. It features both R and Python environments with over 1,000 curated Data Science packages. Anaconda distribution can interface with diverse data sources from flat files to SQL and NoSQL databases, distributed SQL engines, and cloud-based storage. It also integrates with the two widely used high-performance environments – Apache Hadoop and Spark.

The National Center for Biotechnology Information (NCBI)/ U.S. National Library of Medicine maintains a PubMed article database. PubMed is comprised of more than 28 million citations for biomedical literature from MEDLINE, life science journals, and online books. The citations may include links to full-text content from the PubMed Central and publishers' websites. PubMed Central (PMC) is a free, full-text archive of biomedical and life sciences journal literature at the U.S. National Institutes of Health's National Library of Medicine (NIH/NLM). Over 4.7 million articles are archived in PMC. We will download PMC/PubMed articles and craete a *biomedical text corpus* for the AdCyL project. We will also create a *biomedical image corpus*, and associated clinical data and tools using the following resources:

- The National Health and Nutrition Examination Survey (NHANES) is a major program of the Centers for Disease Control and Prevention [9], which assesses the health status of adults and children in the United States. This unique survey combines medical and dental assessments, laboratory tests, dietary questions, and demographics including socioeconomic factors. Survey results have been used to relate dietary habits to health outcomes, health disparities in dental care in children, relationship between TV viewing and physical activity and diet, and risk factors for periodontitis. Over 40,000 articles have been published utilizing NHANES data.
- The Alzheimer's Disease Neuroimaging Initiative (ADNI) dataset is created to help reserchers define the progression of Alzheimer's disease [10]. This dataset is comprised of MRI and PET images, genetics, cognitive tests, CSF, and blood biomarkers.

- The Parkinson's Progression Markers Initiative data [11]. This dataset includes anonymized clinical information and assessment results, imaging data (DaTSCAN Imaging, Structural MRI Imaging, and Diffusion Tensor Imaging), and results from the analysis of clinical laboratory evaluations and DNA sampling.
- Autism Brain Imaging Data Exchange (ABIDE) is a project whose goal is to help diagnose Autism Spectrum Disorder (ASD) in children at earlier ages [12]. Given the complexity and heterogeneity of ASD, large-scale datasets are essential to reveal the brain mechanisms that underlie ASD. ABIDE aggregates functional and structural brain imaging data that is being collected from 24 brain imaging laboratories around the world.
- Insight Segmentation and Registration Toolkit (ITK) is an open-source, cross-platform system which provides an extensive suite of software tools for biomedical image analysis. ITK features leading-edge algorithms for registering and segmenting multidimensional image data.
- The Insight Journal is an open-access on-line publication dedicated to medical image processing and visualization [13]. In addition to articles, open-access is also provided for data, code, and reviews to enable reproducible science via automated code compilation and testing.
- The Visualization Toolkit (VTK) is a cross-platform, open-source system for 3D computer graphics, image processing, and visualization [14]. VTK supports a broad range of visualization algorithms including scalar, vector, tensor, texture, and volumetric methods. VTK also supports advanced modeling techniques – implicit modeling, polygon reduction, mesh smoothing, cutting, contouring, and Delaunay triangulation. VTK supports parallel/GPU computing and integrates with several databases.
- ParaView is a scalable, open-source visualization component of VTK. It runs on a range of computing systems from laptops to supercomputers. VeloView runs on top of ParaView and is used to visualize live or captured 3D LiDAR data from Velodyne's HDL line of sensors.
- 3D Slicer is a VTK-enabled, open-source and extensible application for analyzing and visualizing biomedical images.
- A historical airline flights database of flight arrival and departure details for all commercial flights within the USA, from October 1987 to April 2008 [15]. There are a total of 120 million records in the dataset. Information about each flight includes scheduled departure time, actual departure time, scheduled arrival time, actual arrival time, departure airport code, arrival airport code, flight duration, and day of the week.

6 Accomplishing AdCyL Goals: Specific Actions

Goals 1 and 2: Developing a scalable and personalizable cyberinfrastructure training and integrating innovative and inclusive pedagogy Knowledge discovery, machine learning, data analytics, and AC are the technology enablers of Data Science [16, 17]. Data Science and AC are becoming increasingly intertwined due to the volume and heterogeneity of data. The computing industry is aggressively leading, setting trends and future research direction for Data Science. Computer Science (CS) departments are rather slow in responding to industry workforce needs. However, there is consensus in academia that universities need to integrate Data Science into computing courses and curricula [18]. The current efforts in this direction include targeting non-majors using flipped classrooms [19, 20], gamified scaffolding [21], datathons [22], exploratory visualization [23], and infusing statistics into introductory CS courses [24]. The AdCyL project complements these efforts and **addresses new challenges posed by an interdisciplinary audience, reusability and scalability requirements for creating both short- and long-term broader impacts, and evolving the course to maintain relevancy and currency**.

Six challenges. Developing a first-year course for beginning engineering and computer science

students poses several challenges. First, a programming-centric approach should be avoided, yet provide a strong foundation in data science fundamentals and impart skills to use advanced cyberinfrastructure. Second, projects should be selected in a way to ensure that they are representative of the engineering and computer science disciplines as well as focused on biomedical domain. Third, design the projects with varying levels of challenge so that the level of difficulty matches the varying abilities of the students. Educational research has shown that students have different preferred learning styles [25] and developing content to meet this need is another challenge. The next challenge is how to accomplish personalization for both students and teachers? Achieving reusability and evolution of content to ensure wide adoption for broader impacts and long-term sustainability is the last challenge.

Challenge 1: A non-programming-centric approach to cyberinfrastructure training. It is widely acknowledged that learning programming is a difficult task for many students [26, 27]. We do not go into the details about why programming is complex, but note that, like reading and writing, learning programming requires prolonged periods of practice. Expecting CS freshman to acquire mastery of programming in just one or two courses is like asking mathematics freshman to prove theorems [28]. The recent study done at Cali Poly State University (San Luis Obispo) to discover factors that affect student retention in CS is insightful as well as counter-intuitive [29]. Students do not become expert programmers just by completing one or two courses. It is a slowly developing skill that needs to be nurtured across the entire curriculum. However, many students do not get an opportunity to develop programming skills incrementally as the introductory courses are positioned as *gatekeepers* and many programs associate this with academic rigor.

Guo defines *research programming* as an activity where the objective of programming is to obtain insights from data [30]. He also notes that millions of professionals from diverse fields are engaged in research programming on a daily basis. There is a disconnect between the way programming is taught in classrooms and the way it is practiced in industry [31]. In a recent survey done by Chilana et al. [32], of the 3,151 survey responses from professionals who never or rarely write code, 42.6% invested in learning programming on the job. Though the respondents primarily perform end-user computing tasks such as data analysis, their motivation for learning programming was to enhance the efficacy of technical conversations and to acquire marketable skills. This study coined the term *conversational programmers* to refer to this scenario.

Given this backdrop, we posit that future computer scientists and engineers will perform data scientist functions using more sophisticated tools integrated through Data Science workflow systems [33]. A very small fraction of them will be engaged in the activities of a typical professional software engineer [34]. This is akin to the number of compiler writers, debugging tool developers, and assembly language programmers today compared to their counterparts two decades ago. Furthermore, recently there have been several successful efforts in teaching introductory CS without heavy focus on programming [35–38]. The AdCyL project will use an incremental, contextualized, Data Science-driven approach rooted in computational problem solving to impart AC training [39–41].

