Interdisciplinary STEM Undergraduate Programs and the Effectiveness of Computing Competencies within the Curriculum

Katherine G. Herbert
Department of Computer Science
Montclair State University
Montclair, NJ
herbertk@montclair.edu
orcid: 0000-0001-6663-8187

Thomas J. Marlowe
Department of Mathematics and Computer Science
Seton Hall University
South Orange, NJ
thomas.marlowe@shu.edu

Kees Leune
Department of Mathematics and Computer Science
Adelphi University
Garden City, NY
leune@adelphi.edu
orcid: 0000-0003-1171-6410

Robert M. Siegfried
Department of Mathematics and Computer Science
Adelphi University
Garden City, NY
siegfrie@adelphi.edu
orcid: 0000-0001-9362-4311

Jeanette Wilmanski
Department of Biology
Saint Peter's University
Jersey City, NJ
jwilmanski@saintpeters.edu
orcid: 0000-0002-1681-803X

***Abstract*— Undergraduate interdisciplinary, multidisciplinary and transdisciplinary computing-related STEM programs are proliferating extensively. Each of these programs requires a robust computing component to be integrated into the curriculum. However, including an introductory, programming-oriented sequence designed for CS majors is not always the best fit for these multifaceted programs. In this work in progress paper, we set out to investigate possibilities and issues for the computing component, focusing on three fields: bioinformatics, data science, and cybersecurity. We explore commonalities and differences, and discuss initial plans to test our hypotheses. In doing so, we consider data organization, data acquisition, and a preliminary survey design.**

Keywords— ***CS 0, CS 1, CS 2, introductory computing courses, undergraduate interdisciplinary STEM programs, computing in interdisciplinary programs, bioinformatics, data science, cybersecurity***

I. INTRODUCTION

Over the past thirty years, an ever greater number of interdisciplinary programs have been created, in which computing is an inherent component, or into which computing has been integrated [1,2]. Arguably, computer science (CS) is itself interdisciplinary, arising out of mathematics, physics, engineering, and business in the 1970s and 1980s. These upcoming interdisciplinary fields include bioinformatics, computational chemistry, computational mathematics, data science, cybersecurity, and many others. Incorporating an interdisciplinary approach that includes computing and CS into natural sciences has many demonstrated benefits. These include positive effects on recruitment and retention, as well as preparing undergraduate students for further studies and employment [3]. In many institutions, computer science and these interdisciplinary programs taken together, each enroll a significant population of undergraduate students [4].

Courses in computing play an essential role in these interdisciplinary programs and are typically required of students [5]. Computing content, particularly in data science and cybersecurity programs, is often provided by the department hosting the CS major. In most of those cases, interdisciplinary students begin with an introductory CS programming sequence, which is often the same as what CS majors take. Alternatively, some CS programs start with a computational problem-solving approach to anticipate major topics in the CS major [6, 7].

This use of established curriculum arises in part because of staffing and resource issues. Some universities have a limited contingent of CS faculty, with minimal support, and high demands for research, service, and mentoring of undergraduate students. At other institutions, introductory courses are often taught by adjuncts and teaching assistants. Yet the needs of majors in STEM interdisciplinary programs will not be identical to those of CS students [8]. Additionally, the requirements across these disciplines are not uniform. This is reminiscent of introductory Statistics courses: theory-based for the mathematical sciences, calculus-based or model-driven for other STEM fields, and more applied courses for the social sciences. This paper focuses on undergraduate majors in three areas: bioinformatics, data science and cybersecurity.

These fields of study differ. Cybersecurity students must understand the fundamental internals of computers, operating systems, networks and communication, and software engineering, and should develop programming skills. Data science majors need, in addition to knowledge of analyses, tools, and visualizations, a sense of issues in storage, communication, and databases, and experience with scientific and mathematical programming, perhaps in special-purpose languages [9, 10]. Students in these programs should anticipate substantial computing content in their programs, but may not be served well by a course focused on programming.

