High-Performance Virus Detection System by using Deep Learning

Ying-Feng Hsu[1], Makiko Ito[2], Takumi Maruyama[3], Morito Matsuoka[1], Nicolas Jung[4], Yuki Matsumoto[4], Daisuke Motooka[4], Shota Nakamura[4]

[1] Cybermedia Center, Osaka University, Osaka, Japan
Email: {yf.hsu, matsuoka}@cmc.osaka-u.ac.jp
[2] Fujitsu Laboratories Ltd., Kanagawa, Japan
Email: maki-ito@jp.fujitsu.com
[3] Fujitsu Limited, Tokyo, Japan
Email: takumi_maruyama@jp.fujitsu.com
[4] Genome Information Research Center, Research Institute for Microbial Diseases, Osaka University, Osaka, Japan
Email: {nicolasj, matsumoto, daisukem, nshota}@gen-info.osaka-u.ac.jp

***Abstract*—Metagenomic shotgun sequencing enables us to explore diverse DNA sequences from viruses, bacteria, and eukaryotic microbes in complex samples. As the continuous advancement of sequencing technology generates a massive amount of sequencing data, its overall computational complexity has become a major challenge for traditional database sequence comparison methods. Studies have shown that deep learning-oriented methods have been widely adopted to solve many classification problems, including those in the bioinformatics field, and have demonstrated this method's accuracy and efficiency for analyzing large-scale datasets. The aim of this study attempts to investigate how deep learning (LSTM model) can be used to learn sequential genome patterns through virus detection from metagenomic data. This study provides three major contributions. First, we provide the background and steps for the task of DNA sequencing classification from data collection, preprocessing, and normalization. Second, we analyze the effect of sequence length on LSTM classification accuracy and split the raw sequencing data to proper subsequences to improve the outcome of virus detection. Third, to enhance both the classification accuracy and processing speed, we introduce the concept of discrimination function that enables prediction results for multiple subsequences results and accelerated these processes through GPU parallel computing. Two case studies of HCV and influenza detection were conducted to elaborate upon the accuracy and computational efficiency of our proposed approach. Our test result showed that the proposed LSTM model obtained similar pathogen detection accuracy to the conventional BLAST method with a speed that was about 36 times faster.**

Keywords—Deep Learning, LSTM, GPU Acceleration, Parallel Computing, Metagenomic Shotgun Sequencing, DNA Sequence Classification

I. INTRODUCTION

Deoxyribonucleic acid (DNA) is fundamental to all living species and is based on the order of four nucleotides: guanine (G), cytosine (C), adenine (A), and thymine (T). With the continual growth of low-cost and high-throughput DNA sequence technology, the scale and amount of next-generation sequencing (NGS) datasets are continually increasing in many genomic research areas. With NGS sufficient sequencing throughput, it is also possible to detect rare microbial species and those of low abundance within the microbiome. Metagenomic shotgun sequencing uses NGS technology to sequence DNA strands within a given complex sample randomly. Unlike capillary sequencing or PCR-based approaches, it sequences a large number of genes and shears them into smaller segments. As compared to other types of DNA sequencing, it has many short reads of 50 to 600 base pairs (sequence length) and is sufficient and sensitive enough for clinical pathogen detection. Studies have shown that this method enables the evaluation of the diversity of viruses, bacteria, and eukaryotic microbes, and can help to estimate their abundances in given complex samples. Due to the challenges of large scale data and computational complexity, the traditional sequence comparison method may require a much longer computation time to obtain sequence analyzing results. Commonly used methods, such as the Smith-Waterman algorithm [1] and Basic Local Alignment Search Tool (BLAST) [2] use sequence alignment to measure the similarity between input sequences and reference database sequences. This type of distance comparison-based algorithm is highly time-consuming. Several fast mapping algorithms, such as BWA and Bowtie [1] were developed to handle NGS data. However, those algorithms were able to find almost complete matching only and may not be suitable for the massive amount of metagenomic shotgun sequencing studies.

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

Therefore, the aim of this study attempts to explore how a deep learning (LSTM model) can be used to learn sequential genome patterns through pathogen detection from metagenomic data. We propose a bagging-based ensemble LSTM for the task of pathogen detection through shotgun metagenomics sequence classification. This research makes three major contributions in this area. First, we provide the background and steps for the task of DNA sequencing classification from data collection, preprocessing, and normalization for both using public NCBI reference databases and generating DNA sequencing data from real patients. Second, as a DNA sequence is a type of sequential dataset without specific features, we analyze the effect of sequence length on LSTM classification accuracy and split the raw sequencing data into proper subsequences to improve the outcome of virus detection. Third, to enhance both the classification accuracy and processing speed, we introduce the concept of discrimination function that enables prediction results for multiple subsequences and accelerate the process through GPU parallel computing. Two case studies of HCV and influenza virus detection were conducted to elaborate upon the accuracy and computational efficiency of our proposed approach. Towards the end of the paper, we conducted two viral sequences detection case studies, i.e., HCV and Influenza viral sequences. We elaborate upon the efficiency and computational complexity of our proposed approach in an HPC server, which has eight NVidia Tesla P100 GPUs. Our experimental result shows that we obtained similar accuracy to the conventional BLAST method, but with a computational speed that is about 36 times faster.

The rest of this paper is organized as follows. In Section II, we survey related approaches in the areas of machine learning or deep learning-based sequence comparison and classification. We provide the background of conducting DNA sequencing classification in Section III and the steps of data collection in Section IV, while we elaborate on our methodology in Section V. To evaluate the advantages and performance of the proposed approach; we provide a comprehensive evaluation by using real clinical samples in Section VI. A discussion is proved in Section VII, and we conclude this paper and discuss future work in Section VIII.

