Real-time selective sequencing using nanopores and deep learning

Artem Danilevsky
Tel Aviv University
Avital Luba Polsky
Tel Aviv University
Noam Shomron (nshomron@tauex.tau.ac.il)
Tel Aviv University

Research Article

Keywords: Nanopore-sequencing, selective-sequencing, deep-learning, real-time, classification

Posted Date: May 24th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-540693/v1

Version of Record: A version of this preprint was published at Briefings in Bioinformatics on July 1st, 2022. See the published version at https://doi.org/10.1093/bib/bbac251.

Abstract

Nanopore sequencing is an emerging technology that utilizes a unique method of reading nucleic acid sequences and, at the same time, it detects various chemical modifications. Deep learning has increased in popularity as a useful technique to solve many complex computational tasks. Selective sequencing has been widely used in genomic research; although it introduces several caveats to the process of sequencing, its advantages supersede them. In this study we demonstrate an alternative method of software-based selective sequencing that is performed in real time by combining nanopore sequencing and deep learning. Our results show the feasibility of using deep learning for classifying signals from only the first 200 nucleotides in a raw nanopore sequencing signal format. Using custom deep learning models and a script utilizing "Read-Until" framework to target mitochondrial molecules in real time from a human cell line sample, we achieved a significant separation and enrichment ability of more than 2-fold. In a series of very short sequencing runs (10, 30, and 120 minutes), we identified genomic and mitochondrial reads with accuracy above 90%, although mitochondrial DNA comprises only 0.1% of the total input material. We believe that our results will lay the foundation for rapid and selective sequencing using nanopore technology and will pave the way for future clinical applications using nanopore sequencing data.

Highlights

- First study showing deep learning’s ability to classify raw nanopore sequencing data
- Comparison of 5 neural network architectures for raw nanopore data
- More than 90% accuracy for the classification of mitochondrial DNA
- Enrichment (2.3 fold) of mitochondrial DNA proportion with our method
- Successful use of nanopore selective sequencing feature

1. Introduction

1.1 Next generation sequencing

Next generation sequencing (NGS) has revolutionized DNA sequencing and laid the foundation for a plethora of scientific and clinical opportunities. One recent emerging sequencing technology uses nanopore sequencing (for example, those developed by Oxford Nanopore Technologies, ONT) [1]. In this study we used ONT’s portable MinION sequencer, which was released in 2014. The sequencing is performed by measuring changes in ionic current produced by individual nucleic acids as single DNA strands that pass through an array of protein nanopores. These changes are detected by a sensor and are saved on a computer for later analysis[2]. The recorded ionic current, known as the "raw signal" or "squiggle," is mainly used for basecalling by translating the raw signal into nucleotides. To date, the vast majority of studies that use nanopore sequencers ignore the raw signal after using it to generate a

nucleotide sequence. A few researchers, however, have used this signal for other tasks, such as improving the accuracy of a consensus sequence, or for investigating chemical modifications on the DNA[3,4,5].

1.2 Deep learning

Deep learning is a subset of machine learning methods that have gained increased popularity in recent years after overtaking other methods in the field of image classification[6]. Deep learning has been applied to fields such as image, video, audio, and natural language processing where it has been used to perform tasks such as classification, generation, prediction, and detection[6–8]. Therefore, it is plausible that similar deep learning approaches could be applied to nanopore sequencing data analysis. These approaches include methods such as convolutional neural network (CNN) and recurrent neural network (RNN) architectures that have been used for audio signal analysis[6,9–11]. Initially, raw nanopore signal was translated to nucleotides using a Hidden Markov Model (HMM)[12,13], but recently, deep learning was found to perform the task better and it is now used to translate a raw nanopore signal into a nucleotide sequence [14,15]. Deep learning is also used to perform tasks such as predicting DNA methylation[16] and simulating a raw signal based on a reference genome[17]. These findings reinforce our suggestion to use deep learning in order to classify reads based on their raw signal. Hence, we tested several commonly used deep learning architectures that were previousely applied on similar data in order to select the one that we preferred for our analysis.

1.3 Selective sequencing

Selective sequencing (or sequencing of targeted genomic regions) is a widespread technique used in many applications when the goal is to sequence specific portions of a DNA molecule from a larger pool of genetic material. When targeting only part of the DNA, one can save resources, time, and money. Selective sequencing is traditionally based on physically isolating parts of the DNA during the library preparation steps and prior to sequencing[18–20]. Recently, it was also performed during standard nanopore library preparation[21,22]. Traditional selective methods, however, have been found to introduce bias to the output like lack of evenness of coverage and divergent results from different library preparation kits[23], therefore an alternative method could benefit researchers.

1.4 Nanopore selective sequencing

With the introduction of nanopore sequencing, an exciting new feature, "Read Until", makes it possible to selectively “reject” DNA molecules before the entire molecule has been completely sequenced[24]. The decision to reject the molecule is based on the initial portion of the DNA molecule, potentially saving time and reagents by not sequencing the entire molecule. Several studies have demonstrated real-time selective sequencing using the nanopore “Read Until” feature. In this regard, Loose *et al*. demonstrated in a first published study the ability to perform selective sequencing with the genome of Lambda phage[24]; dynamic time warping (DTW) was used to determine whether the DNA molecule should be sequenced or not. This approach imposed restrictions on the length of the possible target and reference sequences. In

another study, Edwards *et al*. performed real-time selective sequencing by online basecalling the start of the molecules and then deciding which molecule to sequence by mapping it to a reference library using the LAST aligner[25], which is similar to the method used by Payne *et al*. who mapped the base-called nucleotides to a reference genome using minimap2[26]. This approach removed the constraints caused by using the DTW algorithm; however, it introduced two separate steps (basecalling and mapping) into the decision process. Another study used the same concept of basecalling and mapping the reads to a reference genome, but for a different purpose, namely, to achieve more uniform coverage[27]. Finally, in a more recent work, Kovaka *et al*. probabilistically decoded the raw signal into k-mers by using a technique based on an HMM achieving enrichment factor of 4.46[28].

