Consensus Calling and Validation of Single Nucleotide Variant Calling from Nanopore Sequencing with Deep Learning for CyclomicsSeq

Inez Chiara den Hond
i.c.denhond@students.uu.nl
Student number: 6149545
Major Research Project
December 16th, 2022
Master Bioinformatics & Biocomplexity
Universiteit Utrecht

Supervisors: Li-Ting Chen MSc, Dàmi Rebergen MSc, dr. Myrthe Jager
Examiner: dr. Jeroen de Ridder, Center for Molecular Medicine, UMC Utrecht
Second examiner: dr. Gijs van Haaften, Departement of Genetics, UMC Utrecht

Abstract
Cell-free DNA (cfDNA) are small (145 bp) DNA fragments that reside in the human circulation and other bodily fluids. cfDNA is derived from both healthy and tumour cells, in which case it is called circulating tumour DNA (ctDNA). ctDNA harbours genetic and epigenetic characteristics from the tumour genome and is therefore a valuable source of information. Cancer liquid biopsies target these fragments, but as the ctDNA concentration in the blood can be very low, sensitive sequencing technologies are needed. CyclomicsSeq is a novel sequencing technique that uses rolling circle amplification to generate multiple copies of a cfDNA fragment in a long concatemer. They are sequenced with Nanopore sequencing and the consensus sequences for all cfDNA fragments can be used to detect tumour variants and to infer the tumour fraction. The sequencing error rate for Nanopore sequencing is still quite high, but with consensus calling of the multiple cfDNA copies, random sequencing errors are removed. However, systematic sequencing errors remain even with consensus calling. Here, we present a deep learning model that is trained on Nanopore sequencing data. The model can perform accurate consensus calling for CyclomicsSeq and can find tumour variants in ctDNA fragments at a low variant allele frequency.

Layman summary
In each cell of our body, our genetic information is stored in our DNA. When a patient has cancer, the DNA of the tumour cells contain mutations. These are small changes in the DNA that disrupt the normal cellular processes. By sequencing the tumours DNA from the cancer patient, mutations can be found, and this can help in deciding how to treat the disease or to inform about the disease progress. DNA in the tumour is usually not easily accessible for surgery as the tumour can be located deep in the body. Another way to sequence DNA is with cancer liquid biopsies from human blood. In the blood stream of a cancer patient, DNA is present that is released by both healthy and tumour cells. This DNA is called cell-free DNA (cfDNA). The DNA that originates from the tumour bears the same cancerous mutations as the DNA in the tumour cell itself. Liquid biopsies thus provide a non-invasive way to investigate the tumour genome of a cancer patient. CyclomicsSeq is a sequencing technique developed to sequence cfDNA to find tumour mutations. It uses Nanopore sequencing, a sequencing technique that sequences long reads, is relatively inexpensive, and fast compared to other sequencing techniques. Problematically, Nanopore sequencing has a relatively high error rate. CyclomicsSeq therefore sequences the same DNA fragment multiple times and uses the consensus sequence from these copies as the sequence of the input fragment. In this way, sequencing errors can be removed. However, some sequencing errors still persist after this consensus calling step. Therefore, in this thesis we present a deep learning approach that performs this consensus calling step. Deep learning is a form of artificial intelligence, where a model learns from input data how to perform a certain task. If you thereafter ask the model to perform a similar task, it will have learned how to perform this task. By learning the model how to perform correct consensus calling on Nanopore sequencing data, we can use the model to perform the consensus calling step of CyclomicsSeq. We showed that the model performs better consensus calling than the current method that CyclomicsSeq uses. With these consensus sequences of the DNA fragments, we can trace back the mutations in the DNA fragments that originated from the tumours. This consensus calling method is not only useful for CyclomicsSeq but can be used for any DNA that has been sequenced with Nanopore sequencing.

Introduction
Cancer liquid biopsies provide a non-invasive manner for cancer diagnosis, prognosis, or measurement of treatment resistance, in contrast to solid biopsies (Figure 1). Cancer liquid biopsies target cell-free DNA (cfDNA) that is present in the human plasma and other bodily fluids. cfDNA are short DNA

fragments (mainly around 167bp) that end up in the plasma. The biogenesis of cfDNA is mainly through apoptosis and necrosis of cells, but a lot remains unknown about this process. Active release mechanisms have been suggested as well. Tumour cells also shed cfDNA into the circulation, which is then called circulating tumour DNA (ctDNA). Only a small fraction of the total cfDNA pool consists of ctDNA, which complicates detection and somatic variant calling. ctDNA contain valuable information about the genetic and epigenetic properties of the tumour cells. Therefore, accurate sequencing technologies are a requisite to detect and sequence ctDNA in cancer liquid biopsies (1–4).

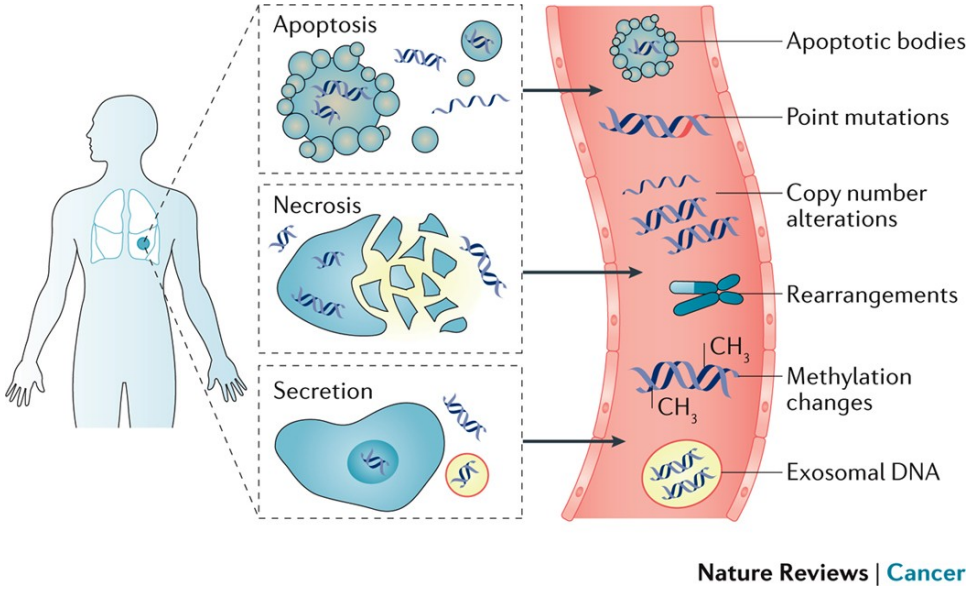


Figure 1 – Cell-free DNA (cfDNA) in the bloodstream. cfDNA can end up in the circulation via multiple mechanisms (apoptosis, necrosis, active secretion) and harbours genetic (e.g. mutations) and epigenetic (e.g. methylation) marks. Adapted from (4).

The last decade, third generation sequencing (TGS) technologies have advanced rapidly. These techniques have enabled long-read sequencing to an extent that had not been thought possible, which has led to numerous breakthroughs in the genomics field. For example, the telomere to telomere (T2T) genome has been completed with long-read sequencing (5,6), whereas the repetitive regions that were incomplete in the previous genome assembly were inaccessible for next generation sequencing (NGS) techniques like Illumina sequencing, that sequences reads up to 300 bp. Illumina sequencing is the current gold standard sequencing technique and provides high read accuracy (>99.9%) (7). Oxford Nanopore Technologies (ONT) and Pacific Biosciences (PacBio) are two major players in the field of long-read sequencing and have overcome these limitations of next generation sequencing in different ways (8).

Single molecule, real-time (SMRT) sequencing has been developed by PacBio. By replication of a DNA molecule that has been circularized with fluorescent labelled nucleotides, the sequence can be recorded multiple times and a consensus sequence is created, which is called circular consensus sequencing (CCS). Raw PacBio reads have a higher error rate (13-15%) than Illumina sequencing but the reads have mostly random errors. The CCS sequences are therefore highly accurate (>99%) (9–13).

Nanopore sequencing, developed by ONT, is the other revolutionary sequencing technique in the field of long-read sequencing. Nanopore sequencing makes use of a nanoscopic pore in a membrane with an electrical gradient where a single-stranded DNA molecule can pass through (Figure 2). When the DNA molecule passes through the pore, small changes in the ion flow result in squiggles that can be translated to a nucleotide sequence. Different nucleotides affect the ion current differently. As the pore accommodates multiple nucleotides at the same moment, the shift in signal is not determined by a

single base but by a kmer of nucleotides (13,14). Some major advantages of Nanopore sequencing over other long-read sequencing techniques are its portability and a low initial cost. The Nanopore sequencing machine fits in a hand palm and works with a normal laptop via USB. One of the unique features of Nanopore sequencing is that the DNA can be sequenced directly without the need for DNA synthesis, as is the case with most NGS techniques (13,15).

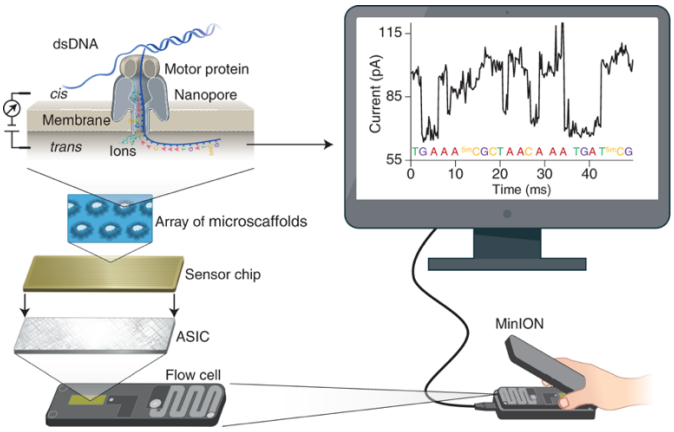


Figure 2 – Technology of Nanopore sequencing with a MinION sequencing device. dsDNA is unwound by a motor protein. single stranded DNA travels through the nanopore that is embedded in a membrane with an electrical gradient. The bases in the pore affect the electrical current, which is measured. Base calling converts the electrical signal to a DNA sequence. Adapted from (13).

Base calling algorithms translate the raw Nanopore signals to nucleotide sequences and form an integral part in the Nanopore sequencing workflow (16,17). Many base calling algorithms have been developed the past decade, both by ONT and by independent researchers (17,18). Currently, Guppy is the most widely used base caller and has been developed by ONT (13).

The past decade, improvements in both pore chemistry and of the base calling algorithms have improved accuracy of Nanopore sequencing. The newest pore release is R10, and the major difference with previous pores is that the R10 pore detects the sequence at two places inside the pore instead of one. Especially for homopolymers, stretches of single bases that are hard to sequence accurately with Nanopore sequencing, the R10 shows improved performance according to ONT (13,19,20).

Despite these advances, Nanopore sequencing comes with some troubles. The R9.4 pore still produces reads with an error rate of 6-15% (13). Homopolymers complicate accurate sequencing and base calling (21), as the signal does not change throughout the homopolymeric region of the DNA molecule. The length of the region could be inferred from the time of the signal, but the time that each kmer resides in the pore is not constant (16). Different publications have shown that systematic errors are introduced in reads produced by Nanopore sequencing and that they are dependent on sequence context, which differs between the forward and revere strand. These errors can arise either in homopolymers or in other genomic regions (22–24).

CyclomicsSeq is a novel sequencing technique based on Nanopore sequencing, developed to sequence cell-free DNA (2). The ctDNA fraction of the cfDNA pool is dependent on the tumour load and ranges from <0.1% to more than 90% (1). This makes it a challenge to accurately detect variants on single ctDNA molecules. CyclomicsSeq uses multiple copies of a single DNA molecule to generate an accurate consensus sequence. Currently, CyclomicsSeq is targeted at the *TP53* gene for disease progression prognosis for head and neck cancer. Other targeted as well as untargeted (genome-wide) approaches are currently under development. CyclomicsSeq uses rolling circle amplification of a cfDNA fragment, to produce a long DNA molecule with multiple copies of the same fragment. These

copies are then processed by a consensus calling pipeline, resulting in highly accurate consensus sequences of the cfDNA molecules. Single nucleotide variants with a variant allele frequency up to 0.02% can be detected with CyclomicsSeq at the time of publication (2). An important feature of the CyclomicsSeq reads are the number of copies used to generate the consensus sequence. Most CyclomicsSeq reads have between 3 and 10 repeats. However, it is seen that more copies improve read accuracy with the current consensus calling method of CyclomicsSeq. Figure 3 shows how the consensus calling process removes random sequencing errors from the consensus sequences. Systematic sequencing errors that are present in each copy persist and interfere with variant calling. Improvement of the consensus calling method may lead to more accurate consensus calling and thus reliable variant detection.

Many polishing methods have been developed to polish Nanopore sequences after assembly of reads, mainly intended for genome assembly but for consensus calling as well. However, systematic sequencing errors are not always resolved with these methods (25). Currently error correction depends mostly on complementing the assembly with short-read sequencing but this approach has limited potential for repetitive regions of the genome (15). Consensus calling methods are often based on deep learning algorithms.