We differentiate between three types of teaching and learning content: core principles, case studies, and projects. **Core principles** are the underlying concepts and methods of AC and Data Science. They encompass algorithms; elementary probability and statistics; matrices; data exploration and preparation; data representation, cleaning, and transformation; formulation of research questions, model development and validating, and visualization and communication of results.

Case studies are complete solutions to problems which demonstrate incremental and iterative solution development, and industry best practices. We will select case studies from engineering/biomedical, and CS domains. **Projects** are real-world problems that students will work on in small teams. The case studies facilitate problem-based learning and serve as exemplars. Students

will model solutions to projects by studying the relevant exemplars.

Challenge 2: Case studies and projects should be representative of engineering/ biomedical and CS domains and are engaging to students with learning diversity.

The AdCyL will feature *case studies* related to the following topics: (i) nutritional differences between rural and non-rural regions that predict cardiovascular health using NHANES data [9], (ii) biomedical image compression [42], (iii) automatic speech recognition [43, 44], (iv) automatic colorization of black and white images [45, 46], (v) automatic machine translation of biomedical text between natural languages [47], (vi) biomedical text modeling and automatic text generation [48], (vii) socioeconomic analysis and health disparities [9, 49], (viii) visualization of biomedical images [50], (ix) diagnosing Carpal Tunnel Syndrome [51], (x) approximating the value of π [37], (xi) analysis of customer loyalty [52], and (xii) determining the latitude and longitude of a location based on address and Google Maps Geocoding API and plotting it on a map of US [53]. We will develop 4 case studies per year and a total of 12 case studies during the project funding period.

The AdCyL will feature *projects* related to the following topics: (1) practical cryptography and ciphers [54, 55], (2) biomedical multimodal information fusion, (3) biomedical image enlargement through spatial filtering [56], (4) Zipf's law and biomedical text processing [57], (5) phoneme classification [58], (6) predicting airline delays [59], (7) spelling correction for the biomedical domain [60, 61], (8) research articles recommendation [62, 63], (9) predicting heart attacks [64], (10) tissue sample classification into cancer classes [64], and (11) predicting disease using data from smart watches [65]. We will develop 4 projects per year and a total of 12 projects during the project funding period. In Section 7, we provide design and development details for four of the above case studies and projects.

Challenge 3: Projects span a spectrum in terms of the level of knowledge and skills required for solving them. This feature guarantees that there are enough projects in the repository to engage students with varying levels of ability and motivation. Our solution to this challenge is shown in Figure 1. The top layer in the figure represents an opaque and complete solution to a project or case study. From students' perspective, this top-level solution unit solves the problem. It is a black box. Given a correctly encoded input, this unit produces a solution. It enables students to experiment with different inputs and observe the outputs. In some cases, this enables students to gain a conceptual understanding of the solution at an end-user level.

The middle layer in the figure depicts multiple solution blocks. They have *coarse granularity* in the sense that the level of abstraction manifested is smaller than the top level solution unit. A certain level of knowledge and skill is required to understand how the middle layer solution units collaboratively solve the problem. More specifically, students need to understand how to compose a solution by stringing together middle layer units. The solution units in the bottom layer feature fine granularity. Multiple units in the bottom layer corresponds to a single unit in the middle layer. Relative to the middle layer units, the level of knowledge and skill required to design and develop bottom layer units is much greater. This three-layer architecture provides a conceptually elegant framework to both understand solutions as well as design and develop solutions. Depending on the academic preparation, motivation, and maturity levels of the students, an instructor may discuss solutions to projects and case studies at an appropriate solutions level. Likewise, an instructor may provide appropriate scaffolding to students. For example, an instructor may provide all or a subset of bottom layer solution blocks and require the students to compose a solution. Likewise, an instructor may choose to provide middle layer solution blocks. In summary, the solution blocks architecture shown in Figure 1 enables engaging diverse student groups whose knowledge and skills vary.

Challenge 4: Training content encompasses needs of students with different learning styles. The need for different types of teaching and learning content to effectively engage academically diverse students and improve learning outcomes is informed by learning research [25]. Ambrose et al. introduce seven principles of learning by drawing on research in cognitive, developmental, and social psychology; educational research; anthropology; demographics; and organizational behavior [66]. Research also suggests that the practice of retrieving and reconstructing knowledge is more effective than elaborate studying with concept mapping [67]. To quote Herbert Simon (a Nobel Laureate and Cognitive Scientist), $\cdots$ *Learning results from what the student does and thinks and only from what the student does and thinks. The teacher can advance learning only by influencing what the student does to learn.* Our approach to developing teaching and learning content will be guided by these principles — goal-directed practice, component skills acquisition and integration, application of acquired knowledge to solving authentic problems. We will design and develop multiple versions of a body of knowledge or skill to meet the learning needs of academically diverse students [68, 69].

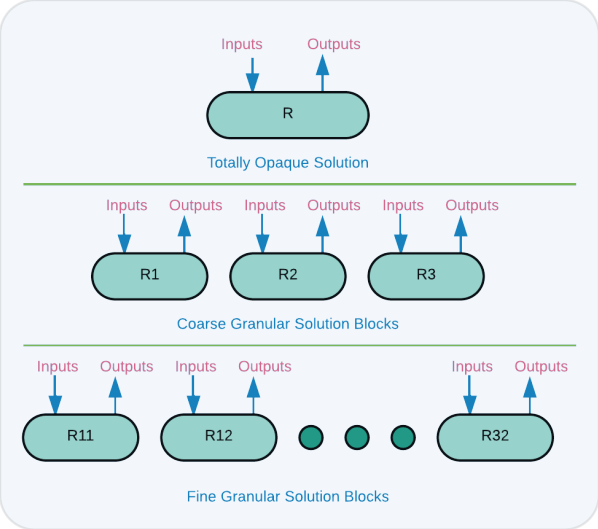

Figure 1: Hierarchical decomposition of solution blocks

Process Oriented Guided Inquiry Learning (POGIL) is an approach to learning based on guided inquiry [70] — a learning cycle of exploration, concept invention and application. CS-POGIL is an adaptation and further development of POGIL for CS education, supported by a 2011 NSF TUES program grant [71]. IntroCS POGIL is another 2017 NSF funded project, headed by Clif Kussmaul of Muhlenberg College. It builds on CS-POGIL and investigates factors that most influence faculty to adopt POGIL in introductory computer science courses, and how the degree of POGIL implementation impacts student learning and engagement. We will incorporate CS-POGIL principles in designing the training content.

Challenge 5: Training content is personalizable by both instructors and students. We accomplish personalization as well as innovative and inclusive pedagogy through the DITA-based approach [72]. Personalization enables students to engage in learning by sequencing topics in ways that reflect their preferences and just-in-time learning needs, subject to prerequisite dependencies among the topics. In other words, students may learn the contents of a course by sequencing the topics differently to suit their own needs.

Darwin Information Typing Architecture (DITA) is an XML-based OASIS open standard for structuring, developing, managing, and publishing content [73]. It supports single-source, versioned content for multi-lingual and multi-channel delivery. It leverages traceability and accountability features to enable evolving and maintaining content using an approach known as *edit it once, edit it everywhere*. Separation of content from target utilization and delivery format, content reuse, semantic markup, extensibility through specialization and inheritance, and topic-based writing are the defining characteristics of DITA. DITA is widely used in industry. For example, *IBM uses DITA as a key part of its 60-million-page Knowledge Center*, which is comprised of 20 million DITA topics [74].