1.5 Our contribution

Here we apply selective sequencing on nanopore sequencing via a unique deep learning approach. We begin by developing a deep learning model capable of accepting only the first 2,000 values of a raw signal, which equates to roughly 200 base pairs as input. We decided to focus on a biologically significant region of human DNA that potentially will provide enough data in whole genome sequencing experiments. This is a prerequisite for training the deep learning model. We chose to perform selective sequencing on mitochondrial DNA. The mitochondrial DNA is a cellular organelle within eukaryotic cells containing about 16K base pairs; it encodes 13 proteins. It has been sequenced many times, has high coverage in publicly available nanopore datasets, and is of biological and medical significance when analyzing human sequencing data[29]. We trained the model to classify sequencing reads into 'mitochondrial' or 'genomic' reads based on the signal. Analysis of the raw signal directly bypasses the error prone basecalling step while also allowing the deep learning model to incorporate additional information present in the raw signal such as DNA modifications[4,5], this potentially could increase accuracy of DNA classification by eliminating data analysis steps and increasing the information volume for the deep learning model. Unlike the previous attempts at real-time selective sequencing, our method neither requires a nucleotide reference nor a generated signal reference. Bypassing a reference decreases the run time and complexity restriction as the reference database expands. We also tested several deep learning architectures for sequence analysis; we tried it on several datasets of nanopore signal data, and applied it for classifying reads of different DNA origins. Finally, we selected the model with the highest classification accuracy and combined it with the "Read Until" API in order to perform a sequencing experiment where we used our model to successfully selectively sequence mitochondrial DNA.

Overall, by developing a new real-time selective sequencing method, we will not only alleviate the challenges caused by the additional steps during library preparation—we can also change the targeted regions during the experiment simply by modifying a parameter in the software. Our method has the potential to increase accuracy, speed up the sequencing process, and it can eventually be applied to any clinical settings where time-sensitive DNA sequencing is of the essence.

2. Methods

2.1 Data organization, preprocessing, and augmentation

For the purpose of training and testing our deep learning models, we used two publicly available nanopore sequencing datasets: (i) Jain et al. produced a human genome assembly using long reads from nanopore sequencing[30]. About 14 million reads were sequenced and aligned to the 1000 genome GRCh38 reference genome[31]. From this dataset, we used 60,000 reads that were aligned to the mitochondria and 200,000 random reads that were aligned to the rest of the human genome. (ii) The "Cliveome" dataset, which was sequenced by ONT and released to the public in 2016 [32]. From this dataset, we used 8,000 reads that mapped to the mitochondria as well as 200,000 random reads that mapped to the rest of the human genome. In each dataset we separated the sequenced reads randomly into training, validation, and test sets containing 80%, 10%, and 10%, respectively, of the total reads. Only the first portion of each raw signal were used to simulate reading the beginning of the molecule with the Read Until feature.

Deep learning requires iterating through the training dataset by mini-batches, which allows handling large datasets and improves the training results[33]. In this research we used the Pytorch[34] deep learning framework, which contains a Dataloader class; we customized this class to allow parallel data loading with custom data transformations. Our custom dataloader applies four transformations to the signal: the first transformation randomly selected a region of 2,000 values from the total 5,000 values. The second transformation changed the signal from the raw values, which represent the electric current level, to differential values in order to eliminate possible bias between voltages of different devices and flow cells. The third transformation cut the signal into a sliding window array, transforming the 1D-long linear signal into a 2D array of stacked sliding windows. The final transformation added Gaussian noise to the sample to mimic the background noise in nanopore sequencing. All of the transformations improved the training process and the final accuracy; further details are in the Supplementary Methods.

2.2 Model architecture, training, and testing

We decided to test 5 neural network varieties for our deep learning model architecture: regular CNN[35], very deep CNN (VDCNN)[36], regular LSTM[37], LSTM with recurrent batch normalization[38], and regular GRU[39]; further model details and justification for their selection are presented in the Supplementary Methods. All models were tested with three different sizes corresponding to the number of hidden parameters: large size, medium size, and small size models. All models were tested extensively with different configurations as explained in the supplementary methods section.

We also attempted to combine a CNN model with an RNN model whose schematic overview can be seen in Supplementary Figure 1. In theory, CNN is good at proximal feature representation and RNN can find long distance dependencies; by combining those techniques, our model could utilize both short- and long-distance information hidden in the raw signal[40]. We combined the VDCNN with regular GRU as well as VDCNN with LSTM with recurrent batch normalization and tested multiple configurations of these models as well as described in the supplementary methods section..

To eliminate any differences in model accuracy due to a different training process, the same python script was used to train all models similarly. We used the training dataset during training, the validation dataset for hyperparameter tuning, and the test dataset was used exclusively at the final stage to measure the accuracy of each model. Accuracy was measured separately for genomic reads and for mitochondrial reads, total accuracy was calculated by averaging the accuracy of the mitochondrial reads and the accuracy of the genomic reads. An Adam (A Method for Stochastic Optimization) optimizer[41] was used; the learning rate and other parameters for the optimizer were determined by a manual search. All models were trained for 300 epochs and the learning curve of each model was assessed to determine whether the loss curve plateaued and whether overfitting became an issue. Supplementary Figure 2 illustrates the learning curve of the model with LSTM and recurrent batch normalization as an example of a successfully trained model.

