Pyroprinting Sensitivity Analysis on the GPU

Douglas Brandt, Aldrin Montana, Bob Somers
Alex Dekhtyar, Chris Lupo
Computer Science Department
California Polytechnic State University
San Luis Obispo, United States
{djbrandt, amontana, rsomers, dekhtyar, clupo}@calpoly.edu

Michael Black, Anya Goodman, Chris Kitts
Biology Department
California Polytechnic State University
San Luis Obispo, United States
{mblack, agoodman, ckitts}@calpoly.edu

***Keywords*-CUDA, GPGPU, pyrosequence, pyrogram**

I. Introduction

Microbial Source Tracking (MST) is a field in which microbial strains are identified and associated with a specific host source [1], [2], [3] (e.g., human, canine, avian, etc). Identifying the hosts of microbial strains lies at the heart of many studies of bacterial contamination in the environment. Being able to determine which host species is responsible, e.g., for fecal contamination of a creek, allows the parties involved to develop specific measures for addressing the contamination.

To address problems of strain differentiation that current MST methods [1], [2], [4] have, our research group has developed a novel pyrosequencing-based MST method [5] called *pyroprinting*. Pyrosequencing is an inexpensive DNA sequencing technique in which a complementary strand is synthesized for a given single-strand DNA fragment. *Pyroprinting* is the simultaneous pyrosequencing of multiple, possibly different, DNA fragments from the same organism. The resulting pattern of emitted light intensities, called a *pyroprint*, is used as a digital fingerprint of the organism. Pyroprints are generated from known highly variable regions, called intergenic transcribed spacers (ITS), located in each copy of the ribosomal RNA operons in the bacterial genome [6], [7], [8]. Initially designed for *E. coli*, these ITS regions have *seven copies* (seven *loci*) in the bacterial DNA. These non-coding regions accumulate more mutations due to lack of selective pressures, which leads to more variability between bacterial strains. In the known, sequenced *E. coli* DNA, these regions are known to vary both from locus to locus in a single genome, and between the genomes of different strains.

In this paper, we present an *in-silico* study to investigate the sensitivity of the pyroprinting method. Given a collection of possible DNA sequences that can be found in the sequenced ITS regions, we construct a collection of *all possible theoretical combinations*. Each such combination represents a theoretically possible strain of *E. coli*. We construct a pyroprint model of each strain, and then build a matrix of pairwise similarities between the pyroprints.

In earlier work [5] we described an *in vitro* study involving collection of fecal samples, isolation of *E. coli* from the samples, and subsequent pyroprinting of the obtained isolates. The pyroprints were then compared to each other, and determination of whether multiple isolates belonged to the same strain was made based on how similar the pyroprints were.

In this work we investigated the distribution of Pearson correlations between pairs of theoretical strains. Theoretical strains yielding a correlation value above 99% are considered *hard to differentiate*. Over two million theoretical *E. coli* strains were generated using sequence data from the National Center for Biotechnology Research (NCBI). We used CUDA to speed up the computation of over two trillion Pearson correlations based on the work by Kijsipongse [9] and Chang [10].

II. In Silico Study

The goal of our *in silico* study is to simulate a wide range of possible isolates using a collection of known alleles, construct their theoretical pyroprints and generate Pearson correlation coefficients for each pair of constructed pyroprints. This section provides a brief outline of the study.

In the context of our simulation study, *an allele* is a single unique DNA sequence that can be found in the DNA region being pyrosequenced. In *E. coli*, the ITS regions 23S-5S and 16S-23S are the pyroprinting targets — each have seven loci in the genome. Additionally, a *theoretical isolate*[1] is any collection of seven alleles from the 23S-5S ITS region. The same allele may be present multiple times in the isolate. The combination describes the DNA material subject to *in silico* pyroprinting. Essentially, a *combination* represents a group of *E. coli* strains that cannot be distinguished via *in silico* pyroprinting.

a) Input: The input to the study is a collection of 24 alleles for the 23S-5S ITS region to be used for construction of different combinations. These alleles have been "harvested" from the NCBI genome database which contains complete genomes for 36 strains of *E. coli*. An additional input to the study is a specific dispensation sequence used to construct all pyrosequences and pyroprints in the study.

Table I shows the total number of theoretical isolates generated for several sizes of the allele collection used in our study. Given m different alleles, the total number N of possible theoretical isolates is the number of unordered combinations of seven alleles selected with replacement:

[1]Also referred to as an *isolate* or *combination*

978-1-4673-2747-3/12/$31.00 ©2012 IEEE

# of Alleles	# of Isolates	# of comparisons
10	11,440	65,431,080
11	19,448	189,102,628
12	31,824	506,367,576
13	50,388	1,269,450,078
14	77,520	3,004,636,440
24	2,035,800	2,072,239,802,100

TABLE I
DATASET SIZES GIVEN NUMBER OF ALLELES

Implementation	Time in kernel calls (s)	Time in host (s)
Initial	4.0263	0.1111
Improved(Global)	16.5592	0.0786
Improved(Constant)	8.0397	0.0746