DITA integrates semantic markup with topic-based authoring – writing small, self-contained, reusable, and context-free content units. These independent units are assembled into documents via external structural specifications called **topic maps**. A map primarily contains a hierarchy of topic references. A map is executed to produce content that can target multiple delivery channels such as HTML, PDF, ePub, and Microsoft Word. This process is depicted in Figure 2.

We will write DITA content as *well-formed* and *validating* XML documents. The *topic* element is the most generic type for topic-based authoring. The following specialized types are available to achieve more granular semantic markup – *concept*, *task*, *reference*, and *glossentry*. **Concept** topics provide the prerequisite conceptual and theoretical foundation necessary to accomplish tasks. An example of a concept topic is *DNS server*. Concept topics are stable and rarely change [75]. **Task** topics are used to enumerate step-by-step instructions for accomplishing tasks such as assessing security vulnerability of a database server. **Reference** topics, as the name implies, capture reference information which is factual in nature. For example, the methods of the Math Java class is a reference topic. Lastly, **glossentry** topics are used to describe glossary entries. The DITA Learning and Training Content Specialization component provides additional support for developing educational content. It includes specific topic types for learning overviews, content, summaries, assessments and plans. Furthermore, support for basic quiz and assessment items is included.

Figure 2: DITA topic-based authoring

The DITA topic-based authoring will result in thousands of XML files. A database management system as well as a Web application are required to create and administer training content via DITA maps. A system architecture for the content delivery is shown in Figure 3, which is referred to as AdCyL delivery system (AdCyL-DS). A critical component of this architecture is a native XML database to store and retrieve DITA topics and maps. There are a few open-source native XML databases such as eXist, Sedna, Tamino, and Virtuoso. We have chosen BaseX due to its light-weight, high-performance, and scalability [76]. It features XPath/XQuery 3.1 processor and an interactive GUI front-end. DITA Open Toolkit (OT) is a vendor-independent, open-source implementation of the DITA standard, which is a publishing engine for DITA maps [77]. Out of the box, DITA OT supports output in formats such as HTML, troff, and PDF. Moreover, the OT enables customization of output including new formats through its extensible plug-in mechanism.

We will implement AdCyL-DS as a Web application using HTML5, CSS3, and JavaScript to enable students and instructors to personalize teaching and learning via DITA maps. The Web application will feature access control and authorization function to enable instructors to administer AdSyL-DS for their classes. Advanced users can also access and create DITA topics and maps via BaseX GUI as well as XQuery and XSLT query languages.

Challenge 6: Reusability, evolution, and sustainability through a community of practice. Reusability and incremental evolution is intrinsic to the DITA infrastructure. Topic-based authoring is ideally suited for team collaboration and work decomposition. By the end of the third year, we expect to create a critical mass of users of the AdCyL teaching and learning content. Once the users see the effectiveness and versatility of the content, it is in their own interest

to evolve the project. We expect some of the users to transition from being only consumers to both producers and consumers. Success of this approach is closely tied to the dissemination efforts discussed in Section 10.

Goal 3: Promoting Interdisciplinary Collaborative Data Science and AC Research. East Carolina University will have both undergraduate and graduate students working on the project along with PI/Co-PIs. Effective and timely communication is essential for the project success. Coordination of the project will rest with the PI to ensure course delivery and research success.

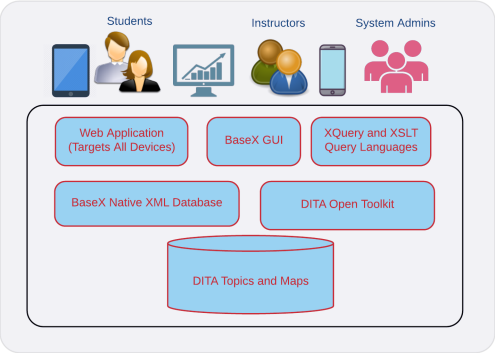

Figure 3: AdSyL delivery system (AdCyL-DS)

Bi-monthly project meetings will be held to discuss project progress and to resolve any roadblocks. This forum will also be used to discuss AC and Data Science research and how to incorporate recent research into the proposed course. Emphasis will be placed on promoting student-peer interactions. The effectiveness of this interdisciplinary research will be measured in terms of student-authored research publications, student presentations at relevant conferences, and joint research proposal submissions to external funding agencies.

Onboarding new students to the project will involve general orientation and online tutorials on how to use DITA-aware tools for developing personalizble course content. It will also include watching pre-recorded lectures on AC literature search, reading and critiquing research literature, bibliography management with cloud-hosted services such as BibSonomy.org, and cloud-hosted collaborative authoring platforms such as ShareLaTeX.com. GitHub will be used for source code version control.

To further enhance the interdisciplinary collaboration and achieve broader impacts, two one-day workshops will be conducted annually during summer months. All personnel involved with the project will make presentations which reflect their role in the project, accomplishments, issues that need resolution, and plans for the upcoming year. This is also an opportunity for the project personnel to collectively reflect on the project and determine any changes to the project direction. These workshops will also feature a training session on how to adapt and personalize AdCyL materials. These sessions are primarily meant for CS teachers/instructors in local/regional high schools, early college high schools, and community colleges.

Goal 4: Cyberinfrastructure Workforce Development with Focus on Underrepresented Groups. An important goal of the AdCyL project is capacity building for cyberinfrastructure workforce development and to substantially increase the participation of underrepresented minorities in CA jobs and research careers. The project will draw students from engineering and computer science disciplines. Other activities discussed under Goal 3 will also contribute to increasing the participation of African American and Hispanic students in cyberinfrastructure and data science workforce, and research engagement in these two areas. The DITA-based implementation of training materials allows both instructors and students to personalize learning, which in turn will help broader adoption and greater impact on workforce development. We will leverage our existing partnerships and collaborations with minority serving institutions in North Carolina – North Carolina Central University, University of North Carolina at Fayetteville, North Carolina A&T State University, and Winston-Salem State University – to achieve greater participation of underrepresented groups by helping these institutions adapt the AdCyL training materials.

7 AdCyL Projects and Case Studies Development

We describe the details of four of the twelve course projects which we will develop for imparting AC training via Data Science. We will use the datasets described in Section 5 for implementing all AdCyL case studies and projects.

7.1 Practical Cryptography and Ciphers for Biomedical Texts

Caesar cipher is one of the oldest encryption algorithms [78, 79]. It is based on a very simple idea – replacing a character by another character which is at a specified *distance* in the alphabet. The distance is called *shift factor*. If the shift factor is 1, the character 'a' is replaced by another character which is at a distance 1 (i.e., the character 'b'); 'b' is replaced by 'c'; 'c' is replaced by 'd', and so on. Note that 'z' is replaced by 'a' as the shift is *cyclic* (i.e., wraps around when the end is reached). The Caesar cipher is a **monoalphabetic substitution cipher** because it uses only one *cipher alphabet* per message. On the other hand, the Vigenère cipher employs several cipher alphabets per message and is an example of *polyalphabetic* cypher. Leon Battista Alberti proposed using two or more cipher alphabets and switching between them during encipherment to confuse potential cryptanalysts. Blaise de Vigenère refined Alberti's idea and turned it into a new cipher – **Vigenère cipher** – which uses twenty-six distinct cipher alphabets to encrypt a message.

