Ishan Anand
6.UAP Proposal
March 10, 2006

Advisors:
Drew Endy,
Assistant Professor, Biological Engineering

Sriram Kosuri
Ph.D. Candidate, Biological Engineering

Genome Viewer/Editor for Refactoring Biological Organisms

Abstract
Refactoring is a technique used by computer scientists for improving program design. The Endy Laboratory has adapted this process to make the genomes of biological organisms more amenable to human understanding and design goals. To assist in this endeavor, I propose a JavaScript/AJAX web application to streamline the process of disassembling an organism's genome into genetic parts and synthesizing new genomes from natural and standardized genetic parts.

1 Background

The MIT Endy Laboratory is interested in genome refactoring. Refactoring is a technique traditionally used by software engineers to redesign computer software (Fowler et. al., 1999). By refactoring, engineers modify the design of an existing program without adding new features or functionality. Instead the new design improves program readability and maintainability:

> Refactoring does not fix bugs or add new functionality. Rather it is designed to improve the understandability of the code or change its structure and design, and remove dead code, to make it easier for human maintenance in the future. In particular, adding new behavior to a program might be difficult with the program's given structure, so a developer might refactor it first to make it easy, and then add the new behavior. (Refactoring, 2006)

Inspired by this technique, the Endy Lab is refactoring naturally occurring genomes into new versions that are easier for humans to understand and modify. In particular, the team recently disassembled a T7 bacteriophage virus and constructed a new man-made "T7.1" virus sequence (Chan et. al., 2005). The T7.1 virus was functionally similar to the natural T7 virus but with a more structured design that removed overlaps between genetic elements. Removing these overlaps allowed the team to independently manipulate each specific gene element, and by doing so, fully describe its function.

The team has now begun work on a "T7.2" virus and is looking for software tools to assist in this process. Instead of rearranging existing genetic elements within the organism (as was done with T7.1), the T7.2 design will actively remove gene elements and substitute other elements with new genes from other genomes:

> Moving beyond our design of T7.1, we will actively erase or delete elements of unknown function. In addition, efforts will be made to made to remove unknown genetic elements... To attempt to make our modeling of gene expression easier, we will use standard synthetic elements in place of the natural elements that regulate transcription and translation. (T7.2, 2005)

T7.1 was designed using existing tools such as Vector NTI and custom perl scripts coerced into ad-hoc refactoring tasks. In particular, these tools do not easily allow the arbitrary definition genetic parts and their reassembly into multiple refactored genome variants. As the goals of T7.2 increase the complexity of gene edits beyond that of T7.1, the team seeks a new tool to streamline this process of part definition and manipulation.

2 Project Description

I will implement a JavaScript/AJAX web application for viewing and editing genetic sequences from a genome database. JavaScript, and its derivative AJAX, are technologies that deliver web applications with the interactivity of a normal desktop program. (Garret, 2005) Most users are already familiar with the popular AJAX application Google Maps. The user interface for viewing gene sequences will be similar to Google Maps in a number of ways:

Feature	Google Maps	Proposed Project
Displays relevant subset of larger a database	Maps a point of interest from a map of the entire world.	Maps gene sequence of interest from an entire genome.
Click-drag	Mouse "click-and-drag" movement allows users to scan adjacent maps without reloading the browser page.	Mouse "click-and-drag" movement allows users to scan adjacent genes or gene base-pairs without reloading the browser page.
Zoom In/Out	Users can zoom in to street level details, or zoom out to a world map view.	Users can zoom in to see an individual gene or basepairs and zoom out to view the whole genome.
Overlayed maps	Satelite views, road map views, and an overlay	Functional, sequence, and overlay views.

However, the application will also have many features that are unlike that of Google Maps:

1. Define arbitrary subsequences as genetic "parts".
2. Indepedently save, edit, and move these gene parts within the genome.
3. Systematically reassemble a genome from a library of gene "parts" created by themselves and other users.

We anticipate it may not be possible to meet all these goal in the alloted timeframe for this thesis, so the project should be well documented, and sufficiently modular for future development.

3 Implementation & Features

The application will consist of two major parts, the client-side graphical front-end and the server database back-end. The front-end will be implemented in the JavaScript programming language and will be the primary development focus of this project. This interface will run within a web browser and allow users to view, edit, create, and save gene sequences with the graphical and interactivity features of a normal desktop program. To impose reasonable limits on the project scope, compatibility may be limited to one or two browsers.

