Libertas Academica
FREEDOM TO RESEARCH

Open Access: Full open access to this and thousands of other papers at http://www.la-press.com.

Cancer Informatics

Supplementary Issue: Array Platform Modeling and Analysis (B)

CARAT-GxG: CUDA-Accelerated Regression Analysis Toolkit for Large-Scale Gene–Gene Interaction with GPU Computing System

Sungyoung Lee[1], Min-Seok Kwon[1] and Taesung Park[1,2]

[1]Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul, South Korea. [2]Department of Statistics, Seoul National University, Seoul, South Korea.

ABSTRACT: In genome-wide association studies (GWAS), regression analysis has been most commonly used to establish an association between a phenotype and genetic variants, such as single nucleotide polymorphism (SNP). However, most applications of regression analysis have been restricted to the investigation of single marker because of the large computational burden. Thus, there have been limited applications of regression analysis to multiple SNPs, including gene–gene interaction (GGI) in large-scale GWAS data. In order to overcome this limitation, we propose CARAT-GxG, a GPU computing system-oriented toolkit, for performing regression analysis with GGI using CUDA (compute unified device architecture). Compared to other methods, CARAT-GxG achieved almost 700-fold execution speed and delivered highly reliable results through our GPU-specific optimization techniques. In addition, it was possible to achieve almost-linear speed acceleration with the application of a GPU computing system, which is implemented by the TORQUE Resource Manager. We expect that CARAT-GxG will enable large-scale regression analysis with GGI for GWAS data.

KEYWORDS: GWAS, gene–gene interaction, logistic regression, GPU, graphics processing unit

SUPPLEMENT: Array Platform Modeling and Analysis (B)

CITATION: Lee et al. CARAT-GxG: CUDA-Accelerated Regression Analysis Toolkit for Large-Scale Gene–Gene Interaction with GPU Computing System. *Cancer Informatics* 2014:13(S7) 27–33 doi: 10.4137/CIN.S16349.

RECEIVED: August 13, 2014. **RESUBMITTED:** October 13, 2014. **ACCEPTED FOR PUBLICATION:** October 14, 2014.

ACADEMIC EDITOR: JT Efird, Editor in Chief

TYPE: Methodology

FUNDING: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (Ministry of Science, ICT and Future Planning (MSIP)) (No. 2012R1A3A2026438). The authors confirm that the funder had no influence over the study design, content of the article, or selection of this journal.

COMPETING INTERESTS: Authors disclose no potential conflicts of interest.

CORRESPONDENCE: tspark@stats.snu.ac.kr

Paper subject to independent expert blind peer review by minimum of two reviewers. All editorial decisions made by independent academic editor. Upon submission manuscript was subject to anti-plagiarism scanning. Prior to publication all authors have given signed confirmation of agreement to article publication and compliance with all applicable ethical and legal requirements, including the accuracy of author and contributor information, disclosure of competing interests and funding sources, compliance with ethical requirements relating to human and animal study participants, and compliance with any copyright requirements of third parties. This journal is a member of the Committee on Publication Ethics (COPE).

Introduction

In recent years, significant advances in both the genotyping and computing fields have enabled us to perform large-scale genome-wide association studies (GWAS).[1,2] GWAS have demonstrated enormous potential in identifying genetic variants for common complex diseases.

In GWAS, regression analysis, including both logistic regression for binary phenotypes and linear regression for quantitative phenotypes, has been one of the most powerful methods to detect associations between genetic variants and target phenotypes. It has successfully identified novel genetic variants, such as single nucleotide polymorphism (SNP), for many complex diseases, including type 2 diabetes,[3–5] bipolar disorder,[6,7] and various cancers.[8–10]

However, these applications are mainly limited to the investigation of association between a single variant and target phenotype, not that between multiple variants and a target phenotype. Note that the association tests are usually performed by fitting a linear or logistic regression model. Although the regression analysis requires inversion and multiplication of matrices with high computational complexity, the computational burden is not so tremendous for the single variant analysis. However, in the case of gene–gene interaction (GGI) analysis in an exhaustive manner, the tremendous computational burden renders regression analysis infeasible, because the number of possible tests is extremely large, say p chosen as 2. Here p denotes the number of SNPs. Furthering this burden is the fact that

parameter estimation in logistic regression usually relies on iterative algorithms such as Newton–Raphson (NR), expectation-maximization (EM), or scoring algorithms. Owing to the computational burden caused by the enormous number of tests, there have been limited applications of regression analysis to multiple SNPs including GGI analysis of large-scale GWAS data.

