Genomic Sequence Analysis Process Improvement

Loran Saggu

Indiana University

lsaggu@indiana.edu

Yang Ruan

Indiana University

yang.ruan.3545@gmail.com

Khaliq Satchell

Elizabeth City State University

kjsatchell026@students.ecsu.edu

Gregor von Laszewski

Indiana University

laszewski@gmail.comSaliya Ekenayake

Indiana University

esaliya@gmail.com

Geoffrey Fox

Indiana University

gcf@indiana.edu

**ABSTRACT**

The genome is the most basic description of an organism as is known today. In an effort to better understand the relationships between organisms, our team is trying to map the similarity between genomic sequences. Through the use of complex algorithms, “distances” between genomic sequences may be defined, and these distances can be scaled and used to create a plot. There is, of course, some work to be done. These processes developed by the lab are heuristic and only understood and accessible by a few. Also they are not automated. However, these issues can be dealt with by setting up a server to run a Python based portal designed to accept user input via the web and run these algorithms on computer clusters. Also, dealing with vast amounts of data requires the use of software capable of dynamically rendering data points. By analyzing, updating, and cleaning code, the visualization software may be made ready to handle larger data sets. Being able to efficiently map the differences in genomic sequences will vastly improve people’s understanding of organisms’ evolutionary histories. This may also have vast implications in the realms of geneology and medicine as a result of the improved understanding of the relationships between various organisms.

# INTRODUCTION

An organism’s genetic code determines everything about that organism: it determines what species the organism will become, how big it will grow, whether or not it will have gills or will be able to convert light into energy. The genetic code of many organisms is quite similar, but of course there are differences which allow for every unique individual. If the similarity between genetic sequences could be mapped, deriving and understanding the relationships between organisms could be done efficiently and effectively.

There are tens of billions of organisms on the earth. Processing even a small subset of genomic sequences means working with thousands of sequences. In order to map these sequences, each sequence must be compared to and “distanced” from every other sequence within the set [4]. Immediately, the set of data is squared in size. The resulting n x n distance matrix, describing the relationship between the n genomic sequences, must be scaled in order to accurately display the relationships between various genomes [4]. As a result, processing this amount of data requires an incredible amount of computing power as well as time. The lab has developed algorithms, pairwise clustering and multi-dimensional scaling, in order to assign distances to gene sequences within a set and ultimately create a set of graph worthy data points [4]. Running these algorithms on the data, though, is not extremely straightforward. Gene sequences must be clustered, distanced, weighted, and scaled [4]. These tasks may be considered map tasks in a map-reduce process. Each map task is administered by its own program developed by the lab. Finally, the data is reduced into a text file used by the visualization software to graphically present the data. Running multiple programs, ensuring the correct environments are present and properly configured, debugging programs or input entry are daunting tasks to those unfamiliar with the world of computing and computer science. A friendly framework that accepts user data and returns processed data is a solution that would allow users beyond the computing community access to these tools.

Furthermore, biologists and other interested parties are interested in looking at the results quickly and efficiently, but rendering these large data sets takes a significant amount of computing power and a toll on processors. The solution for such a problem would be applying parallelization support using multi-core processing which will dynamically change the speed of which the data is processed. Essentially, this will result in the data being ready to be analyzed in a timely manner.

# Submission Tool

In order to accomplish the quantification of genomic similarity and create the desired plots like the one displayed in FIGURE 1, the lab has developed complex algorithms which determine the distances between genomes and adjust these distances in order to provide accurate depictions of these genomic sequences in a 3-D space. These algorithms are embedded within programs designed and written by, once again, the lab. Due to the fact that these programs must be able to process tens of thousands of genomic sequences, these programs require long periods of time to run even on computer clusters, which can process the data in parallel. In addition to being heuristic in nature, the methods used to process this data are only understood by a few, those within the lab. Requests to process the genomic information are manually submitted to computer clusters, and variables must be adjusted uniquely per data set. For a biologist unfamiliar with the computing world, attempting to process genomic data in this manner on his or her own would be quite difficult.