Because of a genome's enormous size, it cannot be stored entirely within the JavaScript interface. Instead the server back-end will contain the entire genome, and the front-end will retrieve the relevant parts of the genome from the server as needed. Users will also be able to save their modified gene parts to the server or to their local machine. The exact form of the database has not been fully fleshed out, but it will likely be a MySQL database running on a Linux Apache/PHP server (the so-called LAMP configuration). Ultimately we would like the database to be a clone or derivative of the MIT Standard Registry of Biological Parts (http://parts.mit.edu) but this may not be feasible for a project of this size.

The application's feature set is driven by the two phases in refactoring a genome:

- *Phase I*: Disassemble the genome sequence into "parts".
- *Phase II*: Assemble the genetic parts into a full genome.

The Phase I features are concerned mainly with viewing and editing of the gene base pairs and can be broadly broken down into editing and viewing features:

Viewing Features

- *Zoom In/Out*: Users will be able to zoom in to see individual basepairs and zoom out to view a gene's position on the whole genome.
- *Click-drag viewing*: Mouse "click-and-drag" movement will allow users to scan adjacent gene basepairs without reloading the browser page.
- *Gene sequence retrieval from the server:* The program will locally cache relevant parts of the genome and retrieve additional sequence data from a server on an as-needed basis.

Editing Features

- *Dissect subsets of the genome into "parts"*: Users will graphically define an arbitrary subsets of the genome as named genetic "parts".
- *Cut and paste basepairs*: Users will graphically cut-and-paste basepairs within the genome sequence.
- *Cut and paste gene parts*: Users will graphically cut-and-paste user-defined genetic "parts" within the genome sequence.
- *Save parts*: Users will save genetic parts to local file or a central database.

These Phase I objectives are the minimal feature set we plan to accomplish during this project. After Phase I completion, the system will be refactored and documented to enable further development by other engineers. Decoupling of program logic and visual presentation will be also be crucial goal of the refactoring process. The refactored Phase I code will then serve as the foundation for implementing the Phase II features, which may or may not be completed during this thesis.

Time permitting, the Phase II features are:

- *Sequence retriveval from a real database*: The application will retrieve sequence data from a true database or server running a database such as MySQL. The database could possibly be a clone of The MIT Registry of Standard Biological parts (http://parts.mit.edu).
- *Drag and drop assembly*: Users will graphically drag-and-drop gene parts and gene basepairs to assemble a genome.
- *Workbench for genome assemblies*: Users will modify and customize previously defined parts and combine multiple parts into newer "super-parts".
- *Save whole genome sequences*: Users will save their genomes to a central database or local file in a common gene sequence format such as GenBank.
- *Edit/Revision history*: The application will track genome and part changes for the user.

4 Current Status and Schedule

Making project estimates is especially tricky given the project goals and the size of a course 6 UAP (nominally 6 hours per week). It is quite likely that completion of the Phase II feature set will be performed by the author or a subsequent developer at a later date.

I am currently implementing the Phase I feature set of the application. A proof of concept demonstrating primitive click-drag viewing and zooming has been successfully implemented. Once the click-drag animation has been refined, retrieval and caching of the gene sequence from the server will be added. For this stage, the server will be very primitive (e.g. a single PHP page running under Apache). This should be completed in the next week or two. The remaining Phase I editing features are expected to take another two to three weeks. Phase I is anticipated to be complete in four to five weeks. The refactoring phase is estimated to take two to three weeks.

Finally, the Phase II feature set is somewhat speculative and we may not be able to achieve all the listed goals. Once Phase I is complete, it will be easier to layout a timetable for the Phase II feature set and the project timetable will be revised. In the meantime, a rough estimate is that each Phase II feature will take two to three weeks. By this schedule, only two, maybe three, Phase II features might be implemented during the course of this semester. The features for implementation will be selected based on the team's priorities and the project progress.

Bibliography

Chan, Leon Y. Kosuri, Sriram. Endy, Drew. (2005) Refactoring bacteriophage T7. *Mol Syst Biol* 13 September 2005; doi:10.1038/msb4100025

Fowler, M. Beck, K. Brant, J. Opdyke, W. Roberts, D. (1999) Refactoring: Improving the Design of Existing Code. Boston, MA, USA: Addison-Wesley Professional

Garret, Jesse James (2005) Ajax: A New Approach to Web Applications. *Adaptive Path, LLC*. 18 Februrary 2005. http://www.adaptivepath.com/publications/essays/archives/000385.php

Wikipedia contributors (2006). Refactoring. *Wikipedia, The Free Encyclopedia*. Retrieved 20:25, March 8, 2006 from http://en.wikipedia.org/w/index.php?title=Refactoring&oldid=41355008.

"T7.2" (2005). http://openwetware.org/index.php?title=T7.2&oldid=12368