Sequence Analysis

GPU-computing Greatly Accelerate Motif Analysis

Pooya Zandevakili[1,*], Ming Hu[2] and Zhaohui Qin[2]

[1]Department of Computer Science and Engineering, Address XXXX etc.

[2]Center for Statistical Genetics, Address XXXX etc.

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Computational discovery and detection of TF binding patterns has become an indispensible tool in functional genomics research. Probabilistic approaches utilizing Position-Specific Weight Matrix (PSWM) have demonstrated superior performance; however, they are generally time-consuming. Given that most of the computation time is spent calculating matching scores position by position, scalable approaches that can take advantage of parallel hardware architectures are highly desirable.

Results: Using NVIDIA Cuda, we have developed a graphics processing unit (GPU)-accelerated motif analysis software library named motifGPU. Extensive performance comparison studies on two different graphics cards from NVIDIA, (GeForce) GTX 260 and GeForce GTX 480, showed that commonly-used model-based motif scan procedures can be dramatically accelerated using our library. *De novo* motif discovery can also benefit from our GPU-accelerated tools.

Availability: The library is available at ???.

Contact: qin@umich.edu

1 INTRODUCTION

Accurately locating the transcription factor (TF)-DNA interaction sites provides key insights into the underlying mechanisms of transcriptional regulation. Since binding sites for a specific TF frequently show sequence specificity, computational prediction of TF binding sites based on such sequence features has demonstrated to be an effective tool in functional genomics research. To capture sequence specificity, position-specific weight matrices (PSWMs) have been proposed to represent the sequence features of TF binding sites, or motifs. PSWM-based approaches have demonstrated advanced sensitivity and specificity. However, scaning large number of long sequences using PSWM is time-consuming since likelihoods need to be calculated for each possible start position.

Recently, advanced parallel computing hardware such as graphics processing units (GPUs) have greatly enabled massively parallel processing on regular desktop computers. Originally designed to accelerate demanding 3D graphics, recently, the power of GPUs has been harnessed for non-graphical, general-purpose applications including bioinformatics (Buckner, et al.,; Schatz, et al., 2007). Due to their unique architecture, GPUs dramatically outperform Central Processing Units (CPUs) when it comes to highly data-parallel applications. Most of the computational time spent in the motif scan procedure is trivially parallelizable. Hence a perfect place to utilize the new GPU hardware.

2 THE LIBRARY

Taking advantage of the powerful GPUs, we have developed a library that greatly accelerates motif search. Our library is in the form of a set of C++ header files that can be "included" from C++ source files to offload motif search to a Cuda-enabled GPU. It consists of a core that is in charge of managing the GPU-related transactions and also easy-to-use abstractions for DNA subsequences and PSWMs.

As our results show, higher speedups can be achieved with longer subsequences since they offer a higher degree of concurrency. Knowing this, our library provides tools to concatenate bulks (e.g. thousands) of shorter subsequences into one long subsequence that can be presented to the GPU for a better speedup. Although due to increased complexity, the separate GPU interface written for this scenario is not as fast as the one for the normal case, it still achieves significant speedups as shown in the results section.

3 RESULTS

All performance comparisons in this study are conducted on a desktop computer with an Intel Core i7 920 processor and 4 GB of RAM running Microsoft Windows XP 64-bit. We perform our experiments on two different graphics cards, NVIDIA GeForce GTX 260 and the new NVIDIA GeForce GTX 480 (Fermi). We measure both the speedup and energy-savings for each of the cards.

Due to granularity limitations of our timing mechanism, we run multiple executions of our code to calculate its execution time. To achieve a fair comparison, data-transfers between the host and the graphics card memory are also repeated with every execution. Figure 1 shows the speedup results for both the cards and both the single- and multi- sequences scenarios (see section 2 for details) with different subsequence lengths and a PSWM width of 12.

*To whom correspondence should be addressed.

Fig. 1. The speed-up of the GPU-accelerated motif search over the original version for NVIDIA GeForce GTX 260 and 480.

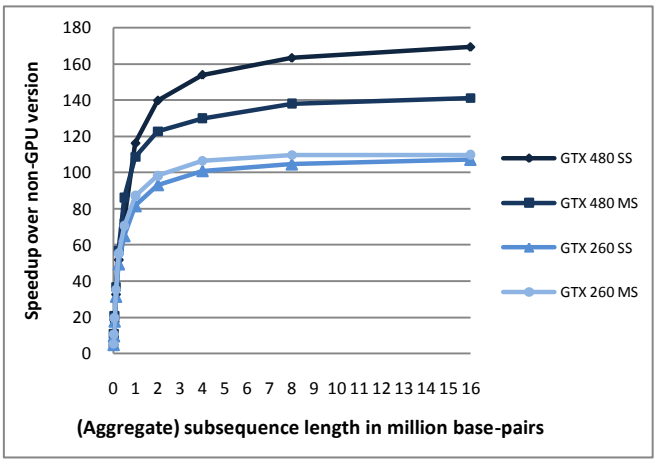
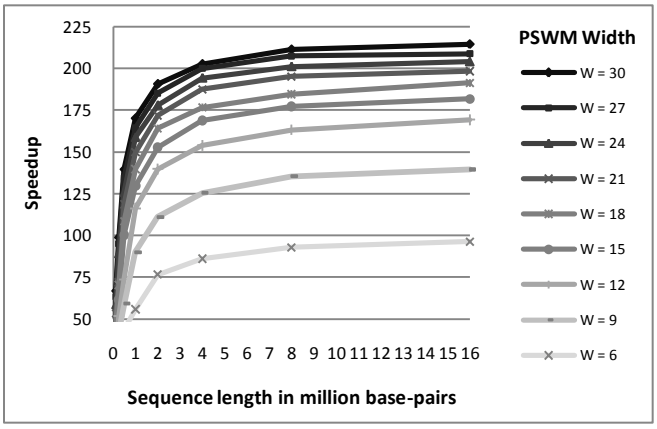


Fig. 2. The speed-up of the GPU-accelerated motif search for different PSWM widths. The results are from NVIDIA GeForce GTX 480.



