Rapid DNA Barcode Demultiplexing

2020 Summer STEM Institute

Michelle Lei

Abstract

Coronavirus has emphasized the necessity for inexpensive and rapid testing, which can be done using a handheld, realtime DNA sequencer, the MinION device. DNA barcodes are unique identifiers that can be attached to different people's DNA samples. Through the use of barcodes, multiple people's samples of DNA can be sequenced with the MinION device at one time. Barcode demultiplexing is the process by which the barcodes are sequenced so that the DNA fragments attached can be traced back to the original person. Currently, barcode demultiplexing is very inaccurate when basecalled, due to the noise in the barcodes, or very complex when using deep neural networks. Here, we present a new method by using raw electrical signals to maintain higher classification accuracy while reducing complexity. We simulate barcode reads of two different classes using DeepSimulator, then extract features including the Hjorth parameters and the Hurst exponent that were previously used in EEG data classification. We utilized supervised learning algorithms to classify the reads and compare the statistics and latency to the state-of-the-art Deepbinner. While maintaining the accuracy of Deepbinner, our model has a much lower latency and is less complex. Our model can classify with a 99.20% f1 score, which is comparable to Deepbinner's results. We can achieve this f1 score using around 625 barcodes to train, and the latency using random forest classification is 5.15 seconds. In the future, more than two barcodes can be classified using our model, and real data can be applied instead of simulated data. By reducing the complexity of Deepbinner, using supervised learning instead of deep neural networks, we can increase the efficiency and decrease the cost of demultiplexing to make testing scalable for future pandemics.

1. Introduction

Currently, SARS-CoV-2 is causing an unprecedented global pandemic that requires efficient and cost-effective testing. To test for viruses, the RNA fragments are converted to complementary DNA and are then sequenced. A third-generation sequencing method, the Oxford Nanopore Technology's (ONT) MinION, was popularized in recent years for its higher speed, longer read lengths, and real-time sequencing (Filloux et al., 2018; Lu et al., 2016). It is a portable, low cost handheld device, which makes it useful in doctors' offices. The MinION contains an electrically charged flow cell, perforated with nanopores. Adapters ligated to the DNA fragments lead the DNA fragment to one of the 512 channels in the flow cell. When the DNA enters, the nucleotides in each DNA fragment are measured as a sequence of electrical current signals called a "squiggle", which can be converted to a corresponding nucleotide sequence through a process called basecalling. Each squiggle obtained from a DNA fragment is called a read. The MinION has been used to test for Ebola, dengue, influenza, and the Ross River virus (Filloux et al., 2018). The MinION has the potential for detecting coronavirus in a reasonable time (Chen et al., 2020). Current testing was insufficient to meet the demands of the Covid-19.

In order to make testing scalable for future pandemics, a MinION feature, barcoding, has gained some popularity. The LamPORE Assay, an assay designed by ONT to sequence more samples per flow cell, is being developed. It can run between 1 to 1,152 samples on one flow cell and takes about an hour to process the first 1 to 96 samples. The process of sequencing the remaining samples is projected to take around 4 hours. The LamPORE assay uses DNA barcodes to multiplex (lam). Barcodes are unique identifiers; for example, DNA barcodes, which are a part of the DNA of different species, are generally used to identify the species (bar). When multiple people's DNA fragments are mixed together, there is no unique identifier to distinguish the fragments from different individuals. For each individual, a

[*] *Research project conducted during the 2020 Summer STEM Institute.* The Summer STEM Institute (SSI) is a six-week virtual summer program where high school students learn how to design and conduct data science and other computational research projects. To view other SSI distinguished projects, please visit www.summersteminstitute.org.

unique barcode can be ligated to that particular individual's DNA fragments so that multiple samples of DNA to be tested for a virus can be inserted into the MinION device at once, or multiplexed. Many different samples can then be tested for a virus, as shown in Figure 1. The DNA sequences can later be matched, or demultiplexed, to the person they belong to through the use of the barcode classification (Wick et al., 2018). If a DNA sample tests positive, the DNA barcodes can identify to whom the DNA sample belongs. This would allow for increased and scalable testing in the future (lam). The barcode, however, becomes very noisy after basecalling. This could be a result of various factors, such as structural similarities in nucleotides, simultaneous influence of multiple nucleotides on the signal, nonuniform speeds of nucleotides through the pore, and the signal not changing within homopolymers (Rang et al., 2018). Current solutions to this are very complex or inaccurate, but the problem itself seems simple.