In bioinformatics, the motivations for the curriculum tend to lean towards solving data-driven problems in the biological/health sciences, with computing objectives often not being the primary thrust of the degree [11,12]. This often attracts students with primary interests in the underlying biology-related questions. Their emphasis is likely to be on data-related skills. CS programs consider data manipulation more advanced than programming, and consequently, it is highly possible that they are introduced in courses that are not included in the interdisciplinary program. Thus, the traditional introductory programming-first approach may not be ideal for introducing computing to them.

In contrast, cybersecurity is commonly defined as a primarily computing-based discipline. The 2017 Joint Task Force Cybersecurity Curricular Guidelines document [13] defines the discipline as involving technology, people, information, and processes to enable assured operations in the context of adversaries. It is an interdisciplinary course of study, including aspects of applied mathematics and statistics, law, policy, human factors, ethics, and risk management, and to a lesser extent, economics and social science. As such, the methods by which computing is taught to cybersecurity students differ from the methods used for disciplines that are less computing-focused. For data science, ACM is currently drafting standards[14]. However, at the university curriculum level, there are many stakeholder departments. Business, CS, mathematics, cognitive science, psychology, and sociology all have significant claims to this curriculum topic. Computing is seeing an explosion of data science undergraduate programs, yet little data has been recorded about their efficacy.

In this research work-in-progress paper, we open a discussion of this topic, and identify important issues. Due to space limitations and the relative newness of these fields, we concentrate mainly on bioinformatics. Section 2 discusses further our observed challenges from our institutional experiences, initial ad-hoc survey, and literature review. Section 3 discusses research design and instruments for further research. Section 4 concludes with final discussion and remarks, and an open invitation to contact our team to collaborate in this ongoing work and to communicate observations and related experiences.

II. Interdisciplinary Curriculum Challenges

Students in interdisciplinary STEM curricula with substantial computing content might best be served by concept-driven, higher-abstraction curriculum modules with emphases appropriate to their intended fields. This goal is difficult to meet with a standard programming-focused sequence, but constructing a sequence appropriate to all such interdisciplinary programs may impose its own challenges. We would argue that a focused introduction to computing emphasizing major-related fluency with high-level concepts will improve interdisciplinary programs, in the same way as major-oriented mathematics courses have contributed. For example, understanding graph structures and networks can help with understanding metabolic pathways and other biological processes. The involvement of computing changes the way problems are formulated, explored, and analyzed. Without this background in computing, students may find it more difficult to utilize these capabilities in tackling questions in their interdisciplinary field. In contrast, students exposed to a sequence focused on programming, unrelated to their major but aimed at concepts in later CS courses, may suffer cognitive overload, so that little is retained in any subject. Naturally, this approach must be interwoven with study of the subject discipline(s) in order to illuminate and understand the fundamental questions of their interdisciplinary major.

Developing the proper set of curriculum modules must be considered in light of institutional resource challenges. Universities face resource constraints in offering any interdisciplinary program, with serious staffing, curricular, and student preparation challenges. Even if such a computing sequence could be designed and staff assigned, challenges for instructors remain: overview knowledge of the interdisciplinary field(s), understanding of the pedagogical objectives, and personal motivation. These challenges are different for institutions where courses are taught by full-time faculty, intruding on their already heavy load, contending with ongoing research or service plans. With content taught by teaching assistants and adjuncts, issues arise from continually developing the interdisciplinary skills of such faculty. Additionally, there is the possibility of continuous attrition of these faculty after such investments.

For bioinformatics, similar challenges and others are discussed in [12,15]. The NIBLSE site is a significant resource considering implementation of bioinformatics curriculum core competencies. This site looks at the goals for the computing component of a bioinformatics program, but not at the desired computing fluency efficacy level. Its goals are specific to bioinformatics, although a few competencies (Role, Concepts, Scripting, Social/Ethical Issues) generalize broadly. The others, with suitable modification, apply mostly to analogous programs like computational chemistry. However, implementing such competencies, especially among resource-strapped institutions or institutions serving under-represented populations, remains a challenge.

There is also likely to be a lag in the implementation of computing curriculum, except perhaps on an experimental basis. Issues include negotiation of objectives, content, and structure with host departments or program offices. Additionally, time and other resources needed to develop and get approval for new courses, and time to orient and prepare faculty must be considered. Until such sequences become commonly accepted, there will be issues when students transfer between programs and between institutions.