Deep learning algorithms are a type of machine learning where the computer learns to perform specific tasks or predictions from large datasets. Deep learning algorithms are neural networks, and they consist of an input and output layer with hidden layers in between. Progression of the data through the layers allows the model to form connections and find patterns in the data. The output layer performs the final prediction task. As neural networks are supervised classifiers that make a prediction based on the input data, labels must be available for the input dataset. The model uses the labels during training to optimize its performance on the prediction task. Many types of neural networks exist. All types have an input and output layer, but the layers in between vary. The simplest form of a neural network is a Dense Neural Network (DNN), and it contains hidden layers and dropout layers. Dropout layers are placed in between the hidden layers and randomly drop a fraction of the connections between two subsequent fully connected hidden layers. This forces the model to learn from slightly different connections each time and provides a form of regularization. Another type of neural network, often used in image classification, is a Convolutional Neural Network (CNN). A CNN consists of convolutional layers and can also contain dropout layers. The convolutional layer uses filters that slide over the input data and learn representations of features and patters in the data. For example, in an image, a filter may recognize the ears while another filter may recognize the nose of a person. CNNs are thought to work well with data that is spatially organized. By training a neural network on input data, the train set, the model learns to perform its tasks. A separate dataset, the test set, is used to assess the model's performance on data the model has not seen before. Thereafter, the model can be used to perform its prediction tasks (26,27).

ONT has developed its own consensus calling and variant calling tool Medaka, which uses a pileup of the reads against a draft assembly and works with a recurrent neural network (RNN). A RNN uses a sequential representation of the data. Medaka is primarily intended for genome assembly and polishing but may be used for single sequence consensus calling as well (25,28).

Recently, DeepMind has developed a sequence correction algorithm (DeepConsensus) for PacBio Circular Consensus Sequencing (29). DeepConsensus is a deep learning model with a transformer architecture. This type of architecture uses self-attention and can capture both long- and short-range interactions in a sequence. The past few years self-attention has found its way from computer science to biology, for example with the Enformer model, that predicts gene expression from DNA sequence (30), or Alphafold, that predicts protein structure from an amino acid sequence (29,31).

To our knowledge, DeepConsensus has not been adjusted to work for Nanopore data, as it is currently specific to PacBio data.

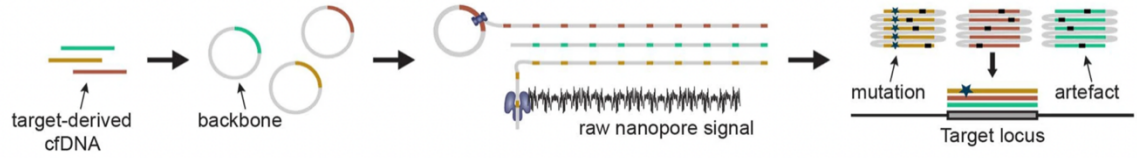


Figure 3- CyclomicsSeq workflow. cfDNA (from either a targeted amplicon or genome wide) is circularized with a DNA backbone sequence and then amplified. In this way, multiple copies of the same fragments are sequenced with Nanopore. Consensus calling of the copies of each original input molecule leads to an accurate consensus sequence. Mutations and systematic sequencing errors are present in every copy of a cfDNA read and persist in the consensus sequence. Mutation is shown with an asterisk. Adapted from (2).

In this thesis, we present a deep learning approach to generate consensus reads of Nanopore sequencing data for CyclomicsSeq (Figure 4). The model will be trained on Nanopore sequencing data and will thus learn to distinguish Nanopore sequencing errors from true variants. With this, it can avoid making systematic errors in the consensus reads. With more accurate consensus reads, mutations in a single ctDNA molecule can be detected more accurately. We train a dense neural network (DNN) and a convolutional neural network (CNN) on Nanopore sequencing data from the CHM13 cell line. The Nanopore sequencing reads will be down sampled to mimic the multiple copies of a CyclomicsSeq fragment. These models will then be used for consensus calling of cfDNA sequenced by CyclomicsSeq, both targeted at the *TP53* gene and genome wide. Finally, we will validate the ability of the models to find single nucleotide variants.

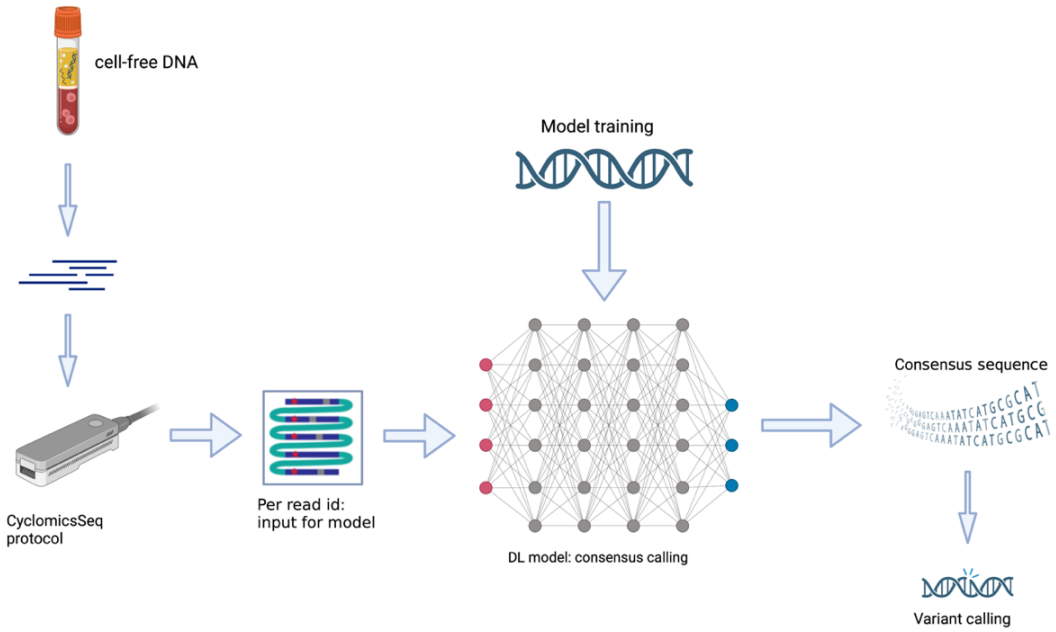


Figure 4 – Workflow presented in this thesis. cfDNA is isolated from the blood and amplified with rolling circle amplification. The concatemers are sequenced by Nanopore, generating multiple copies of the same cfDNA read. These copies are aligned to the human genome and then used as input for a deep learning model to generate a consensus sequence for the original cfDNA read. The model is trained beforehand on DNA of the CHM13 cell line sequenced with Nanopore. Reads are down sampled so models can be trained on different coverage bins. Created with BioRender.com

Results

Performance of DNN and CNN T2T models on train set from chromosome 18

We have trained dense neural networks (DNN) and convolutional neural networks (CNN) on a 300,000 bp region of chromosome 18 from DNA that has been sequenced with Nanopore by the Telomere-to-Telomere (T2T) consortium. The DNA is human genomic DNA from the CHM13hTERT cell line with a haploid genome. A haploid genome guarantees that there is always only 1 allele for all somatic chromosomes. From the reads that map to the forward strand of this region, we have created a sample for each position. A sample consists of the reads that align to the position of interest of a nine base pair window (Figure 5). The rows with sequences are randomly placed throughout the sample, and when the coverage is lower than 20X, the remaining rows are filled with zeroes. The reference sequence is also supplied, but the reference base at the position of interest is left out so the model does not directly learn the reference base for each sample. With each sample, the reference base is stored as the label. The models can thus only predict one of the four canonical bases and cannot predict an insertion or a deletion. We trained five DNNs and five CNNs, each time on a different coverage of the reads. The coverage bins are 3-5X, 6-10X, 11-15X, 16-20X, and 3-20X.

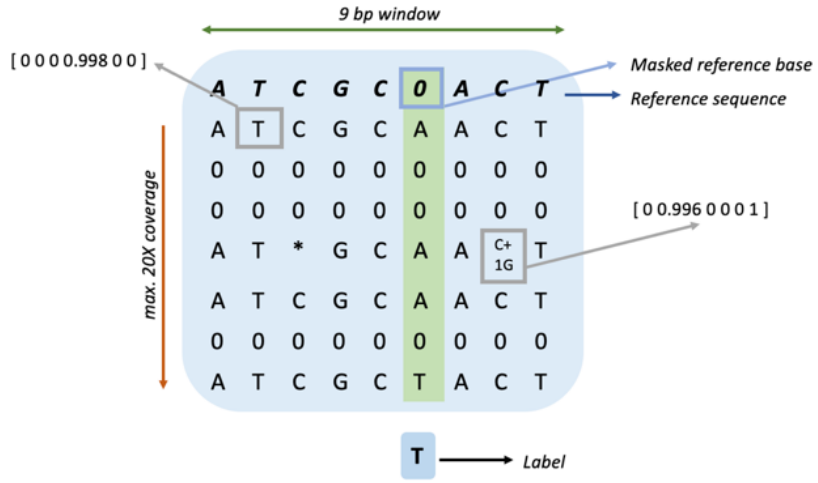


Figure 5 – Matrix design for one sample. Green row: position of interest, sequence in bold: reference sequence, light-blue arrow: masked reference base at position of interest, grey arrows: example of one-hot-encoded base and insertion, red arrow: coverage, green arrow: nine bp window, black arrow: label for position of interest (only for the T2T samples), *: deletion, C+1G: insertion.

After cross validation on the 300,000 train samples to evaluate the model performance, we show that the F1 score of all models is higher than the F1 score of the majority vote (Figure 6). The F1 score is the harmonic mean between precision and recall and is a measure of accuracy. We calculate the macro-average F1 score for each model based on 10-fold cross validation. The majority vote is a simple approach to make a prediction for each sample, where the most frequent base at the position of interest is the predicted base which in theory can remove all random sequencing errors. The F1 score is calculated for predictions of the four bases only.

The F1 score for the majority vote is a few magnitudes lower than the model's F1 score for each coverage bin of the train set (e.g. median F1-score DNN 3-5X: 0.9999648, median F1-score CNN 3-5X: 0.9999604, F1-score majority vote approach 3-5X: 0.997192), showing our models improve consensus calling of a single sample comparing to the majority vote approach. The F1 score of the majority vote approach improves when the coverage of the samples increases (DNN 3-5X: 0.997192,

DNN 16-20X: 0.999385), which is expected as with more reads the chance of making a wrong call decreases.

It is also noted that for each coverage bin of the training samples, the DNN model consistently outperforms the CNN model. For some models the difference is minor (e.g. DNN 3-5X vs. CNN 3-5X) while for other models it is substantial (e.g. median F1-score DNN 6-10X: 0.99996 vs. median F1-score CNN 6-10X: 0.999793). It can also be observed that the F1 scores of each CNN model show a larger spread than the F1 scores for each DNN model, indicating the CNN models are less robust.

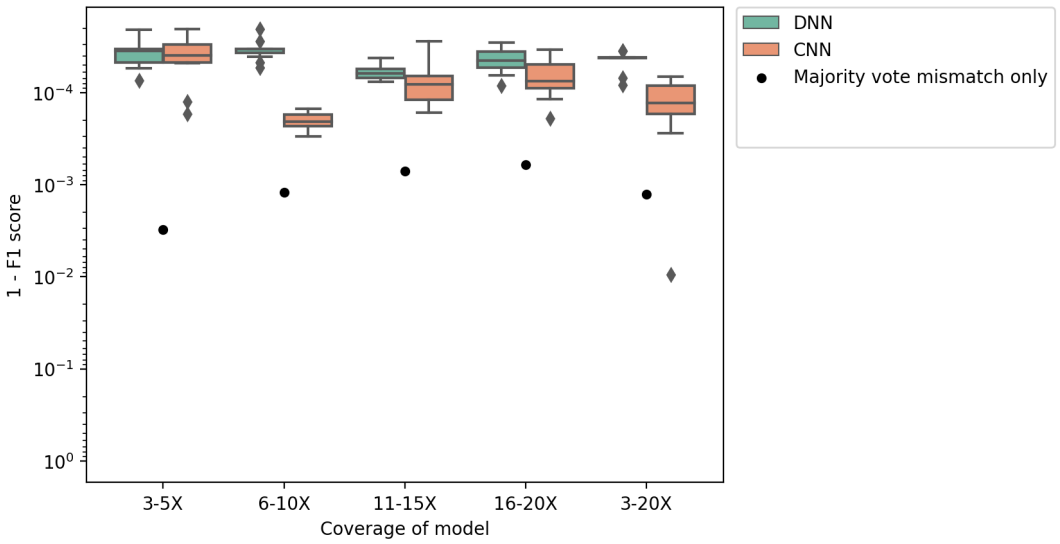


Figure 6 – F1 score of DNN and CNN T2T models trained on different coverage bins. The majority vote F1 score is calculated over all available training samples (where a base is predicted) for each coverage bin. Model F1 scores are determined with 10-fold cross-validation on the train set. Y-axis is plotted logarithmically.

Performance of DNN and CNN T2T models on test set of chromosome 18

Next, we evaluated the performance of all DNN and CNN T2T models on a separate set of 200,000 samples from chromosome 18 (test set), retrieved from the same dataset as the training samples described above. The ten models, each trained on a different coverage bin, are used to predict the samples of the test set per coverage bin (Figure 7). In other words, we want to see how the coverage of the model translates to performance on test data with different coverages. This may be valuable information for the use case of the models, which is prediction of the CyclomicsSeq consensus sequences, where different coverages for each CyclomicsSeq read are expected. The majority of the CyclomicsSeq reads have less than 10 copies per original CyclomicsSeq read (Figure 8).

