BIOINFORMATICS APPLICATIONS NOTE

Vol. 30 no. 7 2014, pages 1031–1033
doi:10.1093/bioinformatics/btt736

Systems biology

Advance Access publication December 19, 2013

GPU-Meta-Storms: computing the structure similarities among massive amount of microbial community samples using GPU

Xiaoquan Su[†], Xuetao Wang[†], Gongchao Jing and Kang Ning[*]

Shandong Key Laboratory of Energy Genetics, CAS Key Laboratory of Biofuels and Bioenergy Genome Center, Computational Biology Group of Single Cell Center, Qingdao Institute of Bioenergy and Bioprocess Technology, Chinese Academy of Sciences, Qingdao 266101, P. R. China

Associate Editor: Igor Jurisica

ABSTRACT

Motivation: The number of microbial community samples is increasing with exponential speed. Data-mining among microbial community samples could facilitate the discovery of valuable biological information that is still hidden in the massive data. However, current methods for the comparison among microbial communities are limited by their ability to process large amount of samples each with complex community structure.

Summary: We have developed an optimized GPU-based software, GPU-Meta-Storms, to efficiently measure the quantitative phylogenetic similarity among massive amount of microbial community samples. Our results have shown that GPU-Meta-Storms would be able to compute the pair-wise similarity scores for 10 240 samples within 20 min, which gained a speed-up of >17 000 times compared with single-core CPU, and >2600 times compared with 16-core CPU. Therefore, the high-performance of GPU-Meta-Storms could facilitate in-depth data mining among massive microbial community samples, and make the real-time analysis and monitoring of temporal or conditional changes for microbial communities possible.

Availability and implementation: GPU-Meta-Storms is implemented by CUDA (Compute Unified Device Architecture) and C++. Source code is available at http://www.computationalbioenergy.org/meta-storms.html.

Contact: ningkang@qibebt.ac.cn

Received on October 31, 2013; revised on November 29, 2013; accepted on December 15, 2013

1 INTRODUCTION

Because most microbes are not isolatable and cultivatable (Jurkowski *et al.*, 2007), metagenomic methods have been used to analyze a microbial community as a whole. Next-generation sequencing techniques (Mardis, 2008) have enabled the fast profiling of structures for a large number of microbial communities. Thus, a rapidly increasing amount of metagenomic profiles for microbial communities have been archived in public repositories and research laboratories around the world, such as MG-RAST (Meyer *et al.*, 2008) and CAMERA2 (Seshadri *et al.*, 2007), while NCBI (http://www.ncbi.nlm.nih.gov/) also contains thousands of metagenomic related projects with >100 000 samples. With such a large volume of metagenomic profiling data, it has become possible and important to compare the complex structures of microbial communities in large scale for in-depth data mining to discover precious biological patterns hidden in those massive data.

A number of methods have been proposed for comparison of different metagenomic samples. MEGAN (Huson *et al.*, 2011) can compare multiple samples based on taxonomy levels without considering phylogenetic relationships among taxa. STAMP (Parks and Beiko, 2010), METAREP (Goll *et al.*, 2010) and MetaRank (Wang *et al.*, 2011) process metagenomic data using standard statistical tests (mainly *t*-tests and principal component analysis with some modifications). Some 16S rRNA amplicon analysis toolkits, such as Mothur (Schloss *et al.*, 2009) and QIIME (Caporaso *et al.*, 2010), include UniFrac (Lozupone and Knight, 2005) and Fast UniFrac (Hamady *et al.*, 2010) algorithms, which use distances among species to make phylogenetic beta diversity measurement more effective at showing ecological patterns. Nevertheless, the restriction in the amount of samples for comparison, as well as running speed, limits the extension of these tools on analysis of huge number of microbial community samples. Moreover, current needs for real-time monitoring of temporal or conditional changes for microbial communities in environment and energy research areas have made these methods inadequate.

Recently, we have proposed a more efficient method, Meta-Storms (Su *et al.*, 2012a), for quantitative comparison between microbial community samples based on a binary phylogenetic tree with a time complexity of $N\log(N)$ (N is the number of species in one sample). However, as the amount of samples increases, the overall time complexity of $M^2\times\ N\log(N)$ (M is the number of samples) for the pair-wise comparison always leads to an unacceptable running time. Therefore, such computing-intensive tasks require novel ultra-high processing throughput.

2 MATERIALS AND METHODS

GPU-Meta-Storms implements the scoring function of Meta-Storms (Su *et al.*, 2012a) with non-recursive transformation, CUDA (Compute Unified Device Architecture)-based parallel programming and optimizations. The synergy of these techniques enabled ultra-high processing speed to evaluate the similarities among large amount of microbial community samples.

