Computational Knowledge Aquisition Strategies Among Environmental Science Graduate Students in a Methods of Data Analysis Course Sequence

Allison Theobold and Stacey Hancock

November 16, 2017

Abstract

Introduction

Computing has become foundational in the fields of Environmental Sciences, however, the gap between Environmental Science education and computation has become more evident. With the growth in computational power, the complexity of Environmental Science models has followed, multiplying the computational, mathematical, and statistical expectations of researchers' abilities. Because of the importance of computational knowledge and abilities, some universities have started to require Environmental Science students to enroll in an introductory programming course (Rubinstein and Chor 2014), while others are providing computing bootcamps for quantitative methods (Stefan et al. 2015).

This research focuses on graduate students enrolled at a mid-size university in the Rocky Mountains, where Methods of Data Analysis I and II are graduate statistics courses required for the completion of a Master's degree in many Environmental Science fields, such as Ecology, Land Resources Environmental Sciences, Animal and Range Sciences, Plant Sciences. [1] This terminal statistics sequence intends to prepare graduate students for the statistical and computational problems they may face as researchers and practitioners. The intention of this study is to describe and understand what factors impact how Environmental Science graduate students gain computational knowledge and the ability to reason through applications in their disciplines. In examining the experiences Environmental Science graduate students face when acquiring computational skills, the present study seeks to capture a more in-depth understanding of the successes and shortfalls these students experience when procuring computational knowledge and skills. The primary research questions that guided this study were:

1. What factors (attitude, background, course content, etc.) impact how graduate Environmental Science students' gain computational knowledge and the ability to reason through computational applications in their discipline?
2. In what ways (if any) do these factors inhibit or foster students' computational understandings and abilities across different backgrounds?

Computational Needs and Abilities of Envrionmental Science Students

Research in computational abilities of Environmental Science majors is in its infancy, with only a handful of institutions performing research that specifically addresses the computation training necessary to prepare students for careers post undergraduate or graduate school. In this section we discuss briefly three literatures that have informed this study. First, we review the literature on the foundational role computation has in

[1] The large assortment of fields this sequence of courses services will hence be labeled under the banner of Environmental Science.

mathematics and the sciences. We then discuss the research efforts on curriculum design for introductory computing courses for non-computer science students. Finally, we describe one institutions development and implementation of computational training for graduate students in the biological sciences.

Importance of Computing in Envrionmental Science

blurb on importance of computing

Computational Courses for Undergraduate Science Majors

Multi-disciplinary efforts have been made at Purdue, Carnegie-Mellon, Harvey Mudd, Princeton, and Winona State ((Cortina 2007, Dodds et al. (2008), (**???**), (**???**), Wilson et al. (2008), Wing (2006))) to create introductory computing courses with a focus on non-computer science majors, in particular science students, in fields ranging from physics to chemistry to biology. These courses are produced in collaboration with science faculty, and are intended to begin each students journey into computing. Students are presented examples in a familiar language that allows for them to focus on the foundational principles of each computational problem. (Hambrusch et al. 2009) The physics, chemistry, and bioinformatics departments collaborated with computer science department, conveying their expectations on what they wished the students would learn. The research on undergraduate level computing courses directly relates to the computational abilities of graduate level science researchers, as these students have most likely graduated from an undergraduate life science program with no computational training. The concepts emphasized in the courses developed at Purdue, and elsewhere, can be used to inform Environmental Science and Statistics departments as to what computational skills other scientific disciplines, such as physics, biology, and chemistry, believe to be the most important for students to grasp.

Computational Training for Graduate Science Majors

Researchers in the Department of Biological and Biomedical Sciences at Harvard have developed an intensive course that introduced graduate students to the "fundamentals of programming, statistics, and image and data analysis through the use of MATLAB." (Stefan et al. 2015) These courses are framed not only with the goal of students developing programming skills, but also emphasizing students learning how to algorithmically reason through a computational problem. The structure of the 50-hour course dedicates the first two days to an introduction to programming using MATLAB, where students learn a variety of topics, including creating variables, performing basic variable operations, indexing, logicals, functions, conditionals, and loops. These courses are given twice a year, once prior to the start of the school year as new graduate students are attending orientation, and a second time for "students who realize the need for such training later in their studies."(Gutlerner and Van Vactor 2013) In introducing beginning graduate students to these concepts, researchers hope to lower the computational barrier for students taking courses, empower student to learn computational tools on their own, as well as allow other courses to "build upon this foundation and integrate quantitative methods throughout the curriculum" (Stefan et al. 2015).