After training all models on the primary dataset, a second dataset (Cliveome) was used to test the models for generalization. At first, the accuracy of the models was tested on the test dataset from the second dataset without any additional training. Later, all models were trained for 30 epochs on the second dataset training data in order to improve the accuracy specifically for the second dataset (fine-tuning). After the additional training, all models were tested again with the second dataset and its accuracy was recorded.

2.3 DNA extraction, library preparation, and MinION sequencing

Monolayer-adherent HEK-293T cells (transformed human embryonic kidney cells, ATCC, USA) were grown in Dulbecco's modified Eagle's medium (DMEM) (Thermo Fisher Scientific, USA) supplemented with 10% (vol/vol) fetal bovine serum (FBS) (Thermo Fisher Scientific, USA), 0.3 g/liter L-glutamine, 100 unit/ml penicillin, and 100 units/ml streptomycin (Biological Industries, Israel). Cells were incubated at 37°C in 5% CO_2 atmosphere. Before use, cells were confirmed to have no mycoplasma contamination using the EZ-PCR Mycoplasma test kit (Biological Industries, Israel). Prior to each experiment, the cells were counted using the Countess automated cell counter (Thermo Fisher Scientific, USA).

Qiagen’s QIAamp DNA mini kit was used to extract DNA from HEK-293T cells. Next, 2.5 x 10^6 or 1 x 10^6 cells were centrifuged at 1,400 x g for 5 minutes, and the resulting pellet was resuspended in 200 µl PBS. DNA was then extracted according to the manufacturer’s protocol and eluted in 200 µl H_2O. The DNA concentration was measured using the dsDNA High Sensitivity assay on a Qubit fluorometer (Thermo Fisher Scientific, USA). DNA purity was assessed using a NanoDrop ND-1000 spectrophotometer (NanoDrop Technologies, Thermo Scientific, USA), to ensure OD 260/280 and OD 263/230 > 1.8.

Approximately 400 ng of purified DNA in a total volume of 7.5 µl in a 0.2 ml PCR tube was used as input for sequencing library preparation using Oxford Nanopore Technologies’ Rapid Sequencing kit (SQK-RAD004, version RSE_9046_v1_revB_17Nov2017) according to the manufacturer’s instructions. For fragmentation and transposase adapter attachment, 2.5 µl FRA was added to the DNA and mixed by inversion. The sample was then incubated at 30°C for 1 minute, followed by 80°C for 1 minute, and finally

cooled on ice. Sequencing adapters were then attached by adding 1 µl RAP to the mixture and mixing by inversion. The sample with sequencing adapters was incubated at room temperature for 5 minutes, and then stored on ice until it was ready for sequencing.

MinION sequencing was conducted according to the manufacturer's instructions using R9.4 and R9.4.1 rev. D flow cells (FLO-MIN106, ONT). After flow cell priming, 4.5 µl nuclease-free water, 34 µl sequencing buffer (SQB), and 25.5 µl mixed loading beads (LB) were added to the library and mixed by gently flicking the tube immediately before loading into the SpotOn port.

Three sequencing experiments were performed under those conditions; they will be referred to as the "HEK1", "HEK2", and "HEK3" runs. The results of the first two experiments were used for testing and training the model, whereas the third experiment was run in conjunction with the read-until script to perform real-time selective sequencing.

2.4 Sequencing data analysis

After data were acquired from the first two sequencing experiments, HEK1 and HEK2, the reads were translated to nucleotides using ONT Albacore version 2.2.5. Although Albacore is currently not supported, a recent comparison between base-calling software indicated that the differences between Albacore and more modern basecallers[42] are miniscule for the purposes of our experiments. The reads were mapped to the GRCh38 human reference genome using minimap2 software[43] version 2.11. Reads were separated into mitochondrial reads and genomic reads based on their mapping, and each group was separated into training/validation/testing groups with proportions of 80%/10%/10% of the total reads, respectively. Initially, the accuracy of the models trained on the first dataset was tested with the HEK1 data. Later, the models were trained for 30 epochs on the HEK1 data and accuracy was tested again (finetunned). The best performing model was determined by the highest accuracy value on the HEK2 data and saved for later use with read-until on the HEK3 sequencing experiment.

To test the performance of read-until, we utilized the developmental API provided by ONT and wrote a custom script to perform selective sequencing based on the "simple.py" file from the GitHub repository of Read-Until. This script receives the raw signal at the beginning of every DNA molecule, the raw signal is analyzed by the deep learning model, and finally the script sends a signal to the MinION device to either keep sequencing the DNA molecule or to stop and remove the unwanted DNA molecule from the pore. Reads that were classified by the model as mitochondrial reads were allowed to be fully sequenced, whereas the rest of the reads had received a signal to terminate their sequencing. In order to gather the validated results, we performed the experiments with 3 technical repeats for 3 different time spans: 10 minutes, 30 minutes, and 120 minutes. In each time span we performed 3 regular sequencing experiments without using read-until and 3 sequencing experiments utilizing Read Until. To account for the deterioration of the flow cell over time and to reduce technical bias, we performed the experiments with read-until and without it sparingly. The reads were translated and mapped to a human reference genome, then for each sequencing experiment the alignment statistics were collected. Logistic regression

with proportions and a random effects variable[44] analysis were performed to test for differences in the proportion of the sequenced mitochondrial nucleotides to the total sequenced nucleotides. A comparison was made between pairs of the technical repeats as follows: 10min_with_read-until_run1 VS 10min_without_read-until_run1, 10min_with_read-until_run2 VS 10min_without_read-until_run2, etc. Additionally, read lengths were collected for each of the experiments and analyzed using the Fisher-Pitman permutation test[45] to check for statistical differences between the read lengths of different groups.