[*] To whom correspondence should be addressed.

[†] The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First Authors.

2.1 Non-recursive transformation

The scoring function of Meta-Storms compares two microbial community samples' structures by bottom–up recursive traversal to their common weighted phylogenetic tree, in which edge weight represents the phylogenetic distance and node weight represents the species abundance, and calculates an overall similarity score (between 0 and 100%).

Though GPU hardware and CUDA technology are good at efficient processing of computing-intensive works in parallel, the above recursive traversal would be limited by the stack size in GPU. Therefore, Meta-Storms scoring function needs to be transformed into non-recursive format (Segovia *et al.*, 2009). Here, we re-sorted all nodes by post-order traversal to the phylogenetic tree, so that all calculations can be processed by serially accessing the sorted nodes without recursive overlap.

2.2 CUDA-based parallel programming implementation

Benefited by the many-core architecture of GPU, scoring function can be invocated in parallel by large number of threads to compute the similarities among massive amount of samples. For the synchronization of many-core programming, we map all samples to Greengenes CoreSet tree (release date: May 2009) (DeSantis *et al.*, 2006) as the reference phylogenetic tree.

To calculate the pair-wise similarity scores (also referred to as 'similarity matrix') of N samples, $N\times N$ threads are launched in GPU to make each similarity score in the matrix processed by one independent thread. Abundance values of species of each sample and phylogenetic distances are firstly loaded from the file system and initialized in RAM by CPU, then sent to graphics processing unit (GPU) on-board RAM for parallel computing. After all threads of GPU finish the tasks, all elements of similarity matrix are sent back to RAM, and stored in file system on hard disk.

2.3 CUDA-based optimization

Limited by the low I/O bandwidth of GPU on board RAM (also referred to as 'global memory' in CUDA), we have designed the following optimization methods to improve the execution efficiency of GPU-Meta-Storms on CUDA.

(1) Global memory alignment: Because all threads calculate the same phylogenetic tree with the same nodes order, their abundance values can be sorted in the same order for global memory alignment to accelerate both the transmission from RAM to GPU and the global memory access by GPU.

(2) Register recycling allocation: All temporary results of internal nodes of the phylogenetic tree are kept into registers, of which the I/O speed is ≈100 times faster than global memory. The register recycling strategy reduces the required register number to only 10 to adapt to the CUDA.

(3) Application of shared memory: For all threads calculating on the same phylogenetic tree, distance values are stored into shared memory, which can be accessed by all threads with low I/O latency.

2.4 Input and output

GPU-Meta-Storms accepts each microbial community sample as an identical plain-text file with the identified reference phylogenetic tree IDs of all species in it, which can be generated by Parallel-META (Su *et al.*, 2012b) from metagenomic shotgun or 16S rRNA sequence data. GPU-Meta-Storms outputs the pair-wise similarity scores of all input samples into a plain-text file, in which each element is a float number that represents the similarity for a pair of samples between 0 and 100%. Based on permutation test results (Su *et al.*, 2012a), a similarity score of 85% or higher indicates significant similarity between two samples.

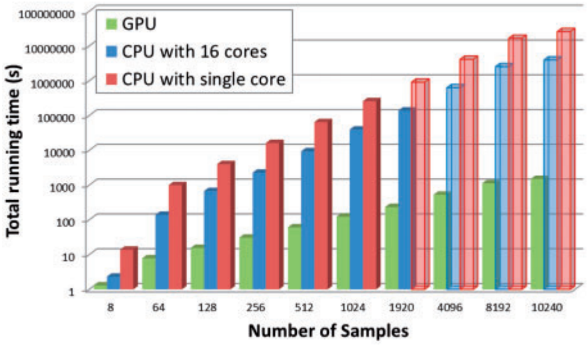

Fig. 1. Overall running time of similarity matrix computing by GPU and CPU. The Y-axis is in 10-based log scale. Framed bars indicate the estimated running time

3 RESULTS AND DISCUSSIONS

Different number of human habitat microbial community samples from the project 'Moving pictures of human microbiome' (Caporaso *et al.*, 2011) have been randomly selected (samples are available on the software website) to evaluate the performance of GPU-Meta-Storms in comparison of microbial community samples. All experiments in this work were completed on a rack server with dual Intel Xeon E5-2650 CPU, 64GB DDR3 ECC RAM, NVIDIA M2075 GPU and 1TB hard drive in RAID 1.

