Genomic Sequence Analysis Automation

Lorander Saggu, Saliya Ekanayake, Yang Ruan, Gregor von Laszewski, Geoffrey Fox

Indiana University

Abstract

In an effort to better understand the relationships between organisms, we are 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 addressed 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. This portal would be integrated into an existing framework, Cloudmesh, lending a user-friendly front-end and a powerful back-end for processing. 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 genetics and medicine as a result of the improved understanding of the relationships between various organisms.

Introduction

There are tens of billions of organisms on the earth. Processing even a small subset of genomic sequences means working with thousands of sequences.

The lab has utilized the Smith-Waterman distancing algorithm and developed pairwise clustering and multi-dimensional scaling algorithms, part of the Twister/Dacidr Pipelilne (Figure 2), in order to assign distances to gene sequences within a set and ultimately create a set of graph worthy data points [1, 4]. Running these algorithms on the data, though, is not extremely straightforward. Gene sequences must be clustered, distanced, weighted, and scaled [4]. 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 as seen in Figure 3. This software, though, must be able to render thousands of points quickly and fluidly.

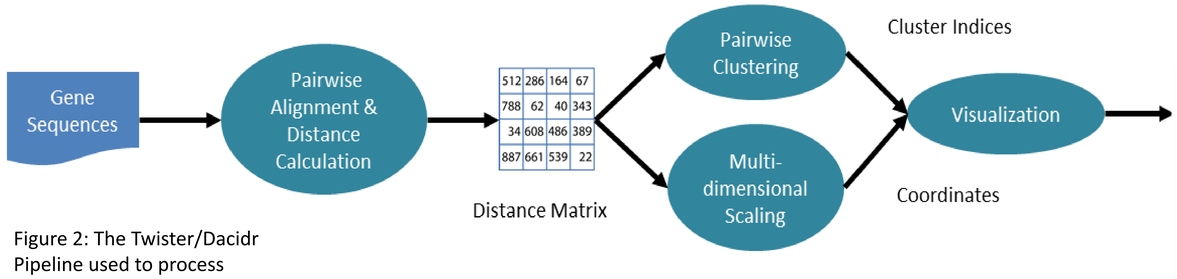

Figure 2: The Twister/Dacidr Pipeline used to process genomic data

Submission Tool

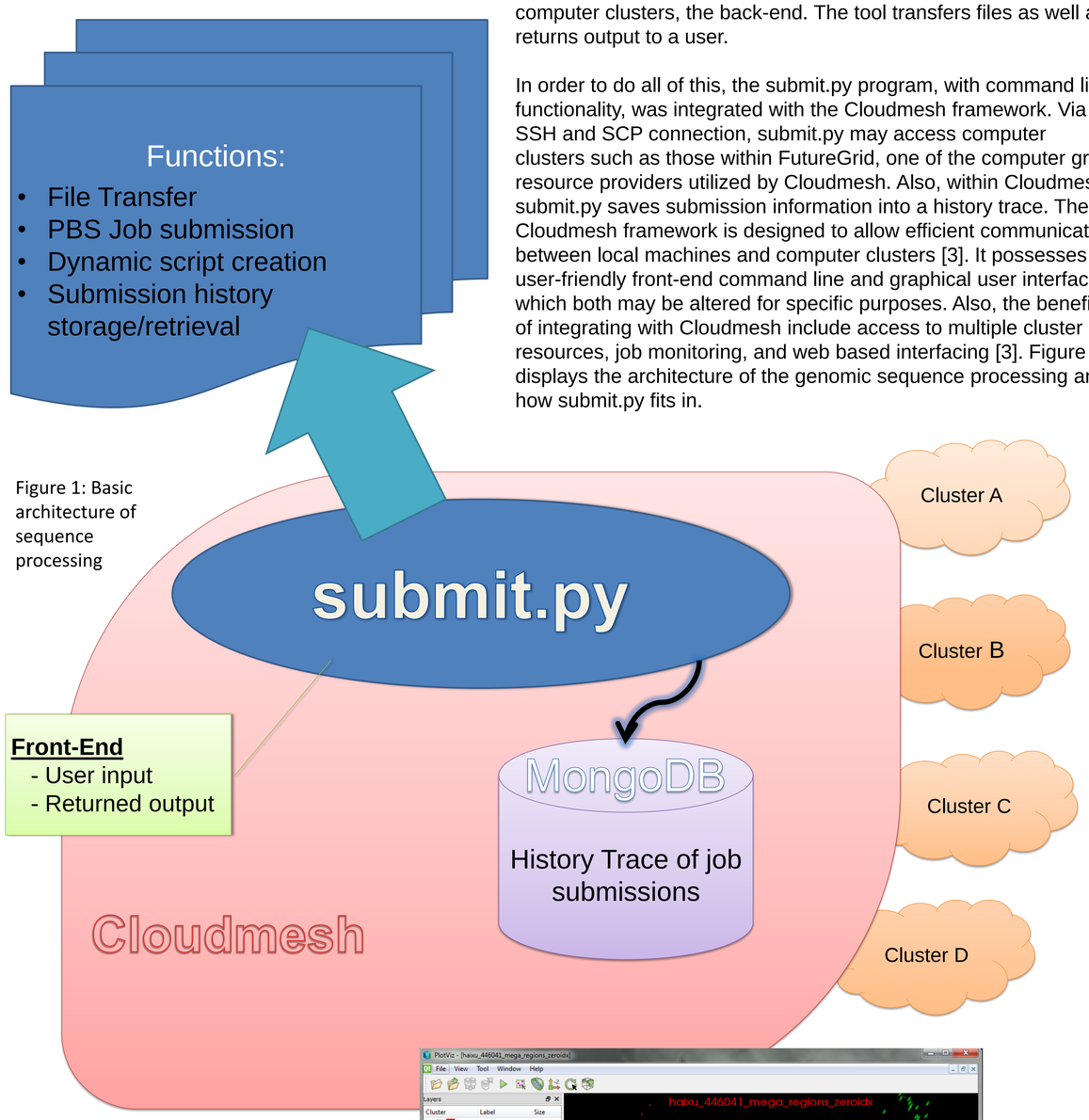

Figure 1: Basic architecture of sequence processing

submit.py is designed to accept Portable Batch System or PBS scripts from a user through the front-end and run them on computer clusters, the back-end. The tool transfers files as well as returns output to a user.

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 computer clusters such as those within FutureGrid, one of the computer grid resource providers utilized by Cloudmesh. Also, within Cloudmesh, submit.py saves submission information into a history trace. 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 processing and how submit.py fits in.

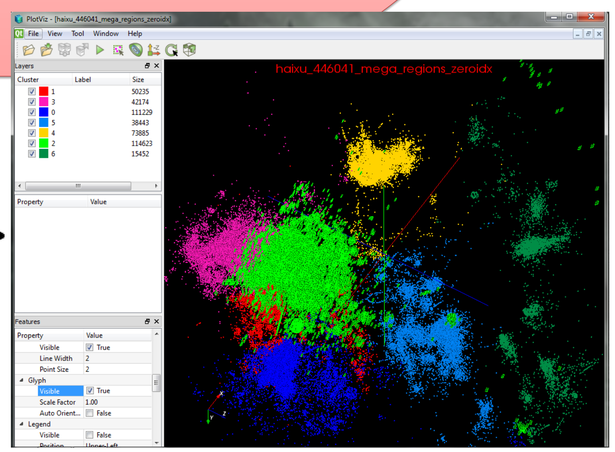

Figure 3: Example of a 3-D plot of genomic sequences

Conclusion

The results of processing genomic sequence 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
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

Primary Contact

Gregor von Laszewski, Indiana University, laszewski@gmail.com