II. Related works

In recent years, huge DNA sequence data have been released to the public. To recognize and identify DNA sequences, various sequence detection or sequence classification techniques have gained a great deal of attention in bioinformatics. Sequence alignment is the foundational method for DNA sequence comparison, and to measure the similarity between two DNA sequences, global alignment and local alignment are widely used. The Needleman-Wunsch algorithm [2] is one of the first methods, which uses dynamic programming to compute an optimum global alignment. In contrast to global alignment algorithms, local alignment algorithms, such as the Smith-Waterman algorithm [3], BLAST [4], and BWA [5] provide fast alternatives to assess the similarity between two sequences by considering the most similar regions, but not enforcing rigid alignment along the full length.

Other than the above distance-based similarity comparison method, Recently, many studies have investigated sequence detection or classification using a machine-learning approach without considering the overall sequence alignment. In [6], the author conducted a study of sequence classification, based on support vector machines (SVM) while [7] extending the standard hidden Markov model (HMM) to classify protein sequences. Since classical machine learning methods cannot operate on the sequence directly, it requires preprocessing work to extract important features from data such as [8] and [9] that use the approaches of k-mer or motif occurrences. Another approach [10] that apply convolutional neural network (CNN) with one-hot vectors to represent sequences for the task of DNA sequence classification. Although the sequence alignment may not need by the above machine learning methods, sequence classification models may still require input sequences with a fixed length. Several machine learning-based studies [11] [12] have proposed alignment and length-free methods for the task of sequence comparison. They applied a fast Fourier transform (FFT) to convert the original sequences to power spectra in the format of a real matrix, based on the frequency of each nucleotide. Since the preceding vector has a higher weight than later vectors, the first few vectors of this matrix are enough to represent a given DNA sequence and used for the sequence comparison.

III. Background

DNA molecules consist of four nucleotides and are defined as a text string of guanine (G), cytosine (C), adenine (A), and thymine (T). The FASTQ format is a standard text-based format that stores DNA text strings and their sequencing quality scores of high-throughput sequencing instruments. A single ASCII character represents both the sequence and quality score. To conduct research in the field of DNA sequencing analysis, we discuss two background studies of sequence analysis with FASTQ data.

A. Format of DNA sequencing data

Although many studies can conduct through public reference databases such as the GenBank database of National Center for Biotechnology Information (NCBI) for the task of species identification in metagenomic studies, a DNA sequencer is required to obtain the sequencing data directly from the patient's biosample such as blood, stool, or tissue. Hence, the data quality is an important preliminary to ensure the confidence of test outcomes. A common type of DNA sequencing data is usually stored in a text format call FASTQ data. Consider Fig. 1 as an example of FASTQ data that is usually structured as four lines for each sequence.

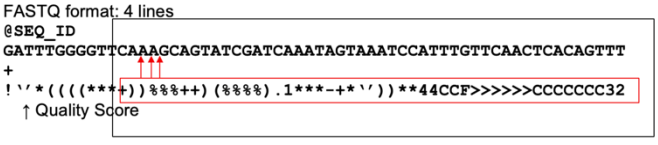


Fig. 1. Sample of FASTQ file (reproduced from Wikipedia)

The quality score is calculated by the equation below, which is used to find the optimal sequence length for the subsequent analysis.

$$\text{Quality Score} = -10\log_{10}(p_{error}) + 33$$

- p_{error}: base-calling error probabilities
- Ascii-code(33) = !

Fig. 2 illustrates the phenomenon of base calling quality (quality score) decreasing from collected real patient samples by using the Illumina HiSeq 2500 DNA instrument. From this quality boxplot, the reading quality started to significantly decrease around the base position of 50 and 110 for influenza and HCV, respectively. Based on this observation and to ensure the accuracy of sequence classification, it is necessary to consider certain preprocessing steps properly, for example, to ignore low-quality base reads or to select several proper fixed-length subsequences. Thus, we were able to ensure that the input subsequences always included a portion of high-quality reads as the input of any sequence analyzing model.

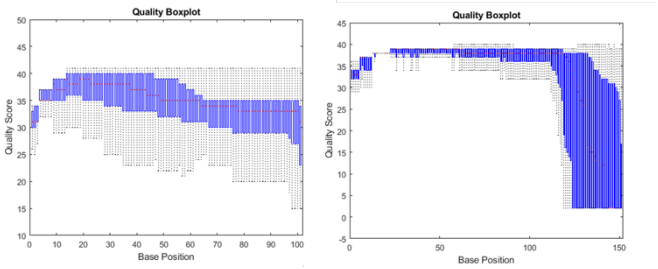


Fig. 2. Base-calling quality (left, Influenza samples; right, HCV samples)

B. *Sequence comparison scenarios:*

Normally, there are three scenarios to consider for processing the DNA sequence comparison:

(1) Reverse complementary sequences.
DNA consists of two polynucleotide chains, named "double-strand," where each A (adenine) is paired with a T (thymine) and vice versa, and similarly, each C (cytosine) is paired with a G (guanine) and vice versa. The reverse complementary sequences are formed by reversing the letters (by interchanging A and T and interchanging C and G) and should also be included in the sequence comparison operation.

(2) The "indel" comparison.
The genetic sequence comparison includes an assessment for the sequence gap, which is often referred to as "Indel": insertion and deletion sequence comparsion.

(3) Sequence alignment.
First is the operation of sequence alignment, where the sequence similarity is calculated from either a global or local optimal sequence alignment scenario. Thus, the match of two sequences is not completed by an exact match, but rather by considering the similarity.

The above three sequence comparison scenarios are often provided as functions in conventional sequencing comparison applications, such as BLAST; therefore, our proposed deep learning solution should consider each of these sequence comparison capabilities.

IV. DATA ACQUISITION AND PRETREATMENT

A large amount of high-quality training data are essential to every machine learning application, especially for approaches that are closely related to deep learning. Our data collection and preprocessing method include both a public database as ground reference data and a real patient's sequencing data generated from our laboratory. In this section, we discuss our procedures of data collection, augmentation, and normalization.

A. *Data collection (Training dataset):*

DNA sequencing data is generated by NGS—an instrument that automates the DNA sequencing processer and generates text strings of A, C, T, and G from the given biosamples. This procedure is commonly referred to as a DNA sequence read. Due to the sequencer's reading capability, there is no DNA sequence that can guarantee to produce 100% accurate DNA sequencer data. In fact, an accuracy of over 99.9% is an acceptable threshold in the industry.

B. *Data argumentation by shifting data*

By directly using the original format of the NCBI sequence, deep learning may not be able to learn enough information from the above three scenarios. In our proposed approach, building a customized shifting reference database provides the functionality of sequence comparison based on the three concepts above. Inspired by split-read aligners from [5], we generated reverse complementary sequences, which are segmented sequences with a proper sequence length based on their quality. To support the sequence comparison in the reverse-complementary sequence scenario, we generated reverse complementary sequences, which double the number of original sequences. DNA sequences are highly dimensional datasets without explicit features; To avoid getting the extremely large data training, we randomly analyze and split the whole sequence proper size of subsequences to extract useful and important features. Our training data includes all shifting sequences for viral sequences and random shifting sampled non-viral sequences (human DNA sequences). This approach enables our model to learn patterns from a sequence alignment scenario.

Fig. 3 illustrates the concept of shifting sequence generation from the NCBI HCV reference database. For each sequence in the database, we generate more subsequences by using the shifting windows = 1. In this manner, those shifting sequences facilitate the sequence alignment operation and allow deep learning extracts the characteristics of those shifting sequences to perform the sequence classification. Please note that the *subsequence_length* is determined by the target testing data set. In our study, we explored subsequence lengths from 20bp to 70bp for our pathogen detection use cases of HCV or influenza to fulfill our system requirement.

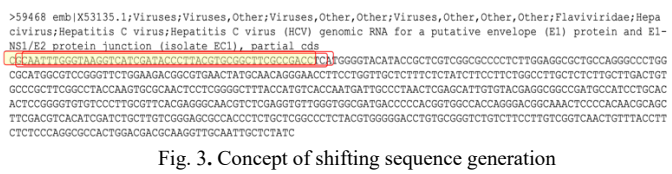


Fig. 3. Concept of shifting sequence generation

C. Data normalization

Data normalization is commonly applied as part of the data preparation process for machine learning. The goal of normalization is to change the values of numeric columns in the dataset to a common scale without distorting differences in the ranges of values. Having an appropriate data normalization can help to accelerate the convergence of the training process. We adopt the common approach of the DNA nucleotide by using one-hot encoding as A = [1 0 0 0], G=[0 1 0 0], C=[0 0 1 0], and T=[0 0 0 1].

D. Data collection (Testing dataset)

1) Statistic of the testing data

We obtained 32 clinical samples for our evaluation dataset, which consist of 9 metagenomic datasets from HCV-positive (hepatitis C viruses) patients and 13 influenza-infected patients. For negative control samples, we used blood samples from 10 healthy donors. Those 32 clinical samples were processed through an Illumina HiSeq 2500 instrument to generate the metagenomics shotgun sequencing data with about 2 to 7 million sequences per sample. Originally, these samples account for about 103 million sequences. In this study, we exclude sequences that are not either viral or human sequences (such as bacterial sequences). Table 1 summarizes the statistics of those samples in this study. In total, there are about 78 million shotgun sequences on average with a different length of reads.

TABLE 1. CHARACTERISTIC OF 32 SHOTGUN SEQUENCE CLINICAL SAMPLES.

	9 HCV blood samples	13 influenza nasal swab samples	10 healthy blood samples
Number of total sequences	9,574,381	40,195,765	29,294,195
Average number of sequences	1,063,820	3,091,982	2,929,420
Sequence length	150bp	100bp	199bp

2) Quality of testing data

Fig. 4 shows the average quality of reads distribution from our 32 testing clinical samples, which have approximately 78 million shotgun sequences. The function considers only sequences with an average quality score equal to or greater than the threshold. In our case, about 86.34% of these samples have an average Phread quality score greater than 30. This high score value indicates that the chances of an incorrect base being called within 1/1000 (accuracy of 99.9%) and ensures the quality and confidence of our testing of these shotgun samples. Recall from the above data argumentation that for each input sequence, we generate multiple subsequences by using the shifting-window method. The final classification outcome is based on the ensemble result from the classifications of these subsequences. Using this high sequence quality ensures the confidence of our data augmentation procedure.

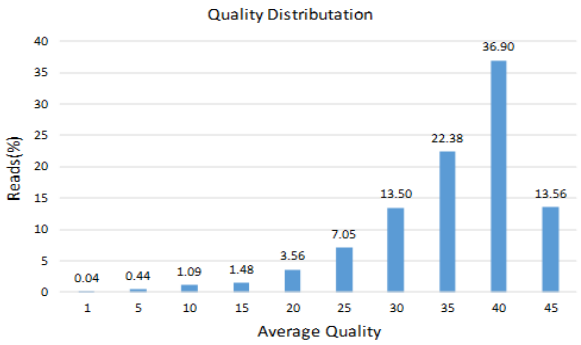


Fig. 4. Sequence quality distribution

V. METHODOLOGY

A DNA sequence encapsulates genetic information from sequential patterns. In this section, we elaborate upon our proposed pathogen detection system. More specifically, we use a deep learning method (the LSTM model) to discover the sequential patterns of given ground reference databases and have adopted it as a disease discriminator to evaluate the given shotgun metagenomics samples. A conventional method like BLAST is highly time-intensive, due to the process of similarity comparison with its reference database. However, it still shows a high degree of accuracy and reliability for the task of DNA sequence analysis and comparison.