3. Results

3.1 Deep Learning model selection

We trained 90 models in total while saving the accuracy statistics (see Supplementary Data File 1). More than half of the models exhibited total accuracy above 70% for all datasets after training. A summary of the results can be seen in Table 1. Larger models, generally, achieve higher accuracy than the smaller versions, as can be seen in Supplementary Data File 1. In addition, the models perform better after fine-tuning on a particular dataset (Table 1 as well as the rest of the results in Supplementary Data File 1). Furthermore, the addition of a dropout or a batch normalization layer generally improved the performance in all models. When comparing different architecture types, the RNN type models: regular LSTM, LSTM with BN and GRU achieved higher accuracy than the CNN-type networks: regular CNN and VDCNN, as seen in Table 1 with the total accuracy scores.

Table 1. Accuracy values of the best performing deep learning models as measured on different datasets.

Data from public datasets			Data from internal experiments				
Primary dataset (NA12878)	Secondary dataset (ONT), without fine-tuning	Secondary dataset (ONT), with fine-tuning	HEK1, without fine-tuning	HEK1, with fine-tuning	HEK2, without fine-tuning	HEK2, with fine-tuning on HEK1	Total Accuracy
56.61%	57.20%	66.30%	62.48%	63.10%	77.38%	55.00%	62.58%
88.70%	72.22%	72.83%	93.83%	77.67%	82.56%	74.79%	80.37%
82.40%	68.14%	73.61%	86.62%	86.26%	81.81%	81.73%	80.08%
89.42%	**82.98**%	87.52%	98.01%	**97.93**%	95.48%	**95.81**%	**92.45**%
94.04%	81.72%	83.64%	**98.61**%	94.55%	**96.13**%	93.60%	91.76%
94.71%	63.07%	**89.40**%	96.45%	97.60%	80.33%	91.70%	87.90%
88.37%	60.76%	77.94%	92.40%	95.39%	87.29%	84.20%	83.76%

Small/Medium/Large – refers to the size of the model, +D – dropout, +S – shortcut, +MP – max-pooling, +BN – regular batch normalization, +LS – last step taken from RNN, +HO – hidden output taken from RNN

3.2 Real-time selective sequencing with Read Until

Based on the previous results, we selected the LSTM with recurrent batch normalization model that achieved the highest accuracy of 95.81% with the HEK 2 data and the highest accuracy of 92.45% overall. The selected model was used in conjunction with Read-Until to perform real-time selective sequencing. The Read-Until script was configured to sequence only molecules that were classified as mitochondrial reads by the model. During sequencing HEK3, the accuracy of the model was above 90%, which corresponds to the accuracy measured on HEK1 and HEK2 without fine-tuning.

The enrichment factor of our method was measured by calculating the difference in the percentage of mitochondrial nucleotides between experiments with and without selective sequencing. This was carried out in order to normalize the samples for total sequencing output and to eliminate any variance due to its interchangability throughout the experiment. We achieved a normalized enrichment factor of 2.3X (p.val < 0.05, Figure 1). When we compared the averages of the mitochondrial nucleotides in all the experiments with and without selective sequencing (normalization or not), we achieved an enrichment factor of 1.34X. When we compared the means of the mitochondrial coverages (as shown in Table 2), we achieved an enrichment factor of 1.32X. Even though most of the molecules were classified as genomic by the deep learning model and should not have been sequenced, most of the molecules classified as genomic were sequenced and saved to the hard-drive (see the Discussion).

Table 2 summarizes the results of all the selective sequencing experiments. Table 2 shows the mean percentage of mitochondrial nucleotides for each time interval, averaged across three experiments performed for each time interval; in addition, the average read length is shown for all of the reads. When comparing the percentage of mitochondrial nucleotides between sequencing experiments with selective sequencing against experiments without selective sequencing, there was a significant difference (p.val < 0.05) between them.

Table 2. The mean coverage of mitochondria and the mean percentage of nucleotides aligned to mitochondria from all the sequenced nucleotides. Also presented are the mean enrichment factor based on percentage and the mean genomic and mitochondria read lengths, for each triplicate of experiments separated by the length of the experiment (the sequencing time), and whether Read Until (selective sequencing) was used.

Sequencing Time	Read Until	Mitochondria coverage	% Mitochondria nucleotides	Enrichment Factor	Genomic read lengths	Mitochondria read lengths
10 Minutes	No	3.26	0.088	2.3	6165	5965
	Yes	4.75	0.170		3450	4563
30 minutes	No	12.95	0.104	1.38	7598	8088
	Yes	13.78	0.143		5106	6382
120 minutes	No	37.6	0.112	3.12	8089	7570
	Yes	53.98	0.269		2488	4319

In addition to differences in the percentage of nucleotides, we examined alterations in read lengths: there was a clear distinction in the read length between the groups. There was no significant variances between the read lengths of the mitochondrial and genomic reads in experiments without selective sequencing (p.val>0.8, difference between means=~75). However, there was a trend towards mitochondrial reads being longer than genomic reads in experiments with selective sequencing (p.val<0.1, diff=~1400). When comparing the read lengths of the mitochondrial reads in experiments with selective sequencing to those without it, the mitochondrial reads with selective sequencing were shorter than mitochondrial reads from experiments without selective sequencing (p.val<0.005, diff=~2100). Larger and more significant difference was observed between the genomic read lengths in experiments with selective sequencing and those without it; the genomic read lengths were significantly longer in experiments without selective sequencing (p.val<0.0005, diff=~3600).

