Pykaryote Research Summary

JAMES LAMINE

with faculty mentors

DR. LOREN HAARMSA and DR. SERITA NELESEN

Summer 2013

For the past three summers, Dr. Loren Haarsma and Dr. Serita Nelesen, along with student researchers, have developed a computer model simulating some aspects of biological evolution. Their aim is to study ways in which complexity, particularly protein complexes, evolves.

The artificial life model is called *Pykaryote*, a portmanteau of *Python*, the programming language in which it is written, and *prokaryote*, a type of simple organism. A typical Pykaryote simulation consists of organisms whose goal is to gather chemicals. At the end of each generation, the most fit organisms - those which gathered the most chemicals - mutate and reproduce.

Organisms live in a two dimensional environment with varying concentrations of chemicals. They can evolve the ability to move to areas with high chemical concentration, building proteins and protein complexes which may improve their chemical gathering capability. Both gathering and protein building are dictated by an organism’s genome.

Almost every aspect of the Pykaryote model is configurable. By changing different settings, we can see under which circumstances complexity does and does not evolve. The majority of the research involves coding the model, running simulations, and analyzing the results.

Pykaryote is progressing nicely, and we are preparing our first manuscript. Simulations show that our organisms evolve the ability to efficiently gather chemicals, build proteins and complexes, and move to chemically rich areas of an environment. We have shown that reducing the rate of mutation impedes evolution, while increasing the chance that a protein is useful accelerates evolution. If the mutation rate is set too high, useful proteins tend to undergo harmful mutations and complexity suffers. Given enough time, complexity can develop even with very low mutation rates.

My work this summer was focused on developing Pykaryote to a point where it is publication ready. To this end I cleaned up the code, fixed bugs, added a number of features, unit tests, and documentation. I also built a tool which automates the task of distributing large batches of simulations over a cluster of computers. I ran many simulations and create useful summaries of the results.

Working on Pykaryote gave me my first chance to develop software full time. As a software engineer, I improved my understanding of a number of technologies, including *git*, *BpX*, and the Python ecosystem. The most valuable lesson I gained during my time here is the importance of test driven development. The experience I gained this summer will serve me well through the remainder of my career.