# Accelerating Processing Speed in Pathway Research Based on GPU

Bo Liao
College of Information Science and Engineering
Hunan University, Changsha, China
dragonbw@163.com

Ting Yao
College of Information Science and Engineering
Hunan University, Changsha, China
yaoting@hnu.edu.cn

Xiong Li
College of Information Science and Engineering
Hunan University, Changsha, China
jt_xli@163.com

***Abstract*—Genome-wide association study (GWAS) has become an effective and successful method to identify disease loci by considering SNPs independently. However, it may be invalid for uncovering the disease loci that not reaching a stringent genome-wide significance threshold. As a result, multi-SNP GWAS is developing rapidly as a complement to traditional GWAS. However, the high computational cost becomes a major limitation for it. The graphical processing unit (GPU) is a programmable graphics processor which has powerful parallel computing ability. And with the development, GPUs have been feasible for many scientific studies. Hence, we are motivated to use GPUs for pathway-based GWAS to improve computational efficiency. The experiment results attained showed the speed-up ratio can reach up to more than 160.**

***Keywords—GWAS; Pathway analysis; Complex disease; GPU; CUDA***

## I. INTRODUCTION

In the past few years, genome-wide association studies (GWAS) have been commonly used to figure out the relationship between the genetic variants and complex diseases, such as diabetes, inflammatory bowel disease, and several cancers. Millions of single nucleotide polymorphism (SNPs) loci on the human genome have been applied for presenting information during modern GWAS. National Cancer Institute–National Human Genome Research Institute (NCI-NHGRI) has completely listed the published GWAS [1].

Typically, GWAS have been an effective means in identifying disease loci for each independent SNP. However, some genes in complex disease may be neglected for not reaching a preset genome-wide significance threshold. As a result, multi loci, systems-based analytical methods have been proposed as alternative or complementary approaches for GWAS. Multi-SNPs analysis at system level analysis turns out to be a useful method that can effectively present the significant association between gene sets, which are organized by biological pathways or networks, and biological mechanisms that hide behind disease pathogenesis.

Several kinds of multi-SNP GWAS methods have been proposed [2]. We summarize some typical multi-SNP GWAS ways and analyze its drawbacks. Gene Set Enrichment Analysis (GSEA) was proposed as a multi-SNP GWAS analysis approach [3, 4]. The authors have been very successful in identifying the association of genotypes and phenotypes by bringing in signals across multiple genes belonging to a set. However, the gene set may not reach significance altogether when two SNPs in a set was purely epistatic interaction. For working out this problems, Yang *et al.* proposed SNPHarvester in [5]. SNPHavester has been widely used as a powerful multi-SNP analysis approach and is able to identify the pathogenic genes which would be neglected by GSEA-type approaches for not reaching a strong marginal association. However, SNPHarvester is confined by the limited number of SNPs that can be examined simultaneously. Because of this issue, the biological mechanisms of the results would be hard to interpret for random groupings of SNPs and the exclusion of SNPs with marginal effects. Multi-dimensionality reduction (MDR) was proposed as a typically distance-based approach based on the idea that cases will be more close to other cases than controls[6, 7]. MDR has effectively recovered the drawbacks of SNPHarvester. However, computational cost turns to be the major limitation of it for the huge number of loci to be explored.

Recently, graphic processing units (GPUs) have been widely populated with Biocomputing on account of the super computing power in Data-Level Parallelism [8]. In this paper, we proposed an project for accelerating the PoDA [9]based on GPU, which designed to address the following problems: firstly, analyzing the relationship between multi-SNPs and disease at systematic level; secondly, improving the computing efficiency of PoDA by employing GPU. For the first problem, PoDA mainly calculates the distance between one sample and the average distance in control/case for every sample. And then combining with the standardized mean across the $l$ loci obtains a standardized distance statistical distribution. The experiment shows that the distribution can distinguish the difference between controls and cases. However, its computing efficient is so low that we can't get the distribution result in a short time when it was running in CPU. As for this, the incoherence among the data set was utilized and GPU has the super computing power in Parallel Data Processing. We modify this method by deserializing the computing process. In our experiment, the computing efficient can be greatly improved and speed-up ratio is reaching up to impressive results.