4. Discussion

4.1 The process and the results of the deep learning model training

When we examined the results of the deep learning model training, the overall high accuracy (>70%) of most of the models indicated that deep learning in general might be an appropriate solution for read

classification based on raw signals. The higher accuracy of larger models is an expected outcome because larger networks have more weight that could be adjusted during the training process and could possibly capture more variability of the data[46]. However, the smaller regular CNN model, which performed better than the medium and large CNN models, is surprising. This could be explained by comparing all of the CNN networks; the larger networks performed well (>90% accuracy) on some datasets but on other datasets the larger models would either over-fit or would not train at all. However, the smaller CNN network had much lower accuracy but performed similarly across all datasets; therefore, the smaller network had a higher total accuracy. This is possibly due to the relative simplicity of a CNN model and the fact that smaller CNN models have fewer weights to train; thus, smaller CNN models have to generalize better than the larger models[47].

VDCNN expectedly outperformed a regular CNN, as was shown in the original paper[36]. Another expected result is the RNN-type architectures (regular LSTM, LSTM with recurrent batch normalization, and GRU), which outperformed the CNN-type architectures. The data in our study could be described as a sequential input, which is the type of data that RNN architecture was designed to analyze[6]. However, we observed that the average accuracies of regular LSTM and VDCNN are similar, which can be explained by the relative simplicity of the one-layered LSTM model against the more complex VDCNN with 17 layers.Even though we expected the combination of the CNN + RNN model to outperform each type individually, based on the fact that convolutional networks are useful for feature extraction[48], and when used in conjunction with RNN, it could produce better results[49]. In our case, the combination of CNN + RNN produced results similar in accuracy to those of LSTM with recurrent batch normalization. These results could be explained either by the very optimal training of LSTM with a recurrent batch normalization model or the sub-optimal training of the CNN + RNN models.

Fine-tuning the models on a small portion of the dataset before analyzing the rest of the dataset improved the results for some models, as seen in Table 1. Each dataset was acquired from a different sequencing experiment and possibly various variables could affect the raw signal such as different chemistry kits, different MinION devices, different library preparation protocols, and different sample qualities. Therefore, by fine-tuning the model to each experiment, we increased the model's accuracy for those specific conditions.

Dropout and batch normalization improved the performance of most models as was expected, based on their contribution to the training process of the deep learning models[6,50]. In addition, the results after training models before the addition of the difference transformation to the raw input were dire; overfitting was a big problem before adding artificial noise, which is known to help with the training of the deep learning models[51]; therefore, those two transformations were applied to the training of all models.

The mechanism by which the deep learning models perform the classification remains unknown, the model could either "simply remember" the relatively short sequence of the mitochondria (only 16.5K nucleotides) and can determine which reads originate from this sequence; or, the models could extract specific features from the reads such as GC content/ k-mer content and more complex features such as

the protein sequence and structure or DNA methylations and, during training, learn which features are present in genomic sequences and which features are present in mitochondrial sequences. We also postulated that the models could have learned more sophisticated features of the mitochondrial DNA, such as a different encoding codond,or the density of the genetic information[52]. Furthermore, deep learning models have been shown to successfully detect circular plasmids, based on their sequencing data, by examining larger chunks of the plasmid sequences achieved by longer reads as well as additional genomic features[53], information that could contribute to a successful classification. We think a thorough analysis of a trained deep learning model from this work, as was done for the visual analysis models[48], could provide useful insights for further research in this field and perhaps new biological features that were not considered important before would be discovered.

4.2 Real-time selective sequencing

Our experiments, which utilized the ability of MinION to perform selective sequencing combined with a deep learning model, demonstrated from several different angles the validity of this method. From the aspect of classification accuracy, our deep learning achieved >90% accuracy in real time which is similar to the results during training and testing on the previous data. Even though we had some variance in the mitochondrial proportions between the experiments, which were caused by the relatively small amount of mitochondrial DNA present in the sample, when statistically examining the mitochondrial nucleotide proportions, the results show significant differences between experiments with and without selective sequencing, thus indicating that our method worked successfully. Also, the differences in the read lengths of the mitochondrial and genomic reads between the different experiments also support that executing the selective sequencing script prioritized the mitochondrial reads over the genomic reads during sequencing.

To assess the results of our selective sequencing script, we can calculate the expected results in a theoretically perfect hardware-software configuration (where each read that was marked for rejection would not have been sequenced): with a 90% accuracy model ( ) and samples where 0.1% of reads are mitochondrial (similar to our samples). We can calculate this theoretical expected mitochondrial percentage using the following formula (the expected “true” mitochondrial reads divided by genomic reads falsely classified as mitochondrial reads):

$$M_{exp} = \frac{\left(M_{samp} \times Acc\right)}{(1 - M_{samp}) \times (1 - Acc)}$$

From this calculation, we could infer that with selective sequencing in theoretically perfect conditions, we should achieve 0.9% mitochondrial reads; therefore, we would achieve an enrichment of 9X when using selective sequencing and with perfect software-hardware performance. In our experiments achieved an enrichment of 2.3X, demonstrating that the hardware is working but also that there is still much room for

optimization in terms of software-hardware interaction with the Read-Until feature of nanopore sequencing.