Essentially, the lab is seeking a more uniform method of submitting genomic sequence data to be processed and plotted using the programs that have already been designed. A system with a more user friendly interface which could be utilized by someone with average computer knowledge is the ultimate endpoint. The goal, then, of this project is to design a program which consolidates the existing programs allowing a user to submit data and obtain results using one system. Instead of having to constantly adjust variables and use multiple programs requesting multiple parameters, a user would only need to input everything once.

The tool developed, submit.py, is designed to accept Portable Batch System or PBS scripts from a user or front-end and run them on computer clusters, the back-end. The tool transfers files as well as returns output to a user. Though the tool was designed with broader uses in mind, in this case, the tool would submit scripts running the genomic processing algorithms on supplied genomic data. The tool would then return the resulting file used to visualize the data within visualization software such as PlotViz. Furthermore, submit.py, uses MongoDB and MongoEngine to create and store user and job submission data. The submission history allows for convenient access and evaluation of old data or old or failed submissions.

In order to do all of this, the submit.py program, with command line functionality, was integrated with the Cloudmesh framework. Via SSH and SCP connection, submit.py may access the clusters within FutureGrid, one of the computer grid resource providers utilized by Cloudmesh. The Cloudmesh framework is designed to allow efficient communication between local machines and computer clusters [3]. It possesses a user-friendly front-end command line and graphical user interface which both may be altered for specific purposes. Also, the benefits of integrating with Cloudmesh include access to multiple cluster resources, job monitoring, and web based interfacing [3]. Figure 2 displays the architecture of the genomic sequence analysis and how submit.py fits in. Ultimately, this tool and Cloudmesh will make the genomic sequence analysis process more accessible beyond the computer science community.

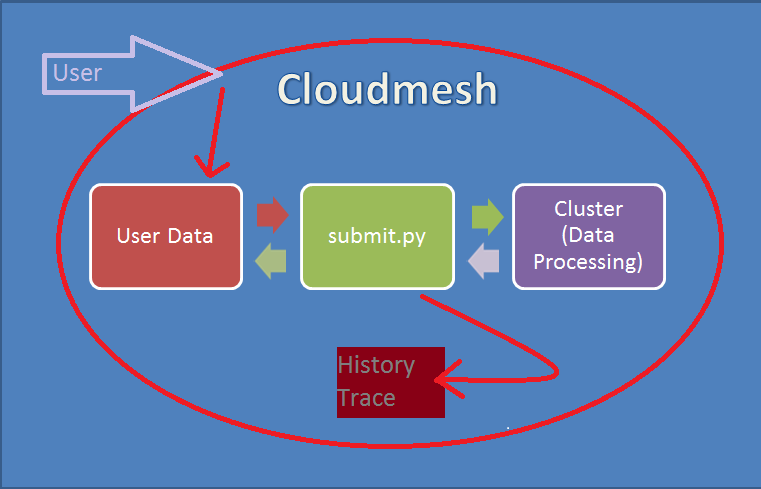


Figure 2: Job submission architecture

# Enhancing PlotViz3

In biology there is a scientific field that develops methods and software tools for organizing and analyzing biological data. That field is bioinformatics and it combines computer science with other fields in order to study biological data and processes which in turn can provide meaningful information on genomic sequences. Currently, there are phylogenetic trees used in bioinformatics but they are 2-dimensional and can be hard to read unlike a software called PlotViz3. PlotViz3 is a tool programmed in the C++ language used for visualizing a large amount of 3-dimensional data points in a virtual space. With this visualization tool, scientists will be able to find the cluster correlations when viewing their genomic sequence data more effectively than previous methods. Although, right now this software is not as fast and efficient as it could be.

Adding in parallelization support for multithreading PlotViz3 using multi-core processing will undoubtedly assist in speeding the process the software goes through when adding data into it. Multi-core processing is a single process running multiple processes all at once rather than one process at a time which is more time consuming when data is needed quickly. In order to find out how to add the parallelization support into the code the environment had to be built onto the system that is being used. Through an Integrated Development Environment (IDE) the code has to be thoroughly read through and understood. Only then can the code be altered and updated with the support. Then a testing a phase is required to make sure that any improvement to the speed of the software has been made.

Essentially, after the testing phase of the research is complete, PlotViz3 will be running its processes quickly and efficiently whenever data is inputted into it. This will cause the software to be more desirable to scientists in the field of bioinformatics who need an easier way to view their genomic sequence data in a suitable amount of time.