Graduate level terminal statistics courses, such as Methods of Data Analysis I and II, are taken by graduate Environmental Science students across the country, potentially acting as the final computational training students receive prior to performing independent research. These such courses provide a natural extension of the research on computational training and thinking of graduate students in the biological sciences, as both provide the computational training used by graduate students in their independent research. Environmental Science and biological science students have similar computational burdens, with computational expectations often being placed on them with little to no training. It is the intention of this study to close the gap in the research and understanding of the resources Environmental Science graduate students invoke when reasoning through computational applications relevant to their field.

Methodology

For this study, a pragmatic phenomenological approach is appropriate as the intention is to understand and describe common experiences in computational thinking and abilities for Environmental Science graduate students in applying their computational skills and understandings to applications in their field. The focus of this study, on the actions taken by students in the process of computationally reasoning through an application, lends itself naturally to a pragmatic framework. This framework allows for a focus on the process of finding a working solution, allowing for varied solutions rather than a single solution (Creswell 2013).

For this study we focus on the factors that impact how Environmental Science graduate students' gain computational knowledge and the ability to reason through applications in their disciplines. Unlike typical definitions of computational knowledge and abilities, which focus on a student's understanding and fluency of computer program, we chose to also include knowledge of data structures, the ability to reason through problems with a given set of tools, as well as knowledge of resources that could provide assistance in solving a particular problem (**???**).

Participants

Students were selected from the Methods of Data Analysis II course in spring of 2017. These students were sampled following their spring break, nearly halfway through the course. Only graduate students from Environmental Science departments were considered. These students were taking the course as a coursework requirement for their respective Master's programs.

Students were requested to complete a survey detailing their previous courses, statistics, computer science, describe the computer languages they were familiar with, and outline their independent research experience. A total of 8 graduate Environmental Science students were enrolled in this course in the spring of 2017, all of which completed the survey. All 8 of these students were requested to participate in the study, of which five agreed. All of the students who agreed to participate identified as female, and exhibited the following characteristics,

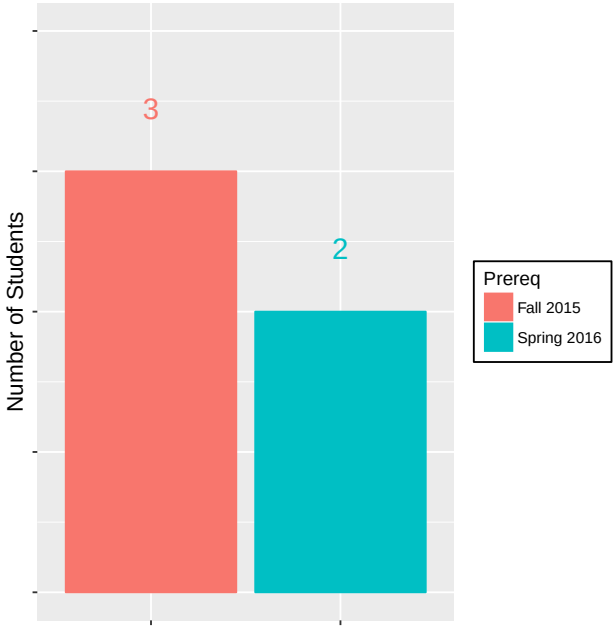
- all had taken Methods of Data Analysis I in the last 2 years,
- all had a variety of programming backgrounds, and
- all had a variety of levels of independent research experience.

Table 1 and Figure 1 describe the demographics of the study participants.

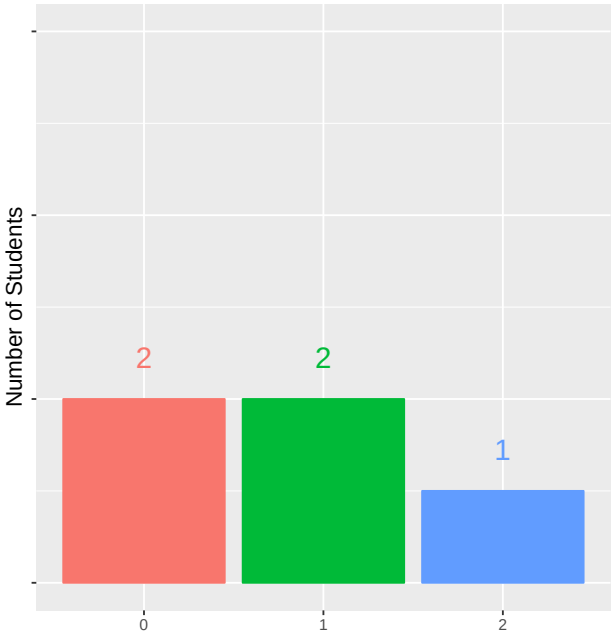
Name (pseudonym)	**Degree**	**Program**
Beth	Master's	Animal Range Science
Catherine	Master's	Environmental Science
Kelly	Master's	Ecology
Robin	Doctorate	Environmental Science
Stephanie	Master's	Environmental Science

Table 1: Sample Description

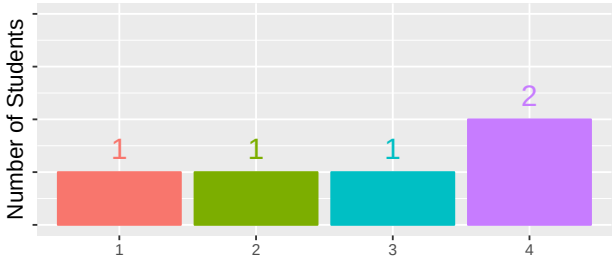
Survey Participant Methods of Data Analysis I



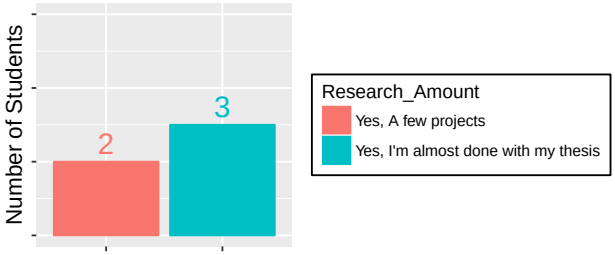
Survey Participant CS Courses



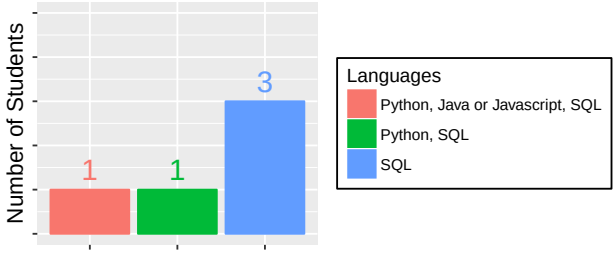
Survey Participant Stat Courses



Survey Participant Independent Research



Survey Participant CS Languages



Data Collection

Following the preliminary survey, students who agreed to be interviewed were asked to describe their coursework, where and how they acquired their computational knowledge, and discuss their experiences in acquiring this knowledge. This interview was modified from surveys administered by researchers investigating computational thinking at Harvard using Scratch, which provided a rich rubric of assessing students' experiences in performing computational applications (**???**).

Following the interview, students were asked to complete an ecological application of statistical computing. These problems assessed students' abilities to both statistically and computationally reason through applications, provide descriptions of how they computationally think through the problems, as well as outline the any gaps noticeable in students' ability to transfer their computational knowledge to applications. Computational problems were developed in collaboration with Statistics faculty at the university, using data from the Blackfoot River, collected by Montana Fish, Wildlife, and Parks in years 1989 to 2006. The analysis in this paper is based on the six interviews; students' abilities to reason through computational applications will be used in future analyses.

Data Analysis

Interviews and survey responses for each participant were transcribed, with participants' names removed and pseudonyms given. The interviews were analyzed so as to describe the thought process of each student working through the application, describing the concepts they used to complete each task, why they believed these to be the correct concepts to use, and how they acquired their knowledge of these concepts. Themes specific to each individual's computational thinking processes were then described, including, but not limited to, synthesis of experiences, coursework, and learned skills. With these individual themes, we were able to compare commonalities that emerged across individuals.

