HIV Evolution Sites Visualization

Kuangyou Yao
University of Washington, Seattle
185 W Stevens Way NE
kuangyou@cs.uw.edu

Liusu Wang
University of Washington, Seattle
185 W Stevens Way NE
liusuw@uw.edu

ABSTRACT
This paper discusses the project we build in a data visualization course, CSE 512, Spring 2015 at the University of Washington. The paper covers the motivation for the project, previous related work, both front-end and back-end implementation, analysis in time and storage complexities, user evaluation, and the projected future development work as an ongoing scientific research tool.

INTRODUCTION
This project is to create a visualization tool in aid of studying one of the deadliest viruses known to mankind, the Human immunodeficiency virus, commonly known as HIV. Currently, there is no cure for HIV and most infected individuals succumb to the end-stage disease known as Acquired Immunodeficiency Syndrome, or AIDS. HIV is a retrovirus, which has certain features making it difficult to treat and eradicate including a high mutation rate. Due to its ability to rapidly generate variation, the human immune system cannot easily target the virus as an outside pathogen. What's worse, the virus also specially infects and destroys human T-cells, which are essential for the immune system to fight diseases. HIV was the culprit responsible for the 1.5 million lives lost globally in the year 2013. (UNAIDS, 2013)

Understanding how the virus evolves in the human body is an important part in defeating this disease. Studies have shown that some individuals possess certain immune genotypes that naturally control virus and significantly delay the progression to AIDS. Hypotheses suggest that it is because their immune system recognize virus through a specific region of the virus shell protein that cannot tolerate mutations very well. Most likely, that part of the protein is crucial to the virus. Without it, the virus cannot survive or maintain its ability to replicate.

Therefore, it is of interest to discover parts in the virus, which are crucial to virus replication. This information could be critical to informing effective vaccine strategies, and to save more people in the future. The project is also projected to support other kinds of virus evolution which might shows similar pattern as HIV.

Michael Dapp from the UW Microbiology Department provides us with HIV protein sequence data collected from patients infected by the virus in the US. As students studying Computer Science, it is an absolute thrill for us to use our skills and knowledge we learn from the Data Visualization Course and give help in the war against a deadly disease. It would be like Edward Jenner conquered smallpox and John Snow defeated cholera at other time in mankind history.

PREVIOUS RELATED WORK
HIV envelope regions were sequenced from time of infection until AIDS in nine adult men infected with HIV type 1 In order to help visualize the location and types of mutations that had arisen, large tables were made using Microsoft Excel. Mutations relative to the original founder virus sequence were color-coded based on specific features.

This is a relatively simple and static visualization of the virus. In order for researchers to highlight a certain cell with color, they would have to click

on every single cell for the color. Also because of the long protein sequence, highlighting the positions with significant variation, researchers would have to collapse the columns without mutation, neglecting a lot of neighboring positions.

METHODS AND IMPLEMENTAION

We designed a general-to-detail interaction tool. We would discuss both the front-end and back-end design.

Front-end

First is an overview of the virus mutation by position. The envelope of the virus is made up of very long sequence of amino acids. The raw data is a collection of viral sequences from different times throughout infection. By displaying the entropy score in a bar chart, where the horizontal axis is the position in the sequence while the vertical axis is the entropy score, users have a broad picture of which certain region of the virus envelope protein has more variability in the amino acid sequence. Since the protein sequence can sometime go up to 500+ position, we incorporated the "Focus + Context Via Brushing" design. We achieved an effect that the user could look at the detail of certain region and the whole picture simultaneously.

Secondly, user can go deeper into one specific position of the virus and how that specific position mutated over the time. By clicking a bar in the "Focus + Context", application displays a detailed longitudinal mutation, also highlighted in 3 different colors, representing whether this is a Forward or Backward mutation, or it is just a statistical noise.

Another feature we added is the adjustment of a hyper parameter determining if the mutation is a "Noise" or "Significant mutation". This feature is implemented using JQuery UI, which listens to sliding event and adjust the color-coding accordingly.

Back-end

For the back-end, which is an Ubuntu machine running on DigitalOcean.com with LAMP server software package. It handles all the data queries and calculation.

On the server, files include numbers of amino acid alignment files in fasta format, a comma-separated-file (csv) file of a database with the official consensus of each column probabilities of the virus. This database table allows comparison of evolving sites in order to characterize them as forward or reverse mutations. We used the BioPython module to parse through the .fasta files. And NumPy module for scientific computing methods, like entropies calculation, and finding probabilities from the probabilities csv file.

Once the python modules finished fetching and calculating the requested data, like entropies and mutation details mentioned above, all the results will be cached in a JSON format. This would save time when a same query comes in. Finally the PHP server will read all the results file and and send back to the client using standard HTTP response for the front end to decrypt and display.

RESULTS

Time complexities

Assuming internet connection is perfect, the most demanding process would be the Context-Focus bar chart filtering in the front end and new file processing in the backend.