In order to overcome this computational burden and identify GGIs within feasible computational time, many approaches have been suggested. They can be classified into two main approaches. One is to reduce the number of possible tests using a variable selection method, and the other is to develop an accelerated algorithm. Variable selection is most commonly used. For example, sure independence screening (SIS) is the most popular variable selection strategy.[11] Ueki and Tamiya proposed an algorithm to identify GGI with SIS variable selection strategy.[12] By selecting the variables in the most parsimonious way, it was shown to be possible to identify more GGIs than other competing methods. However, the variable selection approach still has a chance to miss the true GGI if the variable screened out has a true interaction with the other variable. By motivating from this issue, many researchers have focused on the identification of GGI in an exhaustive manner with a high speed. For example, Wan et al proposed BOolean Operation-based Screening and Testing (BOOST) method[13] in order to make the identification of two-way GGI possible by using the Boolean operation method. However, since this method is based on the fast calculation via Boolean operation, GGIs that cannot be represented via the result of their XOR bitwise manipulation are difficult for BOOST to detect. Alternatively, Kwon et al proposed cuGWAM[14] in order to make the identification of *N*-way GGI possible via a high-throughput graphics processing unit (GPU) for multifactor dimensionality reduction[15] analysis. However, since cuGWAM is based on several accuracy measures that can be obtained from two-by-two contingency tables, it is not straightforward to obtain a statistical *P*-value unless a permutation scheme is applied.

Although BOOST and cuGWAM are very powerful methods for GGI analysis, they are not as flexible as regression analysis in dealing with individual covariates; however, regression-based GGI analysis has not been successfully performed because of the computational burden described earlier. Thus, an acceleration of regression-based GGI analysis is required in order to make an exhaustive investigation computationally feasible. In this work, we thus develop a new regression-based toolkit, CARAT-GxG, which dramatically accelerates the performance of regression analysis of GGI. CARAT-GxG can find an association between a phenotype and single SNP or between several phenotypes and two SNPs with interaction, while adjusting for covariates. CARAT-GxG uses logistic regression for binary phenotype and linear regression for quantitative phenotype. By substantially accelerating the speed of regression-based GGI analysis using GPU, we achieved an exhaustive investigation of GGI from GWAS dataset within a couple of weeks, which was not possible in the traditional analyses. Furthermore, CARAT-GxG can be accelerated by CUDA (compute unified device architecture) using a high-throughput GPU and can be applied to GPU computing systems via TORQUE Resource Manager (http://www.adaptivecomputing.com/products/open-source/torque/), the GPU computing architecture.

In this paper, we describe in detail how regression analysis process can be implemented using GPUs and how much acceleration can be achieved by GPU implementations compared to traditional central processing unit (CPU) implementations. In addition, we describe the factors that affect the performance and accuracy of GPU programs, and evaluate how much these factors affect the performance of CARAT-GxG. Next, we demonstrate how the several adjustable parameters related to regression analysis and GPU-related computational issues affect the accuracy of the results and the execution time. Finally, we introduce a way to apply a GPU computing system with CARAT-GxG and summarize the improvements achieved by using the GPU computing system. CARAT-GxG is freely available from the software website (http://bibs.snu.ac.kr/software/caratgxg/).

Methods and Implementation

CARAT-GxG was implemented in the C/C++ language within a CUDA environment. For performance comparisons, R package and PLINK were used.

Implementation of analysis method. CARAT-GxG is capable of performing all possible SNP combinations in a regression model, including all possible pair-wise SNP–SNP interactions (eg, a dataset with 10,000 SNPs has 49,995,000 SNP–SNP pair-wise interaction combinations in addition to 10,000 single SNP combinations). The SNP variables, including main and interaction effects, can be treated either as nominal or ordinal variables. Depending on the phenotype, CARAT-GxG is implemented by using either logistic or linear regression.

For the continuous phenotypes, CARAT-GxG builds a linear regression model and estimates the parameters using the ordinary least squares (OLS) method for each SNP–SNP combination. This OLS estimation does not require any iterative processes. The significance of model parameters is evaluated via *P*-values of the Wald test. For the binary phenotypes, CARAT-GxG builds a logistic regression model. The model parameters are obtained via iterative algorithms such as the NR algorithm, which is known to converge very quickly, and the scoring algorithm. The significance of model parameters is evaluated via the likelihood ratio test, which is derived from the full and null models. These models are generally represented as depicted below:

$$M_F : logit(P(y=1)) = \beta_o + \beta_i S_i + \beta_i S_i + \beta_{ij} S_i S_j + \sum_k \beta_k C_k$$

$$M_o : logit(P(y=1)) = \beta_o + \sum_k \beta_k C_k$$

where S_i and S_j denote genotypes of SNPs, and C_k denotes the covariates. These equations are applied to all possible combinations.

GPU implementation. The GPU is a circuit specialized for graphical processing, which usually requires large-scale parallelism and floating-point calculations. Thus, GPU has successfully been used to implement several key methods in bioinformatics, such as GGI analysis,[14] sequence alignment,[16,17] and database searching.[18] These GPU applications have been developed using CUDA by Nvidia, Stream SDK by ATI, or in recent years, OpenCL. In order to implement our regression-based GGI application, we chose the CUDA architecture as our development environment.

Generally, a different implementation strategy is required when developing an application that uses the GPU rather than the CPU because of differences in the underlying architecture such as in the memory structure or processing pipeline. These differences make it impossible to port many traditional algorithms based on the CPU to the GPU architecture. Thus, we developed a new implementation strategy to perform regression analysis using the GPU. A key feature of our CARAT-GxG implementation is the ability to execute multiple combinations concurrently. Our implementation consists of the following steps, as shown in Figure 1:

1. As a preliminary step, load the dataset into each GPU memory.
2. Enumerate all possible two-way combinations.
3. Assign the combinations into the GPU memory.
4. Launch a GPU kernel with a given number of combinations from the CPU.
5. Distribute the combinations to threads that are included in each block. The number of threads is automatically determined to maximize performance.
6. Fit a statistical model in each thread, with a given combination through the GPU-optimized algorithm.
7. Calculate the *P*-value for a given combination in each thread.
8. Repeats step 3–7 until all combinations are processed.

In order to accomplish optimal performance using the GPU, many aspects must be considered. Since the main computational burden of regression analysis occurs during matrix calculation, an optimized access of memory, which highly varies by the model of graphics card, is essential to minimize race conditions. CARAT-GxG automatically selects the most appropriate parameter of GPU execution. In order to determine this parameter, a very naïve but fast approach is applied; it is achieved via a sequential test of equally spaced candidates of optimal parameters. As shown in Figure 2C, a concave trend of execution time along with the parameter value justifies this approach.

Methods for performance comparison. Typically, R package and PLINK are used in order to perform regression analysis of GWAS data. Thus, a performance comparison between the former tools and CARAT-GxG is essential. In addition, it is important to note the differences arising from execution on the GPU versus the CPU. Some of these differences are as follows:

1. Calculation architecture and conditions: CPU versus GPU, one way and two way.
2. Acceleration by the number of GPUs: one, two, and three GPUs.
3. Difference of accuracy between GPU and CPU implementations: this arises because the CPU and GPU treat floating-point numbers differently.
4. Performance during repeated iterations of the NR method: it is essential to see how the number of iteration affects the speed and result.
5. Acceleration by the application of CARAT-GxG to the GPU system.

While the ultimate purpose of regression analysis on GWAS data is to find statistically significant associations between variants and target phenotype, the results should also have biological significance. Thus, it is important to

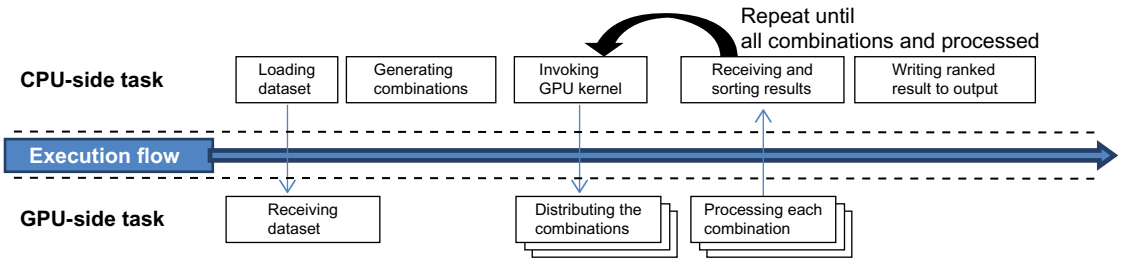

Figure 1. Execution flow of CARAT-GxG. The blue arrow indicates the execution flow between CPU and GPU. The upper and lower sides indicate the task of CPU and GPU, respectively. Data transfer between CPU and GPU is illustrated by an arrow, with description. Both tasks at the same time point can be executed concurrently.