Participants were provided with an itemized detail of how they completed the problem and the transcription of their interview. This allowed participants to check for accuracy of their statements and the interviewer's description of their thought processes in completing the problems. The authenticity of the study, its ability to capture the true experiences of students' abilities to think through computational problems, is enhanced with the lack of researcher engagement with students prior to their participation in the study. This ensured that no student felt more comfortable in the interview environment, working through the application, than any other student.

Results

Three main themes emerged from the interviews during the data analysis process. The names of these themes, their characteristics, and sample responses are described in this section.

The first theme describes how peer support has impacted participants' abilities to transfer computational skills to applications in their field. The second theme provides participants' descriptions of how a singular graduate student or faculty influenced their computational abilities. The third theme recounts how participants' independent research experiences have affected their computational abilities. In addition, their research experiences helped to emphasize the importance of computational literacy to their field. In the following section, themes are delineated with quotations from participants to ensure authenticity of descriptions of their experiences.

Peer Support

All participants spoke of the support they have received from fellow graduate students when performing computational tasks. The students described how, when they are unsure of how to complete a computational

task for their research, they turn to fellow graduate students for help. Participants described instances when the computational tasks required of them were beyond their current knowledge or occasions when they had attempted to complete a problem with all of their knowledge and sought out help from a fellow graduate student. For example, Kelly, an Animal Range Science Master's student, shared that when she reaches a point in coding when she doesn't know how to do something she turns to one of her lab-mates.

> I've been to a point where I didn't know how to do something with my knowledge or what I can find online, and then I'll go to one of my lab-mates.

Catherine, a Master's student in Environmental Science, spoke of the expectations of her advisers that the computational problems she was being asked to perform were "easy, since she had all the information." However, she has had numerous experiences where she did not have the knowledge necessary to perform the task or she was missing "little caveats" that kept her from fully being able to perform the tasks. When faced with these problems, she "reached out to previous students that had taken the course." Robin, a doctoral student in Environmental Science, reiterated Catherine's experiences, describing how she reached out to other graduate students in other labs for help with computational problems. Stephanie, also a doctoral student in Environmental Science, described how when she is faced with computational problems beyond her knowledge she has never been forced to "go beyond talking to her lab-mates" for assistance.

However, Kelly has also had negative experiences when seeking computational assistance from graduate students not of close proximity to her.

> When I'm struggling with something and I go to other grad students, they'll say 'I did this the other day. I'll send you my code.' I've found most of the time I don't understand what they've done enough to plug in what I want and make it work. There have been a few times when making tables and plots and someone sends me their code and I can just plug in my data and it works just fine. I've had less success with that.

Singular Consultant

When describing whom they seek out for computational help, every participant described an all-knowing past or current graduate student whom they seek out for computational assistance. These figures serve as a singular consultant, with whom these students have had the "best," most productive, experiences in finding solutions to computational problems that have arisen. For Beth, this singular consultant comes in the form of a past graduate student from Animal Range Sciences who was hired to help faculty complete projects.

> We have a guy who used to be a student in our department and then he was hired on again to help finish some projects, but he got his Master's in Statistics. He is very helpful with [pointing out what's wrong with your code]. He's very good with code and if I have a quick question he can answer it.

For Kelly, another graduate student on the same project as her serves as this consultant. She described computational problems she has encountered in her thesis, when she turned to this particular graduate student for help, adding that other graduate students in their department also use this person as a consultant for their computational problems. "The other grad student on this project is so well versed in `R` that he's unofficially become the person that people go to with questions." Through her computational struggles, Catherine found assistance from previous graduate students from the department, but she found the most assistance from a previous graduate student "who had left the department and was off professionally somewhere else, but he still took the time to help walk me through [my code]."

One participant, Stephanie, a Environmental Science Doctoral student, serves as the computational consultant for the many members of the Environmental Science department. With her experiences teaching herself `R`, she is able to "explain code in a way that makes sense," says Robin a fellow Environmental Science doctoral student who often seeks out help from Stephanie. With an adviser from a computational background and a project which performs computer modeling, she "has to learn code." Additionally, her laboratory often works in collaboration with faculty from the computer science department, where she and her lab-mates are taught computer science coding practices and jargon. "Stephanie has gotten good at teaching it, because

everyone on our floor is like 'I can't do this, Stephanie help me,' " says Robin. Stephanie stated that graduate students seek her assistance "daily" or "at minimum two to three times a week." In contrast, when Stephanie experiences difficulty in performing computational tasks, she finds solace in her lab-mates and ultimately, when necessary, with her adviser.

> My entire lab works in the same room and my adviser's door is always open. So if someone is having a major issue, whoever is in the room can hear that. If [my adviser] hears me ask [lab-mate] how to do something and he knows how, he just shouts how to do it. So it's a very group oriented dynamic. I've never had to go beyond the people in my lab.

Independent Research Experience

The third theme was the computational knowledge students acquired while participating in independent research. Involvement in independent research helped students to take their course knowledge and transfer it to applications, seeing the messiness of non-classroom applications. These experiences came predominantly in the form of working as a research assistant prior to entering graduate school, collaborating on a project in the first year of graduate school, and performing research for a Master's thesis, or ultimately a doctoral dissertation. Catherine, who still faces everyday computational struggles, attributes the majority of her application specific computational knowledge to her experiences in independent research. She emphasized the importance of understanding how to work in a computing environment, such as `R`, which she learned from performing research, before she could begin to transfer the statistical knowledge she learned in the classroom.

> What I struggled with is [Methods of Data Analysis I] covers theory really well, but since I was new I spent most of my time trying to figure out how to apply that theory in [`R`]. And even now I struggle transferring from `R` into actual statistical theory, when I'm writing my thesis. The way I had to approach it was I had to learn the `R` first, then I was able to look back on what I had actually done, in order to learn the statistics.

Kelly described her experiences with data management for her Master's thesis as producing the most substantial contributions to her computational abilities. Often she attributed her intuition for solving computational problems to experiences she had "merging data sets" and learning to use conditional statements for her project. She emphasized the importance of her classroom Methods of Data Analysis knowledge in understanding "what statistical method to use," but for becoming more computationally fluent she attributes that to her research experiences. "The data management stuff comes from independent research, trial and error, getting myself through." Similar sentiments were voiced by Beth, with the majority of her computational knowledge stemming from her independent research. With the recommendation of her adviser she taught herself how to create an `Access` database to store her data. In storing her project in this manner, she was able to learn important concepts in data structures, subsetting data "using qualifiers and criteria," sorting data, all using SQL statements.

Discussion

The intention of this study is to describe and understand what factors impact how Environmental Science graduate students gain computational knowledge and the ability to reason through applications in their disciplines. Students who participated in the study both described their experiences with computational thinking and their ability to reason through applications related to their field.

Three themes arose from the study, characterizing the factors with the greatest impact on the participants' experiences in acquiring computational knowledge and abilities, as related to applications in their fields. First, participants voiced the importance of their experiences performing independent research as a substantial influence on their abilities to reason through and perform computational problems. Through independent research, the participants were able to play with real-world data and applications outside of what they saw in the classroom. These experiences also opened the door to the unease that comes when one is asked to

perform computational tasks beyond one's knowledge. In these circumstances, the participants discussed their reliance on peers and singular consultants to aid them in accomplishing computational tasks.

The second and third themes both focus on the participants' acquisition of computational knowledge and skills, as well as how they cope when faced with computational expectations beyond their ability. The theme of peer support was initially discussed by the participants in their interviews as a mechanism they use when their "code doesn't run" or when they are asked (or need) to do something outside of their current computational understandings. However, this theme continued to emerge as the participants worked through the computational problems, often attributing their knowledge of computational procedures to a friend or fellow graduate student helping them "do it with their data." These peers offer an avenue for students to seek help, often voiced to be more comfortable than asking an adviser, where participants described both the fear of asking and "feeling dumb," as well as being "brushed off" because their adviser thought they should "be able to figure out how to do it."

In a direct connection to the participants' discomfort in asking for help from an adviser, the third theme of a singular consultant emerged. These singular consultants serve as an all-knowing individual, from whom the participants have either had the "best" experiences with, where the individual spends the necessary time to explain the concepts, or the consultant has always been capable of providing the participant with an answer to their problem. These figures serve a similar role to peers, where the participants are both able to seek computational help and acquire new computational skills and understandings through their interactions. However, as opposed to the help participants received from peers, the students never voiced any negative experiences when seeking help from these consultants.

