Comparison of BWT Construction Algorithms for Very Large Read Collections

Markus J. Bauer[1], Anthony J. Cox[1] and Dirk J. Evers[1]

Illumina Cambridge Ltd.
Chesterford Research Park
CB10 1XL
United Kingdom
{mbauer,acox,devers}@illumina.com

With the advent of next-generation DNA sequencing technologies, datasets containing billions of reads have become commonplace. Advanced text indexing methods are needed to process such large datasets efficiently.

One universal tool in text indexing is the *Burrows-Wheeler transform* (BWT) which is the core of the FM index [2]. Popular alignment programs (like bowtie [4] or bwa [5]) use the FM index of the human genome to perform read mapping. An alternative application was recently presented by Simpson and Durbin [6] who describe an assembler that uses the FM index of a set of reads to find overlaps between them and hence construct the string graph. The BWT of the human genome can be built within hours on a standard PC today, but the computational requirements increase substantially when switching to large read collections.

In recent years, a number of algorithms have been published that construct the BWT either directly, like `bwtdisk` [1], or by constructing a suffix array first and retrieving the BWT afterwards. Examples of the latter approach include `rlcsa` by Sirén [7] or Simpson's implementation of the Ko-Aluru algorithm [3].

In this contribution, we compare various methods to create the BWT and we chart their computational requirements in terms of time and memory. We show what data sets can be processed on a machine with 16Gb of memory and describe the main bottlenecks for creating the BWT on very large read collections.

References

1. P. Ferragina, T. Gagie, and G. Manzini. Lightweight data indexing and compression in external memory. In *LATIN*, volume 6034 of *LNCS*, pages 697–710. Springer, 2010.
2. P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 390–, Washington, DC, USA, 2000. IEEE Computer Society.
3. P. Ko and S. Aluru. Space efficient linear time construction of suffix arrays. *Journal of Discrete Algorithms*, 3(2-4):143 – 156, 2005.
4. B. Langmead, C. Trapnell, M. Pop, and S. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):R25+, 2009.
5. Heng Li and Richard Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics (Oxford, England)*, 25(14):1754–1760, July 2009.
6. J. T. Simpson and R. Durbin. Efficient construction of an assembly string graph using the FM-index. *Bioinformatics*, 26(12):i367–i373, June 2010.
7. J. Sirén. Compressed suffix arrays for massive data. In *SPIRE*, volume 5721 of *LNCS*, pages 63–74. Springer, 2009.