Group Project Report: siRNA Generator

Andrew Rosen

1 Overview

I will first discuss the motivation for our work and our group's broad objective.

1.1 Motivation

Our primary motivation for our project was the 2014 Ebola outbreak. There are five identified species of the Ebolavirus, four of which directly affect humans: Bundibugyo, Taï, Sudan, and Zaire [4]. The last, the Reston virus, does not harm humans. The Zaire ebolavirus is the most deadly and the one responsible for the current outbreak, the most deadly to date.

Symptoms of Ebola virus disease (EVD) include: fever, chills, fatigue, weakness, muscle/join pain, vomiting, reddened eyes, and hemorrhaging [12] [15]. These symptoms appear within eight to ten days, on average [4].

This most recent outbreak primarily affected people in Guinea, Liberia, and Sierra Leone, with additional cases reported in Nigeria, Mali, and Senegal (Figure 1) [4]. Thus far, there have been an estimated 25,907 infected in 2014 outbreak, and nearly 11,000 people have succumbed to the disease [4]. Two cases of Ebola were imported to the United States and was spread to two nurses. This, plus cases Spain and the United Kingdom sparked a large level of concern worldwide [10] [16].

Ebola can be fought, and the most effective method is prevention. Ebola does not spread easily when compared to other viruses such as influenza. Infection requires contact with the bodily fluids of a infected person who is displaying symptoms of Ebolavirus disease [15] [12]. Thus, the most effective way currently available is to quarantine the infected individuals and find those who have had contact with the infected [12]. Meanwhile, research into various treatments such as vaccines [5] or siRNAs [13] has had immense interest due to the severity of this outbreak.

1.2 Our approach

We chose to create a tool to design siRNA to fight viruses since a couple of group members were involved in active research on siRNA and were very familiar with it. siRNA is short for small interfering RNA or silencing RNA as it interferes with or silences the expression of a specific sequence it matches up with. This

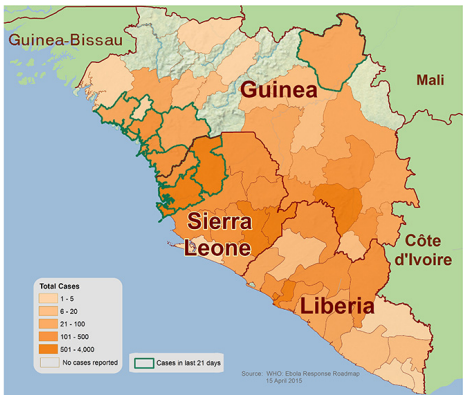

Figure 1: Map of total cases of the 2014 Zaire Ebolavirus outbreak. Source: Center for Disease Control [4].

specificity makes siRNA highly applicable to cancer treatment [**?**], gene therapy [8], and viral treatment [7].

Our idea has merit, given that research is being actively done on siRNA Ebola treatments [13] [9] [6]. In a study published in 2010 in the Lancet, Geisbert *et al* [6] successfully tested a post-exposure siRNA treatment which helped protect against Zaire Ebolavirus in primates.

Needless to say, a tool for evaluating a single genus of viruses to create a siRNA to combat it, such as ebolaviruses, is highly useful, but is rather limited in scope. As a result, we chose to create a system which could analyze any group of viruses for a conserved region and designed siRNA to interfere with it.

2 Contributions

In this section, I will talk about the division of work between each member of the group and how I implemented my portion of the work.

2.1 Assigned Tasks

Our group separated into three rough groups to tackle each of the three tasks we identified.

The first group was tasked with creating a script to identify the conserved region within the species of viruses. These conserved regions would then be compared against the human genome using BLAST [1]. Any sequence that was shared with the human genome would be filtered out of output for the next group, since by targeting that sequence with an siRNA, it could potentially target that sequence in humans.[1] These sequences are output as a FASTA file.

[1] This is generally considered a bad idea as I understand it.