Fig. 5 illustrates the proposed pathogenic virus detection system. After sample preparation and generate DNA sequencing data from DNA sequences, our proposed model was designed to analyze all sequencing data through sequence classification. Depending on the characteristics of input sequences, we provide two operation modes: the direct processing mode and the sequence filtering mode, to ensure results with high accuracy and processing speed. The direct processing mode outputs high homology sequences to speed up pathogen detection, and sequence filtering mode outputs possible sequences (true positive of viral sequences) that are further determined by the BLAST method. Although it may take time to run BLAST, it's important not to miss sequences with low homology. The following subsections elaborate on the components and important concepts of this proposed system.

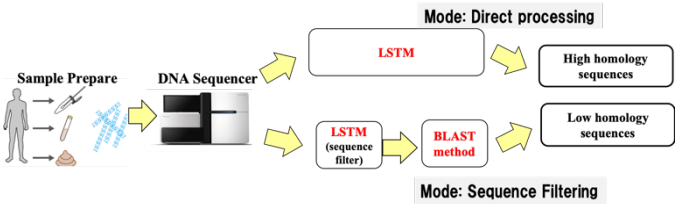


Fig. 5. Data flow in the proposed pathogen detection system

A. Structure of the proposed LSTM model

The LSTM model is the core of the proposed system, and we describe the detail of model construction in the following. To build our model, we obtained the reference sequences of both influenza and hepatitis C viruses from the NCBI viral genome resource [9], and used hg19 for the human reference genome data, as described in the previous section.

Fig. 6 illustrates the architecture of our proposed LSTM network for the DNA sequence classification task. An LSTM network is a type of recurrent neural network (RNN) that is designed to learn the long-term dependencies between time steps of sequential data, which meets the concept of analyzing DNA genomic sequences. This network starts with a sequence input layer followed by an optional dropout layer, which was originally designed to solve the overfitting issue. A sequence input layer inputs DNA sequences data into the network and the LSTM layer learns long short-term dependency patterns from nucleotide sequence representations. A fully connected layer extracts features from its preceding LSTM layer and multiplies it by a weight matrix, and then adds a bias vector to learn non-linear combinations of these features. To classify sequences into virus or human, a sigmoid layer is added to generate the binary classification result at the output layer. It is expected that after the model has learned the genomic pattern of the ground truth database, it should be able to determine a similar pattern for any future input sequence, either as a viral sequence or a human sequence. In this work, we build two binary LSTM prediction models for labeling the input sequences as [HCV, human] and [influenza , human].

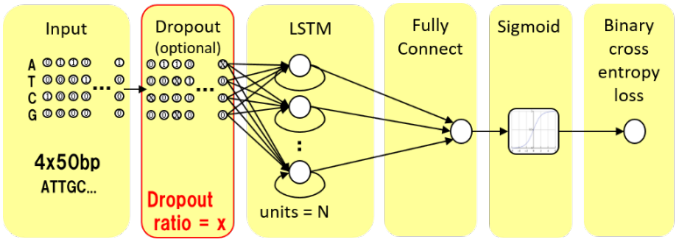


Fig. 6. An illustration of the proposed LSTM model.

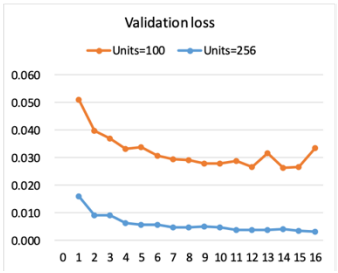


Fig. 7. LSTM training loss decreasing

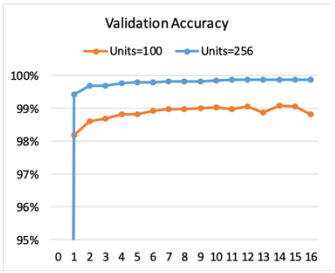


Fig. 8. LSTM training accuracy increasing

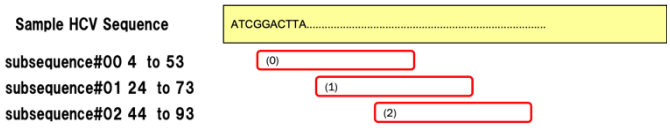
We use the hg19 sequence to represent the human genome for the model training. Furthermore, we explore optimal hyperparameters to improve the accuracy of classification. The parameters of the final networks include Adam as the optimizer, a batch size of 4096, and 256 as *num_units* (size of the LSTM's hidden state). Fig.s 7 and 8 show that we have significantly improved our training loss and accuracy with our optimal parameters and with an increasing number of neurons.

B. *Optimal input sequence length*

DNA sequencers commonly generate varying lengths of sequencing data, which are defined as the base pair (bp). Although the LSTM model is capable of processing inputs as different sequence data, we use a fixed sequence length because of three underlying reasons. First, shotgun sequencing is relatively shorter when compared to other types of genome sequencing datasets, and it is relatively easy to adjust the sequence length instead of directly inputting various amounts of base pairs into the model. Second, when passing data of unfixed lengths through the LSTM network, the program pads shorter sequences with zeros and truncates longer sequences or splits sequences in each batch to provide sequences of the specified length. This procedure increases additional run-time computational overhead. In practice, to reduce the amount of padding or discarded data when padding or truncating sequences, we recommend sorting the input data by their sequence lengths. In fact, the underlying concepts of padding, truncating, and sorting procedures are acutely designed to facilitate the concept of fixed length. Third, sequence base calling is the process that converts the raw image data to nucleotide sequences. The quality of nucleotide reads decreases along with the length of the sequence reads, which are caused by both the sample quality and the capability of the DNA sequencer. It is practical to use only a high-quality sequence for sequence analysis.

C. *Discrimination function*

To cope with the complicated scenario of sequence classification from both sequence alignment and polymorphism, our system first defines the optimal input sequence length as previously mentioned and performs classification tasks from three subsequences. We create a metaclassifier, the discrimination function, to combine those subsequence classification results to form the final classification result. Different subsequence predictions capture information from different angles with different advantages and disadvantages. By adequately leveraging the uniqueness of each prediction, in most cases, it is possible to obtain a higher prediction accuracy than with a single classification [12]. In general, the preceding nucleotides in a whole sequence has a higher base calling score and should be accountable for higher importance in the discrimination function. Fig. 9 illustrates a scenario where we divide an HCV sequence of 151bp to 3 subsequences with a shifting sequence window of 20. A sequence is classified as an HCV-related sequence only when the metaclassifier (the discrimination function) is true.



[Discrimination function]
* Direct processing: (#00>0.99) and (#01>0.99) and (#02>0.99)
* Sequence filtering: (#00>0.99) or (#01>0.99) or (#02>0.99)

Fig. 9. Ensemble subsequences classification results with the discrimination function.

One advantage of this approach over the traditional sequence alignment approach (such as BLAST) is that the machine learning approach is able to find important parts from certain subsequences without comparing the entire input sequence to sequences from the reference database.

D. Dropout layer

Dropout is the most common approach to avoid overfitting and increase the overall prediction accuracy. The term "dropout" refers to dropping out units (both hidden and visible) in a neural network. Intuitively, the procedure involves setting some neurons in the network to be zero (dropped out) during training in each forward pass. The dropout rate corresponds to the probability that a neuron is dropped out. Normally, 0.5 is the default value, and in this study, we explore various possible values from 10% to 50%. The effect of using the dropout layer is provided in section 6B.

E. GPU acceleration

Depending on the reading capability of NGS technology, one data may contain from one million to hundreds of millions of metagenomics shotgun sequences. In a conventional computation use case, sequence comparison normally runs under multiple cores on a single CPU. Analyzing this type of big-data task is highly challenging and time-consuming without efficient sequence comparison algorithms. A GPU has thousands of individual computational cores, which are designed to solve both computational and data-intensive problems by using parallel computing. Parallel computing enhances the computation speed through the concurrent execution of multiple processes. Unlike other deep learning models, LSTM is only allowed to use a single GPU for the model training phase. Depending on the data size and parameter setting of the LSTM model, our model training time varies from 12 to 47 hours. However, for the task of sequence classification, by using multi-GPU acceleration, our proposed method is able to split a hundred million shotgun DNA sequences into many smaller batches of DNA sequences and process them, concurrently, within minutes. The detailed result of this type of analysis is provided in the experimental section.

VI. EXPERIMENTAL RESULT

To evaluate the feasibility of the proposed LSTM model for the task of pathogen detection, we conducted experiments by comparing it with the standard sequence comparison approach, the BLAST, from the aspect of both accuracy and computing speed. BLAST is a database search approach with the capability to report the confidence of its searching result through the indictor expect value (E-value). The E-value indicates the number of expected hits of similar quality (score) that could be found just by chance between the length of the input query sequence and the size of the entire reference database. In other words, the lower the E-value (or closer to zero), the more significant the match, and the higher the evaluation quality.

In our experimental design, we use the DNN direct mode to compare with the BLAST's high-quality output sequences by using a threshold of E-value < $1e^{-30}$. For the sequencing filter mode, and to assess the filtering capability, we use all output sequences from BLAST with a threshold of E-value < $1e^{-10}$ to compare with the output sequences from our proposed LSTM model. Recall from section 4D that our testing data consists of 32 clinical samples: blood samples, i.e., blood samples from 9 HCV-infected patients and 10 healthy individuals, as well as nasal swab samples from 13 influenza-infected patients. The data size is approximately 78 million shotgun sequences with different sequence lengths. Those samples are metagenomic DNA sequencing data that contains all possible genomes from given samples, with over 90% of them being human DNA sequences. HCV and influenza viral sequences should exist only in infected patients (9 HCV and 13 influenza-infected samples). Thus, a successful detection method should properly identify those viral sequences. We implemented and evaluated the proposed LSTM model on an HPC server with dual Intel Xeon E5-2690 v4 (2.60GHz) processors, 768GB 2133 MHz DDR4 LRDIMM of main memory, and eight NVidia Tesla P100 GPUs (PCIE-16GB).

A. Test of direct processing mode

In this test, we compared our DNN's result with the BLAST threshold of an E-value less than $1e^{-30}$, which is considered to be a high-quality hit for homology matches.

Fig. 10 presents the classification result from the 13 influenza-infected clinical samples, which has about 4.8 million shotgun sequences per sample, while Fig. 11 shows the sequences selection result from the 5 HCV-positive samples, which has about 2.1 million shotgun sequences per sample. Since the actual amount of target sequences varies on a scale from one to thousands, to access the result with BLAST, we used a log scale to represent our result. From the evaluation of the influenza test in Fig. 10, it shows that our result is almost identical to BLAST in all samples. More precisely, our method found slightly more influenza-related sequences than BLAST for all samples. Those sequences do not necessarily mean a false positive result since BLAST does also generate erroneous outputs. We provide a detailed discussion of this finding in the false-positive evaluation section.

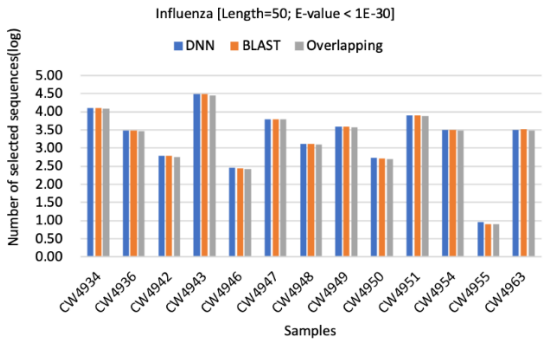


Fig. 10. Influenza test for the direct processing mode.