Statistical analysis of the difference between the percentages of mitochondrial nucleotides sequenced with selective sequencing and those without it revealed a significant difference. The fact that there was a smaller difference between the percentages of mitochondrial reads in experiments with and without selective sequencing when compared to the differences in the genomic reads with and without selective sequencing also supports the idea that selective sequencing prioritizes the mitochondrial reads. Our claim that our selective sequencing method allowed us to sequence more mitochondrial sequences is further supported when we combined the differences in the raw mitochondrial nucleotide counts without normalization, which showed that more mitochondrial sequences are sequenced in experiments with selective sequences, Therefore, even in its current state, our approach could assist researchers in achieving better coverage of a certain region and theoretically save time, resources, and budget by requiring less sequencing to a achieve a similar goal. Furthermore, theoretically, it is possible to change the classification model to target a different region in the genome, thus increasing the utility of this selective sequencing method. We can conclude that the genomic reads were shorter in experiments with selective sequencing probably because our script sent signals to the MinION device to stop sequencing the reads that were classified as genomic, thus non-mitochondrial reads would be shorter than in experiments without the stopping signal. We currently do not know why the signals sent to the hardware to stop sequencing did not entirely prevent the sequencing of reads classified as genomic. However, other studies reported a delayed ejection of unwanted DNA molecules,[28] which shows that the software-hardware interaction is not perfect and could cause “rejected” DNA to be sequenced and saved.

5. Conclusion

From the results of the deep learning models training, we can conclude that the deep learning approach is a valid choice for classifying sequenced reads based on the first 2,000 values of raw signal of the read. There might be better models than those we tested here; however, even using our relatively simple and straighforward approach, when we tested different datasets we achieved good results in terms of accuracy and generalization. Furthermore, for the first time, we showed the ability of deep learning models to classify whole reads based on raw nanopore signals.

The selective sequencing experiments we performed with our script, using the best deep learning model from the previous steps, produced enough evidence to conclude that our script prioritized mitochondrial reads over genomic reads. The deep learning model classified the reads correctly while they were being sequenced; an analysis of the proportion of mitochondrial DNA and the differences in the read lengths revealed that the mitochondrial reads were being prioritized during sequencing and were enriched by a factor of 2.3X.

When combining the results from both parts of the experiment, we concluded that real-time selective sequencing is possible by using deep learning models to analyze the raw signal at the beginning of each

read. It is possible to develop better models to perform classification with increased accuracy; however, the main improvement will come from optimizing the selective sequencing script.

We believe that our findings provide a solid basis for building upon for future research in this field. We hope that our deep learning models will serve as a building block for studies on improving the analysis of raw signals. The discrepancies between our predicted performance and the actual results indicate that further research should be carried out to determine which parts of the selective sequencing scripts and parameters must be optimized. Our method could spark interest in improving the selective sequencing optimization to provide a much better procedure. Our study might also serve as an alternative to other selective sequencing methods.

Abbreviations

CNN – Convolutional neural network

RNN – Recurrent neural network

LSTM – Long short-term memory

GRU – Gated recurrent unit

Declarations

Data availability

The sequencing data produced in this study is available at NCBI's SRA with accession number PRJNA689046.

Code availability

The code used for this study is provided for non-commercial use at: https://github.com/nshomron/Nanopore_Deep_Learning

Author contributions statement

Artem Danilevsky conceived, designed, and implemented the computational part of the project. Avital Polsky developed an optimized sequencing library workflow and implemented the experimental part of the project. Noam Shomron conceived, designed, led the project, and wrote the manuscript.

Additional Information

Artem Danilevsky, Avital Luba Polsky, and Noam Shomron declares that they have no conflict of interest.

Acknowledgments

We thank Dr David Golan, Prof Lior Wolf, and Tzviel Frostig from Prof Yoav Benjamini's laboratory for consultations. We thank the Gertner Institute, the Zimin grant and the Djerassi-Elias Institute of Oncology for financial support. This study was supported in part by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University

References

1. Goodwin, S., McPherson, J. D. & McCombie, W. R. Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.* **17**, 333–351 (2016).
2. Jain, M., Olsen, H. E., Paten, B. & Akeson, M. The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community. *Genome Biol.* **17**, 239 (2016).
3. Rang, F. J., Kloosterman, W. P. & de Ridder, J. From squiggle to basepair: computational approaches for improving nanopore sequencing read accuracy. *Genome Biol.* **19**, 90 (2018).
4. Rand, A. C. *et al.* Mapping DNA methylation with high-throughput nanopore sequencing. *Nat. Methods* **14**, 411–413 (2017).
5. Simpson, J. T. *et al.* Detecting DNA cytosine methylation using nanopore sequencing. *Nat. Methods* **14**, 407–410 (2017).
6. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
7. Angermueller, C., Pärnamaa, T., Parts, L. & Stegle, O. Deep learning for computational biology. *Mol. Syst. Biol.* **12**, 878 (2016).
8. Ching, T. *et al.* Opportunities and obstacles for deep learning in biology and medicine. *J. R. Soc. Interface* **15**, (2018).
9. Çakır, E., Parascandolo, G., Heittola, T., Huttunen, H. & Virtanen, T. Convolutional Recurrent Neural Networks for Polyphonic Sound Event Detection. *IEEEACM Trans. Audio Speech Lang. Process.* **25**, 1291–1303 (2017).
10. Lee, H., Pham, P., Largman, Y. & Ng, A. Y. Unsupervised feature learning for audio classification using convolutional deep belief networks. in *Advances in Neural Information Processing Systems 22* (eds. Bengio, Y., Schuurmans, D., Lafferty, J. D., Williams, C. K. I. & Culotta, A.) 1096–1104 (Curran Associates, Inc., 2009).
11. Huang, P., Kim, M., Hasegawa-Johnson, M. & Smaragdis, P. Deep learning for monaural speech separation. in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 1562–1566 (2014). doi:10.1109/ICASSP.2014.6853860.
12. David, M., Dursi, L. J., Yao, D., Boutros, P. C. & Simpson, J. T. Nanocall: an open source basecaller for Oxford Nanopore sequencing data. *Bioinformatics* **33**, 49–55 (2017).
13. Timp, W., Comer, J. & Aksimentiev, A. DNA Base-Calling from a Nanopore Using a Viterbi Algorithm. *Biophys. J.* **102**, L37–L39 (2012).
14. Teng, H. *et al.* Chiron: translating nanopore raw signal directly into nucleotide sequence using deep learning. *GigaScience* **7**, (2018).

