CS267 Final Project Proposal

Frank Austin Nothaft

1 Proposal

As seen in homework #3, genome assembly is a significant challenge. While homework #3 presented a de Bruijn based genome assembly method [8], "long read" assemblers tend to use overlap-layout concensus (OLC) based approaches [9]. As my CS267 project, I intend to build a distributed OLC assembler using Apache Spark [10]. This assembler targets the PacBio sequencing platform [3] which produces reads of approximately $\sim 10,000$ bases in length.

2 Components

This project involves several components:

1. **Read Overlapper:** I have already implemented a Spark-based overlapper. This tool uses a MinHash signature to compute approximate similarity of the reads [1].
2. **Graph Reduction:** Once the reads are overlapped, we need to simplify the overlap graph of the reads. To do this, I propose to perform a transitive reduction to a string graph-like [6] structure. I have sketched out how to perform this transitive reduction on a message-passing graph API [4] using at most two rounds of message passing.
3. **Assembly Polishing:** Although the PacBio sequencing platform produces very long reads, the reads have a high error rate. Once we have reduced down to a string graph, we plan to use a statistical model to pick the best consensus sequence per segment in the graph. We will start with a simple model, such as a reference-free implementation of the model used in the samtools mpileup variant caller [5].

These components will be implemented on top of the ADAM library [7], which provides primitives for manipulating genomic datasets inside of Spark.

3 Evaluation

We plan to evaluate this project on both performance and accuracy. We will use some of the reference datasets provided by PacBio [2], and will run our evaluation on the Amazon EC2 cloud. We will evaluate:

- Performance:
    1. Strong scaling on small and large genomes
    2. Weak scaling with increased genome size

3. Weak scaling with increased genome coverage

- Accuracy:
  1. Assembly length (N/NG50)
  2. Accuracy against known sequence, where available

References

[1] Berlin, K., Koren, S., Chin, C.-S., Drake, J., Landolin, J. M., and Phillippy, A. M. Assembling large genomes with single-molecule sequencing and locality sensitive hashing. *bioRxiv* (2014), 008003.

[2] Biosciences, P. Devnet—datasets. `https://github.com/PacificBiosciences/DevNet/wiki/Datasets`.

[3] Eid, J., Fehr, A., Gray, J., Luong, K., Lyle, J., Otto, G., Peluso, P., Rank, D., Baybayan, P., Bettman, B., et al. Real-time dna sequencing from single polymerase molecules. *Science 323*, 5910 (2009), 133–138.

[4] Gonzalez, J. E., Xin, R. S., Dave, A., Crankshaw, D., Franklin, M. J., and Stoica, I. Graphx: Graph processing in a distributed dataflow framework. In *Proceedings of the USENIX Symposium on Operating Systems Design and Implementation (OSDI '14)* (2014).

[5] Li, H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics 27*, 21 (2011), 2987–2993.

[6] Myers, E. W. The fragment assembly string graph. *Bioinformatics 21*, suppl 2 (2005), ii79–ii85.

[7] Nothaft, F. A., Massie, M., Danford, T., Zhang, Z., Laserson, U., Yeksigian, C., Kottalam, J., Ahuja, A., Hammerbacher, J., Linderman, M., Franklin, M. J., Joseph, A. D., and Patterson, D. A. Rethinking data-intensive science using scalable analytics systems. In *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '15)* (2015).

[8] Pevzner, P. A., Tang, H., and Waterman, M. S. An eulerian path approach to dna fragment assembly. *Proceedings of the National Academy of Sciences 98*, 17 (2001), 9748–9753.

[9] Rausch, T., Koren, S., Denisov, G., Weese, D., Emde, A.-K., Döring, A., and Reinert, K. A consistency-based consensus algorithm for de novo and reference-guided sequence assembly of short reads. *Bioinformatics 25*, 9 (2009), 1118–1124.

[10] Zaharia, M., Chowdhury, M., Franklin, M. J., Shenker, S., and Stoica, I. Spark: cluster computing with working sets. In *Proceedings of the USENIX Conference on Hot Topics in Cloud Computing (HotCloud '10)* (2010), p. 10.