TEACHING STATEMENT

SCOTT SHERRILL-MIX

The expansion of high-throughput methods have made the analysis of large biological datasets an essential skill for many biologists. In my teaching, I enjoy providing students the bioinformatic tools necessary to effectively deal with these computational challenges. Unfortunately, it seems that some biology-focused students view programming and statistics as uninteresting prerequisites to be rote memorized. However, I think that bioinformatic topics are much better received and retained if the student both develops an understanding of the underlying algorithms and is able to apply their knowledge to solve relevant problems.

In my experience, many students view statistics and programming as a black box. To develop skills in bioinformatics, I think it essential to open the box and understand at least the basics of the algorithms. To help students reach this understanding, one course I would like to develop would focus on self implementation of common bioinformatic algorithms. Such a foundational course would demystify bioinformatics while offering the opportunity for teaching best practices such as documentation, software testing, source control and reproducible research.

When presented with algorithms or equations, it seems that students can feel detached from any biological context. I hope to avoid this disconnect by having students implement solutions to relevant problems and examples. My most memorable classes as a student used real world examples and applied problems to stimulate student interest. For example, these classes had students develop a model to separate out elite baseball players or classify the gender of portraits to strongly reinforce lectures on the theory of Bayesian statistics and machine learning. By following a similar pattern in my classes, I hope students will see immediate progress in their own implementations and retain interest in what they might otherwise have considered a dry subject.

Another way I hope to increase student interest is to add a small amount of competition. During my time as a graduate student, I organized several student "hackathons" programming competitions and saw their effectiveness in providing experience in both collaboration and programming. Adding a hackathon component to my classes would allow students put theory into practice while allowing direct enforcement of best practices by requiring submission through version control for automated testing and evaluation.

In my teaching, I hope to prepare students for the analysis of biological data by providing a solid foundation in algorithmic understanding and analytic skills with direct applications to real world problems. This will allow students to view data analysis less as twiddling knobs and more as an interesting scientific challenge integral to their study of biology.