TABLE II
AMOUNT OF TIME SPENT IN DEVICE VS HOST FOR RUNS WITH 10 ALLELES

$$N = \frac{(m+6)!}{7!(m-1)!}$$

b) Procedure: The study proceeds in four steps:

1) **Isolate generation:** Generate every possible combination of seven alleles (unordered, with replacement). Each combination represents a single unique theoretical isolate in our study.
2) **Pyroprint generation:** For each isolate, construct its pyroprint:
   a) Pyrosequence each individual allele of the isolate.
   b) Combine individual pyrosequences into a pyroprint by adding up the pyrosequence vectors.
3) **Pearson correlation coefficient computation:** for each pair of pyroprints, compute the Pearson correlation between them.
4) **Histogram generation:** separate all computed Pearson correlations coefficients into a number of "buckets" based on their values.

c) Output: The output of the study is the histogram of Pearson correlation coefficient values. The histogram is constructed and reported with respect to a specific collection of *buckets*, which can be considered a *third input parameter* to the process. For each bucket, represented by the *low* and *high* value, the raw count of Pearson correlations that fall between low and $high$ is to be computed and reported.

III. EVALUATION

For our evaluation, we used a machine running Arch Linux 3.3.7-1 that has an Intel Q8200 (Quad Core 2.33GHz) processor, 4GB of DDR3 RAM, and a GTX 260 (192 cores, 896 MB of RAM, and compute capability 1.3). Our simulation was developed using pyCUDA and was tested using CUDA toolkit 4.2 (python2-pycuda and cuda-toolkit packages). At times we had a Tesla2070 available for use, however we were only able to get a limited number of runs and are not fully confident in the results observed and so did not include the results in this paper.

Implementation	occupancy	registers/thread	# kernel calls
Initial	1	12	276
Improved(Global)	0.5	23	276
Improved(Constant)	0.5	23	276

TABLE III
PROFILING DETAILS FOR RUNS WITH 10 ALLELES

Correlation Value Range	% of comparisons
0 - 99%	99.69%
99-99.5%	0.275%
99.5-99.7%	0.0263%
99.7-100%	0.00873%

TABLE IV
PERCENTAGE OF COMPARISON VALUES FALLING WITHIN A SPECIFIED RANGE OF POSSIBLE PEARSON CORRELATION COEFFICIENT VALUES

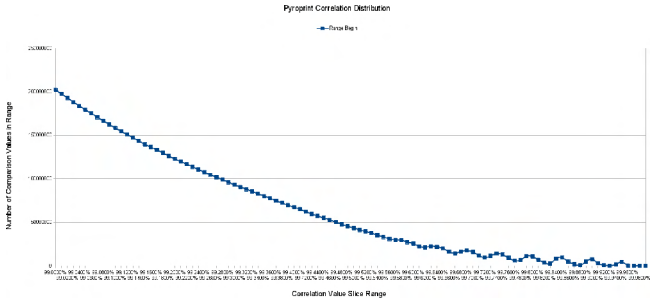

Fig. 1. Distribution of Pyroprint Correlations for 24 alleles

For all runs, our input dataset was a set of reference *E. coli* DNA sequences from the 16s-23s ITS region. We were given the primer `TTGGATCAC` for pyrosequencing as well as the dispensation sequence, `AACACGCGA23(GATC)GAA`. To simulate an *E. coli* isolate we selected, and combined, seven alleles from the set of alleles extracted during pyrosequencing. A pyroprint for each isolate was then constructed as a sum of the seven alleles chosen for the isolate. Given these parameters, our simulation extracted a set of 24 alleles from the reference *E. coli* DNA sequences. As depicted in Table I, 24 alleles yields over 2 million unique isolates, of which there are over 2 trillion pairwise correlations to be computed. For reference, Table I shows how many isolates and comparisons were made given a set number of alleles.

To assess the improvement of our implementation we compare our initial implementation to two versions of a subsequent implementation which use global and constant memory. We refer to our initial implementation as version 0, and constant memory and global memory versions of our new implementation as version 1_c and 1_g, respectively. For each version we observed five runs of five different numbers of alleles (10 - 14). Each run was executed using 16×16 thread-block dimensions and 32×32 grid-block dimensions.

The runtime difference between version 0 and version 1_c is less than the difference between version 1_g and 1_c, as seen in Table V. While this makes it seem that our use of constant memory contributes more to our runtime improvement than

Implementation	10	11	12	13	14	24
Initial	19.1507	52.2426	138.8790	341.9866	813.8992	Not included
Improved(Global)	17.4501	44.6382	110.5517	277.4986	621.5918	122318.7881
Improved(Constant)	8.8498	23.5151	62.6434	156.7317	392.7217	104651.4704

TABLE V
RUN TIMES FOR 10, 11, 12, 13, 14, AND 24 ALLELES IN SECONDS

our modification of the isolate generation method, this is not exactly the case.

