Overlap Graph Reduction for Genome Assembly using Apache Spark

Alexander J. Paul
Bioinformatics and Computational Biology Program
Saint Louis University
St. Louis, MO 63103
apaul7@slu.edu

Dylan Lawrence
Computational and Systems Biology Program
Washington University in St. Louis
St. Louis, MO 63130
dylan.lawrence@wustl.edu

Tae-Hyuk Ahn
Department of Computer Science
Saint Louis University
St. Louis, MO 63103
ahnt@slu.edu

ABSTRACT

The advent of third-generation long-range DNA sequencing and mapping techniques has permitted nearly perfect or very high quality *de novo* assemblies of genomes. However, most overlap graph *de novo* assemblers still require large amounts of computer memory to resolve the large genome graphs. Here, we apply string graph reduction algorithms for genome assembly using Apache Spark on a distributed cloud computing platform.

CCS CONCEPTS

• **Applied computing** → **Life and medical sciences**; *Computational genomics* • **Mathematics of computing**→ Graph algorithms

KEYWORDS

genome assembly, overlap-layout-consensus, apache spark

1 INTRODUCTION

De novo genome assembly programs stitch together fragmented reads of DNA from an organism. Generally, de Bruijn graphs are a suitable approach for high-throughput short reads and overlap-layout-consensus (OLC) algorithm assemblers handle relatively long reads [1]. The assembly of the reads using OLC still struggles when the overlap graph is very large and the Layout step that tries to simplify and reduce the graph requires a vast amount of shared machine memory. Apache Spark is a new framework for fast and efficient data processing [2]. Using the Spark GraphX library, we have applied and tested several string graph reduction algorithms with a very large sample data set for efficient genome assembly on distributed cloud computing platform.

2 GRAPH REDUCTION AND BENCHMARK

3.1. Transitive edge reduction (TER)

Transitive edge reduction is a method of reducing complexity in graphs and helps provide clearer contigs when running OLC. In TER we eliminate extraneous paths in the graph. Formally, TER states that an edge of the graph G from a -> b may be removed if and only if the graph G' with edge a -> b removed still contains a path leading from a to b.

3.2. Composite edge contraction (CEC)

Composite edge contraction (CEC) is another method of reducing complexity. To simplify the overlap graph, a simple vertex, r, along with its only in-arrow edge (u, r) and only out-arrow edge (r,w), are replaced by a composite edge (u,w) in the overlap graph.

3.3. Benchmark

We have validated the scalability of the Layout module using the Spark GraphX library in the Amazon cloud using a sample dataset. This dataset was assembled by the proposed Spark algorithms using 15 virtual machines in 0.5 hours. Compared to the OLC based Omega [3] assembler's run time of 7.5 hours, the Spark based approach greatly improved the Layout step's runtime.

3 CONCLUSION

This work has focused on solving the computationally expensive task of genome assembly Layout. Apache Spark has enabled us to reduce the complexity and runtime of assembling large genomes.

ACKNOWLEDGMENTS

This work is supported in part by startup funds from Saint Louis University and the award NSF CRII-1566292.

REFERENCES

[1] Li, Z., Chen, Y., Mu, D., Yuan, J., Shi, Y., Zhang, H., Gan, J., Li, N., Hu, X., Liu, B., Yang, B. and Fan, W. Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-bruijn-graph. *Brief Funct Genomics*, 11, 1 (Jan 2012), 25-37.

[2] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S. and Stoica, I. Spark: cluster computing with working sets. In *Proceedings of the Proceedings of the 2nd USENIX conference on Hot topics in cloud computing* (Boston, MA, 2010). USENIX Association, [insert City of Publication],[insert 2010 of Publication].

[3] Haider, B., Ahn, T. H., Bushnell, B., Chai, J. J., Copeland, A. and Pan, C. L. Omega: an Overlap-graph *de novo* Assembler for Metagenomics. *Bioinformatics*, 30, 19 (Oct 1 2014), 2717-2722.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the Owner/Author.
ACM-BCB '17, August 20-23, 2017, Boston, MA, USA
© 2017 Copyright is held by the owner/author(s).
ACM ISBN 978-1-4503-4722-8/17/08.
http://dx.doi.org/10.1145/3107411.3108222