Parallel-META: A High-Performance Computational Pipeline for Metagenomic Data Analysis

Xiaoquan Su, Jian Xu, Kang Ning*

Qingdao Institute of Bioenergy and Bioprocess Technology, Chinese Academy of Sciences, Qingdao, Shandong, China

suxq@qibebt.ac.cn, xujian@qibebt.ac.cn, ningkang@qibebt.ac.cn (*corresponding author)

Abstract—**Metagenomics method directly sequences and analyzes genome information from microbial communities. There are usually more than hundreds of genomes from different microbial species in the same community, and the main computational tasks for metagenomics data analysis include taxonomical and functional component of these genomes in the microbial community. Metagenomic data analysis is both data- and computation- intensive, which requires extensive computational power. Most of the current metagenomic data analysis softwares were designed to be used on a single computer, which could not match with the fast increasing number of large metagenomic projects' computational requirements. Therefore, advanced computational methods and pipelines have to be developed to cope with such need for efficient analyses. In this paper, we proposed Parallel-META, a GPU- and multi-core-CPU-based open-source pipeline for metagenomic data analysis, which enabled the efficient and parallel analysis of multiple metagenomic datasets. In Parallel-META, the similarity-based database search was parallelized based on GPU computing and multi-core CPU computing optimization. Experiments have shown that Parallel-META has at least 15 times speed-up compared to traditional metagenomic data analysis method, with the same accuracy of the results (http://www.bioenergychina.org:8800/).**

Keywords: metagenomics, data-intensive computing, high performance computing

I. INTRODUCTION

The total number of microbial cells on earth is huge: approximate estimation of them is 10^{30} [1], and the genomes of these vastly unknown communities of microbes might contain a large number of novel genes with useful functions. However, more than 99% of microbe species were unknown and un-cultivable [2], making traditional isolation and cultivation process non-applicable. Analysis of their metagenomic data is the direct and efficient way to analyze all microbes in the community [3]. The metagenomic approach has made it possible better understanding of microbial diversity as well as their functions and interactions. And the broad applications of metagenomic research, including environmental sciences, bioenergy research and human health, have made it an increasingly popular research area.

Metagenomics research was based on sequencing data from 16S rRNA amplicon, or large-scale shot-gun whole-genome metagenomic sequencing. Early 16S rRNA-based metagenomic survey of microbial communities focused on 16S ribosomal RNA sequences which are relatively short, often conserved within a species, and different between species. The 16S rRNA-based metagenomic survey has already produced data for analysis of microbial communities of Sargasso Sea [4], acid mine drainage biofilm [5] and human gut microbiome [6]. Facilitated with Next-Generation-Sequencing (NGS) techniques [7], current metagenomic research has been advanced rapidly. NGS techniques could produce millions of reads at very high speed with relatively low price, thus it enables sequencing at much greater depth. Based on NGS techniques and high performance computational analysis methods, many large-scale metagenomic research projects have been conducted [8], thus made the large-scale metagenomic research the mainstream in metagenomic research. In this paper, we were focusing on data analysis for shot-gun whole-genome metagenomic sequencing, in which the computational methods play very important roles, especially the similarity-based database search.

The primary goal of metagenomics is the assessment of taxonomic and functional diversity of microbial communities. Based on NGS data, metagenomic data analysis is both data- and computing-intensive. Therefore, high-performance computing is needed for metagenomic data analysis.

Traditional high performance computing platform only use CPU cluster. For high computing speed, CPU computing platform always has large amount of CPUs with high frequent, which also accompanied with high cost and high power consumption. However, with the increase of data size, it becomes more and more difficult for current CPU cluster to satisfy the requirement of the fast-developing metagenomics research. The computing speed of metagenomics data analysis would be accelerated significantly by the combination of GPU computing and parallel CPU computing. For GPU computing, the GPGPU(General Purpose Graphic Process Unit) hardware and CUDA(Compute Unified Device Architecture) software would be the method of choice. CUDA is a massive parallel computing architecture. Based on nVIDIA (Santa Clara, CA) GPGPUs and SIMT (Single Instruction Multiple Threads), it enables dramatic increases in computing performance by parallel computing with huge number of stream processors. For parallel CPU computing, multi-core CPU could be utilized by implementation of multi-threaded parallel programming.

978-1-4577-1666-9/11/$26.00 ©2011 IEEE

In this work, we used both GPU and multi-core CPU to implement the parallel computing to accelerate the computation. We have proposed a high-performance computational pipeline (Parallel-META) for metagenomics research that has the major advantage of efficient process of large metagenomics dataset. The whole system is illustrated in Figure 1. There were two major components of the system: Multi-core CPU and GPU computing facilities, which enabled the hardware support for parallel process of large metagenomics datasets; and high-performance metagenomics data analysis pipeline, which enabled the software support for parallel metagenomic data analysis.

Figure 1.Parallel computing platform based on GPU and multi-core CPU hardware as well as software pipeline.

II. METAGENOMICS AND HIGH-PERFORMANCE COMPUTING

A. Metagenomics

Large databases of reference sequences, such as Greengenes [9], SILVA [10] and RDP [11] already exist for metagenomic sequence analysis. As most of the microbial communities are still unknown, these databases are also updating frequently. For computational analysis of metagenomic data, the most important tasks include taxonomic and functional analyses. A crucial step in the taxonomic analysis of large-scale metagenomic data is "binning", in which the metagenomic sequences were assigned to phylogenetic groups according to their taxonomic origins at different resolutions: from "kindom" to "genus" level. There are two categories of binning methods: similarity-based methods that align reads to reference databases, and composition-based methods that use composition patterns (GC content, k-mer frequency, etc.) to cluster reads. The similarity-based methods classify sequences based on sequence homology, which is determined by reference database searches using general purpose alignment tools such as BLAST [12]. The most frequently used similarity-based metagenomic data binning methods include MEGAN [13], CARMA [14] and Sort-ITEM [15]. Most of these software could be used on PC workstation. However, similarity-based methods rely on reference databases that contain sequences of known genomes, so these methods cannot classify the majority of sequences that were from unknown genomes without close references. In contrast, composition-based methods analyze intrinsic sequence features such as GC content, codon usage and k-mer frequency, and compare these features with reference genome sequence of known taxonomic origins. The most frequently used composition-based metagenomic data binning methods include TETRA [16] and PhyloPhythia [17]. Recently, some all-in-one metagenomic data analysis pipelines were introduced, such as Phyloshop [18] and QIIME [19]. The web-based metagenomic annotation platforms, such as MG-RAST [20] and CAMERA [21] were also designed to analyze metagenomic data. However, the increasing number of metagenome data analysis projects needs more and more computational power, which become an increasingly large huddle for the efficient process of metagenome datasets by current pipelines.

B. GPU computing

CUDA is a massive parallel computing model based on GPGPU (GPU for short) to solve the rapid increasing data computing problem. It is presented by nVIDIA in 2006 with the G80 series GPU. Different form the traditional GPUs, which are consisted of rendering pipeline of Vertex Engine and Pixel Engine, a CUDA enabled GPU is composed by several SMs (Stream Multiprocessors). The amount of SM depends on the model of GPU. For example, nVIDIA Tesla has 30 SMs, and nVIDIA Quadro FX 880M has 6 SMs. In a single SM, there are also several stream processors and a shared memory which can be accessed by these processors in the same SM. For G80/GT100/GT200 series GPU, one SM is composed by 8 stream processors. The latency of the shared memory is quite low, so it is always used as cache. There is also an onboard memory (Global Memory) which can be shared by all the stream processors in a GPU. As GPU cannot directly access the RAM of a computer system, data should be transferred from RAM to Global Memory before GPU computation.

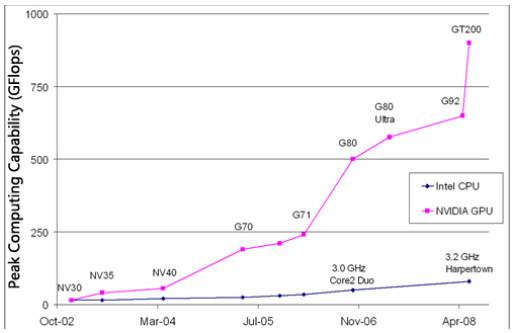

Figure 2. The peak computing capability comparison between CPU and GPU

Based on SIMT (Single Instruction Multiple Threads) structure, GPU can invocate a block of threads on one single SM. Each thread performs a single computation on one stream processor. For one block, the maximum number of thread is 512 for the GPU with computation capability 1.X and 1024 for the GPU with computation capability 2.0. Therefore, Total

978-1-4577-1666-9/11/$26.00 ©2011 IEEE

thread number = (Number of Threads in one single Block) X (Number of Blocks). This number can be very large as there might be huge number of SMs in one GPU, meaning that many threads can be executed parallelly at the same time. That is the main reason for the high computing capability and throughput of GPU rather than CPU (Figure 2).

In a computer system equipped with GPU, the CPU system is called host, and the GPU system is called device. CUDA provides a series of APIs which can be invocated by host programs. As GPU cannot directly access the system memory of CPU and Hard disk, data should be transferred from the system memory (RAM) to the Global Memory of GPU by CUDA APIs. Then the stream processors of GPU can exchange data with the Global Memory and Shared Memory.

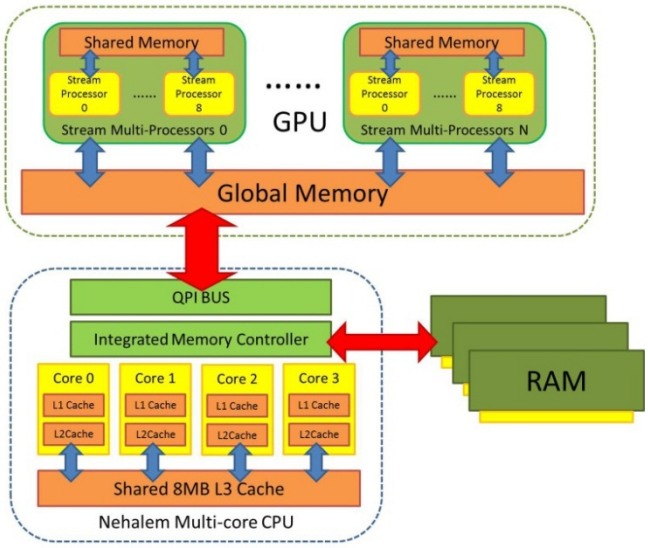

Figure 3. The architecture of GPU & CPU and data transfer in a computer system

C. Multi-core CPU computing

CPU core is the key part made of Monocrystalline silicon on which instructions can be executed. Before 2005, for normal CPUs, there was only 1 core on one single CPU chipset, which limited the development of the computing capability and efficiency. Engineers used the method that integrating several CPU cores on a chipset to solve these problems. Instructions can be executed on those cores parallelly at the same time. This method not only enhanced the computing capability of CPU, but also reduced the TDP (Thermal Design Power) for cutting down the working voltage and clock rate of CPU cores and the application of power management technology.

The latest architecture of Intel multi-core CPU is Nehalem. Nehalem Xeon 5000 series have such architecture features: (a) four cores integrated into one CPU, (b) hyper-threading technology supports 8 threads at most, (c) each core has a 64KB L1 cache and 256KB L2 cache, with an 8MB L3 cache is shared by all cores and (d) turbo Boost technology, dynamically adjusts the work frequent of cores.

A computer system with both GPU and multi-core CPUs is illustrated in Figure 3. This would be a typical hardware architecture for next-generation high performance biological data analysis system, based on which we are testing the Parallel-META metagenomics data analysis pipeline.

III. SYSTEM ARCHITECTURE AND PIPELINE DESIGN

A. Hardware architecture

In this work, the hardware used was one single node of the GPU computing platform of QIBEBT, CAS (Qingdao Institute of Bioenergy and Bioprocess Technology, Chinese Academy Sciences) computing platform that had the following configuration: CPU: Dual Intel Xeon X5645 2.66 GHz with 12 cores, GPU: nVIDIA Tesla C2070 with 448 processors and 6G DDR5 ECC on board memory, RAM: 72GB RDIMM DDR3.

For this system, the total float computing capability of CPU is 89.6Gflops, and the total float computing capability of GPU is up to 1Tflops.

B. Software architecture

The keys to the efficient and parallel process of large metagenomics data is the parallelization of sequence data binning by database search. The Parallel-META pipeline is using the Phyloshop software [18] as the basic framework. Phyloshop is a pipeline which can extract 16S rRNA gene fragments form metagenomic sequences, assign the taxonomy terms for the identified 16S rRNA fragments, and report the taxonomy distribution. This pipeline includes three steps: (1) 16S rRNA prediction by HMM search of HMMER [22], (2) Mapping of 16S rRNA onto the Greengenes [9] core set using megaBLAST [23], (3) Classification of the 16S rRNA fragments based on the mapping to the phylogenetic tree. After these steps, Phyloshop reports the classification, length distribution and the summary of the taxonomic assignments of 16S rRNA sequences at different phylogenetic levels.

To speed up the metagenomic data analysis process, the Parallel-META pipeline is optimized by decomposing large problems into smaller size sub problems and solving them parallelly at the same time on high performance computing devices. Parallel-META mainly optimized the HMM search part and database search part by megaBLAST. The overall pipeline design was illustrated in Figure 4.

In the HMM search part, we used GPU-HMMER [24] component to implement parallel computing instead of traditional HMMER which is based on CPU. The core of HMM search is the Viterbi Algorithm, which is used to compute the most probable path through a given state HMM (Hidden Markov Model). Different from CPU computing that performs HMM search by serially executing the loops of Viterbi Algorithm. In GPU-HMMER, the loops are parallelized and expanded into some sub processes. Then each process was mapped to a thread on a stream processor of GPU. As GPU enable the activities of huge number of threads at the same time, Viterbi Algorithm can be done in a much shorter time on GPU than on CPU.

978-1-4577-1666-9/11/$26.00 ©2011 IEEE

The next step – megaBLAST has been divided into three parts: Problem Decomposition, Parallel Computing, and Result Combination. In the first part, the output data gained from the HMM search were decomposed into sub data files with similar size. Then in the second part, each thread could directly find its input data from the original file and perform the megaBLAST search parallelly by multi-threads programming. After that, sub results are merged together to get the final result.

IV. EXPERIMENTS AND DISCUSSIONS

We have used four sets of Illumina Solexa GAIIx sequencing-based metagenome data [25] to evaluate the performance of Parallel-META. Shotgun pair-end libraries of total saliva genomic DNA was prepared (two from the healthy population and the other two from the caries-active population). Each metagenomic DNA library was then sequenced on one lane of pair-end 100 bp flow cell on Solexa GA-IIx (Illumina, San Diego, CA, USA). After removing the contaminating reads from human hosts, over 7.5 million reads were produced for each of the healthy saliva microbiome, and over 28 million reads were generated for each of the caries-active microbiome. All of the Solexa reads were mapped against the 44 oral reference genomes in Human Microbiome Project [26] to assess the coverage and abundance of these sequenced isolates or their close neighbors in saliva microbiota.

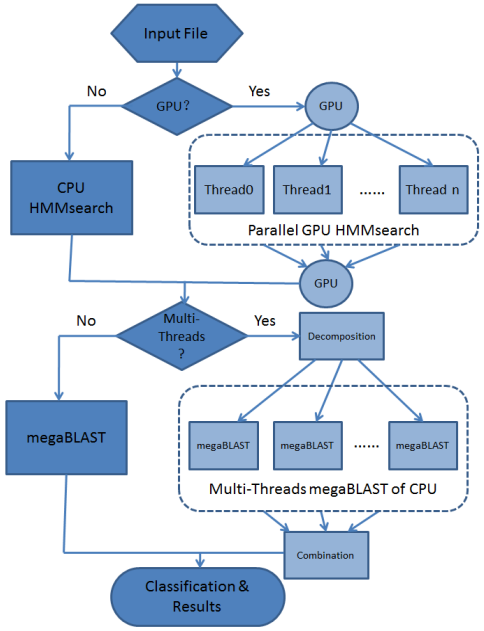

Figure 4.Metagenomic data analysis pipeline by high-performance parallel computing.

These 4 input files used for checking the performance of analyzing different number (from 7.5 million up to 34.4 million) and type of sequences of the optimized pipeline.

Table 1. Statistics of test datasets

	Size(MB)	Sequences	16S rRNA
Input 1	531.86	7,544,950	2406
Input 2	1576.96	17,591,235	2118
Input 3	2775.04	34,405,667	6468
Input 4	2928.64	28,854,628	17119

In the experiment we first measured the speed-up of GPU-based HMMER, then tested the speed-up of multi-thread megaBLAST, and finally evaluated the overall performance improvement of Parallel-META.

A. HMM Search

In the experiment of testing of HMMsearch, we ran the HMMsearch software with each sequence file as input on both CPU and GPU to compare the speed of two different methods. The tests were performed on one single node of the GPU computing cluster with nVIDIA Tesla C2070 GPU and Intel Xeon X5645 CPU. To reduce the effect of system-wise randomness and noises on the results, each input data were executed three times to get the average results, and the average results were compared. From the results (Figure 5), it was clear that a speed-up of at least 13 have been achieved on each input file.

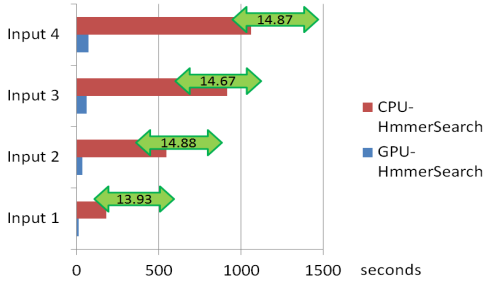

Figure 5.Comparison of running time of CPU & GPU

Then we have compared the speed-up of input file with increasing file sizes. From Figure 5, we could observe that for input file 1 the speed-up was a little smaller than other input files. This might be due to the fact that for the input file 1 the data size was small. In this situation, the data transfer between Global Memory and RAM became a more significant bottleneck than computing. With the increase of the input file size, the computation proportion also became larger, and the data transfer process has less effect on the whole process. The maximum speed-up rate was 14.88. To get the weighted average speed-up, we used the formula as below:

$$\text{Weighted Average} = \sum_{i=1}^{4} Ni * Si \;/\; \sum_{i=1}^{4} Ni \quad (1)$$

Here N_i and S_i were the sequence number and speed-up of input i, respectively. By this we can get the average speed-up of GPU-HMMSearch of 14.71.

B. MegaBLAST

In the experiment on megaBLAST step, for each input of the megaBLAST, we decomposed the data into sub input files. The number of the sub input files has been designed to be the

978-1-4577-1666-9/11/$26.00 ©2011 IEEE

number of CPU threads. The CPU of the computing platform is Dual Intel Xeon X5645 with 12 cores and 24 threads in total; therefore each input data file was divided into 24 sub files, and then each sub problem could be solved on one single thread. We also executed each test dataset three times and provided the average results for comparison. From the results (Figure 6), it was clear that a speed-up of 18 and above have been achieved on each input file. We also observed that with the increase size of the input data, the speed-up rate also increased, though such increase was not significant.

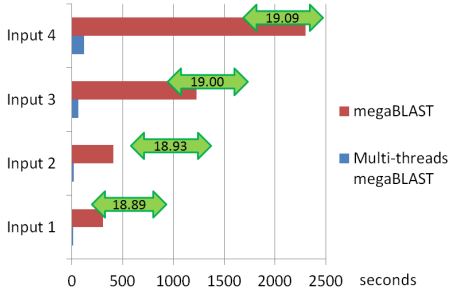

Figure 6. Comparison of Normal and Multi-threads megaBLAST on input files

In theory, to decompose the input data into 24 parts and parallelly solving them will reduce the runtime to 1/24. However, for the implementation of the multi-thread computing, the time cost of problem decomposition and results combination should also be taken into account. In addition, as the CPU system only has 12 physical CPU cores, if the computing throughput was larger than the CPU computing capability, CPU would use the transition algorithm to automatically manage these threads and some threads maybe executed serially when the CPU was very busy. The maximum speed-up was 19.09, and to get the weighted average speed-up, we used a formula which was similar to the one in HMMSearch part:

$$\text{Weighted Average} = \sum_{i=1}^{4} Ri * Si \;/\; \sum_{i=1}^{4} Ri \qquad (2)$$

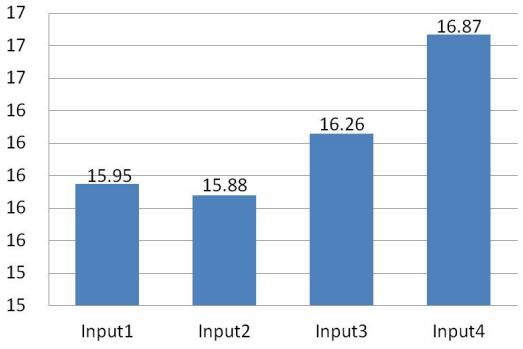

Figure 7. Total speed-up of Parallel-META compared to single CPU for all test datasets.

Here R_i and S_i were the 16S rRNA number and speed-up of input i, respectively. Therefore, the average speed-up of the multi-thread megaBLAST was 19.00.

C. *Overall performance*

Combining these optimization steps, a total speed-up of up to 16.87 has been observed compared to traditional CPU-based methods. More importantly, on all of these datasets, the final results of Parallel-META were identical to the results of the original single CPU-based pipeline, and the taxonomical analysis results were also consistent with the metagenomic data analysis results solely based on 16S rRNA [25].

V. CONCLUSION AND FUTURE WORKS

Traditional metagenomic data analyses were conducted on single PC, based on which handling multiple large metagenome datasets is becoming more and more difficult. In this work, we have tried to utilize GPU computing and multi-core CPU computing to boost the speed of metagenome data analysis, and proposed a novel pipeline that enabled the parallel processing of large metagenome datasets.

The Parallel-META pipeline has been applied on several metagenomic data analysis projects for human-associated bacterial communities, such as oral disease-causing microbial community analysis [27]. Several folds of speed-up has been observed, while the sensitivity and discrepancy power were not compromised. These results have shown that the parallelization of current metagenomic data analysis pipeline is very promising. With current 10 to more than 15 times of speed-up, binning would not be a very time-consuming process any more. Therefore some deeper data mining of the metagenomic data, such as refined gene regulatory network analysis in microbial communities, would be feasible.

Current Parallel-META pipeline could be improved in different ways. Firstly, the megaBLAST search part could also be implemented on GPU architecture, so that the efficiency of this time-consuming part could be further improved. Secondly, as metagenomic datasets are of different types and sources, the parameters for analysis would be different for each metagenomic dataset. These parameters could be trained based on running parallel-META on a large amount of different metagenomic datasets, which in turn could improve the accuracy of parallel-META. Thirdly, the parallel-META framework could be extended to work with multiple search engines and databases so as to be applicable to different types of metagenomic datasets. Finally, as a general-purpose metagenomic data analysis pipeline, parallel-META could also incorporate component-based binning methods, which might also significantly improve the speed for clustering metagenomic short reads [28].

Compliment to the high-performance computational pipeline is the high-performance database management system. The high-performance database management system would not only store large amount of results by high-performance computational pipeline, but also facilitate deeper data mining of metagenome data. Such high-performance database management system would also be incorporated into the next-generation high-performance computational platform for metagenomic data analysis.

978-1-4577-1666-9/11/$26.00 ©2011 IEEE

ACKNOWLEDGMENTS

We thank Huimin Li from USTC and Xingzhi Chang and Yinhe Qiao of QIBEBT, CAS for their support in arrangement of computing facilities. We also thank nVIDIA to provide us with the nVIDIA Tesla C2070 GPU card and their helpful discussion of the problem. This research is supported in part by Chinese Academy of Sciences' e-Science grant INFO-115-D01-Z006 and Ministry of Science and Technology's high-tech (863) grant 2009AA02Z310.

REFERENCES

[1] G. N. Proctor, "Mathematics of microbial plasmid instability and subsequent differential growth of plasmid-free and plasmid-containing cells, relevant to the analysis of experimental colony number data," Plasmid, vol. 32, pp. 101-30, Sep 1994.

[2] A. Jurkowski, et al., "Metagenomics: a call for bringing a new science into the classroom (while it's still new)," CBE Life Sci Educ, vol. 6, pp. 260-5, Winter 2007.

[3] J. A. Eisen, "Environmental shotgun sequencing: its potential and challenges for studying the hidden world of microbes," PLoS Biol, vol. 5, p. e82, Mar 2007.

[4] J. C. Venter, et al., "Environmental genome shotgun sequencing of the Sargasso Sea," Science, vol. 304, pp. 66-74, Apr 2 2004.

[5] G. W. Tyson, et al., "Community structure and metabolism through reconstruction of microbial genomes from the environment," Nature, vol. 428, pp. 37-43, Mar 4 2004.

[6] M. Arumugam, et al., "Enterotypes of the human gut microbiome," Nature, vol. 473, pp. 174-80, May 12 2011.

[7] E. R. Mardis, "Anticipating the 1,000 dollar genome," Genome Biol, vol. 7, p. 112, 2006.

[8] J. Xu, "Microbial ecology in the age of genomics and metagenomics: concepts, tools, and recent advances," Mol Ecol, vol. 15, pp. 1713-31, Jun 2006.

[9] T. Z. DeSantis, et al., "Greengenes, a chimera-checked 16S rRNA gene database and workbench compatible with ARB," Appl Environ Microbiol, vol. 72, pp. 5069-72, Jul 2006.

[10] E. Pruesse, et al., "SILVA: a comprehensive online resource for quality checked and aligned ribosomal RNA sequence data compatible with ARB," Nucleic Acids Res, vol. 35, pp. 7188-96, 2007.

[11] J. R. Cole, et al., "The Ribosomal Database Project: improved alignments and new tools for rRNA analysis," Nucleic Acids Res, vol. 37, pp. D141-5, Jan 2009.

[12] S. F. Altschul, et al., "Basic local alignment search tool," J Mol Biol, vol. 215, pp. 403-10, Oct 5 1990.

[13] D. H. Huson, et al., "MEGAN analysis of metagenomic data," Genome Res, vol. 17, pp. 377-86, Mar 2007.

[14] L. Krause, et al., "Phylogenetic classification of short environmental DNA fragments," Nucleic Acids Res, vol. 36, pp. 2230-9, Apr 2008.

[15] M. Monzoorul Haque, et al., "SOrt-ITEMS: Sequence orthology based approach for improved taxonomic estimation of metagenomic sequences," Bioinformatics, vol. 25, pp. 1722-30, Jul 15 2009.

[16] H. Teeling, et al., "TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences," BMC Bioinformatics, vol. 5, p. 163, Oct 26 2004.

[17] A. C. McHardy, et al., "Accurate phylogenetic classification of variable-length DNA fragments," Nat Methods, vol. 4, pp. 63-72, Jan 2007.

[18] N. Shah, et al., "COMPARING BACTERIAL COMMUNITIES INFERRED FROM 16S rRNA GENE SEQUENCING AND SHOTGUN METAGENOMICS," Pac Symp Biocomput, pp. 165-76, 2011.

[19] J. G. Caporaso, et al., "QIIME allows analysis of high-throughput community sequencing data," Nat Methods, vol. 7, pp. 335-6, May 2010.

[20] E. M. Glass, et al., "Using the metagenomics RAST server (MG-RAST) for analyzing shotgun metagenomes," Cold Spring Harb Protoc, vol. 2010, p. pdb prot5368, Jan 2010.

[21] R. Seshadri, et al., "CAMERA: a community resource for metagenomics," PLoS Biol, vol. 5, p. e75, Mar 2007.

[22] Z. Zhang and W. I. Wood, "A profile hidden Markov model for signal peptides generated by HMMER," Bioinformatics, vol. 19, pp. 307-8, Jan 22 2003.

[23] A. Morgulis, et al., "Database indexing for production MegaBLAST searches," Bioinformatics, vol. 24, pp. 1757-64, Aug 15 2008.

[24] J. P. Walters, et al., "Evaluating the use of GPUs in Liver Image Segmentation and HMMER Database Searches," 2009 Ieee International Symposium on Parallel & Distributed Processing, Vols 1-5, pp. 1010-1021, 2009.

[25] F. Yang, et al., "Saliva microbiomes distinguish caries-active from healthy human populations," ISME J, Jun 30 2011.

[26] K. Mavromatis, et al., "Gene context analysis in the Integrated Microbial Genomes (IMG) data management system," PLoS One, vol. 4, p. e7979, 2009.

[27] F. Yang, et al., "Saliva microbiomes distinguish caries-active from healthy human-populations," ISME Journal, vol. Accepted, 2011.

[28] C. Wei, "MetaBinG: Using GPUs to accelerate metagenomic sequence classification," Personal communications, 2011.

978-1-4577-1666-9/11/$26.00 ©2011 IEEE