Parallelized pipeline for whole genome shotgun metagenomics with GHOSTZ-GPU and MEGAN

Masahito Ohue*, Marina Yamasawa*†, Kazuki Izawa* and Yutaka Akiyama*
*Department of Computer Science, School of Computing, Tokyo Institute of Technology
2-12-1 Ookayama, Meguro-ku, Tokyo 152-8550, Japan
Email: {ohue, akiyama}@c.titech.ac.jp, {yamasawa, izawa}@bi.c.titech.ac.jp
Telephone: +81-3-5734-3645, Fax: +81-3-5734-3646
†Real World Big-Data Computation Open Innovation Laboratory (RWBC-OIL),
National Institute of Advanced Industrial Science and Technology (AIST)
1-1-1 Umezono, Tsukuba, Ibaraki 305-8560, Japan

***Abstract*—Metagenome techniques allow analyses of microorganisms and their genes present in a given environment without isolation and culture. Thus, metagenomics has become a broadly applied tool to study various environments and elucidate the relationship between diseases and the host microbiota. With continuous improvement in the performance of genome sequencers, the number of sequence reads generated has increased exponentially; thus, methods for the efficient processing of such large numbers of sequences are required. To this end, we developed the pipeline system, GHOSTMEGAN, to speed up the processing of large-scale whole genome shotgun metagenome analysis, which integrates the sequence homology search tool GHOSTZ-GPU and the analyzing tool MEGAN. Assuming a cluster-type computer with a job scheduling system, the multi-node parallel processing of GHOSTZ-GPU and MEGAN was pipelined. Performance evaluation of GHOSTMEGAN with a whole genome sequence dataset, the oral metagenome demonstrated that execution of 128 nodes in parallel, which required 15 h on a single node, could be completed in only 20 min, thereby achieving about 45 times faster calculation. This pipeline is expected to greatly accelerate the field of metagenomics and broaden its application potential.**

***Keywords*-whole genome shotgun metagenomics; sequence homology search; parallel computing; metagenome pipeline**

I. Introduction

The metagenome reflects the entire community of microorganisms and their genes in a certain environment. Especially, in the context of human diseases and host microbiota, many studies have suggested an association between the microbiome and pathogenic processes [1]–[3]. There are two main types of metagenome analysis, namely 16S analysis, which analyzes only 16S ribosomal RNA sequences, which are used as a fingerprint of the species and identify only the species of microbes, and whole genome shotgun (WGS) analysis, which analyzes the entire microbial genome. Although 16S analysis incurs only a small computing cost for the sequence search, it provides information only about the microbial taxa in a given environment. In WGS analysis, a sequence homology search is performed for all reads derived from microorganisms; therefore, even if the genome is not sequenced, it is possible to estimate the function of unknown sequences from the alignment with homologous genes. Thus, the biological role in the environment can be clarified with higher-resolution analysis. However, the drawback of WGS analysis is the substantial amount of data and also bigger reference database size required compared to 16S analysis; therefore, speeding up the process by using a parallel distributed technology is crucially needed.

In WGS metagenome analysis, an analysis pipeline is mainly performed by combining two main processes: a sequence homology similarity search and an analysis, which corresponds to a post-processing step, that transforms the search results into a form that is easy to interpret. Traditionally, BLAST [4] has been the most widely used tool for conducting a sequence homology search; however, faster search methods have become available in recent years, such as RAPSearch [5], DIAMOND [6], GHOSTZ-GPU [7], [8], and MMseqs2 [9]. Among these, GHOSTZ-GPU supports multi-GPU processing, which is expected to speed up computation significantly. Suzuki *et al*. reported that a homology search with GHOSTZ-GPU using three GPUs was about 2,000 times faster than a typical BLAST search [8]. In the output of the sequence homology search tool, the search results are arranged for each read sequence. In the case of metagenomic analysis, a text file with more than tens of millions of lines is generated, which requires analysis processing to facilitate interpretation of the specific microbiota in the environment. Examples of standard analyzing tools used for this purpose include MEGAN [10] and HUMAnN [11].