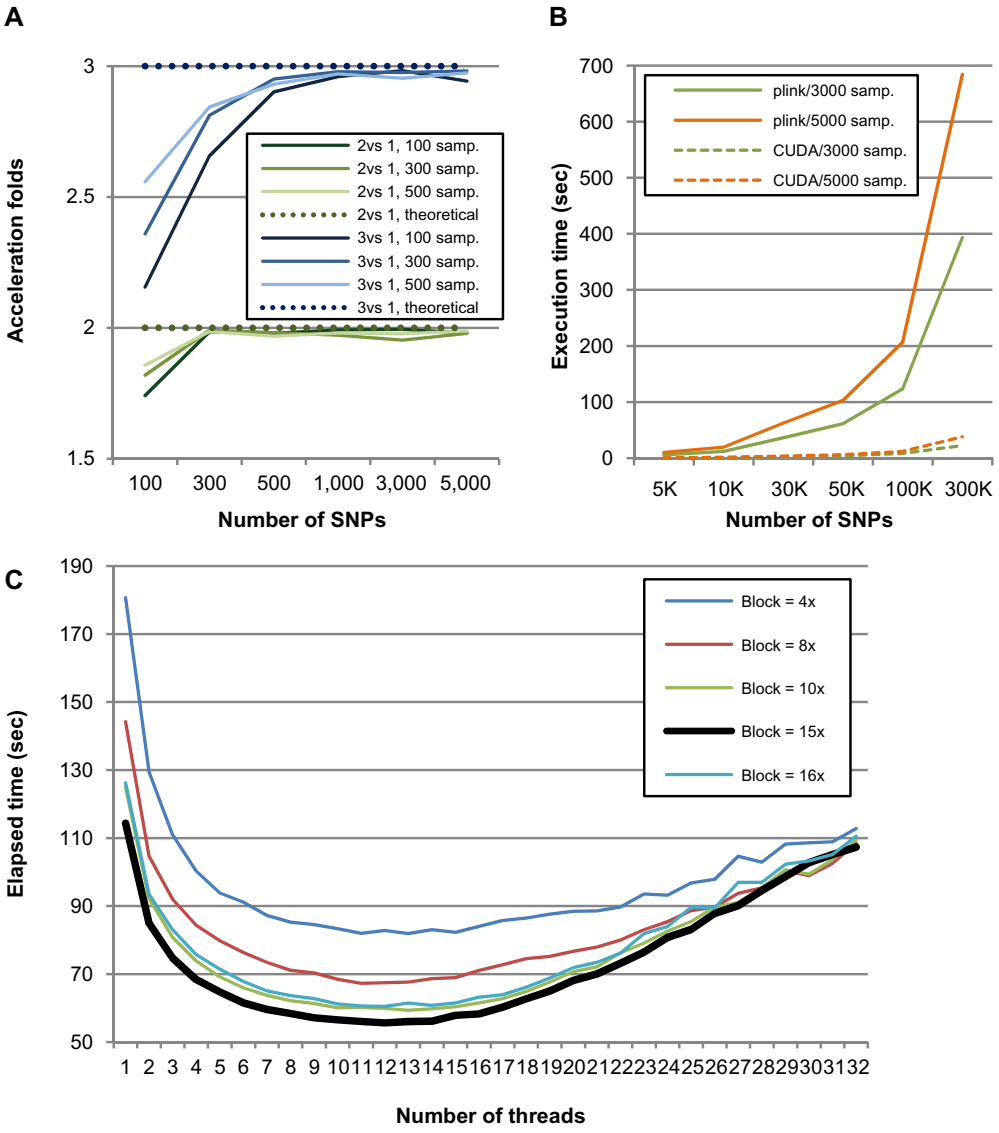

Figure 2. Results of CARAT-GxG performance assessment. (**A**) The dotted line indicates the theoretical acceleration folds by adding a graphics card. The solid line indicates measured acceleration folds in two against one (green) and three against one (blue) graphics cards. (**B**) Execution time between CARAT-GxG and CPU implementations in a single SNP test. (**C**) Execution time of CARAT-GxG according to the number of threads and blocks with the dataset including 1,000 samples with 500 SNPs.

evaluate their biological significance as well. For this reason, an additional parameter to evaluate the biological significance of identified SNP–SNP combinations is required. It can be determined in many ways, such as through a literature search or biological data mining.