For the filtering, which is a O(n) linear time complexity algorithm where n is the length of the entropies file length. It is a simple brute force search through the entire list of entropies, only selecting the ones within the selected region in the brush. We tested in different web browser. Google Chrome and Apple Safari have an amazingly fast JavaScript engine, which can do data filtering in the Context-Focus bar chart almost instantly, no significant lagging can be

observed. Yet Firefox has a very obvious lag when switching different focus region.

In the back-end, which is heavily based on Python modules, a non-compiled programming language. If the requested result is not cached, a new calculation is necessary. The time complexity would be proportional to how big the consensus database csv file is. We don't really know how the method numpy.genfromtxt() is implemented. We are not sure whether it reads the whole file into memories or did some trimming? But we guess it will be scaling up based on the size of the file. Calculations are all 64-bit floating-point arithmetic, with vector algebra.

Then remaining problem would be the network I/O issue. Note that the following time is acquired in an Ethernet connection with 7Mbps speed. Results for the bar chart are stored in the browser when the entropies are received. This file is very small, like around 1.4KB, and it is only requested once, unless the user selected a different alignment file. But for the detailed mutation, with every click on the bar chart, a new AJAX request is sent, a whole new JSON package would be sent from the backend. It is within the size of 60KB, which takes around 180 milliseconds to send back.

Papers and notes may use color figures, which are included in the page limit; the figures must be usable when printed in black and white in the proceedings.

The paper may be accompanied by a short video figure up to five minutes in length. However, the paper should stand on its own without the video figure, as the video may not be available to everyone who reads the paper.

DISCUSSION

We managed to have two users who are researchers, Michael Dapp and his colleague. Siriphan Manocheewam in the UW microbiology to participate our evaluations. We designed a user-experience survey and collected their comments based on our questions. They describe the tool as being "clean, intuitive, comprehensive, informative, and multi-layered" which will greatly help them in the research process.

They also mentioned that the tool very efficiently displays a vast and complex amount of data in an easy to manipulate format, and save a lot of time compared to previous methods of similar data display.

To better understand how users interact and use this tool, we directly observed Michael, researcher in the UW Microbiology, using the tool to complete his tasks. We have perceived that he was interested in the interactions with bar chart by adjusting the brush on the "context", hovering and clicking the bars. He also spent a long time playing with the table and threshold slider. He indicated that this tool potentially helps him with data display because it saves him time comparing with editing the data in Microsoft Excel.

During the conversation with our collaborators, they were impressed with the D3's graphic library. Our tools enabled them to look at the protein from a multi-layered perspective, from general to detailed view. We believe that this catalog style of interaction matches up with how we learn things, from general to detailed. Even though it is a simple design, it still yields great effect.

FUTURE WORK

Design and Appearances

From the feedback we collected from the researchers in the UW microbiology, we concluded the following places needed improvement.

1. Better instruction, preferably merged with the main window instead of creating and showing instruction in a new window.

2. Enhanced Color Scheme. Right now the context-focus bar chart has blue bars. The color is colliding with the blue color that
3. is representing “reverse mutation” in the detailed table to the right hand side.
4. Better positioning for the consensus tooltip. Right now, we can only show a consensus amino acid of a given position in a “div” element in an awkward position. We are planning to implement a fixed header in the top part of the detailed table. So as the user is scrolling down the table, he would always have a reference to the current consensus of the amino acid in that location.
5. Improved navigation through different segments of protein sequence. We hope to create an alternative way (Like a slider) to view different parts.
6. Animation and transition. Right now the tool only has abrupt changes when user switches a different alignment file, or choose a different position to look at different mutation data. It will be nice to include an animated transition for smoothness.

File System Security

As we were developing the project, a fellow Computer Science student happened to look at our backend PHP script code. After 15 minutes, he managed to inject a piece of command, which was executed by the Ubuntu machine. The vulnerability he exploited is that the PHP script directly calls a Linux command and he managed to chain it with another command he wishes. He first tried a simple ‘sudo reboot’ via a AJAX request, forcing the server to shut down and restart. Later, he performed a series of ‘touch’ commands that created a lot of junk file on the server root directory.

Luckily he warned me about the security vulnerability, and we quickly put a sentry code in the PHP script by checking every AJAX request is safe (Without any spaces or special characters). That leads us to think about more on the security matters.

Even though the visualization is built mainly for biology researchers, and users’ backgrounds should be relatively limited when compared to Amazon or Facebook, we should still be careful about potential security risks. The project would be more useful when more users put in valuable data into the server.

Currently the backend did not have any upload file sanity checking. For simplicity, we only put a limit in the file size and file type. We should be laying attention to it as well.

REFERENCE

1. Global report UNAIDS report on the global AIDS epidemic : 2013. (2013). Geneva: UNAIDS.
2. Herbeck, J. T., M. Rolland, Y. Liu, S. Mclaughlin, J. Mcnevin, H. Zhao, K. Wong, J. N. Stoddard, D. Raugi, S. Sorensen, I. Genowati, B. Birditt, A. Mckay, K. Diem, B. S. Maust, W. Deng, A. C. Collier, J. D. Stekler, M. J. Mcelrath, and J. I. Mullins. "Demographic Processes Affect HIV-1 Evolution in Primary Infection before the Onset of Selective Processes." Journal of Virology 85.15 (2011): 7523-534. Web.