The second group takes the conserved regions that did not show up in the human genome and uses those to design siRNA to bind with the virus's mRNA. The resulting siRNA needs to then be compared against the human genome, for the same reasons the first group performed a check. This would output a FASTA file containing the siRNA sequences that would target the genus of virus. These potential siRNA should be further analyzed for any consequences. A member of the group also implemented an algorithm to estimate the amount of mutations it would take for the virus to render the siRNA treatment ineffective.

I was the third group and my responsibilities were quite broad.

- I needed to write a web application to accept a FASTA file from the user. This file would then be passed the beginning of the first group's code. I could accomplish this by spinning up a thread that called the first function needed to run.
- I needed to ensure that the publicly accessible web application did not have write capabilities on the server other than writing the file.[2]
- I was to help setup the webserver we would use for our application: `apollo`.[3]
- I needed to ensure the rest of the group had the tools they needed to run their python programs.
- I made myself available to each of our group members to aid with debugging.

In summary, I needed to do some web programming and act as administrator for our server.

2.2 Implementation

I worked with Brendan to install Ubuntu 14.04 LTS on the server. This was a fairly trivial task.

I then had to acquire the various pieces of software we needed, which included:

- BLAST [1] and clustalw [14] for sequence alignment.
- Biopython [3], which contains many useful biology related python scripts and classes.
- RNAfold [11], which predicts the secondary structures of RNA
- I also obtained the necessary human genome files to perform sequence alignment from NCBI.

[2] And I just thought of an attack! I could upload a very, very, large file. I'm not sure what this would do, but it would probably not be good.

[3] A highly germane name for the server, since Apollo's portfolio includes plagues and medicine.

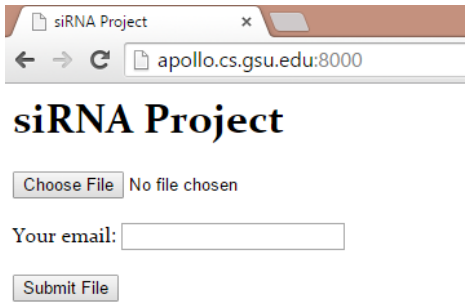

Figure 2: The upload form. This is output from the `listener`. The user uploads a FASTA file and their email address. This information is stored in the server on files and read by `scheduler`.

I will talk specifically about the goal of each part of the application and my implementation of it. You can find my code on github:
`https://github.com/abrosen/bioinf/tree/master/website/`

2.2.1 Application

The first part of the application is the part the user sees. I created a fairly simple webserver using Python, which I named `listener`. The `listener` listens on port 8000 for an incoming get request and then serves a webpage (Figure 2). The webpage is stored as a string in `listener`.

Once `listener` gets a POST from the user, the `listener` timestamps the received file and email. They are stored locally and the FASTA file will be read by the `scheduler`. This separation of exists to prevent malicious users from having write access by compromising the python program that will be performing all the computations.

2.2.2 Scheduler

The scheduler scans the current directory for new files. Upon finding new FASTA files, it feeds them to the first part of the siRNA sequencer.

In order to do this in a responsive manner, I have `scheduler` check every 10 seconds for a new file and boot up a new thread. The thread is not a daemon and it will run independently of `scheduler` in the event `scheduler` crashes.

Once the scripts are done executing, the `scheduler` emails the user with the results. The emails are sent from the server to the Computer Science department's server, which forwards the mail to where it needs to go.

3 Evaluation

3.1 Test Case

We planned on using a FASTA file of the five species of the Ebolavirus as our initial input and test case for the entire program. For my part of the program, I tested `listener` by uploading various files to `apollo` via the webpage form and verifying they were intact on the server side. The most tedious part I had to test was the email portion. Most mail servers I tried to use did not accept connections from computer. I was only able to send mail directly to the department's server when I was logged in to the campus VPN.

3.2 Deliverables

A deadline for combining the code was set for April 19th and not met. There are a number of reasons for this, but I do not believe that the problem was too difficult or large.

I observed a number bad coding practices during the project. One group member was using Python 3.4 despite the rest of the group working in 2.7.[4] Many members hardcoded file names into commands and wrote duplicate components.