The structure of this paper is organized as follows. Section II briefly introduces Compute Unified Device Architecture


978-1-4799-1389-3/13/$31.00 ©2013 IEEE

(CUDA) which is released by NVIDIA Corporation. Section III presents the PoDA and our method. The experimental results using the PoDA and our model are elaborated in Section IV. Meanwhile, we conclude this paper in Section V.

## II. THE HARDWARE OF GPU AND SOFTWARE OF CUDA

### A. The hardware of GPU

Graphic Processing Units (GPU) [8], which is also occasionally called visual processing unit (VPU), is a kind of specialized electronic circuit for accelerating the handling rate of 2D/3D picture. In addition, GPUs have gradually developed into a powerful tool which can be used for non-graphical calculations. In this paper, we fully utilize the GPU's strong ability in paralleling computing combining with the incoherence characters of Biocomputing. NVIDIA GeForce GTX 580 has been employed as the main part of computing core in Device and been linked with PC in charge of managing the task scheduling via PCI-E [10]. Fig. 1 reveals the hardware structure of NVIDIA GeForce GTX 580. 1.5GB global memory can provide enough memory for storing the SNPs data. It has 512 computing units which are called as Streaming Processors (SPs) in GPU's internal. By observing the hardware, we know that single Streaming Multiprocessor (SM) has a group of 32 SPs, which share 48KB memory and execute in the same order.

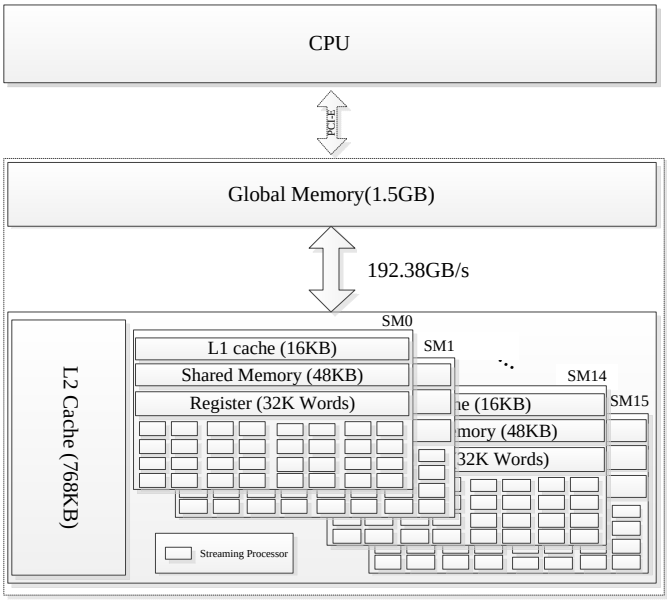


Fig. 1 The hardware of GPU

### B. The software of CUDA

NVIDIA Corporation firstly proposed the concept of Compute Unified Device Architecture [11], which is one kind of software architecture and calls GPU to data parallel computing. CUDA exploits the Single Instruction Multiple Thread (SIMT) as the executing model. According to Fig. 1, disparate threads have respective private registers and the communicating mechanism. A kernel is defined as a parallel computing unit running on GPU. Meanwhile, the form of the grid, which is assembled by numbers of blocks, constitutes the kernel. Every block owns a certain number of threads. Generally speaking, the number is 1024.

