Chapter 6

Conclusions

In this work, I have developed assembly and variant calling algorithms based on the compressed FM-index data structure. Using the algorithms developed, I performed the first overlap-based assembly of a human genome from Illumina sequence reads [Simpson and Durbin, 2012]. Subsequent to the publication of my assembly method in [Simpson and Durbin, 2010], other groups have followed a similar approach. Dinh and Rajasekaran [2011] developed an efficient data structure for representing an exact-match overlap graph. Gonnella and Kurtz extended my idea of directly outputting only the irreducible edges of a string graph to develop a fast string graph construction algorithm [Gonnella and Kurtz, 2012]. Heng Li reformulated the algorithms in chapter 2 and 3 based on a new representation of the FM-index which stores the read sequences and their reverse complement in the same data structure [Li, 2012].

I have extended upon the *de novo* assembly algorithms to perform comparative variant calling between two genomes. The initial results presented in chapter 5 suggest this is a promising approach for finding relatively complex differences between the pair of genomes. I believe that methods which work directly with sequencing reads, rather than relying on alignments to a reference genome, will become increasing important as sequencing technology improves.

High-throughput short read sequencing profoundly changed genomics. New algorithms needed to be developed to cope with the volumes of data. As sequencing costs continue to fall and more genomes are sequenced there will be constant pressure to lower the computational cost of sequence analysis. One approach that

has recently become prominent is to use probabilistic data structures, such as the bloom filter. This approach has been shown to lower the memory requirements of representing a de Bruijn graph [Chikhi and Rizk, 2012; Pell et al., 2012]. I believe these approaches are complimentary to the FM-index algorithms developed in this work. For example, one could use a bloom filter when performing k-mer based error correction and the FM-index when assembling the corrected reads into contigs. These approaches can easily be implemented within our software framework, which is designed as a modular pipeline.

As the third generation of sequencing technology is developed, which is projected to be based on directly reading the sequence of DNA as it passes through a biological nanopore, the algorithmic landscape will change again. Already read lengths of up to 48kbp have been publicly discussed using nanopore approaches [Schneider and Dekker, 2012]. The algorithmic challenge of indexing the data and constructing an assembly graph will remain, and I believe the FM-index and string graph algorithms presented in this work are well-suited for this task. With >10kb reads, genomic repeats become far less of a barrier to reconstructing the complete sequence of a large genome. I believe the core algorithmic challenge in assembly will not be to simply reconstruct the full sequence of a genome but to reconstruct the full haplotype-resolved *phased* genome of diploid organisms. The logical starting point of all sequence analysis should be the complete genetic content of a cell, and I believe this goal is not far off.