In order to assess the analysis result of CARAT-GxG, we analyzed a real GWAS dataset from an age-related macular degeneration (AMD) study, which has several validated GGI.[19] For AMD dataset, several validated association results, including interactions, have been already reported in studies.[20,21] The AMD dataset contains a binary phenotype indicating the status of AMD. The genotypes of 146 samples with 105,980 SNPs were available, which passed quality control process.

GPU computing system setup. Deploying GPUs to computing system infrastructures is unpopular because the method of implementing general-purpose computing on graphics processing units (GPGPU) differs depending on the vendors of graphic cards. However, CUDA has now become a de facto standard for GPGPU. Several open-source-based parallel computing infrastructures, such as TORQUE or Ganglia Monitoring System, now support the integration of CUDA technology. For these reasons, we developed our CARAT-GxG to perform grid computing via TORQUE, an open-source-based infrastructure enabling the control over batch jobs and distributed computing resources.

In order to evaluate the effectiveness of the GPU computing system with CARAT-GxG, an appropriate setup for a given GPU computing system is required to run CARAT-GxG. All setup and comparisons were performed on both the stand-alone GPU system and the GPU cluster. The

stand-alone GPU system contains an Intel Core i7–950 CPU and three NVIDIA GTX480 graphic cards, while the GPU cluster consists of four nodes, and each node includes two physical CPU cores with 24 threads and eight NVIDIA Tesla M2070 GPUs.

According to the system setup of TORQUE, CARAT-GxG automatically generates sequences of tasks that are independently queued by TORQUE and executed. Since the queued tasks are automatically distributed by TORQUE, GPU utilization can be maximized.

Results and Discussion

All performance comparisons between CARAT-GxG and CPU implementations were conducted in a stand-alone GPU system with an Intel Core i7–950 CPU and three NVIDIA GTX480 graphic cards. Owing to enormous time consumption of the CPU version of the R code, the whole execution time of the CPU had to be estimated from the execution time of lower (1,000) combinations in each dataset. In order to perform the comparisons, we simulated a simple dataset consisting of varying numbers of SNPs from 100 to 5,000 and samples from 100 to 1,000 with randomly generated binary phenotype. The genotypes in the dataset were also randomly generated with diverse minor allele frequencies (MAFs) using the uniform distribution between 0.05 and 0.5.

It is important to ensure maximum utilization of the GPU because the performance of GPU is highly dependent on the parameters required by GPGPU. In this context, we investigated the relationship between the analysis efficiency of CARAT-GxG and two parameters given upon executing GPU: the number of threads and the number of blocks. As shown in Figure 2C, it is clear that there is a concave trend as the number of blocks varies. In contrast, it is relatively easy to determine the number of threads, because the number of optimal threads is usually identical to the warp size of the GPU. However, it is difficult to determine the optimal number of blocks to maximize performance because of its dependence on the underlying GPU. Hence, we developed CARAT-GxG to attempt to automatically determine the appropriate number of blocks and threads.

The comparisons between CARAT-GxG and traditional CPU implementations, as shown in Table 1 for SNP–SNP interaction analysis and Figure 2B for single SNP analysis, found that CARAT-GxG performs up to almost 700 times faster. More specifically, our comparison showed that the speedup stabilized as the number of SNPs or individuals increased. The low improvement when evaluating 100 SNPs in Table 1 is caused by the overhead during the initialization and additional processes, such as data transfer between the CPU and GPU or the storage of intermediate results.

From the comparison of acceleration by adding a graphics card to the execution, CARAT-GxG showed steady performance improvement, as shown in Figure 2A. Note that as the size of dataset increases, the degree of efficiency of parallelization becomes crucial. This efficiency was in the range of 95–99% compared to the ideal performance as the size of dataset increased. In addition, CARAT-GxG performed strongly even when using more complex models with additional variables such as interaction and covariates. As shown in Table 1 and Figure 2B, CARAT-GxG outperformed the CPU implementations when calculating the two-way model including interaction.