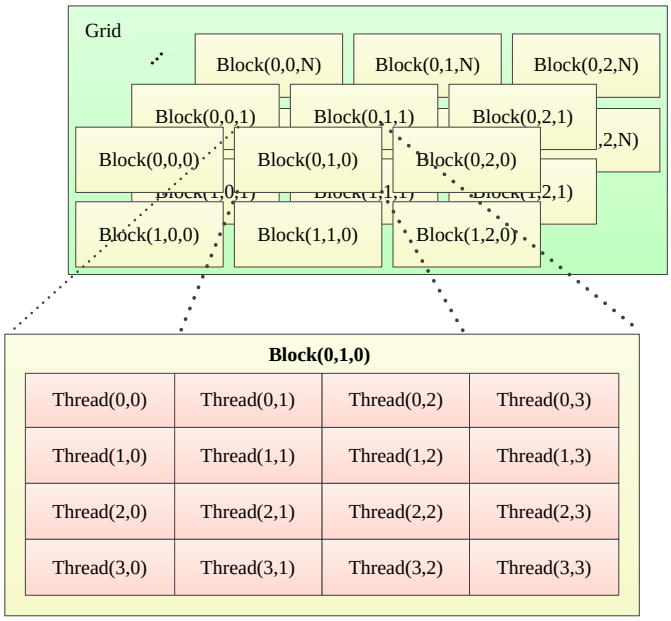


Fig. 1 The construction of CUDA

## III. THE DETAIL OF PODA AND OUR MODEL BASED ON GPU

In this section, we mainly introduce two parts of work: the detail of PoDA and our model based on GPU. Firstly, we show the algorithm of PoDA and conclude that step 3 and step 4 are the most time cost in PoDA algorithm by combining time complexity analyzing with running experiment on CPU. Secondly, a new model based on GPU, which is mainly decreasing the time cost in step 3 and step 4, is proposed. Besides, the analyzing of time complexity for our new model is given.

### A. The decription of the detail of PoDA

In PoDA, the authors selected SNPs by these steps as follows.

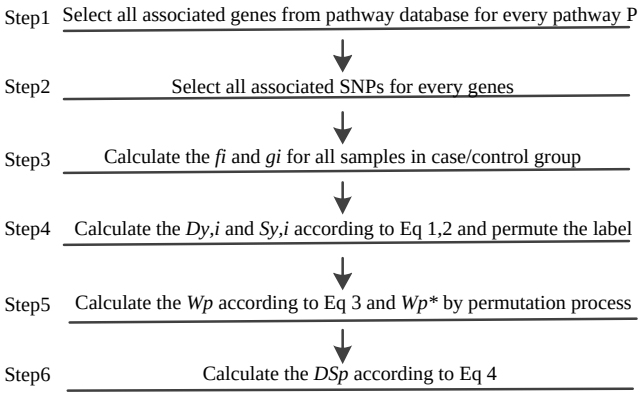


Fig. 2 The detail steps of PoDA

First of all, some diseases were confirmed and the PoDA called Pathway Interaction Database and KEGG for obtaining the associated genes of every pathway in this disease. In the next place, the authors gained SNPs associated with genes in


978-1-4799-1389-3/13/$31.00 ©2013 IEEE

different pathways by using dbSNP and >20% missing data or minor allele frequency <0.05 in case/control was excluded. Fig. 2 shows us the distance process. In allusion to every sample, PoDA defined the mean of its distance to other control minus the mean of its distance to case as its relative distance statistic. For instance, F indicated case group and G presented control group. By computing the mean of F and G for loci i, the authors can get two vectors, fi and gi [12, 13]. In the vector of fi and gi, one column presented a mean distance value for certain loci of case/control groups.

$$D_{Y,i} = |y_i - f_i| - |y_i - g_i| \quad (1)$$

After that, because PoDA needs a value to assess the distance between every sample and the mean of F and G, leave-one-out method, which one sample Y was gotten out and the relative distance between Y and case/control group for all loci i was calculated, was utilized to compute this distance. In addition, PoDA also standardized this distance statistic to SY,i by Eq(2).

$$S_Y = \frac{E(D_{Y,i})}{\sqrt{Var(D_{Y,i}/l)}} \quad (2)$$

Where E(DY,i) means the average value of DY,i for all loci i, Var(DY,i/l) gives the variance of DY,i [13]. Then PoDA proposed a method quantifying the SY,i distribution to non-parametric via calculating the Wilcoxon rank sum statistic, defined as:

$$W_P = \sum_{Y \in case} R_{Y,P} - \frac{n_{case}(n_{case}+1)}{2} \quad (3)$$

Meanwhile, the permuted label is used for calculating fi, gi and Wp and on this basis, Wp* can be achieved. Lastly, the authors yields a distance score (DSp) for pathway P by the Eq (4).

$$DS_P = \frac{W_P - E(W_P^*)}{SD(W_P^*)} \quad (4)$$

By applying PoDA to all pathways with interest, we can distinguish the case and control. The experiment on breast and liver cancer demonstrated its availability for system-level analysis of GWAS data. However, the main expenditure of time in PoDA is from step 3 to step 5. By analyzing the algorithm of PoDA, the time complexity is O(N2) and the number of SNPs in every pathway is very large, so that a small data set can make the program run a long time. Hence, we design a model for accelerating the computing speed based on GPU.

### *B. Our model based on GPU*

We propose a new model for accelerating the computing speed in a pathway research method in this section. The aim of this new model is to distribute the distance process of PoDA among case/control group with no conflicting condition. For the sake of increasing the computational rate on the GPU device, the designed target of our model furthermore is to decrease the number of groups while increasing the amount of processes inside a group. By analyzing the PoDA, we can get the conclusion that the operation of calculating the fi and gi, and Eq(1) are the most time-consuming procedures. So that our model is proposed as a complement of PoDA which can powerfully reduce the time cost by parallelizing these steps.

We first calculate the mean for all case/control samples at loci l as fi and gi by Divide and Conquer algorithm. Because the number of SNPs for different samples in case/control group is the same, the storage of SNPs for every sample employs an array in which one column presents a SNP. Each two samples are divided into one block and calculated in one clock for the operation of Divide and the next step is conquering the two continuous results. The process will be terminated till all results emerge. Fig. 3 shows us a simple example for this operation. Due to its continuous address distribution and achiasmate address accessing, the time cost of each conquer operation is O(1), which means that the time complexity of this part has decreased from O(N2) to O(log2N) (N presents the number of samples).

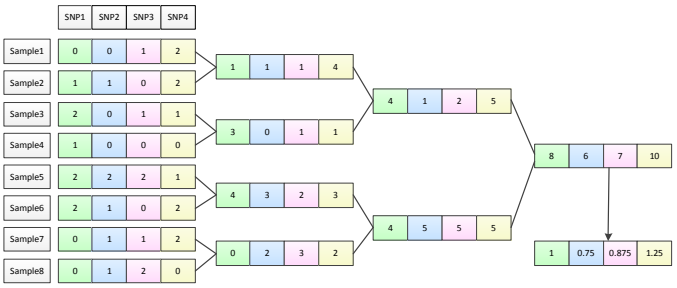


Fig. 3 Computing *fi* and *gi* based on GPU

Furthermore, by observing the Eq(1) we can know that each sample Y shall share the computing result fi and gi, and different samples are independent. Combining with these feathers and the CUDA model, we propose a model to reduce the time cost of Eq(1). The details of our model are presented in Fig. 4. In NVIDIA GeForce GTX 580, every block owns 48KB share memory and 768KB L1 cache, which is enough to store the fi and gi, and every thread can simultaneously access these data. We plan to assign a separate block to every sample, such as the ith block is used for computing the D value of ith sample. Generally, the number of block is 1024, so the number of samples is set as 1024 times, if not, the absence will be filled with zero to reach 1024 times. The operation of threads inside the block is used for computing the different loci. One loci is corresponding to one thread, which calls the same operation by Eq(1). As CUDA provides an effective shared mechanism, it turns to be realizable that all threads in a block share the computing results fi and gi. As a result, Computation for all loci can be finished in several cycles, which time is determined by the loci number dividing the max number of threads in block. Assuming that the number of samples is M and the number of loci is W, the number of blocks is Z and the number of threads in block is K. The running time of Eq(1) in PoDA is MW while it is MW/ZK for our model. Because we propose a model that can effectively deserialize the process of computing fi and gi and Eq(1), which are the most time-consuming procedures. We can predict that it should be an effective method to reduce the computational complexity


978-1-4799-1389-3/13/$31.00 ©2013 IEEE

theoretically, and the experiment results fully improved it. Indeed, the same data running on the model we proposed can be 160 times faster than PoDA on time cost.