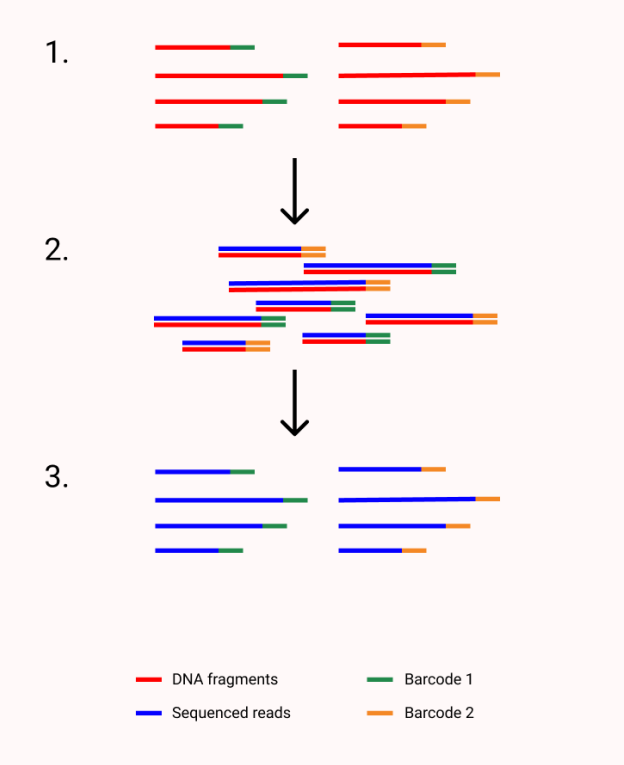

Figure 1. DNA fragments with barcodes attached can be sequenced. The sequenced reads can be traced back to the original individual based on barcodes.

While very useful for making testing scalable, demultiplexing is a challenge. In the past, basecalling algorithms such as Albacore and Porechop were used to identify barcode reads. However, the basecalling methods are unable to classify many reads and have a 20% error rate (Filloux et al., 2018). Many reads could not be binned, or assigned to a barcode, because of how noisy the reads are. Instead of basecalling, the state-of-the-art demultiplexer Deepbinner uses raw electrical signals for classifying reads.

Electroencephalography (EEG) signals resemble MinION reads (Kupkova et al., 2017). EEG signals and MinION reads are both nonlinear and nonstationary, and have spontaneous oscillations (Rahmani et al., 2018). This means that from individual to individual, the frequency of the EEG signal differs (Oh et al., 2014). Similarly, from read to read, the frequency of the read signal changes. EEG data are classified through feature extraction and supervised learning algorithms. A novel approach to classifying barcode reads would be to utilize these EEG data classification methods. Simple features include the Hjorth parameters and the Hurst exponent (Bao et al., 2020). The Hjorth parameters include activity, mobility, and complexity, and they are calculated using the variance of signal waves in a time series (Hjorth, 1970). The Hurst exponent measures the long-memory and predictability of a signal in the time domain (Rahmani et al., 2018). A value of 0.5 indicates that the series is random, while a value less than 0.5 indicates the series is anti-persistent, and a value greater than 0.5 indicates the series is persistent (Qian & Rasheed). Instead of basecalling the barcodes, we focus on signal characteristics and utilize the extra information in the raw data, similar to DeepBinner. This extra information that is lost when converted to base calls will help to classify barcodes faster and more efficiently (Rang et al., 2018). However, instead of utilizing deep neural networks, we will classify barcodes through extracting these four features used for EEG signal data classification and using supervised learning algorithms, reducing the complexity of DeepBinner while maintaining its accuracy.

