

# Teaching Statement

---

I have extensive classroom experience, both in computational biology and in other fields within biology. As a graduate student, I was a teaching assistant in many large classes, some with more than one hundred students. This was very challenging to me, as a liberal arts college graduate. I believe that students learn best when they are encouraged to discuss course material in small groups, and to engage in hands-on practice where appropriate. In these large courses, I preferred to minimize my lecture time, and increase the amount of small group activities and discussions. My strategy was successful, and I earned the College of Natural Sciences Outstanding Teaching Assistant award at the University of Texas at Austin for my work in an upper-division evolution course. Despite this success, I would very much prefer to be in an undergraduate-focused environment where I can get to know my students better on a personal level, tailor the curriculum to their interests, and help them develop the skills that will allow them to thrive in other courses, graduate programs or other careers.

I have also taught a biological computation course that I co-developed with my labmate. This course was enrolled at capacity both times it was offered, and was well-reviewed by the students who took it. My success in teaching this course led me to becoming an instructor with the Software Carpentry Foundation, teaching Python programming with the University of Texas at Austin Statistics and Data Science department, and to my invitation to the What to Teach Biologists About Computation meeting in Annapolis, MD in 2013. I very much enjoyed teaching this course, and would like to adapt the course for undergraduate learners. I would also like to teach general biology courses, and develop computational complements to standard biology course work.

## Evidence-Based Pedagogy

In my teaching, I value insights from educational research, as well as those from evaluations by my students. When I co-developed my own curriculum in computational biology, I worked with smaller groups of students and emphasized a hands-on approach to learning. My goal with this course was to equip my fellow biologists with computational skills that can be used in research, but also translate to industry or other jobs. In my course, I taught the basics of literate programming, before moving on to specific biological software and analyses. When teaching, I like to break up class periods between reviewing material, answering student questions on material from the previous class day, covering new material and doing hands-on exercises. In response to student evaluations and research on memory and cognition, I no longer to speak for more than 20 minutes before assigning a hands-on exercise. Student evaluations from the first iteration of the course had asked for more hands-on exercise time, and the autonomy provided by teaching my own course allowed me to act on

this feedback. Current research<sup>1</sup> shows most learners can hold less than a half-dozen ideas in memory at a time. In a class period, I may touch on a dozen or more ideas. In response to these findings, I have practiced more effective splitting of class between lecture time and exercise time.

I am also enrolled in courses to develop my teaching via ISU's Center for Excellence in Teaching and Learning. I am particularly interested in learning more about formative assessment and using peer review effectively in the classroom.

## Hands-On Learning

Undergraduate-focused learning environments also enable active learning. In teaching computational biology and bioinformatics, I use a tool called the iPython Notebook. This tool allows for code fragments to be annotated with notes, stores the output of commands, and allows the user to call external bioinformatics programs. This tool allows me to write programs and do analyses collaboratively with students in class. This tactic, called 'live coding', is more engaging than copying and pasting pre-written commands, as the students can be active participants in the process. The students can also experiment with their own ideas in these notebooks, then email them to me, and I can annotate them with feedback. In this way, students can structure their in-class learning and out-of-class homework as one continuous experiment in bioinformatics.

Particularly when covering the basics of a science, I also encourage students to develop an independent project, as learning the basics of a discipline can be fairly abstract. Having a biological project in which students have a vested interest allows students to see concrete results from their intellectual labor. Research by Upchurch and Williams<sup>2</sup> also supports that having students work together on a hands-on project is very effective for learning. I recently co-developed a week-long immersive research course on phylogenetic methods. In this course, I emphasized project-based learning. Learners from this course have already begun to incorporate the skills they learned in their own bioinformatic projects<sup>3</sup>.

As a biologist with wide-ranging interests, I find that having basic computation skills allows me to explore many different types of questions, and this freedom is something I'd like to impart on students. One way to do this is to develop computational complements to courses that are not explicitly computational. I would be very excited to develop computational labs to accompany traditional classes. For example, in an evolutionary biology course, laboratory exercises could involve querying publicly-available molecular sequence data on GenBank to build a phylogenetic tree. In a genetics course, students could create small simulations to look at allele frequencies under Hardy-Weinberg equilibrium. In their careers

---

<sup>1</sup> Rouder, J. N., Morey, R. D., Morey, C. C., and Cowan, N. 2011. How to Measure Working Memory Capacity in the Change Detection Paradigm. *Psychonomic Bulletin & Review* 18: 324-330

<sup>2</sup>Williams, L. and Upchurch, R. 2001. In Support of Student Pair Programming. 2001 SIGCSE Conference on Computer Science Education, Charlotte, NC: 327-331

<sup>3</sup>Warren, D. 2016. Age-overlap correlation tests and building an enmtools.clade object. ENM Tools Blog

moving forward, many students will be asking interdisciplinary questions about biology. Courses that have interdisciplinary components can help prepare them to do exactly that.

## A Wholistic Approach to Education

I also think it is important to take teaching out of the classroom. I would like to develop a computational biology speaker series. In the course I co-developed, I would invite members of the University of Texas research community to come speak about how they use computation in their biological research. This is a valuable glimpse at the ‘real world’ of computational biology and bioinformatic careers, and builds communities in which students know to whom they can speak about careers and opportunities. A similar series would be very useful to undergraduate students, and could be expanded to address undergraduate-specific issues, such as finding a good computational biology graduate program, funding for computational biology and working in industry as a computational biologist. I have connections at many universities; if there is a speaker budget, I could arrange for visits from computational biologists. If there is no budget, I can use the experience with effective distance learning via webconferencing I gained as a Software Carpentry instructor to involve remote computational biologists.

Even though teaching is not a part of my current postdoctoral contract, I continue to work with the Software Carpentry Foundation (SWC) and on workshops as part of my own NSF postdoctoral fellowship activities. With SWC, I teach short, immersive workshops aimed at fostering specific computing skills in interested parties, mostly at the novice level. I also maintain the Data Carpentry python educational materials, which use hands-on exercises with real data to help biologists develop programming skills. In my own fellowship activities, I recently taught a week-long course on phylogenetic methods development. The audience for this course was scientists who already have some degree of programming knowledge. The needs of this group of learners is very different than learners who have never interacted with programming before. This course could be combined with my experiences with novice-level instruction to form the basis of a year-long, immersive computational biology research methods course.

July 31, 2018