Fig. 11 gives the HCV test result, for samples CW2591, CW2592, and CW2593, our approach demonstrates a high degree of overlapping with BLAST, and both methods report a significantly larger amount of HCV-related sequences. For the samples CW2589, CW2590, CW2594, and CW2595, only our DNN detected the HCV viral sequences. Both methods do not report any viral sequences for sample CW2596, and reports completed different viral sequences for sample CW2588. This is due to the dynamic amount of virial sequence in the samples. In general, our method seems to more sensitive than BLAST; It reports more HCV-related sequences than the BLAST method. As compared to the test result of influenza (sample from nasal swabs), we consider the comparable level of the virus is different between these two types of samples. HCV samples are the blood sample that has a higher diversity of human-genome–related

sequences; It may increase the noise level and may cause difficulty in discovering the HCV viral sequences.

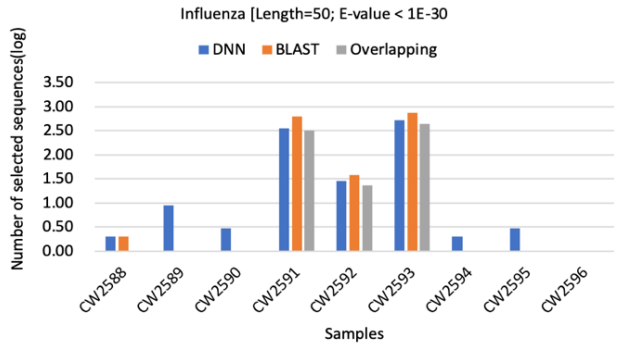


Fig. 11. HCV test for direct processing mode.

B. *Test of sequence filtering mode*

Within this sequencing filtering mode, we expect our system to generate more candidate true positive (viral sequences) and reduce the data load of BLAST. In this test, we analyze the relationship among discrimination function, dropout layer, length of input sequences, and the quality of the output sequences. We compare the output sequences from the LSTM model to all sequences from BLAST (E-value less than $1e^{-10}$).

1) *Effect of the discrimination function*

Fig.s 12 and 13 present the effect of sequence selection from HCV samples by using the discrimination functions of the direct processing mode and sequence filtering mode, respectively. The discrimination function from the direct processing mod, as shown in Fig. 12, is more conservative and outputs less candidate viral sequences. On the other side, as shown in Fig. 13, the sequence filtering mode always outputs more candidate viral sequences than the BLAST method. We observe that the direct processing mode has a high degree of overlapping with BLAST for those high copy number HCV cases(CW2591, CW2592, and CW2593) while the filtering mode can increase the chance of mutual sequences selection results with BLAST such as for CW2588, CW2589, and CW2596

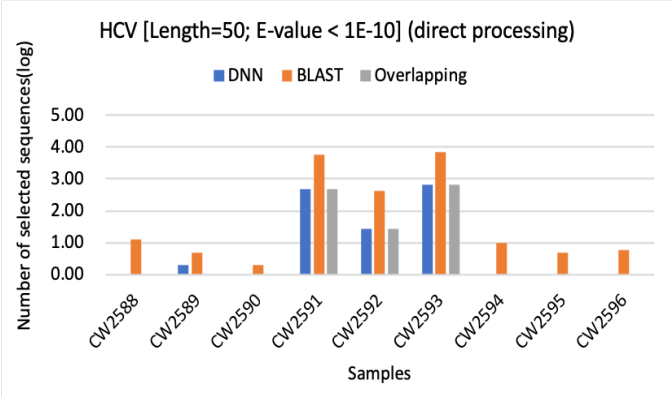


Fig.12 discrimination function for 'direct processing mode.'

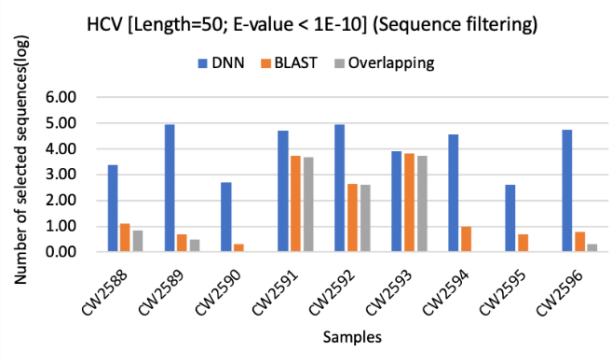


Fig.13 discrimination function for 'sequence filtering mode.'

2) *Effect of adding dropout layer*

Fig. 14 and Fig. 15 depict the effect of mutual sequences with the BLAST for dropout ratio from 10% to 30% for influenza and HCV, respectively. As we discussed in section 5D, we originally expected that adding a dropout out layer would increase the model generalization and output more mutual sequences with BLAST (true positive sequences). However, both tests show that the original approach (without using dropout) consistently has a higher degree of mutually selected viral sequences. This outcome is different from what we expected, and we discuss this phenomenon in the later discussion section.

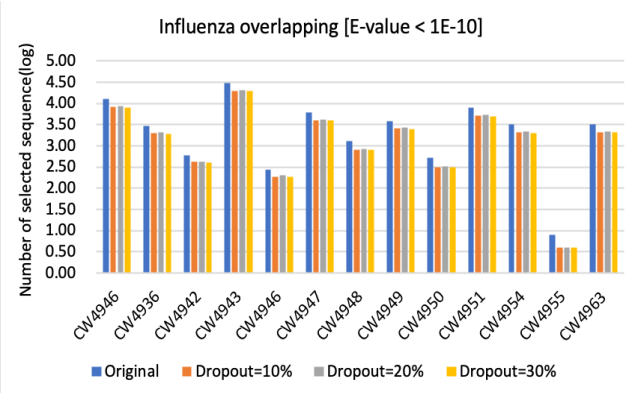


Fig. 14. Effect of adding dropout for LSTM (Influenza test)