3.1 Running time comparison with CPU-based computation

In this work, we compared the running time of GPU-Meta-Storms with CPU based Meta-Storms (version 1.2, single core and 16 cores) using the same non-recursive algorithm based on different number of samples to show the speed acceleration of GPU-Meta-Storms.

Because the similarity matrix with large number of samples cannot be completed in short time for CPU computing, we calculated the expected running time of CPU for >2000 samples based on linear-increase estimation (error rate of 6.81E-12). From Figure 1, it was observed that GPU-Meta-Storms had a maximum speed-up of 17 332 times compared with single-core CPU and 2640 times to 16-core CPU. For real time cost, GPU-Meta-Storms constructed the similarity matrix of 10 240 samples within 20 min, whereas the expected running time of 16-core CPU is 45 days.

3.2 Peak computing throughput of GPU-Meta-Storms

This experiment evaluated the peak computing throughput of double-floating-point of GPU-Meta-Storms and scaled in GFLOPS (Giga Floating-point Operations Per Second). We observed that the peak computing throughput of GPU-Meta-Storms rose up to a stable status of 110 GFLOPS, indicating that >610 000 pair-wise sample similarity scores could be obtained per second.

4 CONCLUSION

GPU-Meta-Storms provides a parallel computing solution for the comparison among massive amount of microbial community samples based on GPU and CUDA with very high speed. Such acceleration techniques based on GPU make it possible to perform in-depth data mining from massive number of samples, thus making real-time analysis and monitoring of temporal or conditional changes for microbial communities possible.

ACKNOWLEDGEMENT

The authors thank NVIDIA for their support through CUDA Research Center in QIBEBT-CAS.

Funding: This work is supported by the grant of National Natural Science Foundation of China (No. 30870572, No. 61303161 and No. 61103167) and National High Technology Research and Development Program of China (863 Program) (No. 2014AA021502).

Conflict of Interest: none declared.

REFERENCES

Caporaso,J.G. *et al*. (2010) QIIME allows analysis of high-throughput community sequencing data. *Nat*. *Methods*, **7**, 335–336.

Caporaso,J.G. *et al*. (2011) Moving pictures of the human microbiome. *Genome Biol*., **12**, R50.

DeSantis,T.Z. *et al*. (2006) Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB. *Appl*. *Environ*. *Microbiol*., **72**, 5069–5072.

Goll,J. *et al*. (2010) METAREP: JCVI metagenomics reports–an open source tool for high-performance comparative metagenomics. *Bioinformatics*, **26**, 2631–2632.

Hamady,M. *et al*. (2010) Fast UniFrac: facilitating high-throughput phylogenetic analyses of microbial communities including analysis of pyrosequencing and PhyloChip data. *ISME J*., **4**, 17–27.

Huson,D.H. *et al*. (2011) Integrative analysis of environmental sequences using MEGAN4. *Genome Res*., **21**, 1552–1560.

Jurkowski,A. *et al*. (2007) Metagenomics: a call for bringing a new science into the classroom (while it's still new). *CBE Life Sci*. *Educ*., **6**, 260–265.

Lozupone,C. and Knight,R. (2005) UniFrac: a new phylogenetic method for comparing microbial communities. *Appl*. *Environ*. *Microbiol*., **71**, 8228–8235.

Mardis,E.R. (2008) The impact of next-generation sequencing technology on genetics. *Trends Genet*., **24**, 133–141.

Meyer,F. *et al*. (2008) The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, **9**, 386.

Parks,D.H. and Beiko,R.G. (2010) Identifying biologically relevant differences between metagenomic communities. *Bioinformatics*, **26**, 715–721.

Schloss,P.D. *et al*. (2009) Introducing mothur: open-source, platform-independent, community-supported software for describing and comparing microbial communities. *Appl*. *Environ*. *Microbiol*., **75**, 7537–7541.

Segovia,A. *et al*. (2009) Iterative layer-based raytracing on CUDA. *In: IEEE 28th International Performance Computing and Communications Conference (Ipcc 2009)*, 2009. pp. 248–255.

Seshadri,R. *et al*. (2007) CAMERA: a community resource for metagenomics. *PLoS Biol*., **5**, e75.

Su,X. *et al*. (2012a) Meta-Storms: efficient search for similar microbial communities based on a novel indexing scheme and similarity score for metagenomic data. *Bioinformatics*, **28**, 2493–2501.

Su,X. *et al*. (2012b) Parallel-META: efficient metagenomic data analysis based on high-performance computation. *BMC Syst*. *Biol*., **6**, S16.

Wang,T.Y. *et al*. (2011) MetaRank: a rank conversion scheme for comparative analysis of microbial community compositions. *Bioinformatics*, **27**, 3341–3347.