15. Boža, V., Brejová, B. & Vinař, T. DeepNano: Deep recurrent neural networks for base calling in MinION nanopore reads. *PLOS ONE* **12**, e0178751 (2017).
16. Ni, P. *et al.* DeepSignal: detecting DNA methylation state from Nanopore sequencing reads using deep-learning. *Bioinformatics* **35**, 4586–4595 (2019).
17. Li, Y. *et al.* DeepSimulator: a deep simulator for Nanopore sequencing. *Bioinformatics* **34**, 2899–2908 (2018).
18. Ng, S. B. *et al.* Targeted capture and massively parallel sequencing of 12 human exomes. *Nature* **461**, 272–276 (2009).
19. Gnirke, A. *et al.* Solution hybrid selection with ultra-long oligonucleotides for massively parallel targeted sequencing. *Nat. Biotechnol.* **27**, 182–189 (2009).
20. Tewhey, R. *et al.* Microdroplet-based PCR enrichment for large-scale targeted sequencing. *Nat. Biotechnol.* **27**, 1025–1031 (2009).
21. Karamitros, T. & Magiorkinis, G. Multiplexed Targeted Sequencing for Oxford Nanopore MinION: A Detailed Library Preparation Procedure. in *Next Generation Sequencing: Methods and Protocols* (eds. Head, S. R., Ordoukhanian, P. & Salomon, D. R.) 43–51 (Springer, 2018). doi:10.1007/978-1-4939-7514-3_4.
22. Gabrieli, T. *et al.* Selective nanopore sequencing of human BRCA1 by Cas9-assisted targeting of chromosome segments (CATCH). *Nucleic Acids Res.* **46**, e87 (2018).
23. Mertes, F. *et al.* Targeted enrichment of genomic DNA regions for next-generation sequencing. *Brief. Funct. Genomics* **10**, 374–386 (2011).
24. Loose, M., Malla, S. & Stout, M. Real-time selective sequencing using nanopore technology. *Nat. Methods* **13**, 751–754 (2016).
25. Edwards, H. S. *et al.* Real-Time Selective Sequencing with RUBRIC: Read Until with Basecall and Reference-Informed Criteria. *Sci. Rep.* **9**, (2019).
26. Payne, A. *et al.* Nanopore adaptive sequencing for mixed samples, whole exome capture and targeted panels. *bioRxiv* 2020.02.03.926956 (2020) doi:10.1101/2020.02.03.926956.
27. Maio, N. D. *et al.* BOSS-RUNS: a flexible and practical dynamic read sampling framework for nanopore sequencing. *bioRxiv* 2020.02.07.938670 (2020) doi:10.1101/2020.02.07.938670.
28. Kovaka, S., Fan, Y., Ni, B., Timp, W. & Schatz, M. C. Targeted nanopore sequencing by real-time mapping of raw electrical signal with UNCALLED. *Nat. Biotechnol.* **39**, 431–441 (2021).
29. Chen, R. *et al.* Comparison of whole genome sequencing and targeted sequencing for mitochondrial DNA. *Mitochondrion* (2021) doi:10.1016/j.mito.2021.01.006.
30. Jain, M. *et al.* Nanopore sequencing and assembly of a human genome with ultra-long reads. *Nat. Biotechnol.* **36**, 338–345 (2018).
31. Consortium, T. 1000 G. P. A global reference for human genetic variation. *Nature* **526**, 68–74 (2015).
32. Clive Brown. ONT-HG1. (2017) doi:10.5281/zenodo.1318628.
33. Masters, D. & Luschi, C. Revisiting Small Batch Training for Deep Neural Networks. (2018).

