Benchmarking Genetically Improved BarraCUDA on Epigenetic Methylation NGS datasets and nVidia GPUs

William B. Langdon, Albert Vilella*, Brian Yee Hong Lam†, Justyna Petke, Mark Harman
Department of Computer Science, University College London, Gower Street, WC1E 6BT, UK.
*Cambridge Epigenetix, Babraham Research Campus, Cambridge, CB22 3AT, UK.
†University of Cambridge Metabolic Research Laboratories, Addenbrooke's Hospital, Cambridge, CB2 0QQ, UK.
W.Langdon@cs.ucl.ac.uk

ABSTRACT

BarraCUDA uses CUDA graphics cards to map DNA reads to the human genome. Previously its software source code was genetically improved for short paired end next generation sequences. On longer noisy epigenetics strings using nVidia Titan and twin Tesla K40 the same GI-ed code is more than 3 times faster than bwa-meth on an 8 core CPU.

Keywords

Genetic programming; Genetic Improvement; GPGPU; SBSE; parallel processing; bisulfite conversion; Bioinformatics;

1. INTRODUCTION

The GI [2] version of BarraCUDA [1, 3] has been incorporated into the main release on SourceForge. The GI version was downloaded more than 1000 times within a year. It was optimised for aligning short next generation (NGS) DNA reads to the reference human genome. The standard way of handling epigenetics data is very similar. However, it requires the creation of a reference epigenetics "human genome" that is approximately twice as big. (See top Figure 2 and Appendix B.) Fortunately, with the advent of nVidia Tesla K40 and Titan GPUs, it is now possible to place epigenetics reference datasets onto GPUs.

The next section describes using multiple GPUs and overlapping processes on multicore computers. Section 3 gives the results. (The epigenetics data are given in appendices.)

2. PIPING 2 GPUS AND 3 PROCESSES

BarraCUDA consists of a number of components (see Figure 2): `aln` aligns hundreds of thousand of short strings against the reference genome in parallel using a GPU. It produces up to ten potential alignments for each query string and passes these on to the `sampe` (SAM paired-end) process, which takes two sets of alignments and reports the best compatible pairings.

Both `aln` and `sampe` have to do more work if the data are noisy. With reasonable DNA data, `sampe` essentially only has to convert the binary data generated by `aln` to SAM format text and runtime is dominated by the two `aln` processes. Hence given two GPUs it is possible to run all three processes in parallel and on a multi-core server generating the SAM output file has little impact on run-time. (A unix bash script is given in Figure 3.)

The epigenetics data are considerably noisier. The usual default for BarraCUDA is to allow only two mismatches. With noisier data, it must be instructed (via the `-n` switch) to allow more. It takes more searching to find strings which diverge from the reference genome in more places so slowing down `aln`. Also typically noise increases the number of po-

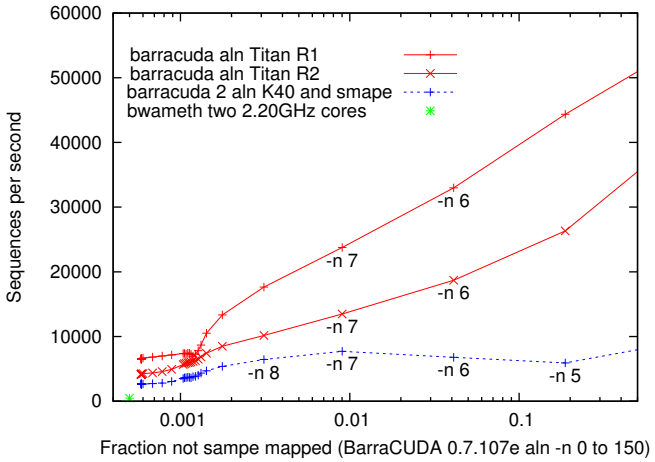

Figure 1: Speed of BarraCUDA on 10 million simulated epigenetics paired end reads. Solid lines are the speed of BarraCUDA alignment on an nVidia 12 GByte GTX Titan. Dotted line is for complete generation of SAM output (i.e. two `aln` and a `sampe` process) running on a two nVidia Tesla K40 on an 8 core server. bwa-meth (*) on twin core CPU.

Table 1: GPU Hardware. Each GPU chip (compute capability, column 2) contains 15 or 24 identical independent multiprocessors (MP, column 3). Each MP contains 192 or 128 stream processors (total given in column 5), whose clock speed is given in column 6. Memory error ECC enabled.

GPU	MP	cores	total	Clock	Mem	Bandwidth
K40	3.5	15 × 192 =	2880	0.88 GHz	11 GB	180 GB/s
Titan	5.2	24 × 128 =	3072	1.22 GHz	12 GB	235 GB/s

GECCO'16 Companion, July 20 - 24, 2016, Denver, CO, USA

ISBN 978-1-4503-4323-7/16/07…$15.00

DOI: http://dx.doi.org/10.1145/2908961.2931687

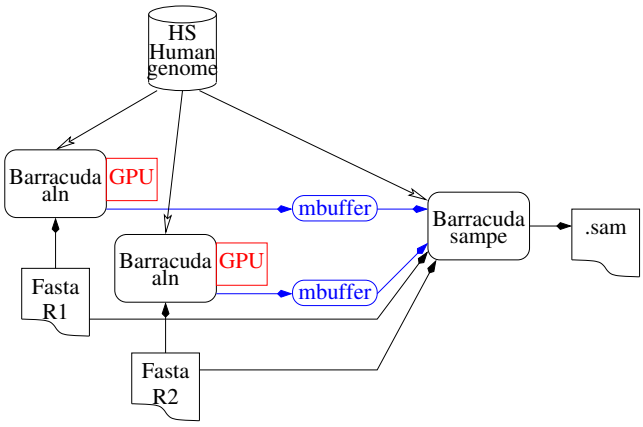

Figure 2: Processing paired end epigenetics reads. `aln` is run twice (once per end) and its alignments are piped (blue arrows) via mbuffer into `sampe`. `sampe` also reads the reference index (HS Human genome) and both ends of each methylated DNA sequence (Fasta R1 and R2) in order to give the combined alignment in SAM format. The two `aln` process each use a GPU and `sampe` uses multiple host threads.