Another challenge lies in holding student interests while developing knowledge and capabilities. The sequence (one or two courses, or modules inserted in existing courses) needs to present a useful but conceptual overview of computing, and a principled exploration of the questions students may face. They must be structured concretely enough to keep student interest. It cannot completely ignore programming, but should concentrate on the foundations, or focus on aspects that are directly useful to the student in their major, without getting entangled in unneeded technical or language details.

Since the programming/CS component in these majors will be at most at the level of a minor, it is arguably also important to incorporate some emphasis on professional skills related to computing: managing requirements, problem solving skills, visualization and presentation, and teamwork. How much emphasis these should receive, and how and where they should be incorporated, is a question that needs further investigation.

On the other hand, while it is fairly easy to enunciate these problems, there may be others we have not identified. Data on interdisciplinary programs is not that easy to find, especially when programs that are very similar or even nearly identical can go by very different names, and data on their computing components, and on evaluation of those components, is even more difficult to derive.

III. Research

Research in pedagogical/programmatic issues impacting the status of one particular field in undergraduate interdisciplinary programs is difficult to find. In academic publications, these concerns are usually deferred to institutional implementation. When searching for data about the current status of the field, sources such as the National Center for Educational Statistics (NCES), Computing Research Association (CRA)'s Taulbee Survey [16], or the National Science Foundation's (NSF) biennial Science and Engineering Indicators [17] have limited data sets. They tend to group interdisciplinary programs together, making it difficult to isolate a given program. This is especially challenging for the NCES data, where interdisciplinary may encompass non-STEM fields.

It is imperative that computing curriculum issues be addressed early and at undergraduate levels. Over 70% of degrees awarded in the sciences are undergraduate degrees, and baccalaureate degrees serve as an entry (and often final) degree in most computing-related fields, although possibly complemented by later professional credentials [18]. Additionally, many under-represented groups often do not have access to graduate degrees until later in life.

Professional societies that encompass such interdisciplinary programs often have curriculum-oriented publications that can support institutions with similar challenges. However, these are skewed towards that professional society's primary discipline. Considering bioinformatics, papers about curriculum look widely different when published in, for example, ACM SIGCSE, IEEE Frontiers in Education, American Biology Teacher, the Journal of College Science Teaching, or PLOS Computational Biology.

Studies of interdisciplinary undergraduate programs also tend to be influenced by graduate student programmatic data. As seen with NCES, education statistics venues tend to group interdisciplinary programs together, making it hard to extract useful trends for understanding program efficacy. Interdisciplinary program data can be obtained from professional societies, NSF S&E reports, and public "ranking" reports like US News and World Report but often have similar problems. Additionally, the concept of graduate transitions, where students start in one discipline but seek graduate studies in another discipline, is also relevant [19, 20]. As a consequence, fields rely heavily on professional society and graduate studies data, which are often more concerned with graduate student outcomes. The Council on Graduate Studies has recognized challenges in NCES data sets, and has proposed its own system for recognizing and characterizing interdisciplinarity [21].

Another issue in furthering this work is proper collection and analysis of data. The National Science Board, in The Skilled Technical Worker [22] publication, discusses the need for more data in all of these areas. The initial step will be identification of the population: determining the set of computing-related STEM interdisciplinary undergraduate major programs, and classify (most of) them into buckets, including bioinformatics, data science, and cybersecurity. For bioinformatics, Sayres et. al. provides a survey for collecting data, but this survey is at too high a granularity to treat the computing concepts needed to understand the efficacy of the introductory computing sequence.

Once the survey population is identified, program administrators and faculty teaching in those programs—both teaching the computing component and others—will be surveyed. Among the questions to be considered are: (1) how significant is the computing component of this program? (2) to what extent do you see it as a CS-related concept? (3) is your introductory sequence shared with CS students or aimed at interdisciplinary students, and if the latter, is it major specific or multi-program based? (4) at what point in a student's program is this sequence taken? (5) to what extent does this sequence exercise soft skills? (6) are these concepts extensible to any other interdisciplinary programs? and (7) does the course/sequence include consideration of ethical, social, global, and related factors? This instrument will be continually reviewed, possibly with involvement of a voluntary advisory committee, to critically reflect on the appropriateness and usefulness of the data collected.