34. Paszke, A. *et al.* Automatic differentiation in PyTorch. (2017).
35. Krizhevsky, A., Sutskever, I. & Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. in *Advances in Neural Information Processing Systems 25* (eds. Pereira, F., Burges, C. J. C., Bottou, L. & Weinberger, K. Q.) 1097–1105 (Curran Associates, Inc., 2012).
36. Conneau, A., Schwenk, H., Barrault, L. & Lecun, Y. Very Deep Convolutional Networks for Text Classification. *ArXiv160601781 Cs* (2016).
37. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Comput*.**9**, 1735–1780 (1997).
38. Cooijmans, T., Ballas, N., Laurent, C., Gülçehre, Ç. & Courville, A. Recurrent Batch Normalization. *ArXiv160309025 Cs* (2016).
39. Cho, K. *et al.* Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *ArXiv14061078 Cs Stat* (2014).
40. Sainath, T. N., Vinyals, O., Senior, A. & Sak, H. Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks. in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* 4580–4584 (2015). doi:10.1109/ICASSP.2015.7178838.
41. Kingma, D. P. & Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv14126980 Cs* (2014).
42. Wick, R. R., Judd, L. M. & Holt, K. E. Performance of neural network basecalling tools for Oxford Nanopore sequencing. *Genome Biol.***20**, 129 (2019).
43. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* doi:10.1093/bioinformatics/bty191.
44. Jaeger, T. F. Categorical Data Analysis: Away from ANOVAs (transformation or not) and towards Logit Mixed Models. *J. Mem. Lang.***59**, 434–446 (2008).
45. Boik, R. J. The Fisher-Pitman permutation test: A non-robust alternative to the normal theory F test when variances are heterogeneous. *Br. J. Math. Stat. Psychol.***40**, 26–42 (1987).
46. Simonyan, K. & Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *ArXiv14091556 Cs* (2014).
47. Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. Understanding deep learning requires rethinking generalization. *ArXiv161103530 Cs* (2016).
48. Zeiler, M. D. & Fergus, R. Visualizing and Understanding Convolutional Networks. *ArXiv13112901 Cs* (2013).
49. Ordóñez, F., Roggen, D., Ordóñez, F. J. & Roggen, D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors***16**, 115 (2016).
50. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *ArXiv150203167 Cs* (2015).
51. Neelakantan, A. *et al.* Adding Gradient Noise Improves Learning for Very Deep Networks. *ArXiv151106807 Cs Stat* (2015).
52. Taanman, J.-W. The mitochondrial genome: structure, transcription, translation and replication. *Biochim. Biophys. Acta BBA - Bioenerg.***1410**, 103–123 (1999).

53. Andreopoulos, W. B. *et al.* Deeplasmid: Deep learning accurately separates plasmids from bacterial chromosomes. *bioRxiv* 2021.03.11.434936 (2021) doi:10.1101/2021.03.11.434936.

Figures

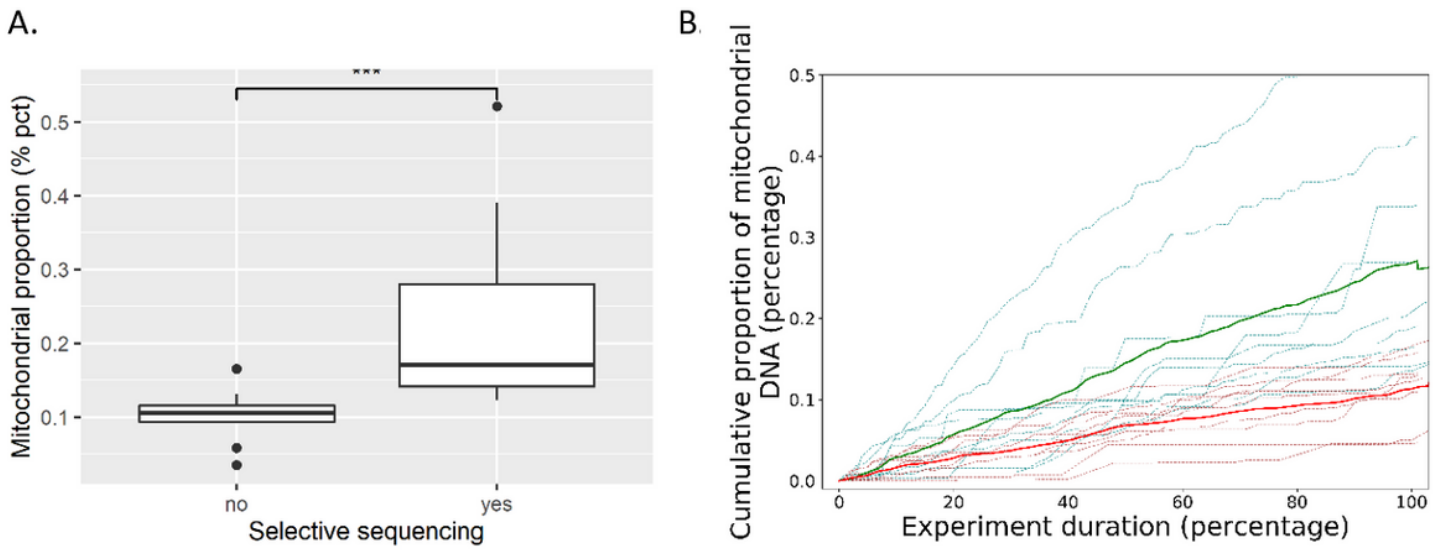

Figure 1

Differences in the percentage of mitochondrial nucleotides between experiments with selective sequencing and without it. A. Box plot illustrating differences in the mitochondrial reads between experiments with and without selective sequencing. B. Cumulative percentage of mitochondrial DNA in relation to the final amount of total DNA throughout the experiments, different timeframes adjusted to a scale of 0% to 100% of the experiment's duration. Green and red solid lines denote the mean percentage of mitochondrial DNA throughout all experiments with and without selective sequencing, respectively. Light green and light red dotted lines denote the percentage of mitochondrial DNA throughout individual experiments with and without selective sequencing, respectively.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryMaterialSupplementaryMethods.docx