UNIVERSITY OF CAMBRIDGE
Metabolic Research Laboratories

IMS Institute of Metabolic Science

BarraCUDA - a Fast Sequence Mapping Software using Graphics Processing Units

Brian Y. H. Lam, Petr Klus, Simon Lam and Giles S. H. Yeo
Metabolic Research Laboratories, Institute of Metabolic Science, University of Cambridge, Addenbrooke's Hospital, Cambridge, CB2 0QQ, United Kingdom
Email addresses: Brian Lam (yhbl2@cam.ac.uk), Giles Yeo (gshy2@cam.ac.uk)

Introduction

High-throughput DNA sequencing (HTS) instruments today are capable of generating millions of sequencing reads in a short period of time, and this represents a serious challenge to current bioinformatics pipeline in processing such an enormous amount of data in a fast and economical fashion.

Modern graphics cards are powerful processing units that consist of hundreds of scalar processors in parallel in order to handle the rendering of high-definition graphics in real-time. It is this computational capability that we propose to harness in order to accelerate some of the time-consuming steps in analyzing data generated by the HTS instruments.

We have developed BarraCUDA, a novel sequence mapping software that utilizes the parallelism of NVIDIA CUDA graphics cards to map sequencing reads to a particular location on a reference genome. While delivering a similar mapping fidelity as other mainstream programs , BarraCUDA is a magnitude faster in mapping throughput compared to its CPU counterparts. The software is also capable of supporting multiple CUDA devices in parallel to further accelerate the mapping throughput.

High-throughput DNA Sequencing

HTS is a technique based on sequencing by synthesis/ligation in a massively parallel fashion. The sequencing throughput has increased dramatically since its introduction in 2005. Today, a state-of-art HTS instrument can generate DNA sequences that cover 30 times the Human genome in just a day.

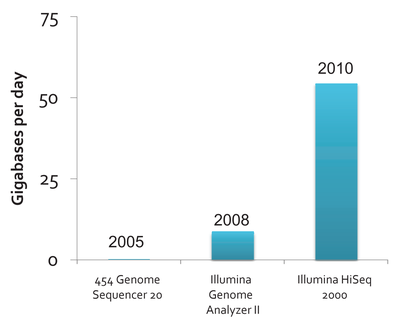

Figure 1 The throughput of DNA sequencing has soared since the introduction of the Genome Sequencer 20 back in 2005. Today's state-of-art HTS instrument can generate 600 gigabases (6 x 10^{11} base pairs, or bp) of DNA sequences in a 11-day run .

Acknowledgments

We would like to thank Graham Pullan, Tobias Brandvik, Ian McFarlane, Dag Lyberg, Simon Lam, Nicole Cheung, Thomas Bradley, and Timothy Lanfear for their help and support for the project. We would also like to thank NVIDIA for providing GPU hardware and access to their PSG cluster. This work is supported by the Medical Research Council Centre of Obesity and Related Metabolic Diseases, EurOCHIP FP7 Consortium and a funding to the Cambridge Biomedical Research Centre, NIHR Cambridge, UK.

Inexact Matching Using FM-index Algorithm

The FM-index[1] algorithm is a substring index based on Burrows-Wheeler transform[2] (BWT) and it allows fast substring matching by prefix *trie* traversal. The algorithm is widely used in DNA sequence mapping programs such as BWA[3] and Bowtie[4], where the string is the reference genome and the query substrings are the sequecing reads generated by HTS instruments

Exact string matching is performed by 'backward search' through a BWT prefix *trie*, e.g. 'banan'.

For inexact matching, a series of substitutions are introduced in the query string such that the string with a substituted character could lead to a match, e.g. 'b' in 'anb' is substituted with an 'a' to give an match 'ana' with one mismatch.