To encrypt a message, the Vigenère cipher requires two strings – a message and a code. For simplicity, we assume both the message and the code are of the same length. Otherwise, the code is appended to itself until its length becomes equal to the message. Letters in the message and the key are denoted by $m_0 m_1 \dots m_{n-1}$ and $k_0 k_1 \dots k_{n-1}$. The corresponding encrypted message is denoted by $e_0 e_1 \dots e_{n-1}$. The following function performs the encryption: $e_i = (m_i + k_i) \bmod 27, 0 \leq i < n$. Decryption is achieved by the following function: *if* $((e_i - k_i) >= 0) m_i = (e_i - k_i)$ *else* $m_i = 27 + e_i - k_i$.

In conformance to the solution block structure of Figure 1, we provide a solution corresponding to the top and middle layers. The top layer will have two solution blocks, one performs encryption (by implementing $e_i = (m_i + k_i) \bmod 27, 0 \leq i < n$) and the other does decryption (by implementing *if* $((e_i - k_i) >= 0) m_i = (e_i - k_i)$ *else* $m_i = 27 + e_i - k_i$). Students will view both of these as black boxes and learn how to encrypt and decrypt messages by exploring with short messages. It is expected that this will generate enough curiosity to understand the inner workings of the back boxes. The middle layer solution blocks are quite similar for both encryption and decryption, and we describe just the blocks needed for encryption. We need two coarse granular solution blocks. The first block enables reading messages and key strings. The second block takes these two strings and returns an encrypted string.

7.2 Automatic Spelling Correction of Biomedical Text

Developing an end-to-end system spell-checking and auto-correction system is both challenging and time-intensive [80]. Norvig designed and implemented a simpler approach that performs well for this task [61, 81]. This implementation uses Python and we will adapt this approach for the biomedical text corpus. An understanding of probability, conditional probability, and Bayes' theorem at a conceptual level are the prerequisites, which will be provided under *core principles* category. The spelling correction problem involves, for a given typed word w, determining what word c (i.e., correction word) was most likely intended. For example, if w is "refered", c should be "referred". The problem is to select a correction word c, from among possible candidate correction words, that maximizes the probability that c is the intended correction word. In other words, compute the expression:

$$\underset{c \in \{\text{candidate correction words}\}}{\operatorname{argmax}} \quad p(c \mid w)$$

Using Bayes' theorem, the above expression is equivalent to:

$$\underset{c \in \{\text{candidate correction words}\}}{\text{argmax}} \frac{p(c)p(w\,|\,c)}{p(w)}$$

Since $p(w)$ is the same for every possible candidate correction word w, the denominator $p(w)$ can be dropped. Therefore, we have the following:

$$\underset{c \in \{\text{candidate correction words}\}}{\text{argmax}} p(c\,|\,w) = \underset{c \in \{\text{candidate correction words}\}}{\text{argmax}} p(c)p(w\,|\,c)$$

The correct candidate word c for a given misspelled word w is that c which results in the highest value for the right hand side (RHS) of the above expression. The RHS is a product of two probabilities: $p(c)$ denotes the probability that c is the intended word, and $p(w\,|\,c)$ is the probability that the word w was typed when the intended was actually c. $p(c)$ is called the *language model* and $p(w\,|\,c)$ is called the *error model*. It is easy to estimate $p(c)$ from the *language corpus data* – the ratio of frequency of occurrence of c to frequency occurrences of all words. However, the corpus does not provide data for estimating $p(w\,|\,c)$. Fortunately, $p(w\,|\,c)$ can be estimated from a list of misspellings and the notion of edit distance. In summary, the solution needs four solution blocks: the *language model*, $p(c)$; the *error model*, $p(w\,|\,c)$; the *candidate model*, $c \in \{\text{candidate correction words}\}$; and the *selection procedure*, which chooses the candidate with the highest combined probability.

Again, in conformance with the solution block structure of Figure 1, we provide a solution corresponding to each of the three layers. The *top layer* will have just one solution block, which returns the correct word c for a given misspelled word w. Students will get a feel for the accuracy of the spell corrector by inputing various misspelled words to the solution block. An instructor may even challenge students to probabilistically estimate the accuracy of the solution. The *middle layer* features four solution blocks, corresponding to the *language*, *error*, *candidate*, and *selection procedure* models. The students may be asked to develop these blocks, or weave a solution using a workflow system and pre-developed solution blocks. Next, consider the solution blocks corresponding to the *bottom layer*. For example, developing the language model, which essentially involves assigning a probability for every possible string in the language. The language model that assigns probabilities for strings of length one is called *unigram model*. Likewise, we can define *bigram* and *trigram* models. Procedures for estimating the model parameters need to deal with data sparsity problems. Smoothing techniques are used to help improve performance of language models.

7.3 Zipf's Law and Biomedical Text

Zipf's law states that for a large corpus, the frequency of any word is inversely proportional to its rank. The word that occurs most frequently is assigned a rank of 1. Students will answers the following two questions for the biomedical text corpus: (i) Do all the letters in the corpus occur with the same frequency? If not, which letter has the highest frequency? Which letter has the lowest frequency? (ii) Does Zipf's law hold for the biomedical text corpus?

First, the number of occurrences of words in the corpus are counted. Next, the words in decreasing order of their frequency of occurrence are listed. For a word w_i, let f_i be its frequency of occurrence and r_i be its rank or position in the sorted list. For example, for the 10^{th} most frequently occurring word, its rank in the list is 10. Zipf's Law states that for any given word, the product $f_i\, r_i$ is equal to some constant k.

We need to first determine major tasks and then come up with solution blocks for each of the three levels. The tasks are: (i) Read one file at a time, extract words, add their frequencies to corresponding cumulative counters. (ii) Sort words in decreasing order of frequency of occurrence, compute word ranks, and determine whether or not Zip's law holds. In conformance with the solution block structure of Figure 1, we provide a solution corresponding to each of the three layers.

There is only one block corresponding to the top layer. Given a list of files from the Gutenberg corpus, the solution block will output a table with three columns and a yes/no answer. The first column is the word, its frequency of occurrence is in the second column, and its rank in the third column. If Zip's law holds, the answer is yes, otherwise no. Students may explore whether or not Zip's law holds for different subsets of the Gutenberg corpus.

The middle layer will provide two solution blocks. The first one will read files, extract words, and increment corresponding frequency counts. Sorting words in decreasing of frequency of occurrence, assigning ranks, computing the product of frequency count and rank, and producing a yes/no answer is accomplished by the second block. The students may be asked to develop these blocks, or weave a solution using a workflow system and pre-developed solution blocks. Each middle layer solution block will be decomposed into two or more blocks of the bottom layer. For example, the first block of the second layer will give rise to three blocks — reading a disk file, extracting words from a file that resides in memory, and incrementing word frequency counters. Our approach prefers conceptual clarity over memory and processor optimizations. For example, though reading one line at a time from a disk file is more efficient than bringing the entire file into primary memory, we chose the approach of reading the entire file into memory. Furthermore, composing a solution using a workflow system is easier when the number of solution blocks is kept to a minimum.

7.4 Tissue Sample Classification into Cancer Classes

Breast cancer is a heterogeneous disease which exhibits high tumor variability in terms of the underlying biological mechanisms, response to treatment, and overall survival rate. Using RNA sequencing techniques, biomedical scientists are collecting gene expression profiles data on a large scale [82]. The large breast cancer dataset developed and provided by The Cancer Genome Atlas project [83] includes more than a thousand breast tumor samples characterized by RNA-Seq technology. Translating these data into biological insights remains a major challenge. Cancer is thought to be driven by gene expression pattern changes due to the accumulation of mutations or epigenetic modifications. A comprehensive characterization of changes in gene expression is central to our understanding of the disease and to make predictions about new data. For example, can we predict the following for a new patient using machine learning and Data Science? How will the patient respond to a particular treatment? How long will the patient live? Will the cancer recur? Big data, data science, and machine learning have the potential to answer such questions.

Unsupervised and supervised learning techniques as well as dimensionality reduction techniques are relevant for analyzing cancer genome data. More specifically, we will use data normalization, clustering and classification, principal component analysis, and exploratory factor analysis methods for analyzing this data. We will use Gene Expression Logic Analyzer (GELA), a workflow/pipeline system for discovering and extracting knowledge from gene expression profiles data [82]. With respect to the solution block structure of Figure 1, this activity corresponds to the top level solution block. CAMUR is a method for eliciting a greater amount of knowledge by computing several reliable classification models on The Cancer Genome Atlas RNA-seq data sets [84]. This enables identifying the most frequent genes related to the predicted cancer class. Our middle level solution blocks will involve developing multiple classification models.

8 Project Plan

The requested start date for this 3-year project is 1 September 2018. Major tasks of the AdCyL project along with PI/Co-PIs responsible for their execution are provided in the *coordination plan* document. Training materials will be piloted in a freshman-level course as core principles, projects, and case studies become available. Shown in Figure 4 is the project Gantt chart.

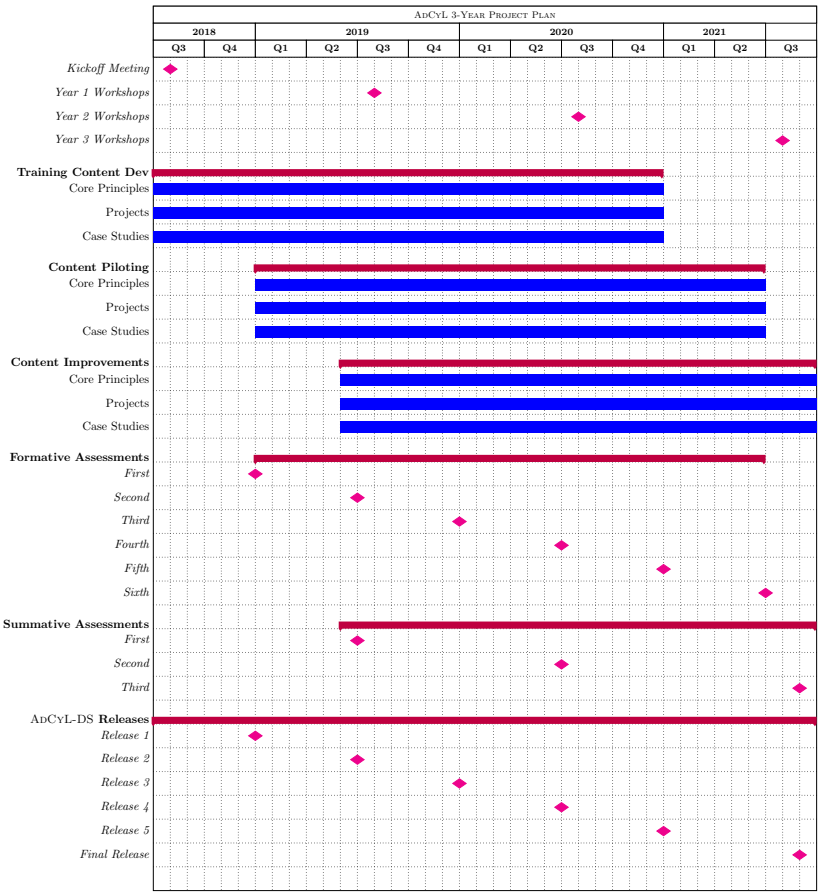

Figure 4: AdCyL project Gantt chart

9 Formative and Summative Project Evaluation

Evaluation Design: The evaluation of AdCyL will examine the implementation and impact of the project activities, with the ultimate goal of realizing **Computational and Data Science for All** education goal for first-year college students beyond East Carolina University. Specific evaluation questions and the approaches used to answer them appear below.

(1) What has been the impact of the project on participating students? One of the important goals of the project is to attract students to Data Science and AC education and eventually into workforce in these areas. The evaluation will look at the impact of the project on this ultimate outcome as well as on a key set of interim indicators. Specific student outcomes include the following.

Students' attitudes toward Data Science and AC. Students will be asked to complete a survey that assesses their attitudes toward Data Science and AC, and their understanding of how these methods and skills can be used in engineering and computer science domains. The survey will be administered at the start of the second year and at the end of the second year, and we will compare their attitudes and interest as well as in the other outcomes that follow. We will also continue to administer the survey during the third year to determine if the training content and pedagogy are successful at increasing students' interest in and knowledge of Data Science and AC.

Increased student success. Currently, less than 5% of East Carolina University students who begin the CS major graduate in 4 years. The goal here is to determine if the training content is

contributing to better retention. The evaluation will track the number of students who continue to remain enrolled over the semesters in both the major. We repeat this process for the engineering majors.

Because underrepresented populations are a specific focus for this project, the evaluation will disaggregate the results for specific sub-groups of students including: female students; low-income students, defined as those who are eligible for PELL grants; and students who are members of racial and ethnic groups underrepresented in college and in CS, including African-American, Hispanic/Latino, and Native American students.

(2) What strategies have been implemented for broad dissemination and greater impact? One goal of the project is to widely disseminate the training content to the CS academic community at-large. To what extent this objective has been achieved? We will survey the professors who have adapted the training materials for their instruction.

(3) What has been the impact of training content and AdCyL-DS **system on faculty instruction?** How did the project contribute to enhancing the teaching effectiveness of instructors? The evaluation will document these effects by conducting phone surveys. Data for this task will also be collected from project workshop participants.

(4) What lessons have been learned about AdCyL **and** AdCyL-DS **implementations?** The evaluator will document lessons that have been learned about implementation through annual reflective interviews with project staff. These interviews will also allow the evaluator to describe how the project has revised itself in responses to feedback and other data collected.

Reporting: The evaluator will provide ongoing formative feedback to the project team through regular meetings and calls. There will be a report provided annually that summarizes the results from the data analyzed to date. A summative report will be submitted at the end of the project. In addition, the evaluator will collaborate with the project staff on more innovative and user-friendly approaches to disseminating the findings of the project, such as through social media, video clips, and policy briefs.

10 Dissemination and Maximizing Broader Impacts

The products of the AdCyL project will be disseminated via multiple channels. Websites at the PI's institutions dedicated to this project are the first channel. The project Website will be designed in a way to ensure proper indexing by the Web search engines. Links will be established to this site from other prominent sites including the National Science Digital Library (NSDL). AdCyL-DS software will be hosted on GitHub for anytime and anywhere download. The software will be packaged as a *docker container* for easy installation on desktops. It will also be packed to enable easy installation on cloud services such as *Amazon Web Services*.

The Consortium for Computing Sciences in Colleges (CCSC) is another channel for dissemination. The ACM SIGCSE and IEEE Frontiers in Education (FIE) conferences are two other venues for achieving broader impact. The investigators will publish research findings and products in archival publications such as IEEE Transactions on Learning Technologies (TLT), IEEE Transactions on Education, ACM Transactions on Computing Education (TOCE), ACM eLearn Magazine, and Taylor & Francis Research in Learning Technology.

Other forums for dissemination include XSEDE, Science Gateways Community Institute, Campus Research Computing (CaRC) Consortium, ACI-REF Consortium, Blue Waters Project, ESnet, Open Science Grid, EGI Foundation, Coalition for Academic Scientific Computation (CASC), and Advancing Research Computing on Campuses (ARCC): Best Practices Workshop. The *Data Management Plan* provides details about data formats and data distribution mechanisms.

11 Sustainability Plan

AdCyL is a three-year project, during which a freshman-level course will be developed, piloted, assessed, enhanced, and disseminated. Currently, we offer a course titled "CSCI 1000 - Explorations in Computing," primarily for students who want to major in CS but are not prepared for the first CS course for majors. We will revise this course and name it "CSCI-1000: Exploring Computing through Data Science," and expand its appeal to all majors in ECU College of Engineering and Technology. The CS department has processes in place for *continuous improvement* of courses, as part of ABET accreditation requirements. CSCI-1000 student learning assessments of CS majors will serve as a baseline for ABET accreditation. We will continue to evolve CSCI-1000 to keep up with advances in AC and Data Science so that the course remains current and relevant.

Given the built-in scalability and personalization features, enhancing and evolving the course to ensure currency will be accomplished naturally in an incremental fashion. Within the project period, we expect to establish a critical mass of professors across the country who will be using the AdCyL teaching and learning content. From this point forward, it is reasonable to expect enhancing and evolving the content also becomes a self-sustaining community effort.

12 Results from Prior NSF Support

NSF REU Award #1560037, $359,986, 3/15/2016 - 3/14/2019, PI: Ding. Project title: REU Site: Software Testing and Analytics. This is a renewed project of previous funded NSF REU award #1262933 (4/1/20133/31/2016). We have hosted the program for five summer semesters at ECU. Fifty-two undergraduate students have participated in the program. **Intellectual Merit**: The sample research projects cover the most important open research topics in software testing and analytics. **Broader Impacts**: The project enabled a diverse pool of undergraduate students from underrepresented groups and academic institutions with limited research opportunities to conduct research. More than 20 REU student-authored research papers have been presented at national and international conferences.

NSF IUSE/PFE:RED Award #1730568, $2,000,000, 07/01/2017 - 06/30/2022, PI: Gudivada. Co-PI: Junhua Ding. Project title: IUSE/PFE:RED: PPSE - Transforming Programmers to Professional Software Engineers through Curricular Innovation, Inclusive Pedagogy, and Faculty Development. **Intellectual Merit**: The PPSE project takes an unprecedented, bold, and systemic approach to the professional formation of software engineers. **Broader Impacts**: The body of research that will be produced through this project will be relevant and equally effective for transforming CS teaching and learning across other CS departments in the country. Two book chapters which discuss the role of analytics in improving student learning are under review for publication in a research monograph.

NSF EEC-1359183, $287,949, 4/01/14 - 03/31/18, PI: George; and NSF EEC-1659796, $356,199, 04/01/2017 - 03/31/2020, PI: George. Both awards are titled REU Site: Research Experience for Undergraduates Biomedical Engineering in Simulations, Imaging, and Modeling (BME-SIM). We have hosted a 10-week summer program since 2014. A total of 35 students successfully completed the REU program. Evaluation results have shown our REU program to be largely successful. **Intellectual Merit**: BME-SIM projects aim to improve modeling of physiological and pathophysiological conditions. The models included various organs or systems; from cells to blood vessels to muscles. **Broader Impacts**: The primary goal of the project is to provide research opportunities to students who may not otherwise have the opportunity. We have successfully met this goal as 46% of our students were from under-represented minorities, 66% of our students were female, and 89% of our students were from R2 institutions or lower based on Carnegie Classification.

References Cited

[1] V. Gudivada, R. Baeza-Yates, and V. Raghavan, "Big data: Promises and problems," *IEEE Computer*, vol. 48, no. 3, pp. 20–23, Mar. 2015.

[2] D. McCreary and A. Kelly, *Making Sense of NoSQL: A guide for managers and the rest of us*. Shelter Island, NY: Manning, 2013.

[3] V. Dhar, "Data science and prediction," *Commun. ACM*, vol. 56, no. 12, pp. 64–73, Dec. 2013.

[4] M. Y. Vardi, "Science has only two legs," *Commun. ACM*, vol. 53, no. 9, pp. 5–5, Sep. 2010. [Online]. Available: http://doi.acm.org/10.1145/1810891.1810892.

[5] T. Hey, S. Tansley, and K. Tolle, *The fourth paradigm: Data-intensive scientific discovery*. Redmond, Washington: Microsoft Research, 2009. [Online]. Available: http://research.microsoft.com/en-us/collaboration/fourthparadigm/4th_paradigm_book_complete_lr.pdf.

[6] CACM Staff, "Science has four legs," *Commun. ACM*, vol. 53, no. 12, pp. 6–7, Dec. 2010. [Online]. Available: http://doi.acm.org/10.1145/1859204.1859206.

[7] A. Szalay, "Data-intensive discoveries in science: The fourth paradigm," in *Proceedings of the Fifth International Workshop on Data-Intensive Distributed Computing*, ser. DIDC '12, New York, NY: ACM, 2012, pp. 1–2. [Online]. Available: http://doi.acm.org/10.1145/2286996.2286998.

[8] Apache Airavata. (2018). A python data science platform, [Online]. Available: https://www.anaconda.com/what-is-anaconda/ (visited on 01/30/2018).

[9] NHANES. (2018). The national health and nutrition examination survey, [Online]. Available: https://www.cdc.gov/nchs/nhanes/index.htm (visited on 01/10/2018).

[10] ADNI. (2018). The alzheimer's disease neuroimaging initiative, [Online]. Available: http://adni.loni.usc.edu/ (visited on 02/10/2018).

[11] PPMI. (2018). The parkinsonï£¡ï£¡s progression markers initiative (ppmi), [Online]. Available: http://www.ppmi-info.org/ (visited on 02/10/2018).

[12] ABIDE. (2018). Autism brain imaging data exchange, [Online]. Available: http://fcon_1000.projects.nitrc.org/indi/abide/ (visited on 02/10/2018).

[13] Kitware. (2018). The insight journal, [Online]. Available: http://www.insight-journal.org/ (visited on 02/08/2018).

[14] VTK. (2018). The visualization toolkit (vtk), [Online]. Available: https://www.vtk.org/ (visited on 01/05/2018).

[15] American Statistical Association. (). Airline on-time performance, [Online]. Available: http://stat-computing.org/dataexpo/2009/ (visited on 01/12/2018).

[16] M. Marttila-Kontio, M. Kontio, and V. Hotti, "Advanced data analytics education for students and companies," in *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ser. ITiCSE '14, New York, NY: ACM, 2014, pp. 249–254. [Online]. Available: http://doi.acm.org/10.1145/2591708.2591746.

[17] K. Morik, H. Durrant-Whyte, G. Hill, D. Müller, and T. Berger-Wolf, "Data driven science: Sigkdd panel," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '15, New York, NY: ACM, 2015, pp. 2329–2330. [Online]. Available: http://doi.acm.org/10.1145/2783258.2788703.

