Scaling ABySS to longer reads using spaced k-mers and Bloom filters

Shaun D Jackman, Karthika Raghavan, Benjamin P Vandervalk, Daniel Paulino, Justin Chu, Hamid Mohamadi, Anthony G Raymond, René L Warren, Inanç Birol
BC Cancer Agency Genome Sciences Centre, Vancouver, BC, Canada

2015-02-21

Adapting to the continually changing landscape of sequencing technology is a particular challenge when maintaining an assembly software package such as ABySS that spans years of development. It also offers opportunities for better assemblies if new algorithms capitalize on the technology improvements.

Illumina read lengths were shoter than 50 nucleotides at the initial release of ABySS, and overlapping MiSeq reads now exceed 500 nucleotides. ABySS and other de Bruijn graph (dBG) assemblers use a hash table to store k-mers, sequences of k nucleotides. A standard hash table requires memory that scales with the value of k. To make better use of longer read lengths without a commensurate increase in memory requires space-efficient data structures. In a new release of ABySS, we use spaced seeds to represent large k-mers while storing a fraction of their nucleotides. For example, two 32-mer separated by a space of 300 nucleotides represents a dBG comparable to a 364-mer dBG, while using the memory of a 64-mer dBG.

We also introduce an assembly finishing tool to close scaffolding gaps in draft assemblies. The Sealer algorithm fills these gaps by navigating a dBG represented probabilistically by a Bloom filter. Because a Bloom filter is space-efficient, we can employ multiple such filters, using smaller k to span regions of low coverage and larger k to resolve repeats.

We assemble *Escherichia coli* overlapping MiSeq reads with ABySS producing an assembly with a contig NGA50 of 176 kbp and no misassembled contigs, shown in Figure 1. We assemble *Caenorhabditis elegans* Illumina TruSeq Synthetic Long Reads and Illumina mate pair reads with ABySS producing an assembly with a scaffold NGA50 of 200 kbp. ABySS is a flexible assembly pipeline that may be used to assemble a variety of sequencing data types and read lengths, from 500 bp overlapping MiSeq reads to 10 kbp pseudo-long reads. The assembly algorithms of ABySS will scale to exploit the length of the long reads from

PacBio and Oxford Nanpore, though correcting the sequencing errors from these technologies remains a challenge.

We present in this work the performance of ABySS, with a detailed look at the data structures used, and the utility of automated finishing. We demonstrate the scalability of these efficient tools to long reads and large genomes.

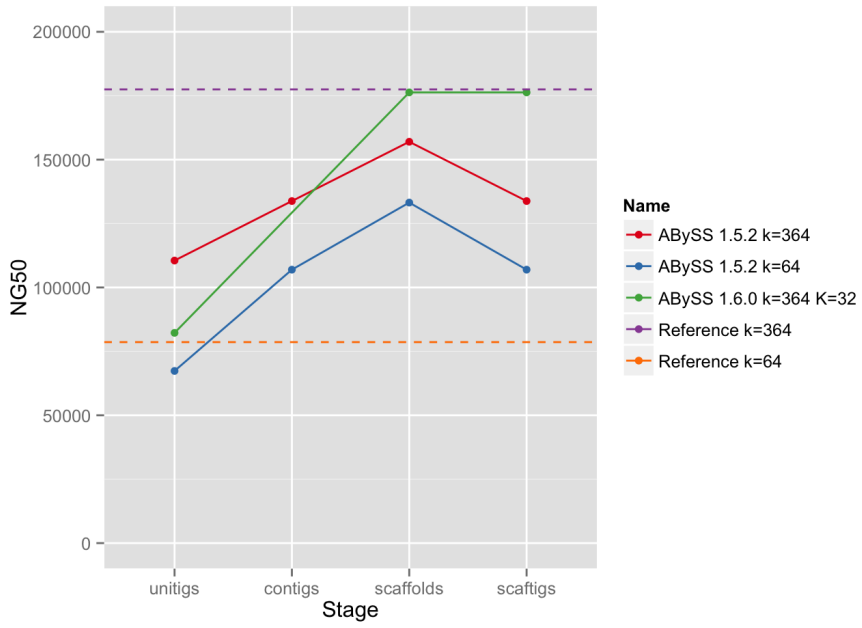

Figure 1: Assembly of *E. coli* overlapping MiSeq reads with both ABySS 1.5.2 using a standard de Bruijn Graph and ABySS 1.6.0 using a spaced-seeds de Bruijn Graph. We also reassemble the reference genome using ABySS, disabling all error-removal algorithms, to show the best possible assembly for that value of k.