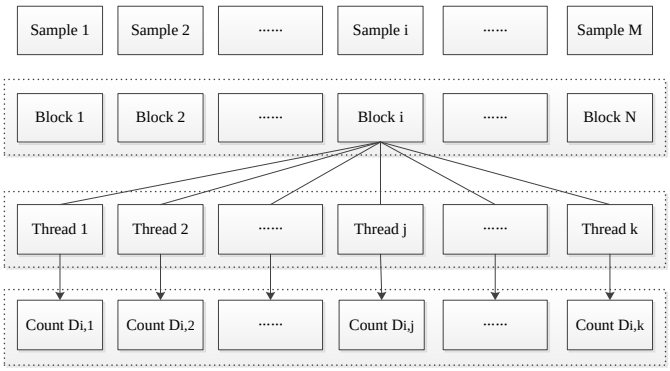


Fig. 4 Proposed our model base on GPU

## IV. SIMULATION EXPERIMENT

The testbed consists of a PC service with an Intel Xeon CPU E5606 @2.13GHz, 12GB of system memory and NVDIA GeForce GTX 580 with 512 processing cores and 1.5GB of global memory. The operation system on the machine is acted by a 64-bit Red Hat Enterprise Linux Workstation release 6.1. All programs consist of host programs complied by G++ 4.4.5 and device programs complied by CUDA release 2.0 which can be downloaded from NVIDIA's official website.

For quantitating the improved effects, we define a speed-up ration as the running time of PoDA which divides the running time of our model on GPU. According to the analysis of CUDA, our experiment time shall add the transmission time between Host and Device. Hence, we employ diverse datasets, which have different sizes, to better describe the speed up ratio. Sample sets with different scales are adopted for computing, such as sample128, sample256, sample512, sample1024, sample2048 and sample4096. On the account of the same sample, we conduct our experiment repeatedly by resetting the number of SNPs (32, 64, 128, 256, 512, 1024, 2048 and 4096). Fig. 6 shows us the simulation experiment results.

In section III, we concluded that the time complexity of PoDA is O(N2) by analyzing the algorithm. PoDA is operated on PC service for the above datasets. By observing the Fig. 5(a), when the program calls the same sample set and the different amount of SNPs, the time cost increases linear and the factor is approximately two. If the SNPs number is held, the running time for different sample sets increases by ratio two. For example, the sample number is 1024, the running time for 1024 SNPs is 8074.6 sec while that of 2048 SNPs reaches up to 16185.31 sec. While the amount of SNPs is set as 2048, the time cost for sample1024 and sample2048 are separately 32416.24 sec and 44100.34 sec.

Fig. 5(b) presents the time cost for the different datasets running on GPU. Its time complexity is analyzed in section III and concluded as MW/ZK. We can conclude that when the same sample is called, if the number of SNPs is less than 128, the time costs are almost the same. For instance, the sample number is 1024, the time cost for 32, 64, 128 SNPs are 9.09, 9.61 and 9.84 sec. The hardware introduced in section II manifests that the running time on GPU includes the transmission time between CPU and GPU and the running time on GPU. When the processing data is large, the accelerating time by GPU can cover the transmission time. Otherwise the accelerating ratio is close to one. In our experiment, with the increase of dataset size, the running time increases by a nonlinearity function.

In order to demonstrate the improvement effect of our model on time cost visually, we draw the speed-up ratio in Fig. 5(c). The software part in section II indicates that the maximum threads in a block and blocks in a grid are both 1024. While the scope of dataset is small, as sample number and SNPs number are less than 1024, the speed-up ratio grows faster than larger dataset. That's because the program calling the GPU's thread has no use for larger time and the memory access time is O(1). But if the number of threads and blocks surpass 1024, the managing threads time and memory access time are increasing. Fig. 5(c) reveals that the increase of speed-up ratio is linear and the factor is larger than 1.

