Resource-efficient Assembly of Large Genomes with Bloom Filter ABySS

Ben Vandervalk, Hamid Mohamadi, Justin Chu, Shaun D Jackman, Golnaz Jahesh, Lauren Coombe, Rene L Warren, Inanc Birol

Michael Smith Genome Sciences Centre

April 29, 2016

Background

Since the introduction of the de Bruijn graph assembly approach by Pevzner et al. in 2001, de Bruijn graph assemblers have become the dominant method for de novo assembly of large genomes. Nonetheless, assembling large genomes remains a challenging task. For instance, the estimated memory requirements for a human genome assembly with the ALLPATHS-LG assembler is 512GB of RAM. While distributed de Bruijn graph assemblers such as ABySS, Ray, and PASHA eliminate the requirement for a single large-memory machine by distributing the de Bruijn graph across multiple cluster nodes, these assemblers still require a computing cluster with a large amount of aggregate memory and a high-speed network fabric. While assemblers typically represent the de Bruijn graph as a hash table of k-mers, the Minia assembler (Chikhi et al., 2012) introduced a more compact probabilistic representation using a Bloom filter, which reduces the memory requirement by orders of magnitude and renders large genome assemblies feasible on a single commodity machine.

Results

Here we present two fundamental improvements to the ABySS assembler that reduce the memory and running time for large genome assemblies. First, as in Minia, we have reduced memory requirements by an order of magnitude through the use of a Bloom filter de Bruijn graph. While Minia

is a standalone unitig assembler, our new Bloom filter assembler is integrated with the existing ABySS pipeline, including downstream stages for contig building, mate pair scaffolding, and long read scaffolding. Second, we have reduced assembly time through the use of a specialized hash function called "ntHash". In our application, ntHash achieves runtimes that are orders of magnitude faster than standard hash functions through the use of a constant-time sliding window calculation, where the hash value of each k-mer is computed from the hash value of the k-mer that preceeds it. On a single 32-core machine with 120GB RAM, the new Bloom filter version of ABySS is able to assemble a modern 76X human dataset (SRA:ERR309932) and scaffold with MPET data (SRA:ERR262997) with an NG50 of 1.7 Mbp, wallclock time of 46 hours, and a peak memory usage of 102GB RAM.

Conclusions

While many implementations of de Bruijn graph assemblers are available, de novo assemblies of large genomes such as Homo sapiens still require heavy computational resources. Here we have demonstrated improvements to ABySS with respect to both memory usage and running time that significantly reduce the cost of assembling large genomes.