The methodology used to administer the survey will be based on existing surveys of introductory programming. We plan to use a compiled list of 4-year institutions across the US, based on the original Reid list work [23]. Additionally, we will review further approaches used across participating subject-areas professional societies, so that all participating domains are recognized. Moreover, we are also cultivating partnerships with publishers to help verify survey data results. The authors recognize that there are challenges in using the methodology. It is US-centric, still CS-centric even given professional society input on methodology, and under-represents all but undergraduate, baccalaureate institutions.

IV. Discussion and Future Work

Understanding the nature of STEM interdisciplinary programs involving CS is a challenging task. CS, by its nature and role in curriculum, runs a spectrum from understanding computing from a general education perspective, through degree programs where it partners with other disciplines, to its own singular major. With the different interpretations and levels, it is understandable but not always satisfactory that many would want to group these fields into a single "interdisciplinary" category.

In this paper, we have begun looking at the problems for pursuing investigation of the introductory computing experience for interdisciplinary STEM majors, using cybersecurity, data sciences, and especially bioinformatics as initial examples. In our research, we see that institutions, particularly those with significant constraints, and that serve populations with multiple challenges, run up against difficulties in delivering computing curriculum. These institutions need curricular pathways so that these groups have a route into interdisciplinary studies with a good chance of success. Additionally, as undergraduate degrees serve as entry-level credentials for employment, it is imperative to find ways forward within the undergraduate curriculum to support this transition for these students.

There are other questions of interest related to the generality of this work. For example, will the conclusions hold in other contexts, and if so, to a comparable, greater, or lesser extent. In particular, do they hold for graduate programs? Or for associate degree (AS) students and those pursuing certificates or industry-specific professional certifications? To adult students returning to change fields or enhance their resumes? It should be noted that roughly half of AS degrees are awarded in computing-related fields, and that many of those students are older students. In each case, are there additional issues to be considered? Is there a reasonable spectrum (possibly multi- but low dimensional) on which one can categorize STEM interdisciplinary programs to characterize their computing needs, and desired content and approach for an introductory sequence in computing? Can such courses be structured so as to support forward reference to important issues in the field, and backward reference to computational concepts, approaches, and techniques?

We invite others to participate in this work, either collaborating or informing our research, or adding to it on their own. Interested parties may contact any of the co-authors.

References

[1] G-A. Amoussou, M. Boylan, and J. Peckham, "Interdisciplinary computing education for the challenges of the future," Proc. of the 41st ACM technical symposium on Computer science education (SIGCSE '10), ACM, New York, NY, USA, 556–557, doi: 10.1145/1734263.1734449

[2] U. Wolz, L. Cassel, T. Way, and K. Pearson, "Cooperative expertise for multidisciplinary computing," Proceedings of the 42nd ACM technical symposium on Computer science education (SIGCSE '11), 2011, ACM, New York, NY, USA, 329–334. doi:10.1145/1953163.1953264

[3] L. Carter, "Interdisciplinary computing classes: worth the effort," Proc. of the 45th ACM technical symposium on Computer science education (SIGCSE '14), 2014, ACM, New York, NY, USA, 445–450, doi:10.1145/2538862.2538882.

[4] Digest of Education Statistics, Institute of Education Sciences, 2019. Retrieved on January 20, 2021 from https://nces.ed.gov/programs/digest/d19/tables/dt19_322.10.asp

[5] J.A. Qualls and L.B. Sherrel, "Why computational thinking should be integrated into the curriculum," Journal of Computing Sciences in Colleges, vol. 25, no. 5, 2010, pp. 66-71.

[6] J.G. Brookshear and D. Brylow, "Computer Science An Overview," Pearson, January 2019, ISBN 978-1292263427.

[7] M.Felleisen, R.B. Findler, M. Flat, and S. Krishnamurthi, "How to Design Programs: An Introduction to Programming and Computing", The MIT Press, 2001.