The second purpose of this study was to describe how these themes impact students' computational abilities across different backgrounds. The theme of research experiences, with its overall positive tone by participants, produced different experiences for students with fewer computational skills and understandings than students with more. The frustrations of simple tasks, such as subsetting data or removing NA's, were felt by the participants who had completed a Bachelor's without any computational elements to their coursework, while those who were exposed to small amounts of computing in their undergraduate coursework, such as a general computer science courses, a GIS course, or experience with `Access` databases, were able to begin computational tasks in their research walking and not crawling.

The largest difference in the impacts of a factor between computational skill groups, came in the theme of a singular consultant. One participant, Stephanie, who entered graduate school after completing a year's work as a research assistant, working in `R`, instead serves as the computational consultant for the Environmental Science department. She still described the theme of seeking help from her peers, predominantly her lab-mates, but, for her, the singular consultant was her adviser. Potentially due to her more substantial computational background, Stephanie voiced that she feels less of a power difference than her peers when seeking help from her adviser.

Conclusion & Implications

The findings of this study, better inform a concept map of how students acquire the skills necessary to perform computational applications in their fields. With the emergent themes describing where students are acquiring these necessary skills, the concept map in Figure 1 reflects these changes.

The computational aspects of this concept map were initially thought to exhibit Bloom's taxonomy. This theory describes the process of learning as a hierarchy of understanding, where students first understand low-level concepts, such as logical statements, using built in `R` functions, making comments in their code, and trouble-shooting error messages, before thinking about them in more complex ways, and learning higher-level concepts, such as user defined functions, defining classes of objects, and implementing loops and conditional statements. We found, however, that the participants do not learn the concepts related to accomplishing computational applications in a hierarchy. The importance of background remains, however. For these

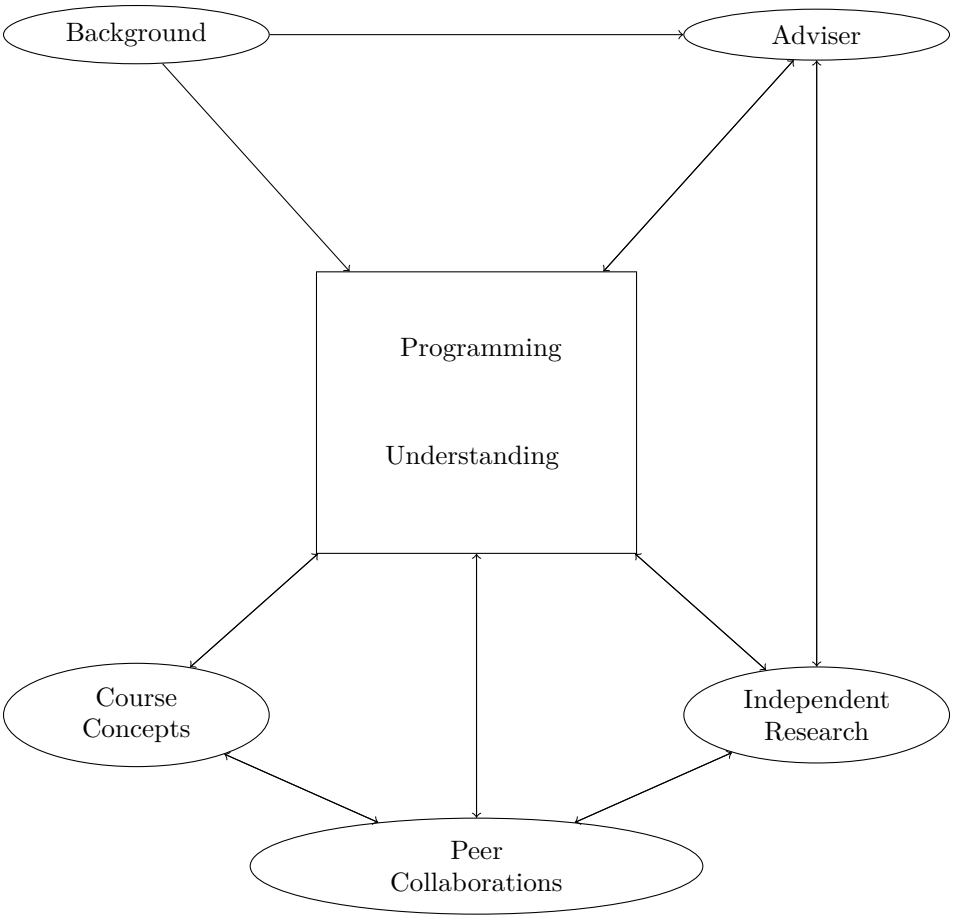


Figure 1: Computational Knowledge Acquisition Concept Map

participants, the elements of background that proved to be of the most help were both undergraduate statistics courses and pre-graduate research. The adviser plays an important role in students acquiring the computational knowledge necessary to perform applications, by both emphasizing the importance of these skills, as well as introductions (or recommendations) for students to store their data in an `Access` database. The ability of many participants to understand both data structures and sorting or filtering data was largely attributed to their experiences working with these types of databases. Additionally, this study found that, although they are not often used, advisers are viewed as an accessible way for students to better understand the computation necessary for their independent research projects, which overall contributes to better computational understanding and skills for these students.

At the center of the map we find programming understanding the bottleneck described by all students in their abilities to successfully accomplish computational applications in their fields. This understanding is informed, albeit weakly, by students' backgrounds and their advisers. However, the majority of the students' understanding comes from their course work, independent research, and collaboration with their peers. The programming understandings that these students attributed to their coursework were primarily the a fore mentioned low-level concepts. These concepts were found to not directly inform a student's research, but instead the concepts were understood through the use of peer interaction prior to implementation in their own research. The programming understandings informed by a student's independent research, in conjunction with peer collaboration, were described by participants to be largely the high-level concepts described previously. These understandings stem from the need to find computational solutions to the applications they are currently working with. These solutions often require higher level programming concepts, but they are accomplished without students learning the lower-level concepts at work within the solutions.

Implications

This study better informs the faculty and staff at this institution of the computational needs of graduate students in the Environmental Science fields. In the discussion of how students are acquiring the computational understandings necessary to successfully perform field related applications, we are able to see how both instruction and learning could be improved. It is possible that students are acquiring computational skills in the classroom, however there is also the possibility that the concepts being taught are not at a low enough level for students to grasp them independent of their peers. To better inform faculty in these departments, a thorough investigation of both the coursework (syllabus) and structure (lecture, laboratory, etc.) of courses completed by these participants could be performed. This will allow for a discussion of how to best teach these computational concepts, so that students leave the classroom with understandings they can implement immediately in their own research.

References

Cortina, T. 2007. "An Introduction to Computer Science for Non-Majors Using Principles of Computation." In *Technical Symposium with Computer Science Education*, 39:218–22. ACM. http://dl.acm.org/citation.cfm?id=1227387.

Creswell, J. 2013. *Qualitative Inquiry & Research Design*. 3rd ed. Thousand Oaks, California: Sage Publications.

Dodds, Zachary, Ran Libeskind-Hadas, Christine Alvarado, and Geoff Kuenning. 2008. "Evaluating a Breadth-First CS 1 for Scientists." In *Technical Symposium with Computer Science Education*, 266–70. Portland, OR: ACM. http://doi.acm.org/10.1145/1352135.1352229.

Gutlerner, Johanna L., and David Van Vactor. 2013. "Catalyzing Curriculum Evolution in Graduate Science Education." *Cell* 153 (4): 731–36. doi:10.1016/j.cell.2013.04.027.

Hambrusch, Susanne, Christoph Hoffmann, John Korb, Mark Haugan, and Antony Hosking. 2009. "A Multidisciplinary Approach Towards Computational Thinking for Science Majors." In, 41:183–87. http://doi.acm.org/10.1145/1539024.1508931.

Rubinstein, A, and B Chor. 2014. "Computational Thinking in Life Science Education." *PLoS Computational Biology* 10 (11). http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003897.

Stefan, Melanie I., Johanna L. Gutlerner, Richard T. Born, and Michael Springer. 2015. "The Quantitative Methods Boot Camp: Teaching Quantitative Thinking and Computing Skills to Graduate Students in the Life Sciences." *PLOS Computational Biology* 11 (4): 1–12. doi:10.1371/journal.pcbi.1004208.

Wilson, Greg, Christine Alvarado, Jennifer Campbell, Rubin Landau, and Robert Sedgewich. 2008. "CS-1 for Scientists." In *Technical Symposium with Computer Science Education*, 36–37. Portland, OR: ACM. http://dl.acm.org/citation.cfm?id=1352151.

Wing, Jeannette. 2006. "Computational Thinking." *Communications of ACM* 49 (3): 33–35.