None of these alone were an issue that completely impeded the implementation, or even together, but were symptoms of the true issues: a lack of communication and miscommunication. It seemed like every member assumed what the other member was doing without consulting anyone else. As a result, each member made their own assumptions about how the components their program were connecting to would work, even though we had determined that FASTA was the common input and output between files.

I do not believe that our final product in completely unusable, but it requires more work to put completely together.

References

[1] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[2] Yunching Chen, Xiaodong Zhu, Xiaoju Zhang, Bin Liu, and Leaf Huang. Nanoparticles modified with tumor-targeting scfv deliver sirna and mirna for cancer therapy. *Molecular Therapy*, 18(9):1650–1656, 2010.

[3] Peter JA Cock, Tiago Antao, Jeffrey T Chang, Brad A Chapman, Cymon J Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, et al. Biopython: freely available python tools

[4] Granted, I did that too, but it was a deliberate choice on my part. That was in the webserver code, and it doesn't directly interact with any code that the group has to write.

for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[4] Centers for Disease Control et al. 2014 ebola outbreak in west africa. 2015.

[5] Thomas W Geisbert. Emergency treatment for exposure to ebola virus: The need to fast-track promising vaccines. *JAMA*, 313(12):1221–1222, 2015.

[6] Thomas W Geisbert, Amy CH Lee, Marjorie Robbins, Joan B Geisbert, Anna N Honko, Vandana Sood, Joshua C Johnson, Susan De Jong, Iran Tavakoli, Adam Judge, et al. Postexposure protection of non-human primates against a lethal ebola virus challenge with rna interference: a proof-of-concept study. *The Lancet*, 375(9729):1896–1905, 2010.

[7] Leonid Gitlin, Sveta Karelsky, and Raul Andino. Short interfering rna confers intracellular antiviral immunity in human cells. *Nature*, 418(6896):430–434, 2002.

[8] Sun Hwa Kim, Ji Hoon Jeong, Soo Hyun Lee, Sung Wan Kim, and Tae Gwan Park. Peg conjugated vegf sirna for anti-angiogenic gene therapy. *Journal of Controlled Release*, 116(2):123–129, 2006.

[9] Jeffrey R Kugelman, Mariano Sanchez-Lockhart, Kristian G Andersen, Stephen Gire, Daniel J Park, Rachel Sealfon, Aaron E Lin, Shirlee Wohl, Pardis C Sabeti, Jens H Kuhn, et al. Evaluation of the potential impact of ebola virus genomic drift on the efficacy of sequence-based candidate therapeutics. *mBio*, 6(1):e02227–14, 2015.

[10] Elizabeth Levin-Sparenberg, Rachel Gicquelais, Natalia Blanco, Miriam D Ismail, Kyu Han Lee, and Betsy Foxman. Ebola: The natural and human history of a deadly virus by david quammen. *American journal of epidemiology*, 181(2):151–151, 2015.

[11] Ronny Lorenz, Stephan HF Bernhart, Christian Hoener Zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F Stadler, Ivo L Hofacker, et al. Viennarna package 2.0. *Algorithms for Molecular Biology*, 6(1):26, 2011.

[12] WHO Ebola Response Team. Ebola virus disease in west africa - the first 9 months of the epidemic and forward projections. *N Engl J Med*, 371(16):1481–95, 2014.

[13] Emily P. Thi, Chad E. Mire, Amy C. H. Lee, Joan B. Geisbert, Joy Z. Zhou, Krystle N. Agans, Nicholas M. Snead, Daniel J. Deer, Trisha R. Barnard, Karla A. Fenton, Ian MacLachlan, and Thomas W. Geisbert. Lipid nanoparticle sirna treatment of ebola-virus-makona-infected nonhuman primates.

[14] Julie D Thompson, Toby Gibson, Des G Higgins, et al. Multiple sequence alignment using clustalw and clustalx. *Current protocols in bioinformatics*, pages 2–3, 2002.

[15] Gary Wong, Gary P Kobinger, and Xiangguo Qiu. Characterization of host immune responses in ebola virus infections.

[16] Karen Yourish and Buchanan Larry. Is the U.S. Prepared for an Ebola Outbreak? *The New York Times*.