The latency of the data-transfers between host and graphics card memory is one of common performance bottle-necks in GPU-computing. One of the reasons we achieve such dramatic speedups is our novel technique of broking down our search into a number of "clusters" and transferring the results from cluster k back to the host memory concurrently with the execution of cluster k + 1. This technique, which we call "clustering", is more efficient when the computation time is at least as big as the data transfer time. For instance as Figure 2 shows, when the PSWM width is too small, clustering is not as helpful to prevent the relative slowdown.

3.1 Energy-efficiency comparison

With respect to energy efficiency, although graphics cards draw more power when active, since it takes less time for them to complete a given computational task they are often more energy-efficient. In the case of our motif search, our original non-accelerated algorithm draws 777.6 Jouls to scan a 16-M.B.P subsequence for a PSWM of width 21. The same search draws 9.4 J and 10.8 J when performed on GTX 480 and GTX 260 respectively.

Table 1. De novo search speedup across various datasets.

Dataset	Orig. Time (sec.)	New Time (sec.)	Overall Speedup	% exec in motif search
ctcf_chip_chip	1489	147	10.1	90.7
ctcf_chip_seq	941	142	6.6	85.1
er_chip_chip	1537	138	11.1	91.9
er_chip_seq	321	45	7.1	87.1
nrsf_chip_chip	1731	153	11.3	91.3
nrsf_chip_seq	146	19	7.7	88.8
statl_chip_seq	919	168	5.5	82.3

3.2 *De novo* motif finding and Amdahl's law

We use our library to improve the performance of *de novo* motif finding. The comparisons are performed using four publically-available ChIP-Seq datasets: NRSF (neuron-restrictive silencer factor) (Johnson, et al., 2007), STAT1 (signal transducer and activator of transcription protein 1) (Robertson, et al., 2007), CTCF (CCCTC-binding factor) (Barski, et al., 2007) and ER (estrogen receptor) (Hu, et al., 2010). We use HPeak (Qin, et al., 2010) to define read-enriched regions from the ChIP-Seq data. The number of peaks identified from these four ChIP-Seq datasets range from 4,982 to 27,470, and cover from 1.4 MB to 8.1 MB.

Table 1 shows our results. The big gap between the speedup of motif search and the overall speedup of *de novo* motif finding is explained by Amdahl's law (Amdahl, 1967) which states that the maximum speedup that can be achieved by parallelizing a portion of a program that constitutes fraction f of the overall execution time is 1 / (1-f). Column 7 of Table 1 shows the time percentage of *de novo* motif finding spent in motif search (f * 100).

4 CONCLUSION

In this paper, we introduce motifGPU, a library that uses GPUs to accelerate motif search. We presented our astonishing speedup results for two NVIDIA graphics cards and compared the cards with respect to energy efficiency. We showed how *de novo* motif finding can benefit from our library and how Amdahl's law can place an upper bound on acceleration of programs through GPUs and other parallel computing hardware.

ACKNOWLEDGEMENTS

We thank Dr. Fan Meng and Joshua Buckner for helpful discussion and advices. We thank ??? for …

Funding: .

REFERENCES

Amdahl, G. (1967) Validity of the single processor approach to achieving large scale computing capabilities, *AFIPS Conference Proceedings* **30**, 483-485.

Buckner, J., Wilson, J., Seligman, M., Athey, B., Watson, S. and Meng, F. The gputools package enables GPU computing in R, *Bioinformatics*, **26**, 134-135.

Matys, V., Kel-Margoulis, O.V., Fricke, E., Liebich, I., Land, S., Barre-Dirrie, A., Reuter, I., Chekmenev, D., Krull, M., Hornischer, K., Voss, N., Stegmaier, P., Lewicki-Potapov, B., Saxel, H., Kel, A.E. and Wingender, E. (2006) TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes, *Nucleic Acids Res*, **34**, D108-110.

Sandelin, A., Alkema, W., Engstrom, P., Wasserman, W.W. and Lenhard, B. (2004) JASPAR: an open-access database for eukaryotic transcription factor binding profiles, *Nucleic Acids Res*, **32**, D91-94.

Schatz, M.C., Trapnell, C., Delcher, A.L. and Varshney, A. (2007) High-throughput sequence alignment using Graphics Processing Units, *BMC Bioinformatics*, **8**, 474.

Vlieghe, D., Sandelin, A., De Bleser, P.J., Vleminckx, K., Wasserman, W.W., van Roy, F. and Lenhard, B. (2006) A new generation of JASPAR, the open-access repository for transcription factor binding site profiles, *Nucleic Acids Res*, **34**, D95-97.

Wingender, E., Chen, X., Hehl, R., Karas, H., Liebich, I., Matys, V., Meinhardt, T., Pruss, M., Reuter, I. and Schacherer, F. (2000) TRANSFAC: an integrated system for gene expression regulation, *Nucleic Acids Res*, **28**, 316-319.