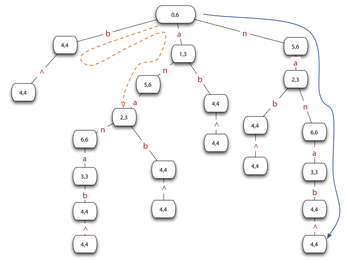

Figure 2 A BWT prefix *trie* for string 'banana'. '^' marks the start of the string. Substring matching is performed by traversal of the *trie* from the root (0,6), in 'backward' direction from the last character of the query string. The blue line indicates the traversal for matching the query 'banan'; The orange line indicates the route to match 'anb'. Since there is no exact match for the query, the 'b' is substituted with an 'a' to give a match 'ana' with one mismatch.

CUDA Implementation

Substring matching is an embarassingly parallel process with no data dependancy. Therefore we used a straight-forward data parallelism and assign a GPU thread to each of the sequencing reads (query strings).

A depth-first search (DFS) strategy is used for prefix *trie* traversal where the memory requirement for each GPU thread is minimal. For long sequencing reads, the query is divided into several short fragments and matching is performed by consecutive kernel runs to prevent thread divergence.

Mapping Accuracy

	BWA	BarraCUDA
% Mapped	96.50	96.64
% Error	0.04	0.06

Table 1 It is important that the mapping accuracy is not compromised by using GPUs. To test this, we compared the mappings generated by BarraCUDA to those from a commonly used mapping software BWA. A library containing 1 million simulated reads of 70 bp was used in this comparison.

Mapping Speed

The mapping throughput was examined by mapping a HTS sequencing library containing 14 million reads (ENA accession: SRR063699) to the genome of of *Drosophila Melanogaster*. The mapping throughput of BarraCUDA with one GPU was about 5X the speed of BWA with a single CPU core. With 8X GPUs, BarraCUDA outperforms BWA using all 12 CPU cores by 2.8 fold.

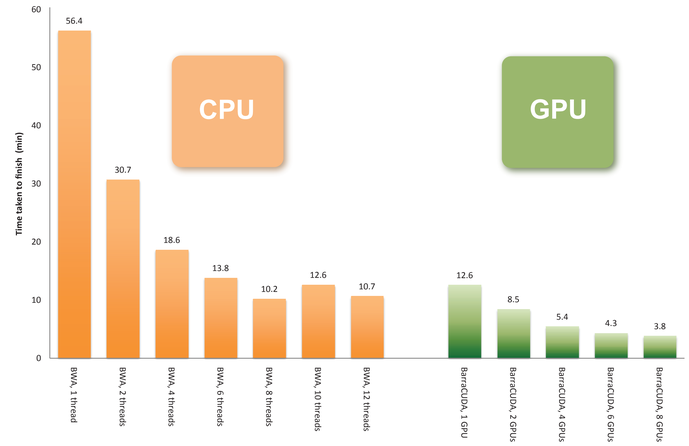

Figure 3 A comparison of mapping performance between BWA (CPU, orange bars) and BarraCUDA (GPU, green bars). A sequencing read library containing 14 million 95 bp reads was mapped onto the *Drosophila Melanogaster* genome. The test was performed on a dual-socket computer node containing 2x 6-core Intel Xeon X5670 (2.93GHz) and 8X NVIDIA Tesla C2050s

Conclusions

BarraCUDA is designed to take advantage of the parallelism of GPU to accelerate the mapping of millions of sequencing reads generated by HTS instruments. By doing this, we could, at least in part streamline the current bioinformatics pipeline such that the wider scientific community could benefit from the sequencing technology. BarraCUDA is currently available at http://seqbarracuda.sf.net

References

1. **Ferragina P, Manzini G.** Opportunistic data structures with applicaions. *41st Annual Symposium of Foundations of Computer Science*, 2000, Redondo Beach, California, IEEE: 390-398.

2. **Burrows M, Wheeler D.** A Block-sorting lossless data compression algorithm. *Technical Report*: Digital Equipment Corporation, 1994.

3. **Li H, Durbin R.** Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 2009, 25(14):1754-1760.

4. **Langmead B, Trapnell C, Pop M, Salzberg SL.** Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 2009, 10(3):R25.