[18] B. Howe, M. J. Franklin, J. Freire, J. Frew, T. Kraska, and R. Ramakrishnan, "Should we all be teaching "intro to data science" instead of "intro to databases"?" In *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, ser. SIGMOD '14, New York, NY: ACM, 2014, pp. 917–918. [Online]. Available: http://doi.acm.org/10.1145/2588555.2600092.

[19] L. N. Cassel, D. Dicheva, C. Dichev, D. Goelman, and M. Posner, "Data science for all: An introductory course for non-majors; in flipped format," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE '16, New York, NY: ACM, 2016, pp. 691–691. [Online]. Available: http://doi.acm.org/10.1145/2839509.2850558.

[20] L. N. Cassel, D. Goelman, D. Dicheva, and H. Topi, "Brainstorming data science as a fluency course for non-majors and as a new specialization," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE '16, New York, NY: ACM, 2016, pp. 708–708. [Online]. Available: http://doi.acm.org/10.1145/2839509.2850498.

[21] P. E. Anderson, T. Nash, and R. McCauley, "Facilitating programming success in data science courses through gamified scaffolding and learn2mine," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '15, New York, NY: ACM, 2015, pp. 99–104. [Online]. Available: http://doi.acm.org/10.1145/2729094.2742597.

[22] C. Anslow, J. Brosz, F. Maurer, and M. Boyes, "Datathons: An experience report of data hackathons for data science education," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE '16, New York, NY: ACM, 2016, pp. 615–620. [Online]. Available: http://doi.acm.org/10.1145/2839509.2844568.

[23] D. R. Hutchings and M. Squire, "Vismap: Exploratory visualization support for introductory data science and visualization," in *Proceedings of the 47th ACM Technical Symposium on Computing Science Education*, ser. SIGCSE '16, New York, NY: ACM, 2016, pp. 163–168. [Online]. Available: http://doi.acm.org/10.1145/2839509.2844572.

[24] O. A. Hall-Holt and K. R. Sanft, "Statistics-infused introduction to computer science," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '15, New York, NY: ACM, 2015, pp. 138–143. [Online]. Available: http://doi.acm.org/10.1145/2676723.2677218.

[25] R. Felder and R. Brent, "Understanding student differences," *Journal of Engineering Education*, vol. 94, no. 1, pp. 57–72, 2005.

[26] G. Bain and I. Barnes, "Why is programming so hard to learn?" In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education*, ser. ITiCSE '14, New York, NY, USA: ACM, 2014, pp. 356–356.

[27] J. Carter, P. Dewan, and M. Pichiliani, "Towards incremental separation of surmountable and insurmountable programming difficulties," in *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '15, New York, NY, USA: ACM, 2015, pp. 241–246.

[28] J. J. Lu and G. H. Fletcher, "Thinking about computational thinking," in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '09, New York, NY: ACM, 2009, pp. 260–264. [Online]. Available: http://doi.acm.org/10.1145/1508865.1508959.

[29] D. Eykholt. (2015). Discovering factors affecting student retention in computer science at cal poly, [Online]. Available: http://digitalcommons.calpoly.edu/cgi/viewcontent.cgi?article=1050&context=laessp (visited on 01/10/2018).

[30] P. J. Guo, "Software tools to facilitate research programming," PhD thesis, Stanford University, May 2012. [Online]. Available: http://pgbovine.net/publications/Philip-Guo_PhD-dissertation_software-tools-for-research-programming.pdf.

[31] P. Guo, "Teaching programming the way it works outside the classroom," *Commun. ACM*, vol. 56, no. 8, pp. 10–11, Aug. 2013. [Online]. Available: http://doi.acm.org/10.1145/2492007.2492012.

[32] P. K. Chilana, R. Singh, and P. J. Guo, "Understanding conversational programmers: A perspective from the software industry," in *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, ser. CHI '16, New York, NY: ACM, 2016, pp. 1462–1472. [Online]. Available: http://doi.acm.org/10.1145/2858036.2858323.

[33] Y. Gil, "Teaching big data analytics skills with intelligent workflow systems," in *Proceedings of the Sixth Symposium on Educational Advances in Artificial Intelligence*, ser. EAAI, 2016, pp. 4081–4088.

[34] ——, "Teaching parallelism without programming: A data science curriculum for non-CS students," in *2014 Workshop on Education for High Performance Computing*, Institute of Electrical and Electronics Engineers (IEEE), Nov. 2014. DOI: 10.1109/eduhpc.2014.12. [Online]. Available: https://doi.org/10.1109%2Feduhpc.2014.12.

[35] V. Allan, V. Barr, D. Brylow, and S. Hambrusch, "Computational thinking in high school courses," in *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, ser. SIGCSE '10, New York, NY: ACM, 2010, pp. 390–391. [Online]. Available: http://doi.acm.org/10.1145/1734263.1734395.

[36] C. Hu, "Computational thinking: What it might mean and what we might do about it," in *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '11, New York, NY: ACM, 2011, pp. 223–227. [Online]. Available: http://doi.acm.org/10.1145/1999747.1999811.

[37] B. N. Miller and D. L. Ranum, *Python programming in context*, Second. Burlington, Massachusetts: Jones & Bartlett Learning, 2014.

[38] T. T. Yuen and K. A. Robbins, "A qualitative study of students' computational thinking skills in a data-driven computing class," *Trans. Comput. Educ.*, vol. 14, no. 4, 27:1–27:19, Dec. 2014.

[39] M. D. DeSchryver and A. Yadav, "Creative and computational thinking in the context of new literacies: Working with teachers to scaffold complex technology-mediated approaches to teaching and learning," *Journal of Technology and Teacher Education*, vol. 23, no. 3, pp. 411–431, Jul. 2015.

[40] B. Kules, "Computational thinking is critical thinking: Connecting to university discourse, goals, and learning outcomes," in *Proceedings of the 79th ASIS&T Annual Meeting: Creating Knowledge, Enhancing Lives Through Information & Technology*, ser. ASIST '16, Silver Springs, MD: American Society for Information Science, 2016, 92:1–92:6.

[41] M. M. Syslo, "From algorithmic to computational thinking: On the way for computing for all students," in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, ser. ITiCSE '15, New York, NY: ACM, 2015, pp. 1–1. [Online]. Available: http://doi.acm.org/10.1145/2729094.2742582.

[42] R. C. Gonzalez and R. E. Woods, *Digital image processing*, Third. Upper Saddle River, N.J.: Prentice Hall, 2008.

[43] S. Jothilakshmi and V. Gudivada, "Large scale data enabled evolution of speech processing research and applications," in *Cognitive Computing: Theory and Applications*, ser. Handbook of Statistics, V. Gudivada, V. Raghavan, V. Govindaraju, and C. R. Rao, Eds., vol. 35, New York, NY: Elsevier, Sep. 2016, pp. 301–340.

[44] H. Li, B. Ma, and K.-A. Lee, "Spoken language recognition: From fundamentals to practice," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[45] R. Zhang, P. Isola, and A. A. Efros. (2017). Colorful image colorization, [Online]. Available: http://richzhang.github.io/colorization/ (visited on 01/15/2017).

[46] ——, "Colorful image colorization," *CoRR*, vol. abs/1603.08511, 2016. [Online]. Available: http://arxiv.org/abs/1603.08511.

[47] P. Williams, R. Sennrich, M. Post, and P. Koehn, *Syntax-based Statistical Machine Translation*, ser. Synthesis Lectures on Human Language Technologies 4. 2016, vol. 9, pp. 1–208. [Online]. Available: http://dx.doi.org/10.2200/S00716ED1V04Y201604HLT033.