When comparing the reliability represented by *P*-values from CARAT-GxG and CPU implementations, we observed that these *P*-values are slightly different; however, these differences are negligible and do not significantly affect the result. The representation of the floating-point number is exactly the same between CPU and GPU. However, the slightly different computation step between CPU and GPU may make a negligible discrepancy of *P*-values. Specifically, these differences were almost proportional to the size of the actual *P*-value, and the range of differences was about 10^{-4}–10^{-10} (data not shown).

The other comparison was made from the difference of rank, sorted by *P*-value for all possible iterations. Since our program supports additional optimization methods, it is possible that the restriction of the possible number of iterations seriously affects the performance. To confirm this, we checked how much the rank has changed from the full iteration, as

Table 1. Execution time of CARAT-GxG and CPU implementation in two-way testing (UNIT: SECOND).

SNPs / SAMPLES	100	300	500	1K	3K	5K
100	0.6239 (75.2351)	3.3846 (771.0163)	9.5516 (3147.0682)	38.1707 (16763.22)	327.1889 (465504.78)	976.2125 (3636110.1325)
300	1.564 (98.3813)	9.1754 (980.2416)	24.2124 (3420.8945)	102.0229 (28763.7075)	883.1818 (965256.6405)	2606.218 (3908843.075)
500	2.5504 (143.2877)	15.4253 (2030.7632)	44.5972 (6609.8788)	168.7734 (40383.0765)	1644.9804 (1008473.73)	4593.415 (5729728.825)
1,000	4.9348 (260.6621)	30.0753 (2732.666)	84.1235 (8333.9238)	327.5836 (42924.5325)	2995.5481 (1052311.613)	8662.1345 (5924214.92)

Notes: All time units are seconds. The upper and lower ones indicate the execution times of CARAT-GxG and CPU implementations, respectively.

shown in Figure 3. The number of conserved ranks was almost identical after 11 iterations, and a list of significant interactions became identical over 10 iterations.

We further evaluated the performance gain of CARAT-GxG on our GPU cluster. In the evaluation, the total analysis time was almost inversely proportional to the number of nodes requested for use by CARAT-GxG, indicating that our approach is highly suitable to implement regression analysis using GPUs with a GPU computing system. In brief, CARAT-GxG on our GPU cluster was capable of performing an exhaustive two-way linear regression analysis with the GGI of 500 K SNP chip in 12 days.

Finally, we tested CARAT-GxG with a real GWAS dataset from an AMD study.[19] We found that the top two combinations have already been identified in other studies,[20,21] and rs380390 is found in the top three combinations except the lowest *P*-value, which is identified to be significant in the initial paper of the dataset.[19] This SNP showed a large marginal effect from the result of our dataset. The top five combinations having the lowest *P*-values are shown in Table 2.

In the analysis of biological data using a bioinformatics approach, the size of data and required computing power are usually very large. Thus, the performance of a given analysis method is a critical factor in providing results for further analysis. However, as the method becomes more complex, the time required to perform analysis increases rapidly.

Motivated by these issues, we posit that GPU programming has the possibility to dramatically increase the efficiency of analysis. Its parallelism is more suited to many bioinformatics methods, which require massive amounts of independent calculation. The benefit of this was evident in our almost 700-fold execution speed performance.

In addition, CARAT-GxG showed stable performance increment by adding a new graphics card, through our GPU-specified optimization and performance tuning processes.

Table 2. A list of top five significant two-way combinations from AMD data.

COMBINATION	*P*-VALUE	PAPER
rs994542, rs9298846	2.95×10^{-14}	[20]
rs380390, rs2402053	1.09×10^{-13}	[19, 21]
rs380390, rs3775640	4.37×10^{-13}	[19]
rs380390, rs10511130	6.25×10^{-13}	[19]
rs380390, rs2125743	7.42×10^{-13}	[19]

As shown in Table 1, researchers who use the CARAT-GxG can easily estimate and control the analysis time. We expect that this predictability can provide an advantage to researchers.

From the aspect of GPU programing, we developed CARAT-GxG to gain more performance and provide more usability by parameter optimization; adjusting the number of blocks and threads is the key to optimize performance. For example, if there are 480 cores that can be utilized in GPU, running a single kernel with more than 480 blocks would take much more time, because some blocks waiting to be executed would remain. Simulation results suggest that we should take these attributes of the GPU into account. We confirmed it before the comparison, by executing CARAT-GxG with various numbers of blocks and threads. It suggested that the execution with appropriate numbers of blocks and threads can boost the performance almost twofold.

Finally, we achieved additional acceleration via an application of a GPU system using TORQUE and showed that our task-based approach analyzed the given dataset in time almost inversely proportional to the number of requested nodes by CARAT-GxG. However, there is a room for achieving more acceleration, because we still do not consider the difference between requested nodes, which can cause speed deceleration.

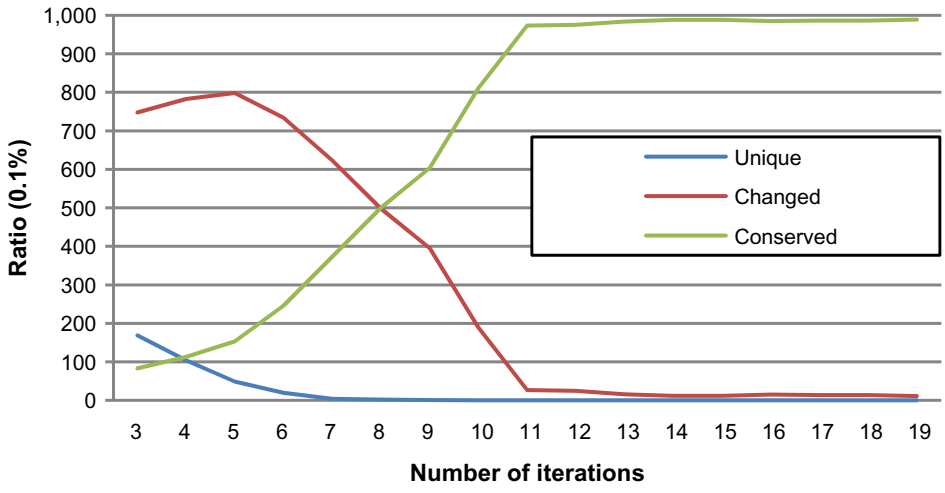

Figure 3. The red, blue, and green lines indicate the number of combinations that change in rank, vanish, and do not change in rank as the number of iterations increases, respectively.

Conclusion

Recently, a number of huge sequencing projects have delivered hundreds of completely mapped sequences, fueled by rapid advancements in sequencing technology. Consequently, the number of available variants has increased significantly. Inevitably, the development of a more computationally efficient method is required in order to investigate the GGI in large datasets. In this aspect, an extensive application of grid computing systems can considerably accelerate the speed of analysis. However, traditional CPU-based grid computing systems are essentially not suitable to perform a huge number of independent tasks. However, the application of GPU can overcome this drawback, with the high-throughput of parallel tasks and computational efficiency. We took advantage of a GPU-based computing system and developed CARAT-GxG in order to address such parallelism and efficiency issues. As a result, we successfully showed that a GPU-based computing system could effectively handle the GGI analysis of large-scale GWAS data in the regression analysis framework in an exhaustive manner. For our best knowledge, it is the first attempt to provide a toolkit that enables an exhaustive investigation of regression-based GGI in GWAS. In contrast to the variable selection-based methods, CARAT-GxG does not screen out any variant from the dataset and, thus, it has a less chance of missing the true interactions.

In addition, we also showed that there are many aspects that GPU analysis can account for, such as the number of blocks. Since regression analysis using GPU is strongly dependent on the frequent access of global memory, the optimal number of blocks is necessarily varied. By optimal allocation adjustment of the number of blocks, CARAT-GxG is dynamically tuned and achieves maximal efficiency. This kind of optimization certainly should be included in many heterogeneous systems, especially in future GPU-based computing systems.

In conclusion, we successfully showed that an exhaustive two-way GGI analysis using regression analysis can be achieved within one-and-a-half weeks in a small GPU computing system. From the comparison of traditional toolkit on a standalone GPU system, we showed that our toolset provides up to 700-folds of acceleration. This fast acceleration enables an investigation of GGI in GWAS dataset in an exhaustive manner. We expect our CARAT-GxG to provide good performance and be practically applicable to the GGI analysis of whole-genome scale datasets. Since the CARAT-GxG supports both continuous and binary phenotypes, it could be applied to many GWAS dataset with various types of phenotypes. We also expect that CARAT-GxG can be easily adapted to future large-scale GPU-based computing systems because of its high scalability.