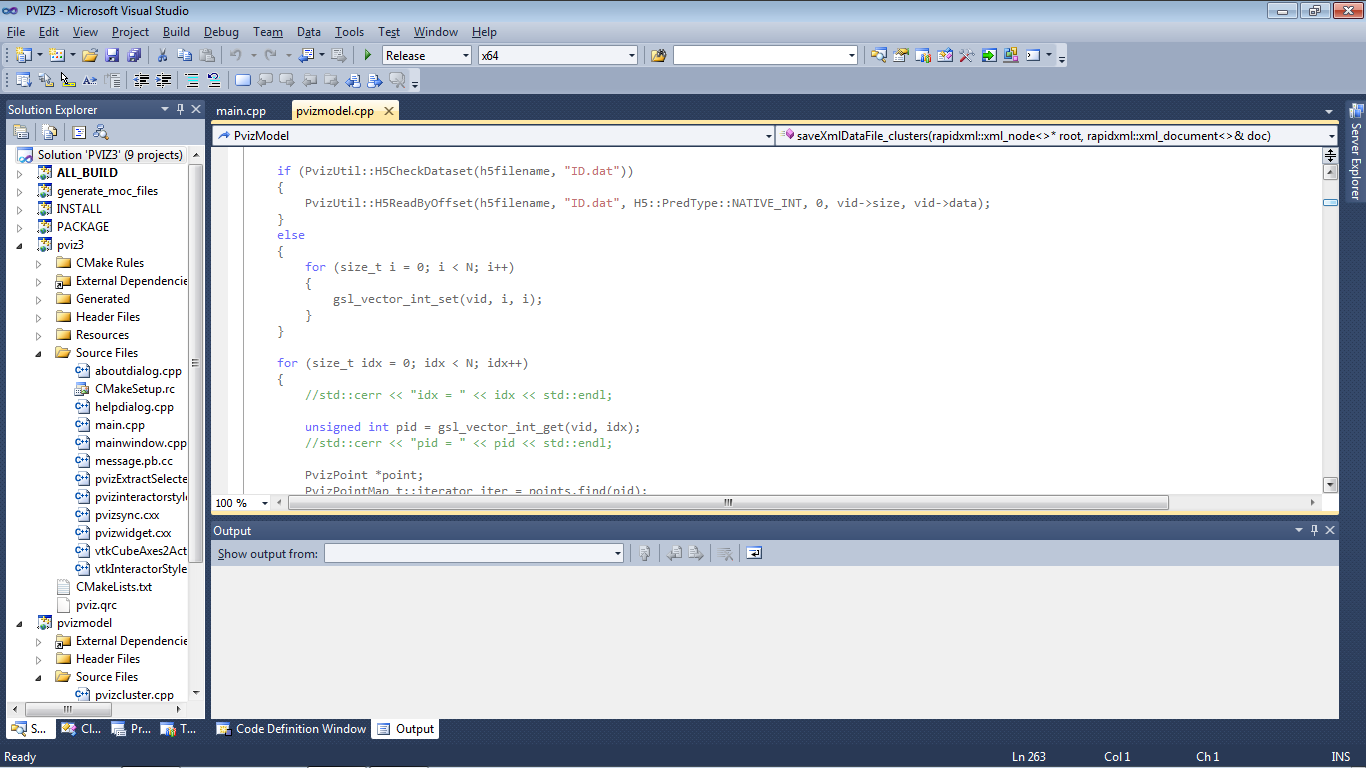


Figure 3: Code behind PlotViz3

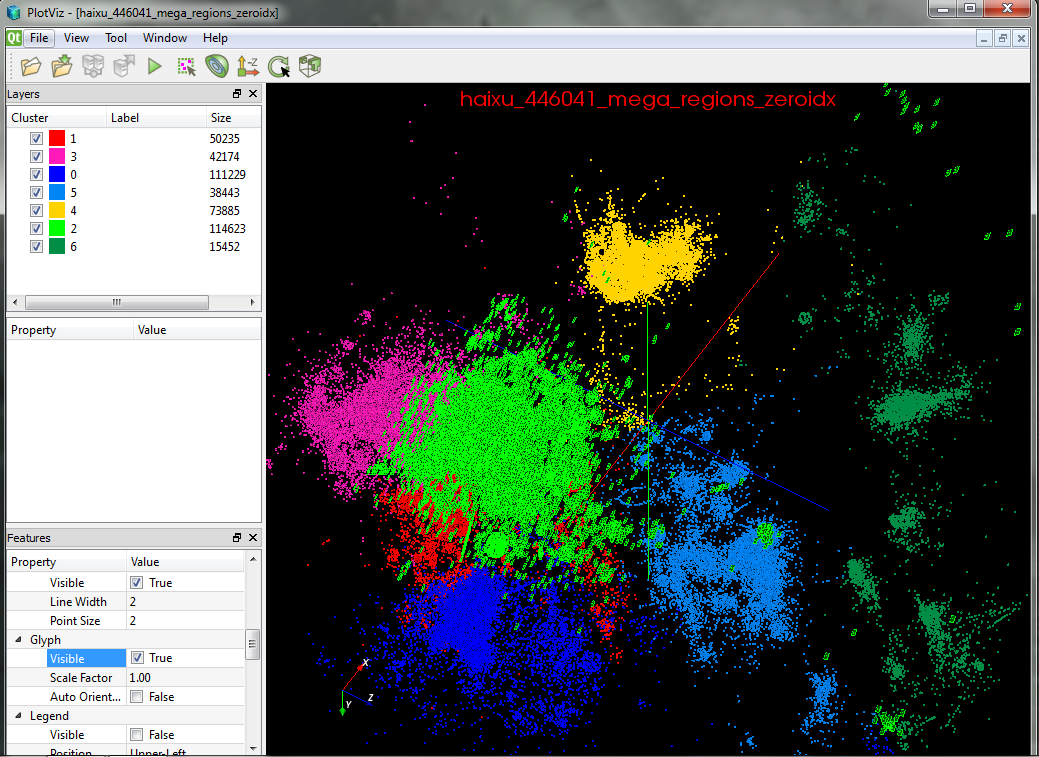


Figure 4: PlotViz3 tool

# Conclusion

As aforementioned, the results of processing this data and creating 3-dimensional visualizations have huge implications within the fields of biology, medicine, and genetics. Being able to quickly and reliably process genomic sequences into friendly visualizations may allow biologists to recognize unnoticed relationships, geneticists to understand and explain evolutionary changes, and medical experts to infer the effectiveness of potential treatments. By working to streamline the job submission process, analysis of genomic data will become more accessible to these groups: biologists, doctors, geneticists, and other interested parties.

# ACKNOWLEDGMENTS

Thank you goes out to the National Science Foundation (NSF) for their funding and support of the FutureGrid project led by Indiana University in conjunction with the University of Chicago. Furthermore, to those involved with the Cloudmesh infrastructure, thank you.