[48] D. Jurafsky and J. H. Martin, *Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition*, Second. Upper Saddle River, N.J.: Pearson Prentice Hall, 2009.

[49] I. Foster, *Big data and social science: A practical guide to methods and tools*, ser. Statistics in the Social and Behavioral Sciences. Boca Raton, Florida: Chapman & Hall/CRC, 2017.

[50] C. N. Knaflic, *Storytelling with data: A data visualization guide for business professionals*. New York, NY: John Wiley, 2015.

[51] M. Hofmann and R. Klinkenberg, *RapidMiner: Data mining use cases and business analytics applications*. Boca Raton, Florida: Chapman and Hall/CRC, 2013.

[52] J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*, Third, ser. Morgan Kaufmann Series in Data Management Systems. Amsterdam, Netherlands: Morgan Kaufmann, 2011.

[53] Google. (2018). Google maps geocoding api, [Online]. Available: https://developers.google.com/maps/documentation/geocoding/start (visited on 01/10/2018).

[54] A. McAndrew, "Teaching cryptography with open-source software," 1, vol. 40, New York, NY: ACM, Mar. 2008, pp. 325–329. [Online]. Available: http://doi.acm.org/10.1145/1352322.1352247.

[55] ——, *Introduction to cryptography with open-source software*. Boca Raton, FL: CRC Press, 2011. [Online]. Available: http://www.worldcat.org/search?qt=worldcat_org_all&q=9781439825709.

[56] S. Tanimoto, *An interdisciplinary introduction to image processing: Pixels, numbers, and programs*. MIT Press, 2012.

[57] V. Gudivada, D. Rao, and V. Raghavan, "Big data driven natural language processing research and applications," in *Big Data Analytics*, ser. Handbook of Statistics, V. Govindaraju, V. Raghavan, and C. R. Rao, Eds., vol. 33, New York, NY: Elsevier, Jul. 2015, pp. 203–238.

[58] S. Burton, R.-M. Déchaine, and E. Vatikiotis-Bateson, *Linguistics for dummies*. Toronto, Canada: John Wiley & Sons, 2012.

[59] V. Gudivada, "Data analytics: Fundamentals," in *Data Analytics for Intelligent Transportation Systems*, M. Chowdhury, A. Apon, and K. Dey, Eds., ISBN: 978-0-12-809715-1, New York, NY: Elsevier, Apr. 2017, pp. 31–67.

[60] T. Segaran and J. Hammerbacher, *Beautiful data*. Sebastopol, CA: O'Reilly, 2009.

[61] P. Norvig. (2016). How to write a spelling corrector, [Online]. Available: http://norvig.com/spell-correct.html (visited on 01/10/2017).

[62] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*. New York, NY: Springer, 2013.

[63] V. Kotu and B. Deshpande, *Predictive analytics and data mining: Concepts and practice with RapidMiner*. New York, NY: Morgan Kaufmann, 2014.

[64] P. D. Lewis, *R for medicine and biology*. Sudbury, Massachusetts.: Jones and Bartlett Publishers, 2010.

[65] A. Chen. (2017). Data from wearables can predict disease, [Online]. Available: http://www.theverge.com/2017/1/12/14251438/data-smartwatch-predict-health-wearables-illness (visited on 01/15/2018).

[66] S. A. Ambrose, M. W. Bridges, M. DiPietro, M. C. Lovett, and M. K. Norman, *How learning works: Seven research-based principles for smart teaching*. San Francisco, CA: Jossey-Bass, 2010.

[67] J. D. Karpicke and J. R. Blunt, "Retrieval practice produces more learning than elaborative studying with concept mapping," *Science*, vol. 331, no. 6018, pp. 772–775, Feb. 2011.

[68] S. Alaoutinen, "Effects of learning style and student background on self-assessment and course performance," in *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, ser. Koli Calling '10, New York, NY, USA: ACM, 2010, pp. 5–12.

[69] E. E. Bachari, E. H. Abelwahed, and M. E. Adnani, "E-learning personalization based on dynamic learners' preference," *International Journal of Computer Science and Information Technology*, vol. 3, no. 3, pp. 200–216, 2011.

[70] POGIL. (2018). Process oriented guided inquiry learning, [Online]. Available: https://pogil.org/ (visited on 01/05/2018).

[71] CS-POGIL. (2018). Process oriented guided inquiry learning in computer science, [Online]. Available: http://cspogil.org/Home (visited on 01/05/2018).

[72] OASIS. (2018). Darwin information typing architecture (DITA) TC, [Online]. Available: https://www.oasis-open.org/committees/tc%5C_home.php?wg%5C_abbrev=dita (visited on 01/10/2018).

[73] DITA XML.org. (2018). Online community for the darwin typing architecture oasis standard, [Online]. Available: http://dita.xml.org/ (visited on 01/10/2018).

[74] K. Soltys. (2018). Lightweight dita: a preview from michael priestley, [Online]. Available: http://techwhirl.com/lightweight-dita-preview-michael-priestley/ (visited on 01/11/2018).

[75] D. Hestenes, M. Wells, and G. Swackhamer, "Concept inventory," *The Physics Teacher*, vol. 30, no. 3, pp. 141–151, 1992.

[76] BaseX. (2018). An open-source, light-weight, high-performance, and scalable native XML database system, [Online]. Available: http://basex.org/ (visited on 01/07/2018).

[77] DITA Open Toolkit. (2018). An open-source publishing engine for XML content authored in the darwin information typing architecture – extensible publishing power for DITA workflows, [Online]. Available: http://www.dita-ot.org/ (visited on 01/10/2018).

[78] S. Singh, *The Code Book: The Secrets Behind Codebreaking*. New York: Delacorte Press, 2016.

[79] ——, *The code book, the science of secrecy from ancient Egypt to Quantum Cryptography*. Anchor, 2000.

[80] C. Whitelaw, B. Hutchinson, G. Y. Chung, and G. Ellis, "Using the web for language independent spellchecking and autocorrection," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2*, ser. EMNLP '09, Stroudsburg, PA: Association for Computational Linguistics, 2009, pp. 890–899. [Online]. Available: http://dl.acm.org/citation.cfm?id=1699571.1699629.

[81] P. Norvig, "Natural language corpus data," in *Beautiful Data: The Stories Behind Elegant Data Solutions*, T. Segaran and J. Hammerbacher, Eds., O'Reilly Media, 2009, ch. 14, pp. 219–242.

[82] E. Weitschek, G. Fiscon, G. Felici, and P. Bertolazzi, "Gela: A software tool for the analysis of gene expression data," in *26th International Workshop on Database and Expert Systems Applications (DEXA)*, Sep. 2015, pp. 31–35. [Online]. Available: http://ieeexplore.ieee.org/document/7406265/?arnumber=7406265.

[83] The Cancer Genome Atlas. (2018). National cancer institute and national human genome research institute, [Online]. Available: https://cancergenome.nih.gov/ (visited on 01/10/2018).

[84] V. Cestarelli, G. Fiscon, G. Felici, P. Bertolazzi, and E. Weitschek, "Camur: Knowledge extraction from RNA-seq cancer data through equivalent classification rules," *Bioinformatics*, vol. 32, no. 5, pp. 697–704, Oct. 2015. [Online]. Available: https://doi.org/10.1093%2Fbioinformatics%2Fbtv635.