Teaching Statement

Saket Navlakha
School of Computer Science
Carnegie Mellon University, Pittsburgh, PA

I enjoy teaching and have had a variety of teaching experiences as a post-doctoral researcher at CMU, as a graduate student at UMD, and as an undergraduate at Cornell.

Experience as a course instructor. My most recent experience has been as a co-instructor with my supervisor (Ziv Bar-Joseph) at CMU for a new graduate course we developed called Algorithms in Nature. This course is designed to introduce students to algorithmic ideas employed by biological systems. I have independently prepared and given roughly half the lectures for this course. These lectures typically come in two parts: the first introduces a fundamental computer science problem (e.g. matrix factorization, network design) and the second presents the same problem from the perspective of some biological process or system (e.g. the brain).

Teaching philosophy. While lecturing, my philosophy is to present a logical story, to create an environment where questions and objections are welcome, and to encourage personal investigation. Because of the broad spectrum of areas covered in this course, I have had to synthesize topics to not only provide students with enough breadth to appreciate the scope of the computational problem, but also enough depth so they can appreciate the challenges and corresponding biological solutions. I also try to vary the medium of delivery from slides with plenty of figures, to board examples, videos and movies, pseudocode, and interactive web applications all as a means to repeatedly drive home key messages. I have also designed problem sets for this course, which are meant to challenge students to apply core concepts learned in class to novel situations. I have also found that teaching and research are related — one feeding into the other — because teaching has forced me to re-think and explain core concepts and how they fit into the overall picture of what is known and unknown.

Experience as a teaching assistant. At UMD, I was a TA for 4 semesters for two different classes: intermediate object oriented programming and an introductory course on web programming for (mostly) non-majors. For the former course, I led recitations twice a week and taught core computer science concepts, including object oriented programming principles, software design and testing, algorithm complexity, and data structures to computer science majors. This was a fun class because I provided students an early glimpse into the excitement, power, and basic methodologies of computer science. For the latter class, I graded and held office hours to aid with issues in CSS, HTML, Javascript, and multimedia programming. This class was interesting because students had little-to-no programming experience, which required patient and careful guidance for those grappling with logical programming concepts. As an undergraduate at Cornell, I was a TA for one course (intermediate programming and data structures), while also co-leading weekly recitations.

The future. I can teach a variety of CS courses, and I would be happy to teach courses in computational and systems biology, data mining, and graph algorithms. I also hope to teach a graduate or upper-level undergraduate course on algorithmic biology (akin to Algorithms in Nature) and am also interested in teaching undergraduate courses in object oriented programming and algorithms.