# REFERENCES

1. Million sequence clustering: [salsahpc.indiana.edu/millionseq](http://salsahpc.indiana.edu/millionseq)
2. von Laszewski, G.; Fox, G. C.; Wang, F.; Younge, A. J.; Kulshrestha; Pike, G. G.; Smith, W.; Voeckler, J.; Figueiredo, R. J.; Fortes, J.; Keahey, K. & Deelman, E. Design of the FutureGrid Experiment Management Framework, Proceedings of Gateway Computing Environments 2010 (GCE2010) at SC10, IEEE, 2010
3. von Laszewski, G.; Wang, F.; Lee, H.; Chen, H. & Fox, G. C., Accessing Multiple Clouds with Cloudmesh, Proceedings of the 2014 ACM International Workshop on Software-defined Ecosystems, ACM, 2014, 21-28
4. Yang Ruan, Saliya Ekanayake, Mina Rho, Haixu Tang, Seung-Hee Bae, Judy Qiu, and Geoffrey Fox. 2012. DACIDR: deterministic annealed clustering with interpolative dimension reduction using a large collection of 16S rRNA sequences. In Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine (BCB '12). ACM, New York, NY, USA, 329-336. DOI=10.1145/2382936.2382978 http://doi.acm.org/10.1145/2382936.2382978