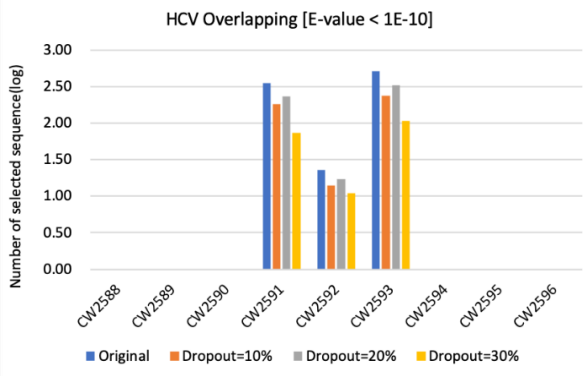


Fig. 15. Effect of adding dropout for LSTM (HCV test)

3) *Effect of sequence length*

The previous test outcome shows that adding the dropout layer in the proposed LSTM model does not facilitate for

returning more candidate true positive sequences. Here, we adjust our approach by changing the length of input sequences with the discrimination function. That is, for each input sequence, we select three of its subsequences and test with various sequence lengths (20 to 70) to check the number of output sequences. Fig. 16 shows the true positive rate (TPR or Recall) from the three most severe HCV patient's blood samples, based on different discrimination function thresholds (0.96~0.99). In general, we observe that using shorter subsequences (20 seems too short) as the input sequence length of the LSTM outputs more mutually selected sequences with the BLAST method. It is also obvious that a lower discrimination function threshold leads to a greater number of selected sequences.

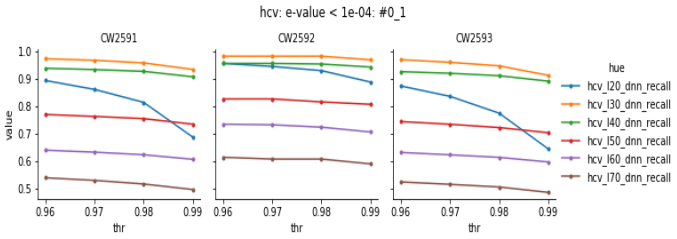


Fig. 16. Degree of sequence selection from various of sequence length

On the other hand, as for the sequence filtering mode, we also expect the system can significantly reduce the amount of data before the sequence flow goes to the BlAST method. Human samples contain more than 90% of human-related DNA sequences; We define those human sequences are true-negative and they should be removed from the LSTM filter. In this study, we assume the BLAST's result is the ground truth since it is the most widely used and most reliable sequence analysis tool. Fig. 17 shows the true negative test result from the Influenza samples.We observe that the use longer sequences length as the input sequences reflect the better result of true negative rate (higher TNR) of detecting human sequences.

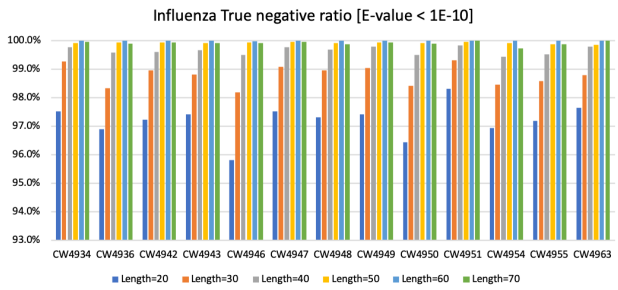


Fig. 17. True-negative test from Influenza samples

C. *Time complexity test*

The above evaluations were based on the detection of viral-related sequences from our collected 32 samples, and our results were reasonably reliable. In general, a metagenomic shotgun dataset can contain anywhere from millions to hundreds of millions of sequences; therefore, the speed of processing is another important factor for model evaluation.Our approach splits large metagenomics shotgun sequences into many smaller DNA batches and processes them with a high-performance multi-GPU. We compared the processing speed in various scenarios from a single CPU (Intel(R) Xeon(R) E5-2690 v4 @ 2.60GHz) and a single GPU to a multi-GPU (Tesla P100-PCIE-16GB). The processing speed of BLAST is mainly depended on the size of the input sequence and its reference database.

In this processing speed test, we use BLAST method as the baseline approach. The BLAST database includes about 147,000 sequences (109 million total bases). Fig. 18 shows the processing speed comparison for both our proposed deep learning approach and the BLAST method by direct processing mode of our proposed system. The y-axis describes how many millions of sequences are processed and analyzed per minute, and the x-axis shows the testing platform. From a single CPU test, our method can process 3.24 million sequences, which is about 8.7 times faster than the BLAST method. The proposed method is ordinarily designed for maximizing GPU acceleration with multiple GPUs for detecting target DNA sequences. We observe the linear increase pattern between the number of GPUs increases and the number of processing sequences per minute. Our best case (8 GPUs) shows that our method can evaluate about 13.38 million sequences per minute, which is about 36 times faster than the BLAST method.

As for the sequence filtering mode shown in Fig. 19, our approach also performs faster than the BLAST only method. Unlike the direct processing mode above, we do not see a linear increase pattern from the number of GPU and the amount of processing sequence after accelerating from the 5th GPU. This limitation is caused by the BLAST method due to the constrain of reaching the max IO threshold. Overall, our method shows the advantage of faster processing speed from 4.6 (use 1GPU) to 7.8 times (use 8 GPU) which also demonstrates the advantage of this proposed system.