Author Contributions

Contributed to the writing of the manuscript: SL, MSK. Jointly developed the structure and arguments for the paper: SL, MSK, TP. Contributed to the writing of the manuscript: SL, TP. Agree with manuscript results and conclusions: MSK. All authors reviewed and approved of the final manuscript.

REFERENCES

1. Hirschhorn JN, Daly MJ. Genome-wide association studies for common diseases and complex traits. *Nat Rev Genet*. 2005;6:95–108.
2. Cho YS, Go MJ, Kim YJ, et al. A large-scale genome-wide association study of Asian populations uncovers genetic factors influencing eight quantitative traits. *Nat Genet*. 2009;41:527–34.
3. Zeggini E, Scott LJ, Saxena R, et al. Meta-analysis of genome-wide association data and large-scale replication identifies additional susceptibility loci for type 2 diabetes. *Nat Genet*. 2008;40:638–45.
4. Sim X, Ong RT, Suo C, et al. Transferability of type 2 diabetes implicated loci in multi-ethnic cohorts from Southeast Asia. *PLoS Genet*. 2011;7:e1001363.
5. Yasuda K, Miyake K, Horikawa Y, et al. Variants in KCNQ1 are associated with susceptibility to type 2 diabetes mellitus. *Nat Genet*. 2008;40:1092–7.
6. Ferreira MA, O'Donovan MC, Meng YA, et al. Collaborative genome-wide association analysis supports a role for ANK3 and CACNA1C in bipolar disorder. *Nat Genet*. 2008;40:1056–8.
7. Scott LJ, Muglia P, Kong XQ, et al. Genome-wide association and meta-analysis of bipolar disorder in individuals of European ancestry. *Proc Natl Acad Sci U S A*. 2009;106:7501–6.
8. Turnbull C, Ahmed S, Morrison J, et al. Genome-wide association study identifies five new breast cancer susceptibility loci. *Nat Genet*. 2010;42:504–7.
9. Tomlinson IP, Webb E, Carvajal-Carmona L, et al. A genome-wide association study identifies colorectal cancer susceptibility loci on chromosomes 10p14 and 8q23.3. *Nat Genet*. 2008;40:623–30.
10. Landi MT, Chatterjee N, Yu K, et al. A genome-wide association study of lung cancer identifies a region of chromosome 5p15 associated with risk for adenocarcinoma. *Am J Hum Genet*. 2009;85:679–91.
11. Fan J, Lv J. Sure independence screening for ultrahigh dimensional feature space. *J R Stat Soc*. 2008;70:849–911.
12. Ueki M, Tamiya G. Ultrahigh-dimensional variable selection method for whole-genome gene–gene interaction analysis. *BMC Bioinformatics*. 2012;13:72.
13. Wan X, Yang C, Yang Q, et al. BOOST: a fast approach to detecting gene–gene interactions in genome-wide case-control studies. *Am J Hum Genet*. 2010;87:325–40.
14. Kwon MS, Kim K, Lee S, Park T. cuGWAM: genome-wide association multifactor dimensionality reduction using CUDA-enabled high-performance graphics processing unit. *Int J Data Min Bioinform*. 2012;6:471–81.
15. Ritchie MD, Hahn LW, Roodi N, et al. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. *Am J Hum Genet*. 2001;69:138–47.
16. Manavski SA, Valle G. CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment. *BMC Bioinformatics*. 2008;9 (suppl 2):S10.
17. Liu Y, Popp B, Schmidt B. CUSHAW3: sensitive and accurate base-space and color-space short-read alignment with hybrid seeding. *PLoS One*. 2014;9(1):e86869.
18. Liu Y, Maskell DL, Schmidt B. CUDASW++: optimizing Smith-Waterman sequence database searches for CUDA-enabled graphics processing units. *BMC Res Notes*. 2009;2:73.
19. Klein RJ, Zeiss C, Chew EY, et al. Complement factor H polymorphism in age-related macular degeneration. *Science*. 2005;308:385–9.
20. Wan X, Yang C, Yang Q, Xue H, Tang NL, Yu W. Detecting two-locus associations allowing for interactions in genome-wide association studies. *Bioinformatics*. 2010;26:2517–25.
21. Han B, Chen XW, Talebizadeh Z. FEPI-MB: identifying SNPs-disease association using a Markov Blanket-based approach. *BMC Bioinformatics*. 2011;12 (suppl 12):S3.