The performance difference between version 0 and version 1_c is much smaller using a Tesla2070 than it is using a GTX260. The reason for this is that the GTX260 is much more memory constrained. It has only 896MB of global memory available where a Tesla2070 has up to 6GB of global memory available. Each version of our implementation must copy data into global memory on device, so the more memory available, the less times data will have to be copied across the PCI-e bus to device. However, version 1_c (and version 1_g) amortizes this cost by only requiring a single data transfer to device. This is particularly emphasized by the fact that the runtime of the pearson correlation kernel for version 1_c is ≈ 29.9ms whereas it is only ≈ 15.3ms for version 0. These results suggest that our performance increase is primarily in memory and bus usage.

The distribution of Pearson correlation coefficient values, depicted in Figure 1, shows that the amount of comparison values above 99% correlation value significantly drops off for higher correlation values. This distribution can also be seen in Table IV. For our simulation, over 99.6% of all correlation coefficients fall below 99%. In preliminary studies conducted on actual pyroprints, multiple pyroprints of the same isolate showed correlations above 99.7%. However table IV shows that only 0.00873% of correlations fall into that category. This suggests that the pyroprinting methodology is sufficiently sensitive to distinguish different bacterial strains.

IV. CONCLUSION

Despite the drawbacks of our new implementation, its scalability is far beyond our initial implementation. Our new implmementation requires minimal preparation for the GPU to begin computation: 24 alleles $\times$ 104 bytes (1 byte per pyroprint index), boundary contraints (number of unique isolates), and the range of isolates to compare. This means that the kernel will be able to compute all comparisons once preparation for the kernel has been made without the need for additional data transfers. Our initial implementation may have a fast kernel, but memory constraints prevent it from being scalable and portable.

V. FUTURE WORK

Work is being done to implement our simulation with multi-GPUs in a single machine as well as using open MPI to be distributed across many machines. Although this data shows the most significant advantage our new implementation has over our initial implementation, due to time and availability constraints, the appropriate modifications could not be made to our simulation to be included here.

ACKNOWLEDGMENT

This work was supported in part by an undergraduate education grant from the W.M. Keck foundation.

REFERENCES

[1] T. M. Scott, J. B. Rose, T. M. Jenkins, S. R. Farrah, and J. Lukasik, "Microbial source tracking: Current methodology and future directions," *Appl. Environ. Microbiol.*, vol. 68, pp. 5796 – 5803, dec 2002.

[2] J. M. Simpson, J. W. S. Domingo, and D. J. Reasoner, "Microbial source tracking: State of the science," *Environmental Science and Technology*, vol. 36, pp. 5279 – 5288, dec 2002.

[3] T. R. Desmarais, H. M. Solo-Gabriele, and C. J. Palmer, "Influence of soil on fecal indicator organisms in a tidally influenced subtropical environment," *Applied and Environmental Microbiology*, vol. 68, pp. 1165 – 1172, mar 2002.

[4] D. M. Gordon, "Strain typing and the ecological structure of *Escherichia coli*," *Journal of AOAC International*, vol. 93, pp. 974 – 984, may 2010.

[5] A. Montana, A. Dekhtyar, E. Neal, M. Black, and C. Kitts, "Chronology-sensitive hierarchical clustering of pyrosequenced dna samples of e. coli: A case study," in *2011 International Conference on Bioinformatics and Biomedicine*. IEEE, 2011, pp. 155–159. [Online]. Available: http://www.computer.org/portal/web/csdl/doi/10.1109/BIBM.2011.99

[6] S. L. Boyer, V. R. Flechtner, and J. R. Johansen, "Is the 16s-23s rrna internal transcribed spacer region a good tool for use in molecular systematics and population genetics? a case study in cyanobacteria," *Molecular Biology and Evolution*, vol. 18, no. 6, pp. 1057 – 1069, june 2001.

[7] A. Roth, M. Fischer, M. Hamid, S. Michalke, W. Ludwig, and H. Mauch, "Differentiation of phylogenetically related slowly growing mycobacteria based on 16s-23s rrna gene internal transcribed spacer sequences," *Journal of Clinical Microbiology*, vol. 36, no. 1, pp. 139 – 147, january 1998.

[8] S. Tyler, C. Strathee, K. Rozee, and W. Johnson, "Oligonucleotide primers designed to differentiate pathogenic pseudomonads on the basis of sequencing of genes coding for 16s-23s rrna internal transcribed spacers," *Clinical and Diagnostic Laboratory Immunology*, vol. 2, no. 4, pp. 448–453, july 1995.

[9] E. Kijsipongse, S. U-Ruekolan, C. Ngamphiw, and S. Tongsima, "Efficient large pearson correlation matrix computing using hybrid mpi/cuda," in *Computer Science and Software Engineering (JCSSE), 2011 Eighth International Joint Conference on*, may 2011, pp. 237 – 241.

[10] Dar-Jen Chang, A. H. Desoky, M. Ouyang, and E. C. Rouchka, "Compute pairwise manhattan distance and pearson correlation coefficient of data points with gpu," in *Proceedings of the 2009 10th ACIS International Conference on Software Engineering, Artificial Intelligences, Networking and Parallel/Distributed Computing*, ser. SNPD '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 501–506. [Online]. Available: http://dx.doi.org/10.1109/SNPD.2009.34