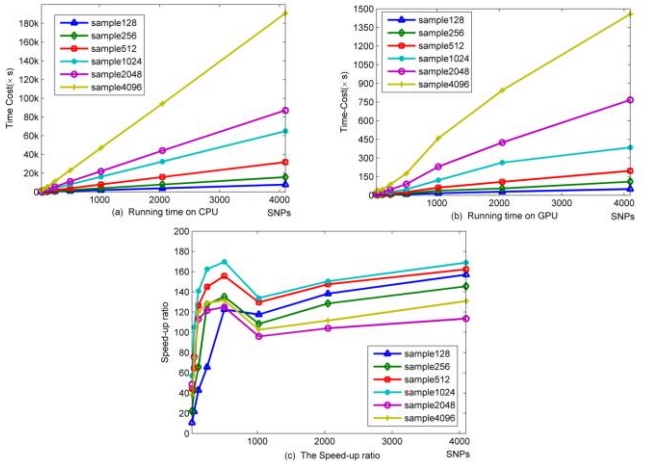


Fig. 5 The simulation experiment results

## V. CONCLUSION

In this paper, the authors propose a new calculating model for accelerating computing speed in the pathway research of GWAS. Firstly, we thoroughly introduce the contributions of PoDA in pathway research and conclude that its method can effectively distinguish case/control group via a pathway. Besides, we propose a new model based on GPU for solving the drawback of PoDA, which has low computing effective. The process of computing is deserialized in data-level. Finally, adequate simulation results on anolog data set certificate that our model is highly feasible and efficient.

## ACKNOWLEDGMENT

This work is supported by the Program for New Century Excellent Talents in University (Grant NCET-10-0365), National Nature Science Foundation of China (Grant 60973082, 11171369), the Planned Science and Technology Project of Hunan Province (Grant 2009FJ3195, 2012FJ2012) and supported by the Fundamental Research Funds of the Central Universities, Hunan university.


978-1-4799-1389-3/13/$31.00 ©2013 IEEE

## REFERENCES


[1] Hindorff LA, S.P., Junkins HA, Ramos EM, Mehta JP, et al.. "Potential etiologic and functional implications of genome-wide association loci for human diseases and traits". PNAS, 2009. 106: 9362–7.

[2] Ramanan V K, S.L., Moore J H, et al., "Pathway analysis of genomic data: concepts, methods and prospects for future development[J]". TRENDS in Genetics, 2012.

[3] Wang K, L.M., Bucan M., "Pathway-based approaches for analysis of genomewide association studies". Am J Hum Genet, 2007. 81: 1278.

[4] Holden M, D.S., Wojnowski L, Kulle B., "GSEA-SNP: applying gene set enrichment analysis to SNP data from genome-wide association studies". Bioinformatics, 2008. 24: 2784–5.

[5] Yang C, H.Z., Wan X, Yang Q, Xue H, et al., "SNPHarvester: a filtering-based approach for detecting epistatic interactions in genome-wide association studies". Bioinformatics, 2009. 25:504-11.

[6] Motsinger A, R.M., "Multifactor dimensionality reduction: an analysis strategy for modelling and detecting gene–gene interactions in human genetics and pharmacogenomics studies". Human Genomics, 2006. 2:318–328.

[7] Moore J, G.J., Tsai C, Chiang F, Holden T, et al., "A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility". Journal of theoretical biology, 2006. 241: 252–261.

[8] Wikipedia. http://en.wikipedia.org/wiki/GPU.

[9] Rosemary Braun, K.B., "Pathways of Distinction Analysis: A New Technique for Multi–SNP Analysis of GWAS Data". PLoS Genetics, 2011. 7(6).

[10] Nvidia, GeForce GTX 580. http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-580.

[11] CUDA, Compute Unified Device Architecture. http://www.nvidia.com/object/cudahome.html.

[12] Braun R, R.W., Schaefer C, Zhang J, Buetow K, "Needles in the haystack: Identifying individuals present in pooled genomic data". PLoS Genet, 2009. 5:e1000668.

[13] Homer N, S.S., Redman M, Duggan D, Tembe W, et al., "Resolving individuals contributing trace amounts of DNA to highly complex mixtures using high-density SNP genotyping microarrays". PLoS Genet, 2008. 4:e1000167.




978-1-4799-1389-3/13/$31.00 ©2013 IEEE