Accurately classifying barcodes quickly and simplistically will improve the LamPORE Assay. Multiple DNA samples can be simultaneously sequenced in a single flow cell, increasing the speed and reducing the cost of DNA sequencing for SARS-CoV-2. As future pandemics threaten to infect globally, scalable, fast, and inexpensive testing becomes more important.

The paper will proceed as follows. In Section 2, we review current literature on barcode demultiplexing. In Section 3, we explain the methodology of our research, including its design and implementation. Next, in Section 4, we provide our results and discuss them. Finally, in Section 5, we conclude and describe future directions.

2. Literature Review

Currently, DNA barcodes are being identified using basecalling methods, which are unable to classify many reads and have a 20% error rate (Filloux et al., 2018). The state of the art barcoding technology right now is Deepbinner,

which looks at raw electrical signals to classify reads into the proper bins and operates using deep neural networks. By using signal-space instead of base-space tools, there is greater accuracy because the signals would not first be converted to base calls, where noise could be introduced (Wick et al., 2018). Deepbineer has high sensitivity, can bin lower quality reads, and has higher precision. However, training Deepbinner's networks is computationally intensive and they does not generalize well over different flow cells and library preparation kits. The network is composed of groups of convolutional layers, which are the result of randomized trials of various architectures. The "blackbox" nature of Deepbinner makes it difficult to identify the reason behind an error in classification (Wick et al., 2018). Because of the complexity, Deepbinner requires supercomputers to operate and a high level of computing for real-time analyses. The latency of Deepbinner is very high. These disadvantages introduce an opportunity to improve upon barcode demultiplexing. While maintaining the high precision and sensitivity of Deepbinner, we look to reduce its complexity.

In 1970, the Hjorth parameters had been discovered in order to classify EEG data. They were able to describe qualities of a time series through the use of variances, which allows them to inherit the additive properties of variance so that the parameter values measured pertain to the average basic elements in the complex curve (Hjorth, 1970). In 2000, they were used to distinguish between three different mental states of normal EEG, where the lower dimensions of the feature vectors allowed for simplicity in computation and stability in classification (**?**). The Hjorth parameters and another EEG feature extraction method Fisher ratio were also used to associate EEG patterns with certain physical tasks (Oh et al., 2014).

Recently, research was done exploring Hjorth parameters in classifying *E. coli* and bacteriophage DNA. The modulations of the ionic currents at each pore can be graphed as a squiggle when they pass through the Nanopore through Python packages such as Poretools (Loman & Quinlan, 2014). Feature extraction is simple and undemanding when applying Hjorth parameters, requiring only a linear time complexity. The Hjorth descriptors consist of three features, which are activity, mobility, and complexity (Kupkova et al., 2017). Activity represents the variance of a time function, and it is large if there are many high frequency components to the signal. Mobility is the proportion of the standard deviation of the power spectrum. Complexity indicates how the shape of a signal compares to a pure sine wave, converging to one as the shape of the signal becomes more similar to the sine wave (Hjorth, 1970). These values can be calculated by using the variance of the squiggles.

$$Activity = var(y(t))$$

$$Mobility = \sqrt{\frac{var(\frac{dy(t)}{dt})}{var(y(t))}}$$

$$Complexity = \frac{Mobility(\frac{dy(t)}{dt})}{Mobility(y(t))}$$

The three features can be visualized as three-dimensional feature vectors that form clusters, and the DNA can be separated into two different organisms (Kupkova et al., 2017). The transformation is also deterministic, so supervised algorithms can be trained to identify the phage DNA for automatic classification of the reads as the position of the clusters and their centroids will remain constant for a single organism. Possible supervised learning algorithms include Support Vector Machine (SVM), which finds an optimal hydroplane between two sets of datasets (Ghandi); random forest, which assesses various binary trees that split data along nodes to eventually end in the right class (Koehrsen); and K-Nearest Number (K-NN), which classifies a data point based on the data points around that point (knn). The success of the Hjorth parameters in identifying phage DNA when mixed with bacteria DNA introduces the possibility of using the parameters for DNA barcode classification.

3. Methods

Instead of basecalling the barcodes into nucleotide sequences, we looked at raw electrical signals over a time-domain and utilized the extra information in them beyond the corresponding nucleotides. This allows for higher sensitivity, which can reduce the number of unclassified reads (Wick). To assess classification of barcodes from feature extraction, we simulated a dataset using DeepSimulator, extracted features, and used supervised learning algorithms such as K-NN and SVM.

Simulating a Dataset

We obtained DNA barcodes through DeepSimulator, which generates barcode signals similar to those of real barcodes without sequencing real fragments of DNA. DeepSimulator is the state-of-the-art Nanopore simulator right now, and it is widely known for its accuracy and is commonly used in place of real reads. Two nucleotide barcode sequences, AAGAAAGTTGTCGGTGTCTTTGTG and TCGATTCCGTTTGTAGTCGTCTGT, were inputted into DeepSimulator, with 40,000 reads being generated for each.

DeepSimulator first samples sequences from the genome that correspond to DNA sequences which pass through the Nanopore, with the read length being alterable by the user. The length of DNA barcodes is twenty-four nucleotides, so the read-length set was twenty-four (Wick et al., 2018).

A context independent pore model, based on the official Nanopore Tech stats of 6-mers, helps generate signals quickly. The signal generator makes generated raw electrical signals in the form of a time series that are very similar to real ones and are of variant qualities (Li et al., 2020). Because barcodes, at the ends of reads, are inherently noisier than regular reads, noise was added to the barcodes acquired from DeepSimulator. Based on average barcode noise levels, the noise on the barcodes was increased two standard deviations. When the signals were read, they appeared as a list of numbers. There were around 240 signals per read. Sometimes, DeepSimulator generated barcode reads of less than 100 signals. When this was the case, the barcode reads were discarded because the short reads could not give adequate information, and the sequencing quality was insufficient.

The barcode signal data stored in Fast5 files were extracted using the ONT Python library. 4,000 signal events were generated through a single second of time. Median/Median Absolute Deviation (MAD) normalization was then applied to the data, following by removing duplicates from the samples. A total of 3410 (1710 each) unique reads were included in further analysis.

Extracting Features

Features are commonly extracted from EEG data through the use of time domain properties. Squiggles from MinION reads are comparable to those from EEG data, being nonlinear and non-stationary. Given the physical similarities between the two time series, we extracted common features for barcode classification. However, because MinION reads are only twenty-four nucleotides long and much smaller than EEG data, resulting in around 600 signals per read, choosing a limited number of features to extract should be sufficient. Hjorth parameters have previously been used to classify between bacteria and bacteriophage DNA (Kupkova et al., 2017), so we used them to classify the barcodes.

A common tool for evaluating features to classify EEG data is Pyeeg. This Python library consists of several functions and can calculate Hjorth parameters, entropy, and Hurst exponent, amongst other features. Because our data was standardized using the median and median absolute deviation and the Pyeeg source code operates off of different assumptions, we wrote our own functions similar to that of Pyeeg to find the Hjorth parameters: activity, mobility, and complexity. The Hjorth parameters and the barcode class were added to a dataframe, which was then split into train and test sets, using a random seed of 31.

We classified the reads using SVM and random forest, finding the f1 scores to be around 80%. We added another feature commonly used for EEG data, the Hurst exponent. With the additional feature, the total number of dimensions became four. Since classifying twenty-four nucleotide barcode reads should not be a complex task, we removed one feature. After looking at scatter plots of activity against the other three features, we found that activity did not show a large distinction between the barcode classes. We removed activity and discovered that the f1 score and other assessment statistics increased for SVM and random forest. Activity is the variance of the signal, whereas the other features extracted require more calculation. Therefore, variation between the two barcode classes may have been too similar for activity to have distinguished between them.