[8] V. Barr, "Computing education will not be one size fits all," ACM Inroads, vol. 9, no. 4, December 2018, pp. 73–76. doi:10.1145/3276305

[9] P. Anderson, J. Bowring, R. McCauley, G. Pothering, and C. Starr, "An undergraduate degree in data science: curriculum and a decade of implementation experience," Proc. of the 45th ACM technical symposium on Computer science education (SIGCSE '14), 2014, ACM, New York, NY, USA, 145–150. DOI: 10.1145/2538862.2538936

[10] J. C. Adams, "Creating a Balanced Data Science Program," Proc. of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20), 2020, ACM, New York, NY, USA, 185–191. DOI:10.1145/3328778.3366800

[11] L. Welch, F. Lewitter, R. Schwartz, C. Brooksbank , P. Radivojac, B. Gaeta, M. V. Schneider, "Bioinformatics Curriculum Guidelines: Toward a Definition of Core Competencies", PLOS Computational Biology, March 2014, DOI: 10.1371/journal.pcbi.1003496

[12] M. A. Wilson Sayres, C. Hause, M.l Sierk, S. Robic, A. G. Rosenwald, T. M. Smith, et al. "Bioinformatics core competencies for undergraduate life sciences education", PLOS One, June, 201, DOI: 10.1371/journal.pone.0196878.

[13] Cybersecurity Curricula 2017, Computing Curricula Series Joint Task Force on Cybersecurity Education, December, 2017. Retrieved on January 21, 2021 from https://cybered.hosting.acm.org/wp-content/uploads/2018/02/newcover_csec2017.pdf.

[14] P. M. Leidig and L. Cassel, "Efforts on Computing Competencies for Undergraduate Data Science Curricula," Proc. of the 2020 ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE '20). ACM, New York, NY, USA, 519–520. DOI:10.1145/3341525.3393962

[15] NIBLSE Conference. Bioinformatics Competencies. NIBLSE Bioinformatics Core Competencies, 2019. Retrieved on January 21, 2021 from https://qubeshub.org/community/groups/niblse/core_competencies.

[16] S. Zweben and B. Bizot, "The CRA Taulbee Survey," Computing Research News, vol. 32, no. 5, May, 2020, pp. 3-63. Retrieved on January 21, 2021 from https://cra.org/wp-content/uploads/2020/05/2019-Taulbee-Survey.pdf

[17] B. Khan, C. Robbins, and A. Okrent, "The State of U.S. Science and Engineering", 2020, National Science Foundation. Retrieved on January 20, 2021 from https://ncses.nsf.gov/pubs/nsb20201

[18] J. Trapani and K. Hale, "Trends in Undergraduate and Graduate S&E Degree Awards," Higher Education in Science and Engineering, National Science Foundation, September 2019. Retrieved on January 20, 2021 from https://ncses.nsf.gov/pubs/nsb20197/trends-in-undergraduate-and-graduate-s-e-degree-awards#undergraduate-degree-awards

[19] T. J. Marlowe, K. G. Herbert, "An Interdisciplinary View of Education in the Formal and Natural Sciences – From STEM to STREAM to …", Journal of Systemics, Cybernetics, and Informatics, vol. 17, no. 5, December 2019, pp. 75-87.

[20] K. G. Herbert, T. J. Marlowe, "An Interdisciplinary Graduate Certificate in the Formal and Natural Sciences – A Proposal", Journal of Systemics, Cybernetics, and Informatics, vol. 17, no, 5, December 2019, pp. 88-92.

[21] N. E. Bell, "Data Sources: Measuring Interdisciplinarity," Council of Graduate Schools, June, 2012. Retrieved on January 21, 2021 from https://cgsnet.org/data-sources-measuring-interdisciplinarity-0

[22] D. L. Souvaine, E. Ocha, et al. "The skilled technical workforce: Crafting America's Science & Engineering Enterprise," National Science Board Report NSB-2019-23, September, 2019. Retrieved on January 20, 2021 from https://nsf.gov/nsb/publications/2019/nsb201923.pdf.

[23] R. M. Siegfried, D. Liporace, and K. G. Herbert, "What Can the Reid List of First Programming Languages Teach Us About Teaching CS1?", SIGCSE '19: Proc.s of the 50th ACM Technical Symposium on Computer Science Education, Minneapolis, Minnesota, February 2019.