tential approximate matches so causing more work for the `sampe` process. Although BarraCUDA `sampe` can use multiple CPU threads, noisy data slows it considerably, until `sampe` becomes the bottle-neck.

With paired epigenetics reads, the R1 and R2 DNA typically have different amounts of noise meaning one of the `aln` processes has much more work to do than the other. (See the separation of the two `aln` plots in Figure 1.)

2.1 Unix Pipes, mbuffer

The two `aln` processes generate several megabytes of binary data every time they finish aligning a group of about a hundred thousand reads. As `sampe` slows down it fails to read these data as fast as they are generated so causing both `aln` to wait until their output is read.

On our unix server a system pipe only provides a 4096 byte buffer. We increased this 10 000 fold by using `mbuffer -b 10000` with each `aln` output pipe. This does not solve the problem entirely, `sampe` must still eventually process all the data output by the two `aln` but it does mean even when the `aln` are actively finding alignments the `sampe` process can continue and can deal better with the now different speeds of the two `aln` processes.

3. RESULTS

BarraCUDA was run on two cluster nodes. The first housing a GTX Titan and the second an 8 core server with twin Tesla K40s. The upper lines in Figure 1 show the raw performance of the Titan (output stored on disk to be processed by `sampe` later). With the default number of mismatches (`-n 2`) `aln` is blisteringly fast but almost no matches are found. `-n` must be increased to 7 to get reasonable matches, whereupon the Titan processes 24 000 sequences per second (R1, K40 15 000) and 13 000 (R2, K40 8 600). On the 8 core server, the process is dominated by `sampe` but it still achieves 7 700 pairs of sequences per second.

Bwa-meth (`https://github.com/brentp/bwa-meth`) is a leading open source python script for aligning bisulfite DNA

```
./sampe sampe $hg19 \
<(mbuffer -b 10000 < <(./aln1 aln -n 7 -C 0 $hg19 $seq1) \
<(mbuffer -b 10000 < <(./aln2 aln -n 7 -C 1 $hg19 $seq2) \
$seq1 $seq2 > $sam
```

Figure 3: Bash command line using process substitution, pipes, mbuffer and input-output redirection to run two `aln` processes in parallel with `sampe`. `$hg19` the location of the reference genome index, `$seq1` and `$seq2` are the files holding the pairs of epigenetics fasta sequences and `$sam` is the output. (`./aln1`, `./aln2` and `./sampe` are all BarraCUDA. Renaming it makes unix `top` etc. easier to understand.)

strings. On a twin 2.20 GHz core cluster node bwa-meth processed 475 pairs of epigenetics sequences per second. (Note it maps almost all of them). Even assuming linear scale up to 8 cores and proportionate clock speeds, the two Tesla K40s would be 3.2 times faster.

4. CONCLUSIONS

For typical epigenetics DNA on modern GPUs the released version of BarraCUDA (which was GIed last year) is faster than a leading human coded CPU only alternative and gives acceptable performance. Mapping noisy next generation sequences is dominated by human coded heuristics. For GPU and multi-core CPUs the appendices offer labelled data for principled exploration of the performance efficiency speed v. accuracy Pareto trade-off in BarraCUDA, bwa-meth, etc.

Acknowledgements

I would like to thank Bobby R. Bruce, Neil Daeche and Tristan Clark of UCL, and Thomas Maier-Komor for mbuffer. Teslas donated by nVidia.

References

[1] P. Klus et al. BarraCUDA. *BMC Res Nts*, 5(27), 2012.

[2] W. B. Langdon. Genetically improved software. *Handbook GP Applications*, pp181–220. Springer, 2015.

[3] W. B. Langdon, Brian Y. H. Lam, J. Petke, and M. Harman. Improving CUDA DNA analysis software with genetic programming. In *GECCO 2015*, pp1063–70

A. EPIGENETICS FASTA SEQUENCES

We generated ten million pairs of synthetic epigenetics data, each end comprising 150 base pairs. The separation between the two ends was randomly chosen to be between 150 and 500 base pairs. The overall error rate was 0.8% with a bisulfite conversion rate of 95%. Data available via: `https:// s3-eu-west-1.amazonaws.com/cegx-test-001/tmwg-example-files/SIM03_S1_L001_R1_001.fastq.gz` `https://s3-eu-west-1.amazonaws.com/cegx-test-001/tmwg-example-files/SIM03_S1_L001_R2_001.fastq.gz`.

B. EPIGENETIC (HS) HUMAN REFERENCE GENOME

BarraCUDA was used to build its index file from the Epigenetics fasta sequences generated from the reference human genome (38DH). It (hs38DH) is 6 Gbytes. I.e. about twice the size of usual BarraCUDA index. It may be down loaded from `https://s3-eu-west-1.amazonaws.com/cegx-test-001/tmwg-example-files/hs38DH_bwameth.tar.gz`.