Supervised Learning Algorithm

Several supervised learning algorithms were implemented to classify the barcodes, including K-NN, random forest, and SVM. Through plotting various f1 scores from different k values, the k value of 3 resulted in the highest f1 score. Tenfold cross validation was used for random forest. After using GridSearchCV, the RBF kernel with a tenfold cross validation was determined to have the highest f1 score for SVM. We set a constant random seed of 31, and we took statistical metrics after training 20% of the reads and testing on the remaining 80%.

We reserved half of the data for testing and used the other half for training, increasing the training number of reads from 50 to 1700 incrementally by 50 to find the minimum training dataset number for the f1 score to remain using convergence plots. The convergence plots are seen in Figure 1.

The state-of-the-art is Deepbinner, which has a high accuracy, but also a large classification latency. We ran Deepbinner on a Linux machine to assess its latency and accuracy on the same barcodes used for feature extraction and classification. Because we ran the feature extraction classification without a Linux machine, we ran Deepbinner on a single thread so as to measure the prediction latency of the model, not the aptitude of the machine. The barcode sequences used in the current study belong to the sequences in the native barcoding kit, for which a pre-trained model has been built for the Deepbinner software (Wick). Therefore, we used Deepbinner to perform classification on the two barcodes with the native model. Had the nucleotide sequences used in the study not been included in the pretrained models, a lot of data and heavy computation would have been needed to train Deepbinner (Wick).

Four Barcodes

We evaluated the performance of our model when using four barcodes. Using DeepSimulator, we generated two additional twenty-four nucleotide signal sequences. The nucleotide sequences inputted were GAGTCTTGTGTCCCAGTTACCAGG and TTCGGATTCTATCGTGTTTCCCTA. Two standard deviations of noise were added to these sequences as well. After removing duplicates and reads

that were less than 100 nucleotides, we standardized the signals. We then extracted mobility, complexity, and the hurst exponent from these reads. Using these features, we tried different supervised learning algorithms, including K-NN, random forest, and SVM. Through plotting the macro f1 score across different k values, we found that a k value of 3 performed the best. We found through GridSearchCV that the polynomial kernel, with degree 3, performed the best. We compared tenfold polynomial SVM, random forest, and 3-NN.

4. Results and Discussion

To evaluate Deepbinner and our model, we looked at statistical metrics, including the f1 score, sensitivity, specificity, and precision. We also looked at the latency of each classification method. The latency of classification is measured as the delay between when the user inputs the command to classify and when the program produces an output. An attribute of an efficient classification method is a short latency. Sensitivity is the proportion of positive reads binned correctly. Specificity is the proportion of negative reads binned correctly. Precision is the proportion of reads binned positive that are true positives. In all of these metrics, a true positive can be defined as Barcode 1 being classified as Barcode 1 and a true negative can be defined as Barcode 2 being classified as Barcode 2. The f1 score can be defined as

$$f1\ score = \frac{2 \times precision \times sensitivity}{precision + sensitivity}$$

The latency for classification of random forest is 5.15 seconds while the latency of Deepbinner, which is 107.56 seconds. The sensitivity, precision, specificity, and f1 scores of random forest were all above 99%, and Deepbinner's were 100%. In addition, 24.77% of Deepbinner's reads were not binned, which were not considered in the table.

After evaluating RBF SVM, 3-NN, and random forest, we found that out of all the supervised learning algorithms used, random forest was shown to have the highest statistical metrics except for sensitivity and f1 score, with values of above 99% for every metric. It achieved a 99.20% f1 score while only training 625 out of all 3420 reads, which is around 19% of the total reads, as seen in Figure 2.

We attempted to classify two more barcodes along with the other two as an extension, for a total of four barcodes. For four barcode classification, we used sensitivity, precision, and macro f1 score, which is the average of the f1 scores for each barcode class, as statistical metrics. We found that random forest performed the best, with sensitivity, precision, and macro f1 score all above 90% as seen in Table 2.

Using random forest and tenfold cross validation, the DNA barcodes could be classified with comparable accuracy, sensitivity, and precision to DeepBinner, as seen in Table 1. The latency of the supervised learning algorithm classification was much shorter than DeepBinner. DeepBinner must classify using its neural network hidden layers in a very time expensive process, a process which is much simpler when training and classifying using a supervised learning algorithm. The 100% statistical metrics of Deepbinner were the result of Deepbinner being pretrained with a package of reads. If a read is too short, there is not enough information for Deepbinner to bin the read, and these reads that were not binned were not considered for the table.

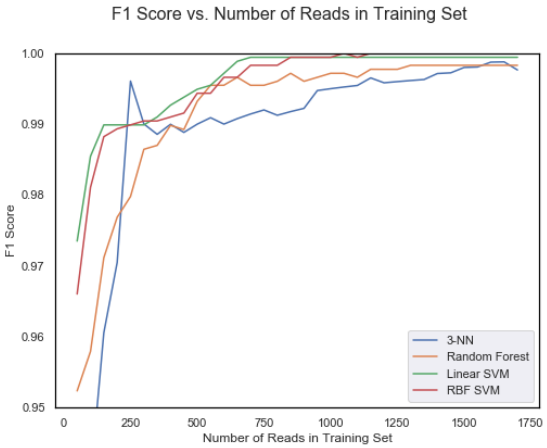

Figure 2. A convergence plot showing f1 scores of RBF SVM, Linear SVM, Random Forest, and 3-NN across different training set sizes.

Our data came from a simulator rather than actual MinION reads because of the lack of accessibility in obtaining real reads. While our model outperformed Deepbinner in latency and maintained similar sensitivity, precision, and f1 accuracy, we did not have real reads to test the two models against, though DeepSimulator is well-known for its accuracy in simulating reads. In our research, we did not perform barcode extraction as would be necessary in real reads; instead, we simulated only barcodes. This is a limitation in the performance of our model, as we do not know how this might affect the latency or the statistical metrics.

Both DeepBinner and demultiplexing using feature extraction utilize raw electrical signals as opposed to basecalling. These raw electrical signals carry extra information in them besides the nucleotides they correspond to that allow them to be binned more often and with higher accuracy. Deep convolutional neural networks, which are used for DeepBinner, are extremely accurate, but they are very complex. They have many hidden layers that require a lot of time and data to train. Using feature extraction of two Hjorth parameters, mobility and complexity, and the Hurst exponent, the data could be plotted with three dimensional feature vectors in

Algorithm	Sensitivity (%)	Specificity (%)	Precision (%)	F1 Score (%)	Latency (s)
3-NN	99.54	97.15	97.30	98.41	0.29
RBF SVM	99.71	94.77	95.09	99.31	0.03
Random Forest	99.27	99.12	99.13	99.20	5.15
Deepbinner	100	100	100	100	107.56

Table 1. The sensitivity, specificity, precision, f1 score, and latency values across different classification algorithms and Deepbinner.

Algorithm	Sensitivity (%)	Precision (%)	Macro F1 Score (%)	Latency (s)
3-NN	86.94	86.74	88.93	0.66
Polynomial SVM	88.27	88.16	88.16	0.13
Random Forest	90.23	90.24	90.22	12.96

Table 2. The sensitivity, precision, macro f1 score, and latency values across different classification algorithms for four barcode classification.

Figure 3. They formed evident clusters which could be separated using deterministic algorithms. For similar accuracy to DeepBinner, 3-NN only needed to train 625 reads. Given the simplicity of classifying 24 nucleotide reads, using deep neural networks appears unnecessary. Supervised learning algorithms work just as well with less training, less time, and lower costs.

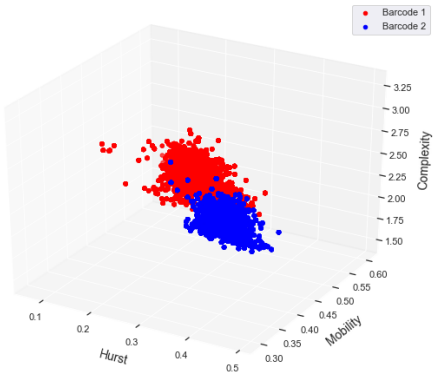

Figure 3. A convergence plot showing f1 scores of RBF SVM, Linear SVM, Random Forest, and 3-NN across different training set sizes.

DeepBinner is computationally intensive and does not generalize well across different flow cells. It is also intensive to install. DeepBinner has many layers to train, so it takes a long time to train and classify barcodes. Right now, there are only two pre-trained models included in Deepbinner software. To train barcodes other than the ones in the models requires a lot of data and a fast computer, preferably with TensorFlow on a big GPU (Wick). Its ”black box” nature makes the classification process difficult to understand or resolve if there is an error. When using DNA barcodes to multiplex DNA samples for efficient testing, it is important to demultiplex as quickly as possible, with as much accuracy as possible, and with as little training as possible. Instead of making the problem more complex, as is the case with going from basecalling to deep neural networks, the problem should be made simpler by using supervised learning algorithms and extracting features, including the Hjorth parameters and the Hurst exponent. This novel approach to classifying DNA barcodes will allow these barcodes to then be demultiplexed rapidly and with small amounts of training data while resulting in high macro f1 scores, precision, and sensitivity. Barcode demultiplexing using feature extraction will contribute to the efficiency and cost-effectiveness of the LamPORE Assay, providing faster and less expensive testing for diseases.

While classifying between two different classes of barcodes works well with feature extraction, the LamPORE Assay has the ability to multiplex up to 1,152 samp in a single flow cell. Currently, Deepbinner is capable of demultiplexing

twelve barcodes, which is the number of distinct barcodes in barcode kits, with high accuracy (Wick et al., 2018). In the future, more barcodes could potentially be demultiplexed using supervised learning algorithms. We have begun to classify four barcodes, with macro f1 scores from random forest being 90.22%, shown in Table 2. We notice that very few barcodes are misclassified to be barcode 1, while barcodes 3 and 4 are frequently classified as the other. This could be a result of similar nucleotide sequences throughout the barcodes. While multiple barcode demultiplexing can be done, with this model, the barcodes should ideally be very different from each other. The next step is to utilize real data to classify reads, and to explore other features to better classify multiple barcode classes.

5. Conclusion

DNA barcodes allow for fast and inexpensive testing since the price of a single flow cell can be held constant and multiple DNA fragments can be sequenced simultaneously. To demultiplex the noisy DNA barcodes, we use raw electrical signals, for the raw electrical signals contain more information than basecalled sequences. We are not interested in the exact sequence of the barcodes; rather, we want to classify them, to identify to which person's DNA they are attached. Because of this, basecalling becomes unimportant.

Using data generated by DeepSimulator, we were able to reconstruct squiggles from a list of signals. By taking the variances of these signals, we then extracted Hjorth parameters and the Hurst exponent from the reads. We were able to obtain a 99.20% f1 score, a 99.13% precision, a 99.12% specificity, and a 99.27% sensitivity from using random forest, in Table 1. These values were comparable to the performance of the state-of-the-art, Deepbinner, which utilizes deep neural networks. Deepbinner is computationally intensive, so its classification latency was high. When compared with our model, which had a latency of 5.15 seconds, Deepbinner had a higher latency of 107.56 seconds. Deepbinner also requires much more training than our model, which only required 19% of the data for training to achieve a 99.20% f1 score.

Our model outperformed Deepbinner in latency, being a much simpler solution to barcode demultiplexing. At the same time, our model maintained the statistical performance of Deepbinner. In future research, multiple barcodes can be classified using feature extraction instead of just the two in this research.

The LamPORE Assay has provided an opportunity for multiplexing barcodes through the MinION device to sequence DNA samples faster and for less cost. This will make testing scalable for future pandemics. In the past, because DNA barcodes are noisy, demultiplexing has either been inaccurate or very complex. Our model is able to maintain the state-of-the-art's accuracy while reducing complexity when classifying between two barcode classes.

References

DNA barcoding - International Barcode of Life.

K Nearest Neighbor Algorithm In Python — by Cory Maklin — Towards Data Science.

LamPORE Assay.

Bao, F. S., Liu, X., and Zhang, C. Pyeeg: An open source python module for eeg/meg feature extraction. *Computational Intelligence and Neuroscience*, 2011:1–7, 2020. doi: 10.1155/2011/406391.

Chen, B., Tian, E.-K., He, B., Tian, L., Han, R., Wang, S., Xiang, Q., Zhang, S., El Arnaout, T., and Cheng, W. Overview of lethal human coronaviruses. *Signal Transduction and Targeted Therapy*, 5(1):1–16, 2020. doi: 10.1038/s41392-020-0190-2.

Filloux, D., Fernandez, E., Loire, E., Claude, L., Galzi, S., Candresse, T., Winter, S., Jeeva, M. L., Makeshkumar, T., Martin, D. P., and Roumagnac, P. Nanopore-based detection and characterization of yam viruses. *Scientific Reports*, 8(1):17879, 2018. doi: https://www.nature.com/articles/s41598-018-36042-7.

Ghandi, R. *Support Vector Machine – Introduction to Machine Learning Algorithms*.

Hjorth, B. Eeg analysis based on time domain properties. *Electroencephalography and Clinical Neurophysiology*, 29(3):306–310, 1970. doi: 10.1016/0013-4694(70)90143-4.

Koehrsen, W. An implementation and explanation of the random forest in python.

Kupkova, K., Sedlar, K., and Provaznik, I. Reference-free identification of phage dna using signal processing on nanopore data. In *2017 IEEE 17th International Conference on Bioinformatics and Bioengineering (BIBE)*, pp. 101–105, Washington, DC, 10 2017. IEEE.

Li, Y., Wang, S., Bi, C., Qiu, Z., Li, M., and Gao, X. Deepsimulator1.5: a more powerful, quicker and lighter simulator for nanopore sequencing. *Bioinformatics*, 36(8):2578–2580, 2020. doi: 10.1093/bioinformatics/btz963.

Loman, N. and Quinlan, A. Poretools: a toolkit for analyzing nanopore sequence data. *Bioinformatics*, 30(23):3399–3401, 2014. doi: https://academic.oup.com/bioinformatics/article/30/23/3399/207172.

Lu, H., Giordano, F., and Ning, Z. Oxford nanopore minion sequencing and genome assembly. *Genomics, Proteomics & Bioinformatics*, 14(5):265–279, 2016. doi: http://www.sciencedirect.com/science/article/pii/S1672022916301309.

Oh, S.-H., Lee, Y.-R., and Kim, H.-N. A novel eeg feature extraction method using hjorth parameter. *IJEEE*, pp. 106–110, 2014. doi: 10.12720/ijeee.2.2.106-110.

Qian, B. and Rasheed, K. Hurst exponent and financial market predictability. pp. 7.

Rahmani, B., Wong, C. K., Norouzzadeh, P., Bodurka, J., and McKinney, B. Dynamical hurst analysis identifies eeg channel differences between ptsd and healthy controls. *PLOS ONE*, 13(7):e0199144, 2018. doi: 10.1371/journal.pone.0199144.

Rang, F. J., Kloosterman, W. P., and de Ridder, J. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. *Genome Biology*, 19(1):90, 2018. doi: 10.1186/s13059-018-1462-9.

Wick, R. rrwick/deepbinner.

Wick, R. R., Judd, L. M., and Holt, K. E. Demultiplexing barcoded oxford nanopore reads with deep convolutional neural networks. Technical report, Bioinformatics, 2018.