Teaching Statement

While I admittedly have limited experience with course creation and lecturing, I am genuinely eager and excited for the opportunity to teach. Throughout my education, I have taken note of what works and what doesn't, what is engaging and what is boring, and what builds lasting and practical knowledge.

Teaching Philosophy

I believe that **learning and teaching should be fun and exciting**. At Brown University's Computer Science department, where I earned my Bachelor's degree, there is an undergrad teaching assistant program which I participated in and loved for six of my eight semesters. We had a culture of making entertaining skits and interactive games that engage the class while teaching them core concepts. I recall one of my favorite professors, Pascal Von Hentenryk, having a sword fight with one of the TAs as part of a very memorable first lecture for Introduction to Computer Systems, one of the most foundational courses in the field. In that same lecture, he got out a bucket of sand and a tin of coffee beans and said in that course we would turn sand (silicon) into coffee (java, the programming language). In my robotics course, in order to demonstrate particle filter mapping, the professor had everyone stand up and act as a particle in the algorithm. In one of the more dry courses, Theory of Computation, Eugene Charniak--a mathematician always dressed in tweed and a bowtie--told us jokes between topics.

I also believe that **learning the history and evolution of a field is important**. It gives a contextualization of how we got to where we are, gives examples of how real scientific progress works, and gives the students a sense of connection to the field.

Another principle I employ in my own learning and will employ in my teaching, is **"learn by doing"**. I will create projects in which, by design, the students will not initially have all of the information they need to complete them. I will point them to the necessary resources they need to find that information and allow them to build the knowledge for themselves. In my experience this creates much more long-term and practical comprehension than any other learning technique.

Mentorship Philosophy

Of course I enjoy making innovative advances, technical breakthroughs, and visually appealing and insightful data visualizations. But what I enjoy more than that is sharing them with others and having an interchange about them. **It is the people I work with and the relationships we develop that give me the most joy from my work.** Equally, I enjoy sharing the fundamentals of my interdisciplinary field as well as each field individually with others. I am passionate about the basics because I use them everyday and know just how powerful simple ideas are when combined in novel ways. I am especially passionate about teaching practical skills and linking abstract topics to the real world through examples. To this end, I have created a YouTube channel named Data Democracy with the slogan "We bridge the gap between what you learn in the classroom and the skills you need to thrive in industry and academia." which seeks to teach these concepts. An example video can be seen at **https://tinyurl.com/yxhev3qt.**

I supervised a PhD student, Yichen Wang, during her rotation in Mara Lawniczak's lab due to its close relation to my souporcell project. This was her first rotation and first time living in

another country and she was clearly nervous. She would often not speak up in meetings with PIs even when she had the most expertise on a subject of anyone in the room. I encouraged her to express her ideas confidently and ask any questions she had. Throughout the rotation she gained confidence and did great work. She discovered that the biggest cause of false positive variant calls in single cell RNAseq data came from post transcriptional modifications which could be filtered out using a database of known modification sites taking variant calling specificity in scRNAseq from 75% to over 90% with almost no reduction in sensitivity. She then followed up on this discovery assessing the cell variation in post transcriptional modifications and found that these did not vary significantly across cell types within a single tissue. While this was a negative result, it was a surprising negative result and her use of statistics and visualization techniques to show it were impressive.

My computer science and biotech industry experience and medical background will not only give me a unique perspective in research and course design, but they will additionally be a valuable resource in advising undergraduate students who seek to follow any of these career paths.

Teaching interests

In my primary field of computational biology, a core competency is data analysis. But while the techniques of basic programming, statistical tests, and data visualization are fairly easily taught, in reality there is a darker side. What happens when your data doesn't conform to your assumptions? Hint--it never does. This is where we get into the subtopic I call data debugging. This is the process of figuring out why your data is not what you expect and what caused that difference. In this course, I would first teach the basics of various data analysis techniques, but then give students datasets with some fatal flaw explainable by a biological mechanism, instrument error, experimental design flaw, or simple human failure. In some circumstances this could be removed and analysis would continue. In others, the students would need to submit the flaw to get the "repeated and fixed experiment" data.

I have several other course ideas, but this statement is already too long. In addition to the course outlined above, I would be comfortable teaching the following courses.

In computer science I would feel comfortable teaching any courses outside of programming languages, operating systems, and graphics. Some might include

Introductory Programming, Machine Learning, Data Structures, Algorithms, Computational Biology, Intro to Computer Systems, Discrete Mathematics, Software Engineering, Scientific computing, Data science, Probabilistic algorithms, Probabilistic Machine Learning, Combinatorial optimization, Statistical modeling, Robotics, and the list goes on and on.

I am also interested in designing cross-department courses between CS, biology, and statistics. In a cross department appointment or cross department courses I would be comfortable teaching the following.

Computer science for non CS majors, Introductory Probability Theory, Bioinformatics, Genetics, Genomics, Biostatistics, Physiology, Cell Biology, Data Analysis