Overview. My interest in the application of computational methods to the physical sciences initially led me to work in computational biology. My experiences have since led me to become more interested in working on the computational tools that make such applications possible.

Tracing the evolution of presynaptic receptors *(Summer 2006, Computational Biology Initiative, Harvard Medical School)*. I worked with Dr. Dennis Wall, director of the Computational Biology Initiative, and Professor David Van Vactor, associate professor of cell biology at Harvard Medical School, to track domain evolution within presynaptic receptors and to test hypotheses about the coevolution of the receptors.

The *presynaptic region* is the area at the end of a neuron's axon before the synapse, which is the space between two neurons through which neurotransmitters are released. The presynaptic and postsynaptic regions contain receptors playing various roles in communication between neurons. *Presynaptic receptors* are particularly important during the development of the nervous because messages conveyed to the receptors determine the direction and duration of axon growth. We were interested in gaining insight to the role of these presynaptic receptors by examining their evolution, with the ultimate goal of determining the minimal requirements for a functional nervous system. Since these receptors were made up of *domains* corresponding to patterns in the gene sequences, we could formulate hypotheses using computational methods to examine organism sequences. In our analysis we were particularly interested in the evolutionary boundary between *cnidaria* (hydra) and *porifera* (sponge): while sponges are not known to have central nervous systems, cnidarians are known to have fully functioning central nervous systems. Understanding the role of these receptors and their interactions is useful because many of these receptors are implicated in autism, a little-understood disease.

My initial goal was to trace the evolution of LAR and Liprin-α, two presynaptic receptors thought to play a role in the development of the central nervous system, to their points of emergence along the evolutionary time line. By mid-summer I had successfully combined existing methods of statistical sequence alignment and domain identification to develop a method for tracing the evolution of the receptor domains. After completing my analyses of LAR and Liprin-α, I applied the method to the larger set of presynaptic receptors. I presented my procedures and results at UC Santa Barbara at a mini-symposium of scientists interested in the evolution of the central nervous system. After being delayed waiting for the release of a clean *nematostella* (sponge) genome sequence, Dr. Wall and I are now preparing our results for publication.

Though my interests have moved away from computational biology, my experience at CBI was formative. From working in a multi-disciplinary area, I learned that the most relevant problems are not self-contained in a single field and often involve the expertise of scientists of different backgrounds. Observing how others approached problems taught me how to ask questions and how to develop methods for answering them. Because I was given a good deal of independence, I learned to make my own decisions about how to proceed, to communicate my results often and effectively to my collaborators, and to ask for feedback and assistance when it was necessary. Since I previously had little background in evolutionary biology or statistical methods in computational biology, I also learned to take advantage of textbooks and scientific papers to learn what was necessary as quickly as possible.

Developing a decentralized distributed computing model *(Harvard University, Fall 2006)*. As I moved away from my interests in biology to my interest in computational tools, I became interested in the uses of biology to inspire computing. As the final project for Professor Radhika Nagpal's graduate-level course in biologically-inspired distributed systems I did a project with a partner and in collaboration with Dr. William Butera[1] at IBM investigating how to apply amorphous computing concepts to decentralizing Google's MapReduce paradigm for distributed computing. As processor manufacturers have begun increasing the number of processors rather than the number of transistors to keep up with Moore's Law, it becomes necessary to find good ways of distributing tasks among the processors.

Google's MapReduce algorithm, which automates the organization of labor among a cluster of machines, suggests a solution to this problem. MapReduce allows the user to define the problem in terms of a **`map`** function, which specifies how the problem should be solved on arbitrary chunks of the input, and a **`reduce`** function, which specifies how the **`map`** results should be combined into a single solution. The one flaw with MapReduce, we pointed out, was that it required the existence of a centralized master node. As the number of computing agents increases, the bottleneck of information flow through the master would be significant. Thus, we proposed applying an amorphous computing method, which is inspired by leaderless systems such as bacteria colonies, in which communication occurs by the passing of gradients through the system. Our solution eliminated the role of the master and involved the agents determining independently whether to take on work and aggregating results with an “output portal” when finished. My project partner is now continuing this work as a full-time IBM employee.

Interlude: experiencing software engineering in industry *(Google Inc., Summer 2007)*. My experience at Google with software engineering in industry increased my motivation to return to academia and solve relevant problems in computational tools, of which I discovered many during my internship. At Google I worked on video search, but due to my non-disclosure agreement I am not at liberty to elaborate.

Because of the experimental nature of my internship project, I had the freedom of trying many things, including testing the effectiveness of various classifiers and other methods. It was rewarding to be given a relevant problem, identify the questions necessary to solve it, and set about solving it without knowing the ideal solution. Seeing the how the pressure of production prevented many of the full-time employees with whom I worked, including those working on build and language tools, from having this freedom led me to conclude that academia is more appropriate for solving the problems I want to solve.

Type systems and polymorphism *(Harvard University, Fall 2007—Senior Thesis)*. I am working on a senior thesis with Professor Greg Morrisett on programming languages. I write more about these topics in my research proposal.

---

1 Butera wrote his MIT Ph.D. dissertation, *Programming a Paintable Computer*, on applying amorphous computing concepts to solving relevant problems.