Traditionally, the sequence homology search step was the main bottleneck for WGS metagenome analysis [12]. However, the development of high-speed sequence homology search tools using GPU implementation and parallelization such as GHOSTZ-GPU has resolved this issue; thus, the analysis step has now become the main bottleneck for conducting such analyses efficiently. Since substantial at-

2471-7819/19/$31.00 ©2019 IEEE
DOI 10.1109/BIBE.2019.00035

tention was paid to improving sequence homology searches, attention must now focus on the development of an analysis tool capable of processing a large number of WGS analysis results in parallel; however, there is currently no parallel pipeline tool that can handle a series of processes continuously.

MetaWRAP [13], MEGAN [10], and MiGAP [14] were developed as pipeline tools for WGS metagenome analysis. However, MetaWRAP does not handle sequence homology searches and preferably performs metagenome assembly to connect read sequences. It does not support multi-node parallelization. As mentioned above, MEGAN analysis processing consists of the following three functionalities; totaling the output of the sequence homology search using NCBI taxonomy database [15], phylogenetic classification, and calculation of the relative abundance of each taxon based on the lowest common ancestor [10]. Moreover, MEGAN has a sequence homology search function powered by DIAMOND, which can be used as a pipeline [10]. However, DIAMOND does not support GPU acceleration, and thus a high-speed sequence homology search is not possible. MiGAP uses BLAST, and thus also cannot perform high-speed sequence homology searches.

Therefore, to overcome these hurdles and limitations of current tools, we here report a novel WGS metagenome analysis system called GHOSTMEGAN that pipelines the sequence homology search and analysis processing and performs distributed computation on parallel computers by linking GHOSTZ-GPU, which is the fastest sequence homology search tool that supports multi-GPU computation, and MEGAN's powerful analyzing function. The performance of GHOSTMEGAN was evaluated using an actual WGS metagenome dataset by parallel execution on the TSUBAME 3.0 supercomputing system.

II. GHOSTMEGAN: A New Pipeline Tool for WGS Metagenome Analysis

GHOSTMEGAN is a pipeline system designed to perform a series of WGS metagenome analysis processes in parallel on a multi-node and multi-GPU heterogeneous cluster. GHOSTMEGAN consists of the following four steps (Fig. 1): *A.* dividing query file (single-node, CPU)—dividing query read sequences into a predetermined number, *B.* sequence homology search by GHOSTZ-GPU (multi-node, multi-GPU)—performing a sequence homology search by data parallelization of query sequences, *C.* analysis processing by MEGAN (multi-node, CPU)—executing the analysis process based on the sequence homology search results for query sequences, and *D.* and integrating results (single-node, CPU)—integrating the results of queries processed by multiple nodes.

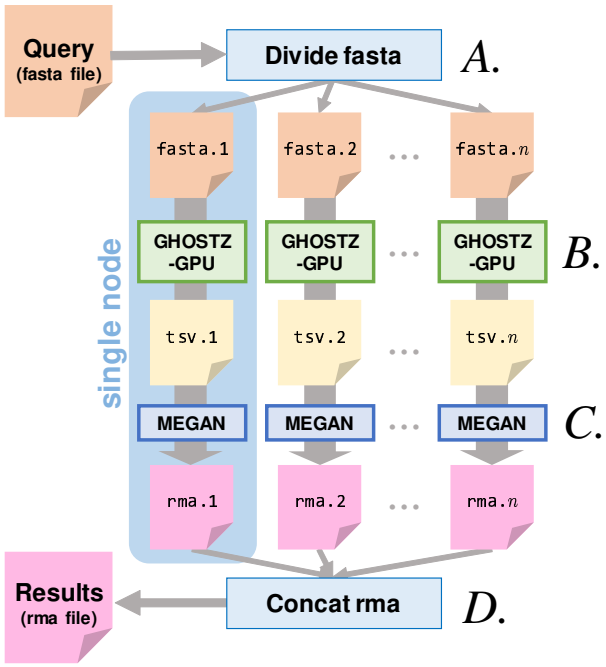

Figure 1. GHOSTMEGAN pipeline workflow

A. Dividing query

The input file (query) for WGS metagenome analysis is a huge single text file (fasta format) of read sequences. In this step, this query file is divided using an equal number of divisions that are represented as nodes n. Since the processing time for dividing queries is extremely small compared with that required for the other steps, this step was not implemented in parallel.

B. Sequence homology search by GHOSTZ-GPU

GHOSTZ-GPU [8] is executed for individual divided query files using one compute node. GHOSTZ-GPU performs thread parallel computation using all of the CPU cores and GPUs in one compute node. Since the reference genome database file is typically about 100 GB, it can be stored in the local storage in all nodes that execute the sequence homology search. The output of GHOSTZ-GPU is a tab-delimited text file, which is the same format provided by a BLAST search. The analysis process only targets search results that satisfy E-value $< 10^{-5}$, so that any search results exceeding this threshold are pruned at this step. Note that the search results may slightly differ from those obtained without the dividing procedure owing to the presence of sequences with the same alignment score. However, this has almost no effect on the final biological analysis results.

C. Analyzing by MEGAN

MEGAN [10] works on the output file of GHOSTZ-GPU as the input. Although standard MEGAN processing

Table I
TSUBAME 3.0 COMPUTE NODE SPECIFICATION (F_NODE)

CPU	Intel Xeon E5-2680v4 2.4 GHz (14 cores) × 2
GPU	NVIDIA Tesla P100 NVLink (16 GB) × 4
RAM	256 GiB
Local storage	Intel SSD DC P3500 2 TB
Network	Intel Omni-Path 100 Gb/s × 4
Job scheduler	Univa Grid Engine 8.5.4C104_11

is performed on the GUI environment, analysis can be accomplished on the CUI by using the MEGAN `blast2rma` command. The computation is then performed independently for each read sequence search result in the rma file, which will not be affected by dividing of queries.

D. Integrating results

After all the analyses performed by MEGAN in parallel are completed, the MEGAN `compute-comparison` command is run, which integrates multiple analysis results into a single file, and MEGAN `extract-biome` is then used to summarize the integrated whole results.

Ensuring usability as a pipeline

To ensure usability, only one parameter file needs to be edited. The parameter file is in a format that can be read by Python configparser, and all necessary options can be described. Main procedure is described in bash script file, and it is possible to complete a series of executions such as generating and executing Python programs and wrapper shell scripts and submitting jobs to the queueing system.

III. EXPERIMENTAL SETTINGS

A. System and software environment

We evaluated the performance of our pipeline GHOSTMEGAN on the TSUBAME 3.0 supercomputing system (https://www.t3.gsic.titech.ac.jp/en) at Tokyo Institute of Technology, Japan. Table I shows the hardware specification of TSUBMAE 3.0 compute node (f_node). Each compute node has 28 CPU cores and 4 Tesla P100 GPUs. GHOSTZ-GPU ver. 1.1.0 and MEGAN ver. 6.12.6 were used in these experiments. The `$ ghostz-gpu aln -d [DB] -b 1 -q d -a 1 -g 3 -i [query]` option was used in GHOSTZ-GPU, and the `$ blast2rma --in [GHOSTZ output] --out [MEGAN rma file] --format BlastTab` option was used in MEGAN `blast2rma`.

B. Dataset

The query used for performance evaluation was a portion of the human oral WGS metagenome dataset obtained by Duran-Pinedo *et al.* [16]. The query used a random sample of 1,000,000 reads from periodontally healthy individual samples. The query file size was 145 MB. The NCBI nr genome sequence database was used as the reference database, including 166,109,435 sequences with a file size of 101 GB (ftp://ftp.ncbi.nih.gov/blast/db/, accessed August 18, 2018).

C. Evaluation method

We performed GHOSTMEGAN with n nodes running in parallel using n of 1 (no division), 2, 4, 8, 16, 32, 64, and 128 as the query division number, respectively, and compared the execution times and speedup rates. The execution time did not include the waiting time from when the job was submitted by the queueing system until the job was started.

IV. RESULTS AND DISCUSSION

A. Overall pipeline execution time

The execution times of the entire GHOSTMEGAN pipeline for each number of nodes n are shown in Fig. 2, and the corresponding speedup rates are shown in Fig. 3. The results clearly showed that the computing time was successfully reduced with an increase in the number of parallel computing nodes, each equipped with four GPU cards. The processing took about 15 h for one node and was completed in about 20 min with 128 nodes, representing an approximate 45-times acceleration in the computation speed. However, as shown in Fig. 3, the speedup rate showed a gradual slope with the increase in the parallel number n, and linear speed improvement could not be obtained. Thus, based on this evaluation dataset, it is presumed that speedup cannot be obtained by parallelization with $n = 128$ or more.

B. GHOSTZ-GPU execution time

With GHOSTZ-GPU, a good speed improvement was obtained up to $n = 32$, with no significant reduction in execution time at $n = 64, 128$. The appropriate value of n depends on the size of the query sequence. Therefore, with about 1 million reads, $n = 32$ can be selected from the perspective of efficiency, and $n = 64, 128$ can be chosen if the reduction in calculation time is a priority with inferior efficiency. Note that GHOSTZ-GPU was too fast, and the speedup rate was saturated with the query data in this study. However, if the amount of query data is increased, the measurement of the execution time on a single node cannot be performed.

C. MEGAN execution time

In MEGAN, the speed improvement rate was nearly linear (strong scaling was 0.87) even at $n = 128$. In this study, the measurements were only made up to $n = 128$ owing to the limitations of the TSUBAME 3.0 system; however, even higher speeds can be expected at $n = 256$, for example.

D. Execution times of other processes

The execution times for dividing queries, submitting jobs, and integrating results are shown in Table II. When n was small, these execution times were almost negligible; however, at $n = 64, 128$, they were non-negligible compared with the total computation time. Concerning query

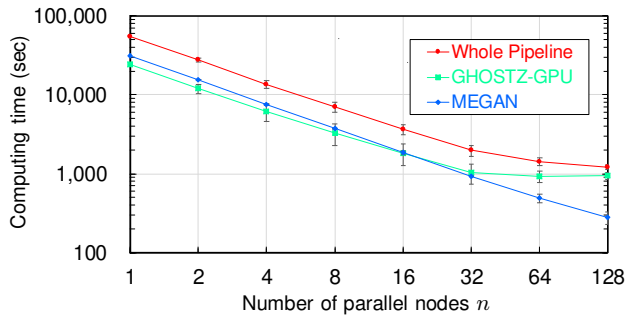

Figure 2. Execution time of the whole pipeline, GHOSTZ-GPU, and MEGAN for different numbers of nodes run in parallel

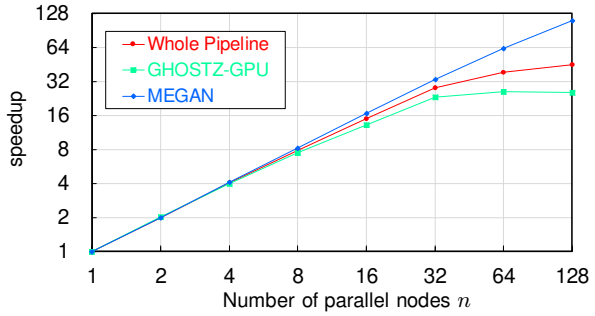

Figure 3. Speedup rates of the whole pipeline, GHOSTZ-GPU, and MEGAN

Table II
EXECUTION TIMES FOR DIVIDING QUERIES, SUBMITTING JOBS, AND INTEGRATING RESULTS

n	1	2	4	8	16	32	64	128
time (sec)	-	40.4	45.8	50.3	64.8	102	151	239

decomposition, the file I/O time increased as n increased, but this only represented a small fraction of the total time. The reason for the increase in execution time with an increase in the number of parallel nodes is considered to represent the delay of the TSUBAME 3.0's queueing system when submitting jobs.

E. Differences in search results

GHOSTZ-GPU search results may slightly differ depending on the query sequence decomposition. As mentioned above, this occurs because of the presence of sequences with the same alignment score along with the limited number of higher-ranked sequences that are stored internally. We found only two reads with different homology search results out of 1 million reads in the parallel computing of GHOSTMEGAN for the dataset. Table III shows the results for each number of parallel nodes. For read a, the search

Table III
READS WHICH HAVE DIFFERENT SEARCH RESULTS ACCORDING TO PARALLEL EXECUTION

n	1	2	4	8	16	32	64	128
read a	s_a	s'_a	s_a	s_a	s_a	s_a	s_a	s_a
read b	s'_b	s_b	s_b	s_b	s_b	s_b	s_b	s_b

s_a	XP_019376199.1 PREDICTED: tigger transposable element-derived protein 1-like, partial [*Gavialis gangeticus*]
s'_a	XP_025968818.1 LOW QUALITY PROTEIN: tigger transposable element-derived protein 1-like [*Dromaius novaehollandiae*]
s_b	EHH57573.1 hypothetical protein EGM_07242, partial [*Macaca fascicularis*]
s'_b	BAD18412.1 unnamed protein product [*Homo sapiens*]

results differed only with $n = 2$ divisions, and for read b, the calculation results only differed when no division was performed. s_a and s'_a, the hit sequences of read a, were annotated as widely conserved equivalent genes "tigger transposable element-derived protein 1-like," and the result did not affect the WGS metagenome analysis. Besides, both s_b and s'_b were annotated as function-unknown genes, and this result also did not affect the metagenome analysis at this time. The evaluation demonstrated that GHOSTMEGAN's parallel computation virtually showed no difference in the final biological analysis results.

F. Potential for speed up according to the number of parallel compute nodes

The speed of the sequence homology search process began to saturate at around $n = 64$ for this one-million-sequence query, but the analyzing process showed an excellent speedup rate even with $n = 128$ in parallel. The main contributor to this degradation in the parallelization performance of GHOSTZ-GPU is considered to be the increase in the calculation time of the ratio of a specific I/O such as database chunk reading owing to the smaller query size per process. However, the query size used this study is for evaluation of parallel efficiency and is smaller than the size used for the extensive metagenomic research. For example, the periodontal disease study [16] used data of about 37 million reads, which is much larger than the one million reads used in this study. Therefore, in an actual application, calculation with better parallel efficiency should be possible even with 128 or more nodes. In this connection, the calculation speed of GHOSTZ-GPU for $n = 128$ was not slower than that for $n = 64$ in this study. When 128 nodes are ensured as the computational resource, the parallelization performance is inferior, but calculation with 128 nodes offers the advantage of obtaining metagenome analysis results in the fastest time.

V. Conclusion

GHOSTMEGAN, a pipeline of parallelized sequence homology search and analysis processing, was developed and evaluated to improve large-scale metagenomic analysis. GHOSTMEGAN achieved parallel computing by dividing the query sequences and executing GHOSTZ-GPU and MEGAN on multiple computing nodes. The evaluation experiment showed that execution of 128 nodes in parallel could be achieved in only 20 min, representing a 45-times increase in the speed compared to the processing of one node.

The GHOSTZ-GPU used in this evaluation is a sequence homology search tool that was significantly accelerated by the use of multiple GPUs, whereas MEGAN is not boosted by GPU. Since it will be essential to withstand further increases in data sizes in the future, GPU implementation of analysis tools will be crucial.

Acknowledgment

This work was partially supported by KAKENHI (Grant No. 17H01814, 18K18149) from the Japan Society for the Promotion of Science (JSPS), the Program for Building Regional Innovation Ecosystems "Program to Industrialize an Innovative Middle Molecule Drug Discovery Flow through Fusion of Computational Drug Design and Chemical Synthesis Technology" from the Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT), the Research Complex Program "Wellbeing Research Campus: Creating new values through technological and social innovation" from the Japan Science and Technology Agency (JST), and AIST-Tokyo Tech Real World Big-Data Computation Open Innovation Laboratory (RWBC-OIL), National Institute of Advanced Industrial Science and Technology (AIST). We would like to thank Editage (www.editage.com) for English language editing.

References

[1] C. Quince, A.W. Walker, J.T. Simpson, *et al.* Shotgun metagenomics, from sampling to analysis. *Nat Biotech*, 35(9), 833, 2017.

[2] J.R. Marchesi, D.H. Adams, F. Fava, *et al.* The gut microbiota and host health: a new clinical frontier. *Gut*, 65(2), 330–339, 2016.

[3] J. Wirbel, P.T. Pyl, E. Kartal, *et al.* Meta-analysis of fecal metagenomes reveals global microbial signatures that are specific for colorectal cancer. *Nat Med*, 25(4), 679, 2019.

[4] S.F. Altschul, W. Gish, W. Miller, *et al.* Basic local alignment search tool. *J Mol Biol*, 215(3), 403–410, 1990.

[5] Y. Zhao, H. Tang, Y. Ye. RAPSearch2: a fast and memory-efficient protein similarity search tool for next-generation sequencing data. *Bioinformatics*, 28(1), 125–126, 2011.

[6] B. Buchfink, C. Xie, D.H. Huson. Fast and sensitive protein alignment using DIAMOND. *Nat Methods*, 12(1), 59–60, 2015.

[7] S. Suzuki, M. Kakuta, T. Ishida, Y. Akiyama. Faster sequence homology searches by clustering subsequences. *Bioinformatics*, 31(8), 1183–1190, 2015.

[8] S. Suzuki, M. Kakuta, T. Ishida, Y. Akiyama. GPU-acceleration of sequence homology searches with database subsequence clustering. *PLoS ONE*, 11(8), e0157338, 2016.

[9] M. Steinegger, J. Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotech*, 35, 1026–1028, 2017.

[10] H.H. Daniel, B. Sina, F. Isabell, *et al.* MEGAN community edition - interactive exploration and analysis of large-scale microbiome sequencing data. *PLoS Comput Biol*, 12(6), e1004957, 2016.

[11] S. Abubucker, N. Segata, J. Goll, *et al.* Metabolic reconstruction for metagenomic data and its application to the human microbiome. *PLoS Comput Biol*, 8(6), e1002358, 2012.

[12] S.V. Angiuoli, J.R. White, M. Matalka, *et al.* Resources and costs for microbial sequence analysis evaluated using virtual machines and cloud computing. *PLoS ONE*, 6(10), e26624, 2011.

[13] G.V. Uritskiy, J. DiRuggiero, J. Taylor, MetaWRAP-a flexible pipeline for genome-resolved metagenomic data analysis. *Microbiome*, 6(1), 158, 2018.

[14] H. Sugawara, A. Ohyama, H. Mori, K. Kurokawa. Microbial genome annotation pipeline (MiGAP) for diverse users. in *20th Intl Conf Genome Inform 2009*, S-001, 1–2, 2009.

[15] S. Federhen. The NCBI taxonomy database. *Nucl Acids Res*, 40, D136–D143, 2012.

[16] A.E. Duran-Pinedo, T. Chen, R. Teles, *et al.* Community-wide transcriptome of the oral microbiome in subjects with and without periodontitis. *ISME J*, 8(8), 1659–1672, 2014.