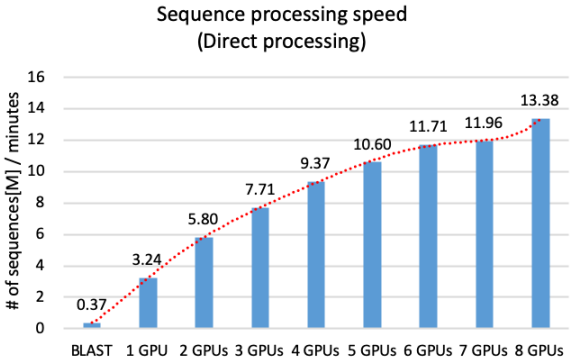


Fig. 18. The processing speed of direct procession mode

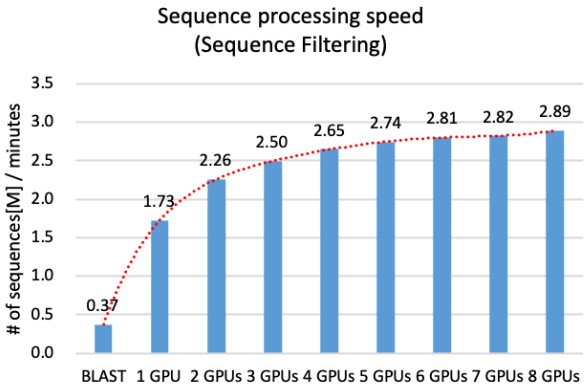


Fig. 19. The processing speed of sequence filtering mode

VII. Discussion

Our study includes discovering the effect of adding dropout layer, number of total base pairs (sequence length), and discrimination. The dropout layer is commonly used to regularize deep neural networks to avoid model overfitting and improve model performance. However, from our experimental result, we find that adding a dropout layer in the proposed LSTM model decreases the overall classification performance. One explanation may lie in the characteristic of our training data from NCBI reference data. These datasets consider the most completed (HCV, Influzensa) reference data. In other words, it is unlikely to find new HCV or influenza nucleotide sequences from given samples unless a new type of virus appears. Besides, since adding the dropout usually leads to additional training time, without considering dropout, our approach also reduces the time complexity.

In the case of dealing with low homology sequences, we proposed sequence filtering mode to select more candidate true positive sequences (virus sequences) by combining our proposed LSTM model with the conventional method, as described in Section V. Our study shows that using shorter base-pair sequences as an input can have more accurate detection for virus sequences, which is the higher TPR. A partial explanation for this may lie in the fact that the character of the viral DNA sequence is due to a shorter structure. On the other hand, we also found using longer sequences as input can help the system to remove more human-related sequences (TNR). By considering the trade-offs between TNR and TPR, we believe that not missing viral sequences is more important than reducing human sequences. From this aspect, we recommend our system adopt a shorter sequence for the sequence filtering mode by default. Moreover, the proposed prediction model was based on combined results from multiple subsequences with a discrimination function. Thus, by lowering the threshold of this function, the system can increase the number of output sequences.

VIII. Conclusion

In this work, we explored how an LSTM network can be used to learn sequential genome patterns through pathogen detection from metagenome data. Our method includes steps of data collection, preprocessing, and normalization for both public NCBI reference databases and shotgun sequences from clinical samples. We also conducted a base-calling quality analysis to find the optimal subsequence length (base pair) for our proposed model, and this further facilitated the accuracy of our pathogen detection. The proposed prediction model was based on a discrimination function that enables multiple subsequence prediction results to increase classification accuracy. We collected and conducted case studies that analyzed influenza, HCV viral sequences, and healthy samples. As for the high-performance processing algorithm, it splits large metagenomic shotgun sequences into many smaller DNA batches and processes them with a multi-GPU. We evaluated the accuracy of pathogen detection by using an HPC server that included eight NVidia Tesla P100 GPUs. Our experimental result shows that we obtained similar accuracy to the conventional BLAST method, but at a speed that was about 36 times faster.

Acknowledgments

The authors are members of the Fujitsu next generation Cloud Research Alliance Laboratory (FCRAL). This research and development work was partially supported by the MIC/SCOPE #172107106 and by Fujitsu Ltd.

References

[1] B. Langmead, C. Trapnell, M. Pop and S. L. Salzberg, "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome," *Genome biology,* vol. 10, no. 3, 2009.

[2] J. W. Masek and S. M. Paterson, "A faster algorithm computing string edit distances," *Journal of Computer and System Sciences ,* 1980.

[3] O. Gotoh, "An improved algorithm for matching biological sequences," *Journal of Molecular Biology,* 1982.

[4] NCBI, "BLAST: Basic Local Alignment Search Tool," [Online]. Available: https://blast.ncbi.nlm.nih.gov/Blast.cgi.

[5] BWA, "Aligner Burrows-Wheeler (BWA)," [Online]. Available: http://bio-bwa.sourceforge.net/.

[6] T. Seo, "Classification of Nucleotide Sequences Using Support Vector Machines," *Journal of Molecular Evolution,* 2010.

[7] S. Blasiak and H. Rangwala, "A Hidden Markov Model Variant for Sequence Classification," in *International Joint Conference on Artificial Intelligence*, 2011.

[8] R. Rizzo, A. Fiannaca and M. L. Rosa, "A Deep Learning Approach to DNA Sequence Classification," in *Proceeding of International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics(CIBB)*, 2015.

[9] G. Aoki and Y. Sakakibara, "Convolutional neural networks for classification of alignments of non-coding RNA sequences," *Bioinformatics,* vol. 34, no. 10, 2018.

[10] N. G. Nguyen, V. A. Tran, D. L. Ngo and D. Phan, "DNA Sequence Classification by Convolutional Neural Network," *Journal of Biomedical Science and Engineering,* vol. 9, 2016.

[11] h. Yin, Y. Chen and S. S.-T. Yau, "A measure of DNA sequence similarity by Fourier Transform with applications on hierarchical clustering," *Journal of Theoretical Biology ,* 2014.

[12] T. Hoang, C. Yin, H. Zheng, C. Yu, R. L. H. and S. S.-T. Yau, "A new method to cluster DNA sequences using Fourier power spectrum," *Journal of Theoretical Biology ,* 2015.