Overall, the false positive rate (FPR) of samples predicted with the DNN models is lower than for samples predicted with the CNN models. Therefore, we will use mainly the DNN models for consensus calling for CyclomicsSeq in the next sections. Benchmarking the performance of the DNN models with the false positive rate of the majority vote approach shows that all combinations perform better than the majority vote. When looking at the performance on prediction of the samples with 3-5X coverage with the DNN models, we see the lowest false positive rate is achieved with the DNN 3-5X model (FPR: 0.00122%). The DNN models trained on higher coverage (FPR DNN 11-15X: 0.00253%, FPR DNN 16-20X: 0.00592%, FPR DNN 3-20X: 0.00243%) perform less on prediction of the samples with 3-5X coverage. As the majority of the CyclomicsSeq reads will have less than 10X coverage, we expect that the models trained on a lower coverage will be best at predicting the CyclomicsSeq consensus sequence.

Taken together, these results suggest that the DNN T2T models can perform accurate consensus calling for Nanopore sequencing data. The next section therefore moves on to bring these models into practice for consensus calling for CyclomicsSeq sequencing data.

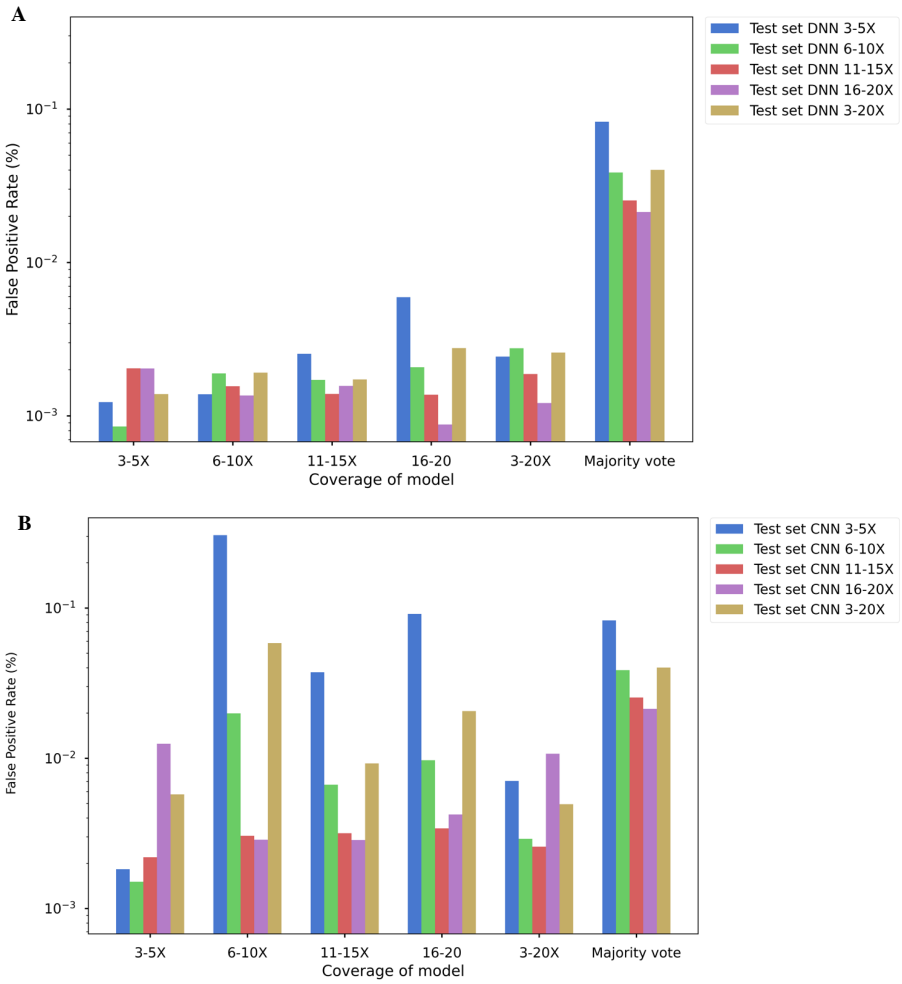


Figure 7 – False positive rate of prediction of the test set for all coverage bins by all DNN T2T (A) and CNN T2T (B) models. The majority vote is calculated over all samples of the test set and is similar for A and B. Y-axis is plotted logarithmically.

Consensus calling for CyclomicsSeq reads from the *TP53* gene using the DNN T2T models

We proceeded to use the DNN T2T models for consensus calling of CyclomicsSeq reads. CyclomicsSeq dataset A contains synthetic DNA of five regions of the *TP53* gene, with five mutations incorporated at low variant allele frequency (Table 1). The DNA is sequenced with CyclomicsSeq and base calling was performed with Guppy high accuracy base calling. Converting each CyclomicsSeq read with *n* copies of the original DNA fragment into a sample per position with *n*X coverage while retaining the order of the positions, the consensus sequence can be predicted for each original CyclomicsSeq read with each model (DNN 3-5X, DNN 6-10X, DNN 11-15X, DNN 16-20X, DNN 3-20X) and mapped to the CHM13 v2 reference genome. As expected from the CyclomicsSeq workflow, most reads have a lower number of copies per read and reads with more repeats are less frequent (Figure 8). All reads with a coverage higher than 20 are in the category '20+X' and the coverage of those reads is subsampled to 20. We compare the consensus calling accuracy of our model to the performance of the current consensus calling from CyclomicsSeq, the Cycas Consensus method. This is a majority vote approach that is weighed by base quality. The per-read error rate is calculated as $\frac{NM}{alignment\ length}$, where the NM value

indicates the number of mistakes in the alignment of the predicted consensus sequence to the reference sequence.

Table 1 – Known mutations in CyclomicsSeq dataset A & B.

Region	Position	Mutation	Expected VAF
Region 3	chr17:7,577,900	G>A	0.5%
Region 3	chr17:7,577,926	C>T	0.5%
Region 3	chr17:7,577,927	G>A	0.5%
Region 4	chr17:7,578,324	T>A	0.25%
Region 4	chr17:7,578,387	A>C	0.25%

Looking into the per-read error rate split per coverage number for the consensus sequences generated by the Cycas Consensus method, we see that that the per-read error rate decreases sharply when the number of repeats increases from 3X (0.4169%) to 10X (0.064%) (Figure 9). From 10X to >20X (0.0336%), the per-read error rate does not drop drastically anymore. This is in line with previous findings from CyclomicsSeq. The same pattern can be found for the consensus sequences generated with the DNN T2T models but holds true especially for the models trained on a higher coverage (DNN 11-15X, DNN 16-20X): a lower coverage number is associated with a higher error rate. We see that for CyclomicsSeq reads with 3X copies, the per-read error rate for sequences predicted with the DNN 3-5X model is 0.062% while it is 0.4169% for the consensus sequences predicted with the Cycas Consensus method. This is in line with the cross-testing of all DNN T2T models with different coverages of the test set (Figure 7). From 10 copies and higher, the different DNN models show smaller differences in per-read error rates and the difference in the per-read error rate with the consensus sequences from the Cycas Consensus method decreases. Still, the Cycas Consensus method results in consensus reads with a higher mean per-read error rate for each number of copies compared to the consensus reads predicted with the DNN models.

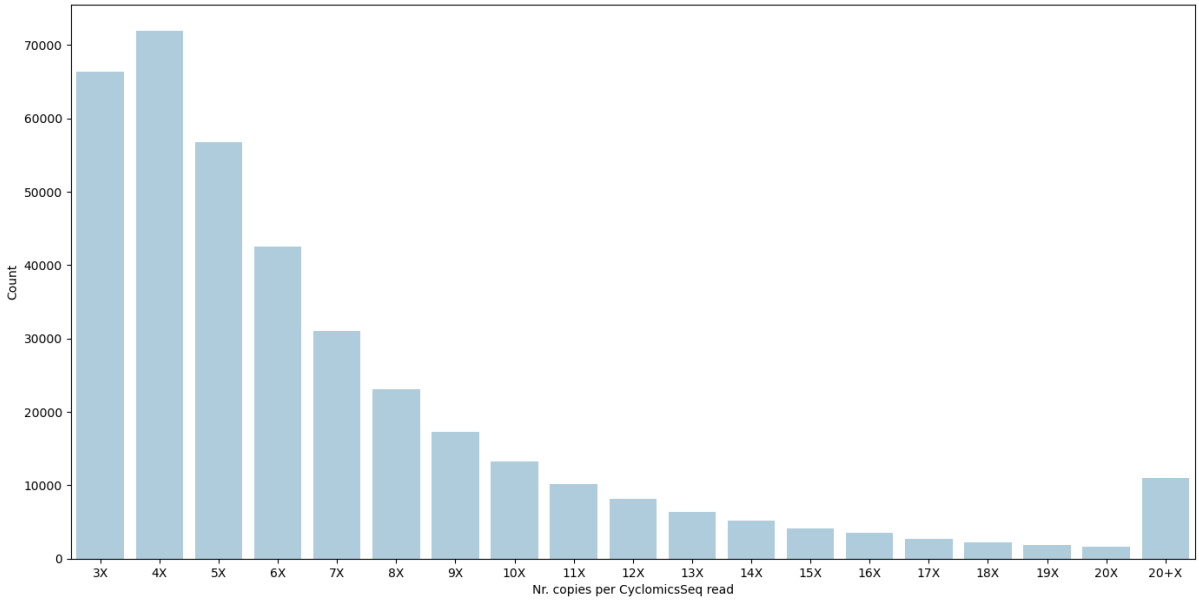


Figure 8 – Number of copies per CyclomicsSeq read of the Cycas Consensus method from CyclomicsSeq dataset A. Only the overlapping subset of reads is analysed. Reads with more than 20 copies are binned in the 20+X category and the first 20 copies based on the start coordinate are used.

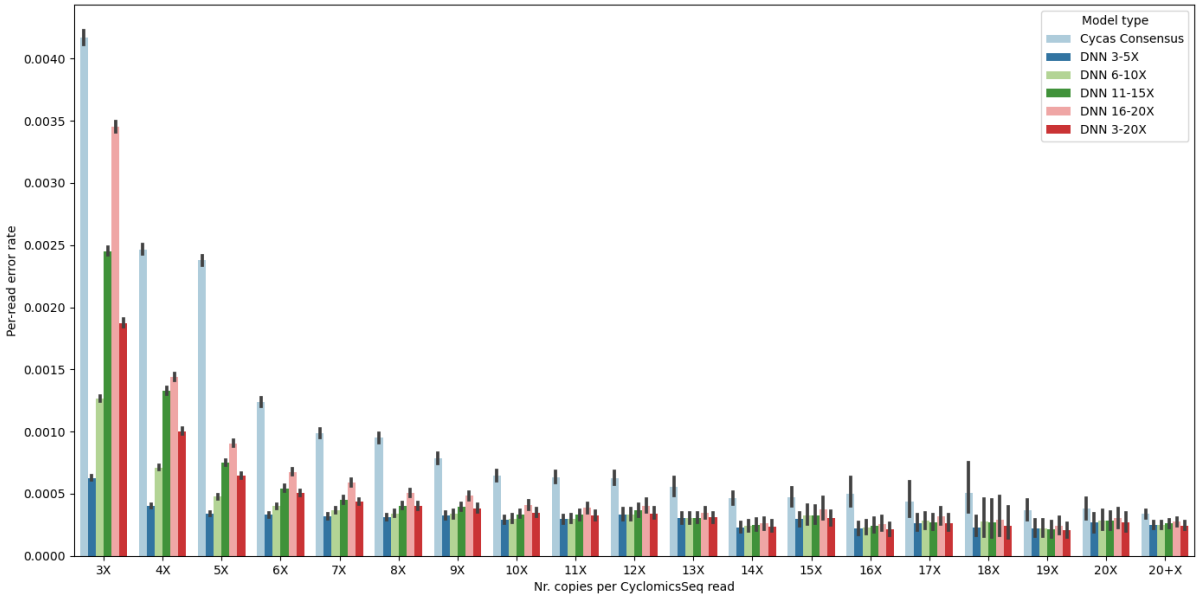


Figure 9 – Per-read error rate for consensus sequences generated with the DNN T2T models or with the Cycas Consensus method. Coverage on the x-axis indicates number of copies in the original CyclomicsSeq repeat, i.e., the number of reads used for consensus calling. Bars show 95% confidence interval. Only the overlapping subset of reads is analysed.

Consensus calling for CyclomicsSeq reads from the *TP53* gene using the DNN and CNN T2T models

Next, we used the DNN and CNN T2T models to predict the consensus sequence for each CyclomicsSeq read of another CyclomicsSeq dataset, dataset B. This dataset contains synthetic DNA of five regions of the *TP53* gene, with five mutations incorporated at a low variant allele frequency (VAF) (three mutations at 0.5%, two mutations at 0.25%). Therefore, this dataset can be used to validate the consensus calling performance as well as the ability of our model to predict mutations. As the DNA has been amplified with PCR during the CyclomicsSeq protocol, the expected error rate is at least 0.20%, which is the error rate for errors that occur during the PCR step. The consensus sequences are predicted with each model (DNN/CNN 3-5X, 6-10X, 11-15X, 16-20X, 3-20X) and mapped to the CHM13 v2 reference genome. The predicted consensus sequences did not all map to the reference genome, indicating these reads have too many errors to be mapped (Figure S1A). The Cycas Consensus sequences all mapped to the reference genome or to the backbone sequence. In this section we will first compare the consensus calling performance of the models to the Cycas Consensus method, and then investigate the ability of the model to accurately call mutations at the single molecule level.

The error rate per read is calculated as $\frac{NM - \sum mutations}{alignment\ length}$ with $\sum mutations$ being the number of known mutations found in the read. By calculating this adjusted NM score, the mutations do not interfere with the error rate. A read with a mutation and no other alignment mistakes is thus considered perfect. However, indirectly, a mutation can affect base calling of neighbouring bases in the read because the sequence context changes with a mutation. The DNN models predict the consensus sequence for each CyclomicsSeq read with a lower average per-read error rate than the Cycas Consensus pipeline (Figure 10A). Again, we see that the DNN models trained on samples with low coverage (DNN 3-5X, DNN 6-10X, DNN 3-20X) achieve a lower per-read error rate for the predicted consensus sequences than the DNN models trained on samples with a higher coverage (DNN 11-15X, DNN 16-20X). All CNN models perform less accurate consensus calling than the Cycas Consensus method. The DNN models predict a higher percentage of reads without alignment mistakes than the Cycas Consensus method, whereas the CNN model predicts relatively fewer perfect reads (Figure 10B). Looking at the per-read quality score for the reads with at least one alignment mistake, again the CNN models output reads with

higher error rates. The DNN models and the Cycas Consensus method both achieve around Q20 as the median quality score with the DNN 3-5X model achieving the lowest median per-read error rate (Figure 10C). From the accumulative distribution of reads, we show that the predicted consensus sequences of the DNN 3-5X and DNN 6-10X model have a higher proportion of reads with a higher per-read quality score than the reads of the Cycas Consensus method (Figure 10D). Keeping in mind that the models are trained on forward sequences only, we analysed the error rate per read direction. For all models except CNN 3-5X, the reads that map to the forward strand show a lower per-read error rate than those mapping to the reverse strand (Figure 10E). This section has shown that the DNN T2T models can perform accurate consensus calling for CyclomicsSeq and outperform the current Cycas Consensus method.

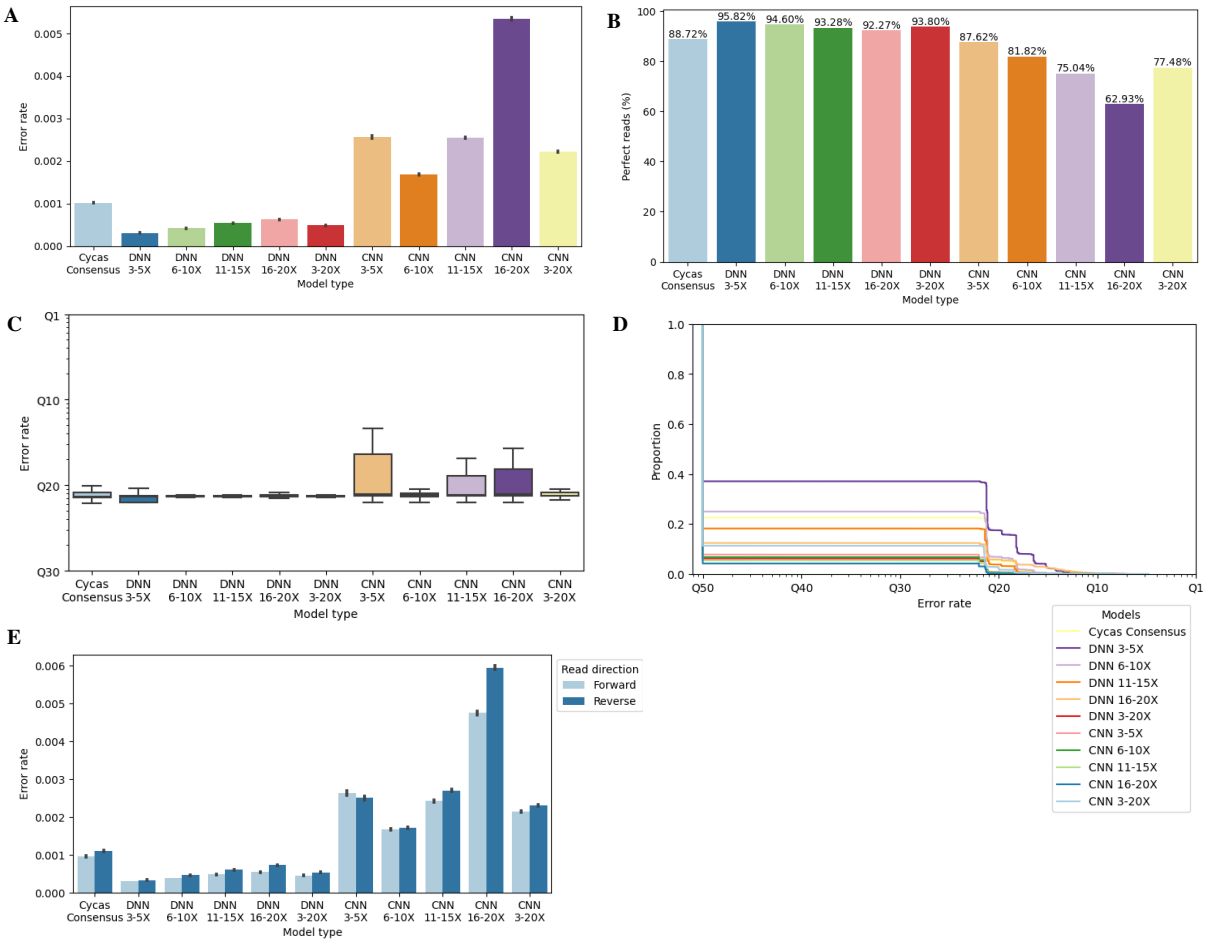


Figure 10 - Analysis of the consensus reads of CyclomicsSeq dataset B predicted with DNN or CNN T2T models trained on different coverages and with the Cycas Consensus pipeline. All error rates are corrected for presence of known mutations and only the overlapping subset of reads is analysed. A: Mean per-read error rate. Error bars show 95% confidence interval. B: Percentage of predicted reads with no alignment mistakes. C: Median per-read quality score for reads with at least one alignment mistake. Y-axis is plotted logarithmically. D: Accumulative proportion of reads based on quality score. X-axis is plotted logarithmically. E: Median per-read error rate split for reads mapping to the forward or reverse strand.

Validation of mutation calling for CyclomicsSeq reads mapping to the *TP53* gene

Moving on, we sought to find out whether the T2T models can accurately predict mutations on a single molecule in addition to consensus calling. Five mutations are present at a low variant allele frequency (Table 1) in CyclomicsSeq dataset B. Mutations 1, 2 and 3 are designed to appear on a single molecule that maps to region 3, and mutation 4 and 5 are designed to appear on reads that map to region 4.

First, we show that all models can predict the mutations at these five positions (Figure 11A, 11B). This indicates that the models have learned to look at the base context of the reads and is not just

predicting the reference base that it might have learned during the training process. When comparing the expected VAF of the mutations with the observed VAF in the consensus reads from the DNN models and from the Cycas Consensus pipeline, it is shown there is a discrepancy between the expected VAF and the found VAF for all models including the Cycas Consensus method (Figure 11A, 11B). The two mutations on region 4 are found at a VAF around 0.08% for all models and the Cycas Consensus pipeline, whereas the expected VAF is 0.25%. The three mutations on region 3 are found at a VAF around 0.90% for all models (except CNN 3-5X) and the Cycas Consensus pipeline, where the expected VAF is 0.50%. During preparation of the synthetic DNA, two separate DNA solutions are diluted to achieve the expected VAF. The discrepancy between the expected and the observed VAF can occur due to unavoidable dilution errors.

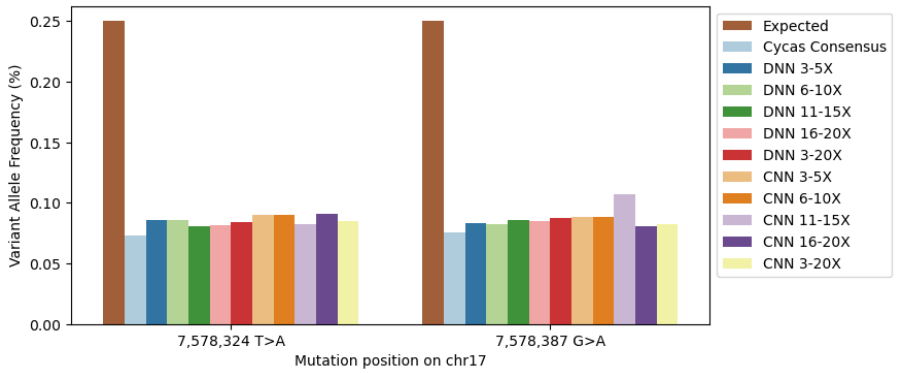
In both regions, the mutations are most often found on the same molecule for the consensus sequences predicted with each DNN model (Figure 11C, 11D) and CNN model (Figure S3), as well as the consensus sequences from the Cycas Consensus method (Figure 11C, 11D). When only one or two of the three mutations are present in the case of region 4, or only one mutation in the case of region 4, we cannot be sure whether those are actually sequencing errors (false positives) or that they are true mutations. In that case, the other mutations have been missed (false negatives).

Then, we analysed if we would be able to detect mutations on the *TP53* gene without a priori knowledge of the mutation position. For the consensus sequences predicted with the DNN 3-5X model, we observed that the allele fraction of the mutated positions in region 3 (observed VAF 0.90%) is higher than the per-nucleotide error rate for all regions (Figure 12). With a variant allele fraction detection limit of 0.30%, we can thus detect mutations without calling false positive mutations in these regions of the *TP53* gene. For the consensus reads of the Cycas Consensus method in region 3, a similar pattern can be observed (Figure 12). However, at the ends of the region 3 and 4 a few positions with a higher error rate are observed. These positions may interfere with mutation calling when the detection limit is set at 0.30%. For the consensus sequences predicted with the DNN 3-5X model, the VAF of the mutations in region 4 (0.08%) is lower than the per-nucleotide error rate of the surrounding positions. Thus, these mutations cannot be detected without prior knowledge. This holds true for the consensus sequences from the Cycas Consensus method as well. As the targeted CyclomicsSeq protocol comprises a PCR step, the expected error rate attributable to PCR errors is 0.20%. However, most of the errors are below 0.20% for both the consensus sequences predicted with the DNN 3-5X model and with the Cycas Consensus method.

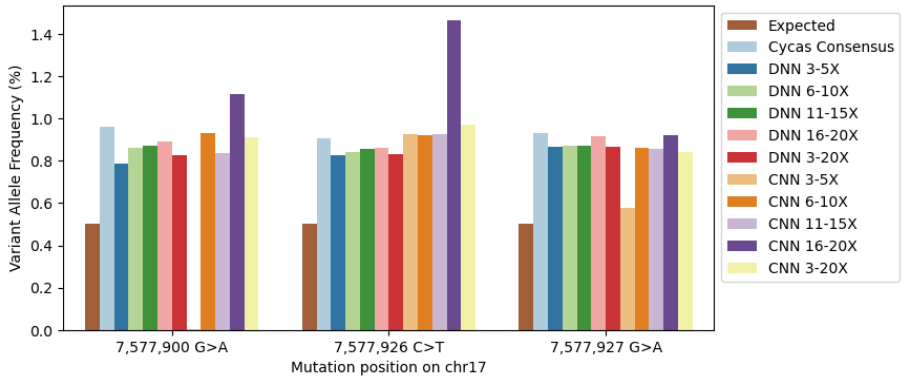
The other DNN models and all CNN models show an error rate equal to or higher than the observed mutation rates, so the mutations cannot be found back in these reads without prior knowledge (Figure S4). The model also predicts the base quality score. However, we did not filter for read or base quality. This may reduce the error rate at all positions, as it is hypothesized that wrongly predicted bases have a lower base quality score. Overall, these results indicate that the DNN 3-5X model can validate single nucleotide variants on the *TP53* gene. In previous sections it has been shown that all DNN models can perform accurate consensus calling for the *TP53* gene.

Figure 11 (next page) – Mutation analysis of CyclomicsSeq dataset C predicted with DNN and CNN T2T models trained on different coverage bins or with the Cycas Consensus method. A and B are based on all reads output by each model, C and D are based only on the overlap between all datasets. A: Expected and observed variant allele frequency (VAF) for three mutations in region 3. B: Expected and observed VAF for two mutations in region 2. C: Frequency of each combination of mutation 1, 2, and 3 in consensus reads predicted with the DNN T2T models or the Cycas Consensus method. Based on 117923 reads mapping to region 3. D: Frequency of each combination of mutation 4 and 5 in consensus reads predicted with the DNN T2T models or the Cycas Consensus method. Based on 65582 reads mapping to region 4.

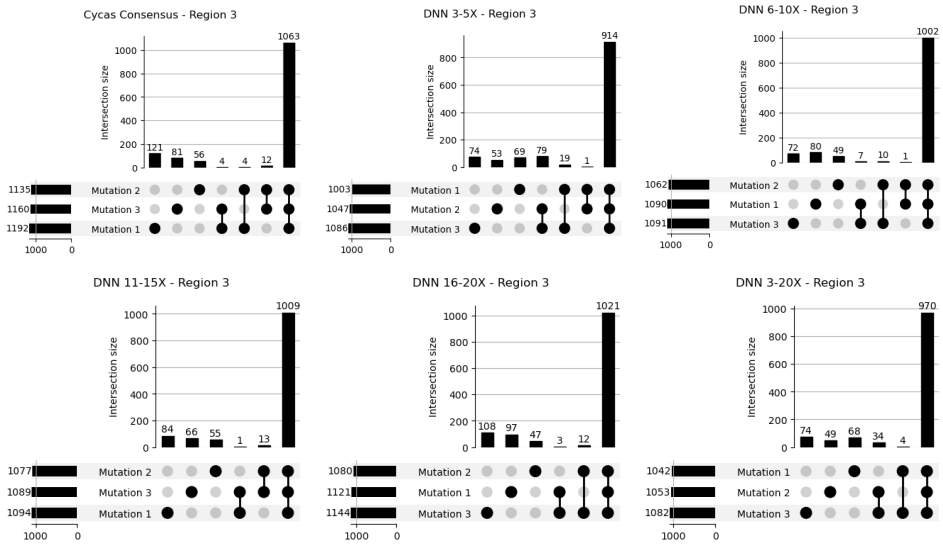
A


Variant Allele Frequency (%)
0.25
0.20
0.15
0.10
0.05
0.00
7,578,324 T>A
7,578,387 G>A
Mutation position on chr17
Expected
Cycas Consensus
DNN 3-5X
DNN 6-10X
DNN 11-15X
DNN 16-20X
DNN 3-20X
CNN 3-5X
CNN 6-10X
CNN 11-15X
CNN 16-20X
CNN 3-20X

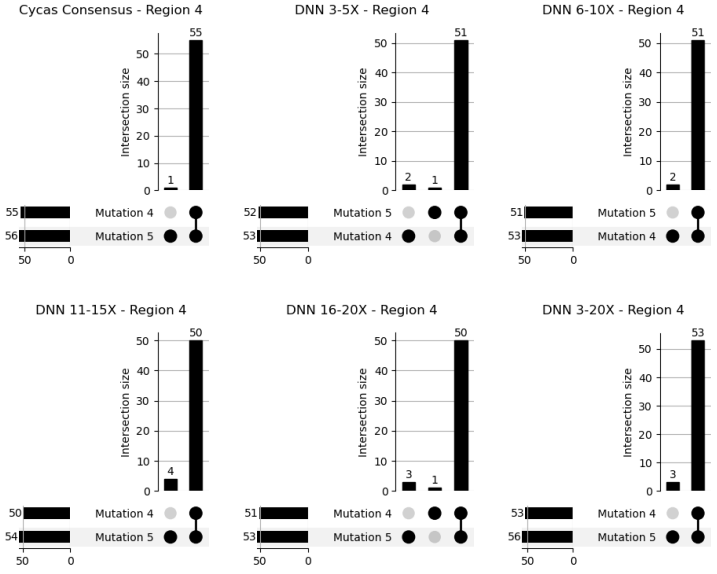
B


Variant Allele Frequency (%)
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
7,577,900 G>A
7,577,926 C>T
7,577,927 G>A
Mutation position on chr17
Expected
Cycas Consensus
DNN 3-5X
DNN 6-10X
DNN 11-15X
DNN 16-20X
DNN 3-20X
CNN 3-5X
CNN 6-10X
CNN 11-15X
CNN 16-20X
CNN 3-20X

C


Cycas Consensus - Region 3
DNN 3-5X - Region 3
DNN 6-10X - Region 3
DNN 11-15X - Region 3
DNN 16-20X - Region 3
DNN 3-20X - Region 3
Intersection size
Mutation 1
Mutation 2
Mutation 3

D


Cycas Consensus - Region 4
DNN 3-5X - Region 4
DNN 6-10X - Region 4
DNN 11-15X - Region 4
DNN 16-20X - Region 4
DNN 3-20X - Region 4
Intersection size
Mutation 4
Mutation 5

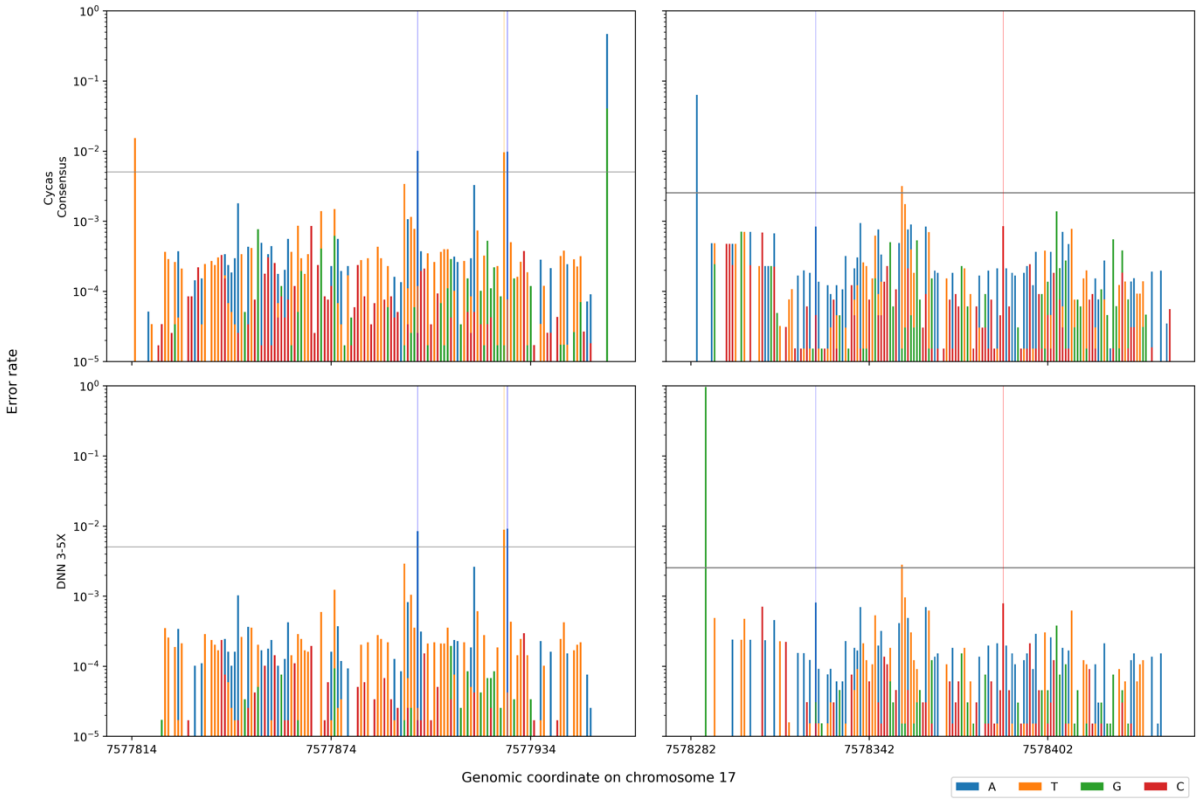


Figure 12 – Error rate per position per base for consensus reads predicted with the Cycas Consensus method or with the DNN 3-5X model. Only the overlapping subsets of reads between all models and the Cycas Consensus method are used. Region 3 (left) and region 4 (right) are shown here. Vertical lines indicate mutation 1-4 coloured according to the alternative base. Horizontal grey lines indicate the expected variant allele frequency (0.5% for region 3, 0.25% for region 4). Position 7,578,287 has a known homozygous mutation T > G and should be present in the reads of the Cycas Consensus method as well.

Consensus calling of genome wide CyclomicsSeq reads with the DNN T2T models

To evaluate the consensus calling performance of our model on other human chromosomes, we were provided with a whole genome sequencing dataset of cell-free DNA of a healthy individual sequenced by CyclomicsSeq. All DNN T2T models were used to predict the consensus sequence for each Cyclomics read (examples in Figure 14). Predicted sequences are mapped to the hs37d5 reference genome as this genome was used in the CyclomicsSeq pipeline. Each model's output contains approximately 0.7% unmapped reads (Figure S5A). Reads overlapping with known SNPS and indels of the individual are discarded so that no variants are expected in the remaining reads, besides variants that potentially have been missed with variant calling. The per-read error rate for this dataset is calculated only for nucleotide mismatches as $\frac{NM - nr.\ indels\ found}{Alignment\ length}$.

The Cycas Consensus reads now show a higher percentage of perfect reads (90.02%) than the DNN models (DNN 3-5X: 89.05%, Figure 13A). The mean error rate for the predicted consensus sequences is lower for the DNN 3-5X and DNN 6-10X model than for the consensus reads from the Cycas Consensus method (Figure 13C). Looking at the per-read quality scores for reads with at least one single nucleotide mismatch (Figure 13B), the distribution appears very similar with the per-read quality scores of the Cycas Consensus method showing a larger spread towards both lower and higher quality scores compared to the DNN models trained on a low coverage (DNN 3-5X, DNN 6-10X). For all models and the Cycas Consensus method, the reads with the lowest quality score have a quality score at Q10 (Figure 13B). For the DNN 3-5X and the DNN 6-10X model, a per-read comparison of the error rate of the consensus sequence predicted with the model or the Cycas Consensus method, it can be seen

that the quality scores are lower for the reads predicted with the model (Figure 13D). A point below the diagonal indicates the error rate of that read is lower for the consensus read predicted with the model than with the Cycas Consensus method. In the second panel of figure 14, it is seen that the Cycas Consensus method makes consensus calling mistakes that are removed with the best models (DNN 3-5X, DNN 3-20X) but not by the models that work less well (DNN 16-20X). All these results suggest that our method is able to predict accurate consensus sequences for cell-free DNA reads that are derived from anywhere on the genome. Again, we see that the DNN 3-5X model returns the most accurate consensus sequences.

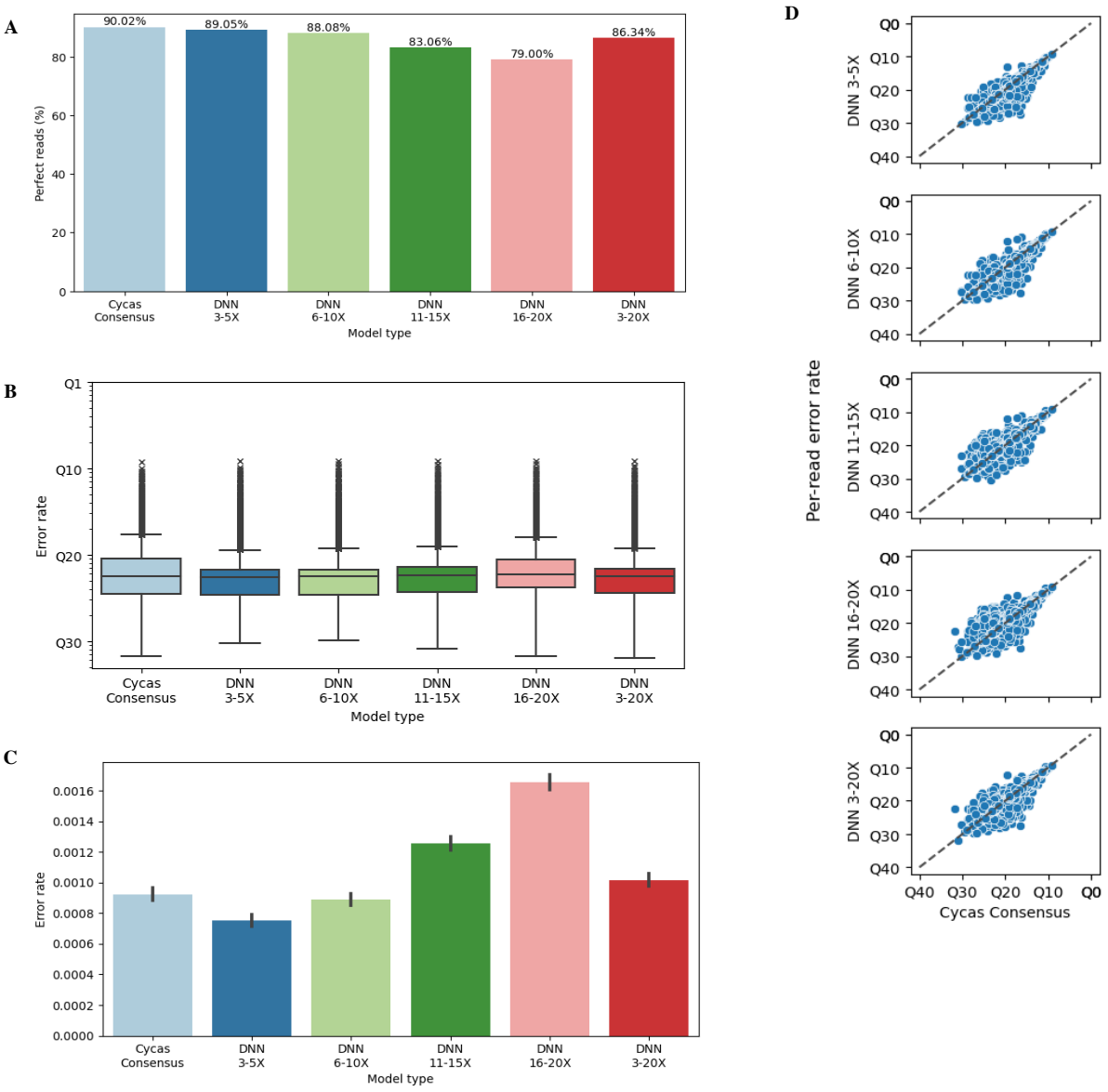


Figure 13 – Analysis of the consensus reads of genome wide cell-free DNA sequenced with CyclomicsSeq. The consensus reads are predicted with DNN models trained on different coverages or with the Cycas Consensus method. The per-read error rate is corrected for the number of insertions and deletions per read. Only the input reads that occur in each dataset are analysed. A: Percentage of reads with no single nucleotide alignment mistakes. B: Median per-read quality score for reads with at least one single nucleotide alignment mistake. Y-axis is plotted logarithmically. C: Mean per-read error rate. Error bars indicate 95% confidence interval. D: Comparison of the per-read quality score for each read predicted with one of the DNN models and the Cycas Consensus method. Only reads with at least one single nucleotide alignment mistake are shown. Diagonal equals where the per-read quality score is similar for both methods.

Figure 14 – Consensus calling examples shown in IGV (Integrative Genomics Viewer) of cfDNA reads derived from chromosome 1. The cfDNA fragments originate from the genome wide CyclomicsSeq dataset (dataset C). Purple triangles indicate positions of insertions in at least one of the tracks. Reads are coloured by read direction (red indicates forward strand, blue reverse strand). Deletions are indicated with dashes and insertions with purple vertical bars in the alignment. Mismatched bases are coloured, and transparent when their base quality is below 20. In each example, the first panel shows the copies of a single cfDNA fragment sequenced with CyclomicsSeq. The other panels show consensus sequences for these input copies as predicted by DNN 3-5X (panel 2), DNN 3-20X (panel 3), DNN 16-20X (panel 4), and the Cycas Consensus method (panel 5).

Discussion

In this thesis, we presented a proof-of-concept method to improve consensus calling of cfDNA reads sequenced with CyclomicsSeq. CyclomicsSeq uses rolling circle amplification to sequence concatemers of cfDNA fragments with Nanopore to reduce random sequencing errors. However, some sequence contexts (e.g. homopolymers) result in systematic sequencing errors with Nanopore sequencing, which cannot be removed by consensus calling of rolling circle amplificated repeats. These errors interfere with single nucleotide variant calling in a cfDNA molecule. We have showcased a series of deep learning models trained on Nanopore sequencing data to perform consensus calling that reduces systematic sequencing errors while retaining true variants.

We have trained a set of dense neural networks (DNN) and convolutional neural networks (CNN) for different coverage bins (3-5X, 6-10X, 11-15X, 16-20X, 3-20X) based on DNA reads sequenced with Nanopore. The DNA originates from a cell line with a haploid genome and thus phasing of the reads prior to using the reads as model input is not necessary. We have shown that our model performs better consensus calling than a simple majority vote approach. The models are then used to perform consensus calling for several CyclomicsSeq datasets. We have shown that a model trained on reads with a low coverage (3-5X, 6-10X) outperform models trained on a higher coverage for consensus calling of CyclomicsSeq reads. Next, we validated variant calling of five mutations at a low variant allele frequency. We have shown that the DNN 3-5X model can call mutations above a variant allele frequency of 0.30%, but that a further decrease in error rate is required for determining tumour allele fraction based on single molecules for mutations with a lower allele frequency. Finally, we have shown that we can accurately perform genome wide consensus calling. Our model, that is trained only on sequences from a region of chromosome 18, is thus capable of consensus calling for all chromosomes.

In general, we saw that prediction using the models trained on lower coverage resulted in less errors than using the models trained on higher coverage, when looking at samples with a low coverage. We shuffled the rows ('reads') of the input matrix (Figure 5) to make sure the model pays attention to all the rows, but this effect persisted. Perhaps the model that is trained on high coverage utilizes all available information during training, and during prediction of samples with low coverage, this model cannot deal with missing information. For samples with a higher coverage, model performance was similar across all models. We would therefore recommend using a model trained on low coverage reads (3-10X), perhaps including a low number of reads with a higher coverage (10-20X) for consensus calling of cell-free DNA reads. With techniques that seek to explain the inner workings of 'black box' machine learning models (32), like saliency maps (33), node importance can be visualized. This may indicate which rows (the reads) of the input sample the model uses to make its prediction. Saliency maps are usually used for images classification with CNN models, but the 3D input sample may be regarded as an image with pixel values between 0 and 1. Using this information, it can be investigated why the DNN 3-5X model is better at predicting a sample with low coverage than the DNN 16-20X model, and this information may aid in future model development.

Surprisingly, the DNN models constantly outperformed the CNN models. We would have expected that convolution over the 3D matrix that is used as input for the CNN models, would allow the model to capture sequence patterns better than the DNN model, that uses a flattened matrix as input. However, further investigation into the CNN architecture may improve performance.

Currently, the models are trained on the forward sequences of a 300,000 bp region of chromosome 18. This region contains 50.5% of the possible 9-mers (the window of bases around the position of interest in the input sample) on the forward strand. Here, we note a few things that may be improved about our

method. The first thing to expand on would be to add the reverse sequences and allow the model to predict sequences that map to the reverse strand. Second, the 300,000 bp region could easily be extended to a bigger region, including other chromosomes. This would increase the number of 9mers present in the train set. One of the reasons that we have not done this, is the long runtime of the script that generates the input matrices. Generation of 10,000 samples takes between 30 and 60 minutes, depending on the coverage. Multiple batches of samples can be created simultaneously. However, model training itself does not take very long (~15 minutes for the DNN models, ~1 hr for the CNN models on a single GPU), so optimization of the script to generate the input matrices would highly increase the efficiency of our method. A few ideas on how to optimize the script are prepared. The long runtime of the complete workflow impedes implementation of our method in (clinical) practice. Additionally, we have chosen a simple approach of model training with little hyperparameter optimization for time limitation. We believe an increased number of samples in combination with a more refined training strategy will improve the model's performance. Finally, our method does not contain additional read filtering after prediction of the consensus sequence. The model already outputs base quality, but this is not utilised besides during the mapping. A filter for read quality based on the quality score of all positions in the read would further decrease the per-read error rate, which in turn will improve variant calling.

This method has been developed initially to predict single nucleotide variants in a single ctDNA molecule. However, as it also serves as a consensus calling method, the fact that our method currently does not predict indels is a limiting factor for its utility. Given that the input samples for our method are based on a pile-up approach, here we present some adjustments that could be adapted to include these features in the future. The method could relatively easily be extended to predict deletions. Deletions are correctly represented in the pile-up approach, but in the current method the label is always one of the four canonical bases. The reference genome can be edited to include extra bases. Realignment of reads to this reference will lead to deletions in the reads at these positions, where samples can be created with deletion as label. It would be even better to create deletions at kmers where the Nanopore sequencer is prone to skip bases. In the current format it is not possible to predict insertions with the reference sequence supplied in each sample. One change would be needed for this method to predict insertions, which is the removal of the reference sequence from each sample. This will allow for extension of the pile-up when an insertion occurs, as there is no more limitation where each position has to be aligned to the reference genome.

DeepConsensus, a sequence correction algorithm developed for PacBio sequencing (29), uses a novel alignment-based loss, which has helped them overcome the problem of predicting indels. They have switched from using a sample design where reads are aligned to the reference or other contigs to a multiple sequence alignment (MSA) from the PacBio reads and the draft consensus that has been created by the PacBio pipeline. They use a variable-length prediction which allows their method to represent insertions and deletions errors in the alignment. DeepConsensus also leverage metadata from the sequencing process like interpulse duration and pulse width (parameters specific to PacBio sequencing).

ONT has developed Medaka, a consensus calling and variant calling tool for Nanopore sequencing (25,28). Due to time constraints, we have not come round to benchmark our model's performance against Medaka. However, this has been done internally at CyclomicsSeq before and Medaka has not proven to be a candidate to replace the Cycas Consensus pipeline because of computational problems. Medaka has been mainly developed for genome assembly, so has to restart every time a new CyclomicsSeq read enters the pipeline. There seems to be a module of Medaka specific for consensus calling of single molecule reads, but this is not well documented (34,35). Future research may be done to investigate this method's ability for consensus calling of a small set of reads like the

CyclomicsSeq reads. This will also allow benchmarking of our model's performance to another consensus calling method besides the Cycas Consensus method.

Two additional ways of improving our consensus calling method, would be to use the raw Nanopore squiggles or to use a different model architecture. Currently, our method is built on base called sequences. The raw Nanopore squiggles contain more information than the base called sequences, but implementation of this will not be straightforward as the DNA strand does not move through the pore at a single speed, so it is difficult to correlate a base called DNA sequence to a raw squiggle (16). For example, Nanopolish, a method for genome assembly, uses an algorithm that uses the raw signal for improving the consensus sequence (36). Potentially the raw signal can be added to the input matrix in a way similar that DeepConsensus implements additional sequencing information (29). Secondly, a more complex model architecture may improve performance if properly implemented, for example with self-attention layers like DeepConsensus or Enformer, or a recurrent neural network (RNN) architecture like Medaka. In this study, the DNN models shows the best results. They have a straightforward architecture and show more robust performance during training than the CNN models.

The method developed in this thesis is not limited to consensus calling and variant calling for CyclomicsSeq. In principle, all aligned reads can be used with our model, up to a coverage of 20X. Phasing may be necessary if DNA derived from a diploid genome is used. The model is trained on human Nanopore sequencing data, so most likely best results will be achieved when the input material has been sequenced with Nanopore and is derived from human. This greatly extends the use case of our model beyond the application for CyclomicsSeq, ranging from consensus calling for Nanopore sequencing data to polishing of genome assemblies.

In this thesis, we have presented a promising deep learning approach to consensus calling and variant calling on a single molecule level. We have shown to be able to predict consensus sequences with high accuracy. The best model (DNN 3-5X) increased the per-read accuracy compared to the current Cycas Consensus method of CyclomicsSeq. There is abundant potential for improvement of our method, and further development will improve consensus calling and single molecule variant calling. Broadly speaking, such a consensus calling method retains the benefits of long-read Nanopore sequencing while overcoming its main limitation that is its high error rate. For CyclomicsSeq, our consensus calling method will help with more accurate detection of mutations in cfDNA and will thus aid in cancer diagnosis and prognosis.

Methods

1. Data

1.1 Nanopore T2T dataset

The Telomere-to-telomere (T2T) consortium has sequenced the CHM13hTERT human cell line with Nanopore and base called with Guppy (version 5.0.7) (6). This dataset is publicly available in fastq format (37). This cell line has a homozygous genome and can thus be regarded as haploid. The base called data was mapped to the CHM13 v1.1 reference genome with minimap2, resulting in 120X coverage. Throughout this project, the nanopore preset option (-x map-ont) is always used for minimap2. The reads are filtered for primary mapping reads and reads mapping to the forward strand only. Reads are randomly subsampled to 30X for storage purposes.

1.2 CyclomicsSeq datasets

Three datasets from CyclomicsSeq based on regions of the *TP53* gene or genome wide are used in this study (Table 2). All DNA is processed and sequenced with the CyclomicsSeq protocol, which is shortly described in section 1.3.

CyclomicsSeq dataset A

CyclomicsSeq dataset A consists of synthetic DNA from five amplicons of the *TP*53 gene. Two mutations are incorporated in the synthetic DNA at 0.25% variant allele frequency (VAF) and three mutations at 0.5% VAF (Table 3). Base calling is done with Guppy high accuracy base calling (version 6.1.5). Reads are mapped to the CHM13 v2 reference genome.

CyclomicsSeq dataset B

This dataset consists of synthetic DNA from five amplicons of the *TP*53 gene. Two mutations are incorporated in the synthetic DNA at 0.25% variant allele frequency (VAF) and three mutations at 0.5% VAF (Table 2). Base calling is done with Guppy super high accuracy base calling (version 6.2.7). Reads are mapped to the CHM13 v2 reference genome.

CyclomicsSeq dataset C

Cell-free DNA of a healthy individual has been sequenced untargeted with CyclomicsSeq, resulting in genome-wide coverage. Base calling is done with Guppy super high accuracy base calling (version 6.2.11). Reads are mapped to the hs37d5 reference genome.

Table 2 – Overview of datasets. WGS = whole genome sequencing, N.A. = not applicable.

Name	Species	Reference genome	Targeted or WGS	Sequencing	Basecalling	Used for	Source	CyclomicsSeq serial number
Nanopore T2T	Cell line	T2T v2	WGS (we used only chr18)	Nanopore R9.4.1	Guppy high accuracy basecalling (version 5.0.7)	Model training	Public	N.A.
CyclomicsSeq dataset A	Synthetic	T2T v2	Targeted (chr17)	Nanopore R10.4.1	Guppy high accuracy base calling (version 6.1.5)	Consensus calling	Cyclomics Seq	CYC000025
CyclomicsSeq dataset B	Synthetic	T2T v2	Targeted (chr17)	Nanopore R10.4.1	Guppy super high accuracy base calling (version 6.2.7)	Consensus calling	Cyclomics Seq	CYC000025
CyclomicsSeq dataset C	Human	Hs37d5	WGS	Nanopore R10.4.1	Guppy (version 6.2.11)	Consensus calling	Cyclomics Seq	CYC000048: HC002

Table 3 – Known mutations in CyclomicsSeq dataset A and B.

Region	Position	Mutation	Expected VAF
Region 3	chr17:7,577,900	G>A	0.5%
Region 3	chr17:7,577,926	C>T	0.5%
Region 3	chr17:7,577,927	G>A	0.5%
Region 4	chr17:7,578,324	T>A	0.25%
Region 4	chr17:7,578,387	A>C	0.25%

1.3 CyclomicsSeq protocol

The CyclomicsSeq workflow starts with native cfDNA or synthetic DNA. The cfDNA is enriched for five amplicons on the *TP53* gene (targeted approach). These products are then amplified with PCR for *TP53* sequences in case of a targeted approach. A synthetic backbone sequence is ligated to each cfDNA fragment and together they circularize. Each circular DNA fragment then undergoes rolling circle amplification (RCA) to create multiple copies of the same DNA fragment on one strand. These products are sequenced with Nanopore sequencing, using a R10.4.1 pore. The resulting reads are mapped to a reference genome including the backbone sequences. Then, consensus calling is performed with a base quality score weighted majority vote approach to generate a consensus sequence from all copies of the same DNA fragment. This method is referred to as the Cycas Consensus method in this project. The genome-wide cfDNA is not enriched for specific amplicons, and the PCR step is thus not required. Further processing is similar to the targeted approach. Further details can be found in the publication of CyclomicsSeq (2).

2. Model input

The aligned reads (in bam format) are used to generate training and test samples. Using a pile-up approach, a sample consists of *n* reads aligned to a nine-base-pair window around the position of interest (Figure 15). There are five bases upstream and three bases downstream of the position of interest. The number of reads that align to the position of interest, or the coverage, ranges between 3X and 20X. If the coverage of a position is higher than the desired coverage, reads that have a higher starting coordinate on the forward strand are excluded. When the coverage is lower than 20X, the remaining rows are zero-padded to ensure each sample has the same size. In addition to the aligned reads to the position of interest, the reference sequence is also supplied. At the position of interest, the base of the reference sequence is replaced by a zero ('masked reference base'). For the train and test samples, the label for each position of interest is the reference base.

Each base is one hot encoded with six categories (A, C, G, T, deletion, insertion). The base quality score is converted to a probability of being correct with $p = 1 - 10^{-\frac{Phred\ quality\ score}{10}}$ and is used as the one in the one hot encoding. In this way, the base quality score is coupled with the sequence information. An insertion is encoded with two categories: the quality score for the base and one at the insertion. The label is also one-hot-encoded, but as the reference base can never be an insertion or a deletion, the label is always one of the four canonical bases. For one-hot-encoding of the label, we use a 1 and not a quality score. Finally, the rows (excluding the row with the reference sequence, so only the rows with reads and the rows with zeros) are shuffled randomly. This is done to ensure that the model is paying attention to all rows and is not just focusing on the first few reads.

For the dense neural network (DNN), an individual sample is stored in a 1D matrix with a size of (1134,). This can be calculated as 9 bases wide * 21 bases long * 6 for the encoding. The matrix depicted in figure X is flattened. For the convolutional neural network (CNN), a sample is stored as a 3D matrix. The shape of the matrix is (21, 9, 6) with (height, width, depth).

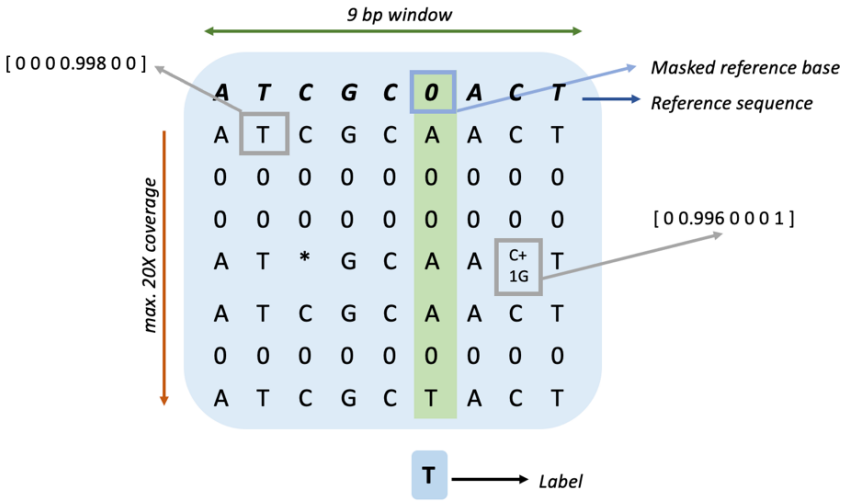


Figure 15 – Matrix design for one sample. Green row: position of interest, sequence in bold: reference sequence, light-blue arrow: masked reference base at position of interest, grey arrows: example of one-hot-encoded base and insertion, red arrow: coverage, green arrow: nine bp window, black arrow: label for position of interest (only for the T2T samples), *: deletion, C+1G: insertion.

2.1 T2T Samples

A 600,000 bp region of chromosome 18 of the T2T dataset is used to generate samples for training, testing, and hyperparameter optimization (Figure 16). The samples are created for reads mapping to the forward strand only. For each position a sample is created so in total we have 600,000 samples. Each sample is created with 20X coverage. Later, the coverage of a sample can be adjusted by dropping the last *n* rows of the matrix and padding it with zeros again. The samples and corresponding labels are stored in hdf5 format.

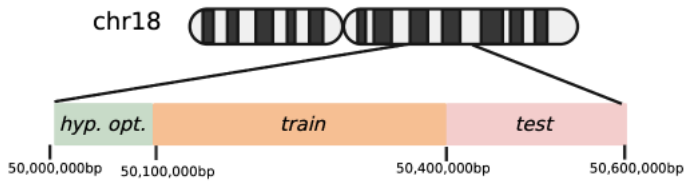


Figure 16 – Division of chromosome 18 into hyperparameter optimization (hyp. opt.) set (100,000 samples, green), train set (300,000 samples, orange), and test set (200,000 samples, pink).

2.2 CyclomicsSeq samples

The raw CyclomicsSeq reads are also converted into samples. Here, a label is not stored as these samples will be used to predict the consensus sequence for the CyclomicsSeq reads. For each read id, a sample is created for each position and stored in the correct order in the hdf5 file with the read id as unique identifier (Figure 17). The coverage of each sample is determined by the number of repeats in the original CyclomicsSeq read. If the coverage on a certain position is lower than three, the sample is not created. If the coverage is higher than 20, only the first 20 reads are used. The 20 reads with the highest start coordinate on the forward strand are taken. We do not perform read quality filtering to pick the 20 best reads. If the coverage is between 3 and 19, the remaining rows are filled with zeros.

As each sequence in the bam file is stored in the forward direction regardless of the original read orientation, reverse reads are converted to their forward sequence in the matrix. Information about

the original read orientation is retrieved later based on the unique read id from the Cycas Consensus reads.

The reference sequence for the reads of CyclomicsSeq dataset A is retrieved from the CHM13 v1.1 reference genome, and for the reads of CyclomicsSeq dataset B from the CHM13 v2 reference genome. For the genome wide CyclomicsSeq reads (dataset C), the reference sequence is retrieved from the hs37d5 reference genome.

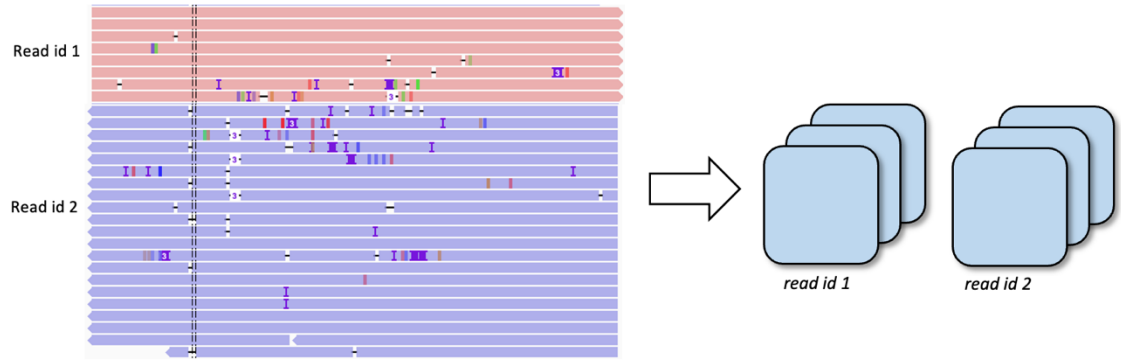


Figure 17– Two CyclomicsSeq reads with 8 (read id 1) and 21 repeats (read id 2). Per read id, a sample is created for each position and stored sequentially. For read id 2, the last read will not be included in the matrix as only the first 20 reads are considered.

3. Models

3.1 Model architecture

The dense neural network (DNN) consists of fully connected dense layers alternated with dropout layers (Figure 18A). The ReLu activation function is used at each hidden layer. The final layer uses the softmax activation function to calculate the probability of each class for the input sample. The class with the highest score will be the predicted class. The convolutional neural network (CNN) consists of 2D convolutional layers alternated with dropout layers (Figure 18B). The kernel size in each 2D convolutional layer is 2 by 2. ReLu activation is used at each 2D convolutional layer. For the final classification, flattening is performed to get a fully connected layer. After the last dropout layer, the softmax activation function is used to calculate the probability of each class for the input sample. The class with the highest score will be the predicted class.

3.2 Model training

We use the Keras framework in Python for model training and testing. For all models, we use a learning rate of 0.001 and the Adam optimizer. The used loss function is categorical cross-entropy loss. We train a DNN and a CNN for each coverage bin (3-5X, 6-10X, 11-15X, 16-20X, 3-20X) so we have 10 models in total. Each model is trained for a maximum of 20 epochs, but early stopping is allowed based on validation loss with a patience of four epochs.

We performed hyperparameter optimization on 100,000 samples (Figure 16) for the dropout rate in the first, second and last dropout layer for each model (Table 4). The samples are split into a train set (25%), validation set (25%), and a test set (50%). Hyperparameter tuning for the DNN model is computationally less expensive than for the CNN model, so we gave the DNN model a larger range of parameters to tune than the CNN model. The optimized dropout parameters are used during cross-validation and final model training.

We randomly split the training samples (300,000 samples, Figure 16) into a train set (25%), validation set (25%), and test set (50%) 10 times to perform 10-fold cross-validation for evaluation of

model performance. We report the macro-average F1-score for each fold as the mean of the F1-score for each class. The F1-score per class is calculated as $\frac{2 * precision * recall}{precision + recall}$. Precision is calculated as $\frac{TP}{TP+FP}$ and recall as $\frac{TP}{TP+FN}$, with TP as true positives, FP as false positives, and FN as false negatives.

We then randomly split the training samples again into a train set (80%) and a validation set (20%), to train the final models. The best model based on the validation loss metric is saved.

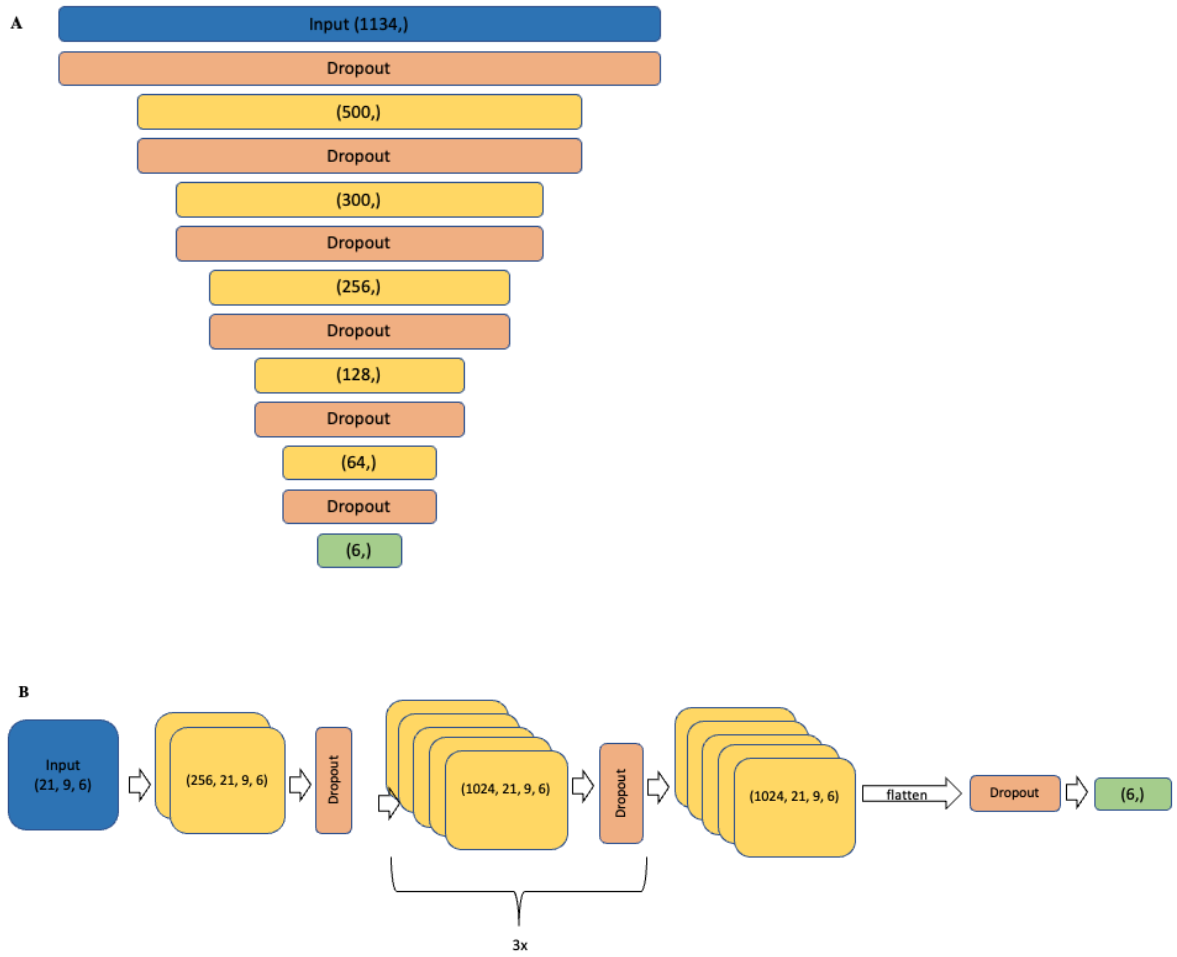


Figure 18 – Model architecture. Layers are not drawn to scale. Input layers are shown in blue, dropout layers in orange, and output layers in green. A: DNN, dense (fully connected) layers shown in yellow. The number indicates the number of nodes for each layer. B: CNN, 2D convolutional layers shown in yellow. The kernel size is (2,2) for each 2D convolutional layer. The numbers indicate the shape of each 2D convolutional layer with (nr. Filters, height, width, depth).

Table 4 – Results of hyperparameter tuning of first, second, and last dropout layer for each T2T model. Range of values indicates the numbers that were tested for each dropout layer. Tuning is performed with the keras tuner package (Table 11).

Model type	**Dropout layer**	**Range of values for tuning**	**Value after hyperparameter tuning**
DNN 3-5X	**First**	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.2
	Second	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.1
	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.2
DNN 6-10X	**First**	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.3
	Second	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.8
	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.4
DNN 11-15X	**First**	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.2
	Second	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.2

	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.2
DNN 16-20X	**First**	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.3
	Second	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.2
	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.3
DNN 3-20X	**First**	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.7
	Second	0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8	0.6
	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.2
CNN 3-5X	**First**	0.1, 0.2, 0.3, 0.4	0.3
	Second	0.1, 0.2, 0.3, 0.4	0.3
	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.1
CNN 6-10X	**First**	0.1, 0.2, 0.3, 0.4	0.3
	Second	0.1, 0.2, 0.3, 0.4	0.1
	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.3
CNN 11-15X	**First**	0.1, 0.2, 0.3, 0.4	0.4
	Second	0.1, 0.2, 0.3, 0.4	0.4
	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.4
CNN 16-20X	**First**	0.1, 0.2, 0.3, 0.4	0.4
	Second	0.1, 0.2, 0.3, 0.4	0.2
	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.3
CNN 3-20X	**First**	0.1, 0.2, 0.3, 0.4	0.3
	Second	0.1, 0.2, 0.3, 0.4	0.4
	Last	0.1, 0.2, 0.3, 0.4, 0.5	0.2

3.3 Model testing

We use all 10 models to predict the test set (200,000 samples, Figure 16) for each coverage bin (3-5X, 6-10X, 11-15X, 16-20X, 3-20X). Because for the DNN and the CNN samples, each sample has the same size regardless of the coverage, a model trained on coverage 3-5X can be used to predict a sample with 16-20X coverage. Using the final models, we test every combination once. For each prediction, we use the mean false positive rate as the metric. The false positive rate per class is calculated as $\frac{FP}{FP+TN}$, with FP as false positives and TN as true negatives.

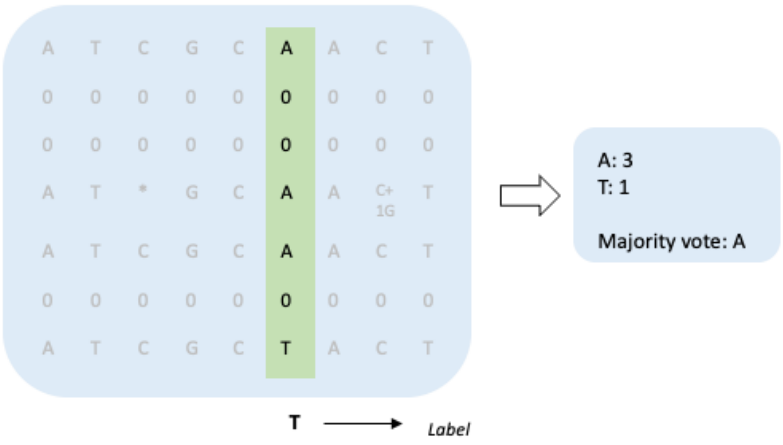


Figure 19 – Example of majority vote approach for the position of interest (green). The base predicted by the majority vote is compared to the label, and wrong in this example.

4. Majority vote

To benchmark each model's performance, we use a naive approach to predict each sample. With the majority vote, we consider the most frequent base on the position of interest as the predicted base (Figure 19). We calculate the majority vote for each coverage bin (3-5X, 6-10X, 11-15X, 16-20X, 3-20X). The majority vote is the same for the DNN and the CNN models. As the samples are created with a masked reference base at the position of interest, we do not take the reference base into account for the majority vote. The predicted bases are compared to the labels to calculate the performance metrics (F1 score) similarly to the way they are calculated for the predictions of the model for model training (section 3.2) and model testing (section 3.3). For the training set, the majority vote is calculated for all samples where a base is predicted and not those where a deletion is predicted to evaluate the single nucleotide mismatch majority vote performance. For the model testing, the majority vote is calculated for all samples.

5. Prediction of CyclomicsSeq consensus sequence

The T2T DNN and CNN models are used to predict the consensus sequence for each CyclomicsSeq read. Each CyclomicsSeq read consists of multiple copies of the same fragment. The long concatemer CyclomicsSeq read with multiple copies of a cfDNA fragment has been pre-processed with the CyclomicsSeq pipeline. This includes discarding reads with more than 50 mismatches to the reference (NM > 50), reads with a mapping quality below 20, and reads shorter than 500 bp. Hereafter, the concatemerized sequence is split up into multiple short reads that are copies from the same input fragment. We use three datasets provided by CyclomicsSeq (Table 2) that have been converted into samples for each position per read id. The model takes a matrix for each position as input and outputs one of the four canonical bases. The Phred base quality score is derived from the softmax probability distribution with the following formula: $Q = -10 * \log(1 - p)$ with p being the softmax probability for the predicted base. The Phred base quality score is capped at 93, with 93 being the maximum reported Nanopore quality score (38). As the matrices are stored per read id, the predicted sequence for each read id is stored in fastq format.

6. Analysis of prediction of CyclomicsSeq dataset A

The resulting fastq files from the prediction of each CyclomicsSeq read of dataset A with all DNN models are mapped to the CHM13 v2 reference genome using minimap2. From the reads that map to chromosome 17, the following information is stored: region, read id, NM tag, alignment length, cigar string, the type of model that is used for prediction, and the original coverage/number of repeats for each read. The NM tag depicts the number of mismatches of the aligned read with the reference sequence. The soft-clipped bases are not included in the alignment length. As the reads are predicted per region and this was semi-manually annotated, we excluded one read id (10 reads in total) with unknown region from further analysis.

It appeared duplicate reads were present in each region, we found reads with similar read ids that map to the same region. This may be an artefact of the CyclomicsSeq protocol as one would expect a unique read id for each CyclomicsSeq read. For each set of duplicated reads, we kept the read with the longest alignment length.

The consensus reads of the CyclomicsSeq pipeline for dataset A are retrieved in bam format and have been mapped to the CHM13 v2 reference genome using minimap2. From the reads that map to chromosome 17 and that have more than three repeats (based on presence of YM tag), the following information is stored: region, read id, NM tag, YM tag (repeat number in original CyclomicsSeq read), YR tag (read direction), alignment length, and the cigar string. The YM and YR tags are specific for CyclomicsSeq. For duplicate read ids, we kept the read with the longest alignment length.

We calculated the per-read error rate ($\frac{NM}{Alignment\ length}$). We then selected the reads that occur in each dataset (each model + Cycas Consensus). This resulted in 379,500 unique reads per model. The non-overlap read set is analysed separately, these results can be found in the supplementary figures. The number of reads per step can be found in table 5 (predicted reads) and table 6 (Cycas Consensus reads).

Table 5 – Number of reads for each step for the predicted consensus reads of CyclomicsSeq dataset A. All unmapped reads originate from a read with three copies of the original fragment (coverage 3X).

	DNN 3-5X	**DNN 6-10X**	**DNN 11-15X**	**DNN 16-20X**	**DNN 3-20X**
Number of split reads	8796174	8796174	8796174	8796174	8796174
Number of reads for model input	399611	399611	399611	399611	399611
Number of reads outputted by model	399611	399611	399611	399611	399611
Mapped reads (all reads mapped to chr17)	399610	399609	399605	399605	399609
Unmapped reads	1	2	6	6	2
After removal of 10 reads with unknown region	399608	399607	399603	399603	399607
After removal of duplicate read ids	397641	397640	397636	397636	397640
Perfect reads – non overlap	377566	366510	346960	337486	356527
Non perfect reads – non overlap	20075	31130	50676	60150	41113
Overlap between all read sets (including Cycas Consensus)	379500	379500	379500	379500	379500
Perfect reads - overlap	360375	349938	331625	322635	340496
Non perfect reads – overlap	19125	29562	47875	56865	39004

Table 6 – Number of reads for each step for the Cycas Consensus reads of CyclomicsSeq dataset A. No unmapped reads were produced by the Cycas Consensus pipeline.

	Cycas Consensus
Mapped reads to all chromosomes (no unmapped reads)	1008117
Mapped reads (backbone)	544203
Mapped reads to other chromosomes	Chr1: 14, chr2: 18, chr3: 8, chr4: 12, chr5: 11, chr6: 8, chr9: 12, chr10: 11, chr11: 6, chr12: 11, chr13: 8, chr14: 4, chr15: 3, chr16: 2, chr18: 7, chr19: 7, chr20: 8, chr21: 2, chr22: 3, chrX: 8
Mapped reads to chromosome 17	463728
Mapped reads to chromosome 17 & presence of YM tag	443133
After removal of duplicate read ids	440938
Perfect reads (non-overlap)	305270
Non-perfect reads (non-overlap)	135668
Overlap between all read sets	379500
Perfect reads (overlap)	345678
Non-perfect reads (overlap)	33822

7. Analysis of prediction of CyclomicsSeq dataset B

The resulting fastq files from the prediction of each CyclomicsSeq read of dataset B with all 10 models are mapped to the CHM13 v2 reference genome using minimap2. From the reads that map to chromosome 17, the following information is stored in tabular format: region, read id, NM tag, alignment length, cigar string and the type of model that is used for prediction. Additionally, it is stored whether one or more of the five mutations is present in the read. The number of reads for each model can be found in table 6. For each duplicate read id, we kept the read with the longest alignment length. We noticed that the number of reads that mapped to region 2 is very low compared to the other regions. Therefore, all reads mapping to region 2 are discarded and region 2 is left out of the analysis.

The consensus reads of the CyclomicsSeq pipeline for dataset B are retrieved in bam format and have been mapped to the CHM13 v2 reference genome using minimap2. From the reads that map to chromosome 17 and that have more than three repeats (based on presence of YM tag), the following information is stored: region, read id, NM tag, YM tag (repeat number in original CyclomicsSeq read), YR tag (read direction), alignment length, and the cigar string. The YM and YR tags are specific for CyclomicsSeq. Additionally, it is stored whether one or more of the five mutations is present in the read. All reads mapping to region 2 are discarded. We manually removed 23 reads that span both region 2 and 3 with a large deletion between the two regions. This is probably a mapping artifact as it is practically impossible for the CyclomicsSeq protocol to generate reads that span multiple regions.

We calculated the error rate per read ($\frac{NM}{Alignment\ length}$) and the error rate corrected for the number of mutations found per read ($\frac{NM-nr.mutations\ found}{Alignment\ length}$). We manually removed two reads from the Cycas Consensus dataset where the error rate corrected for the number of mutations found was -1. We then selected the reads that occur in each dataset (each model + Cycas Consensus) so we could perform a per-read analysis. This resulted in 291,100 unique reads per model. The non-overlap read set is analysed separately, these results can be found in the supplementary information. The number of reads per step can be found in table 7 and 8. Reads are considered perfect when no alignment mistakes are made besides the known mutations. The per-read error rate is calculated both at normal scale and at the Phred scale: $Q = -10 * \log_{10} P$ with P as the per-read error rate. The Phred Q-score of perfect reads (P = 0) is set at Q50 (P = 0.0005), to allow plotting at a logarithmic scale.

Table 7 – Number of reads for each step for the predicted consensus reads of CyclomicsSeq dataset B

	DNN 3-5X	**DNN 6-10X**	**DNN 11-15X**	**DNN 16-20X**	**DNN 3-20X**	**CNN 3-5X**	**CNN 6-10X**	**CNN 11-15X**	**CNN 16-20X**	**CNN 3-20X**
Split reads – raw input (MapQ NM filtered)	10378363	10378363	10378363	10378363	10378363	10378363	10378363	10378363	10378363	10378363
Nr. reads – Matrix input (chr17, >3x repeat)	409342	409342	409342	409342	409342	410122	410122	410122	410122	410122
Fastq reads – Model output	409342	409342	409342	409342	409342	410122	410122	410122	410122	410122
Mapped reads (all mapped to chr17)	409338	409338	409335	409336	409331	385490	410111	410103	410097	410113
Unmapped reads	4	4	7	6	11	24632	11	19	25	9

Mapped reads to chromosome 17	409338	409338	409335	409336	409331	385490	410111	410103	410097	410113
Mapped reads without region 2	408783	408783	408780	408781	408776	384978	409556	409548	409542	409558
After removal of duplicate read ids	395957	395957	395954	395955	395950	374460	396703	396695	396691	396705
Perfect reads – non overlap	379251	374592	369748	365902	371650	327472	325932	298585	257754	308427
Non perfect reads – non overlap	16706	21365	26206	30053	24300	46988	70771	98110	138937	88278
Overlap between all read sets (including Cycas Consensus)	292100	292100	292100	292100	292100	292100	292100	292100	292100	292100
Perfect reads - overlap	279889	276330	272474	269507	273987	255935	238997	219204	183832	226308
Non perfect reads – overlap	12211	15770	19626	22593	18113	36165	53103	72896	108268	65792

Table 8 – Number of reads for each step for the Cycas consensus reads of CyclomicsSeq dataset B.

Steps	**Cycas Consensus**
Mapped reads	1070547
Unmapped reads	0
Mapped reads to chromosome 17	513179
Mapped reads to other contigs	chr1: 18, chr2: 20, chr3: 9, chr4: 12, chr5: 11, chr6: 7, chr7: 12, chr8L 9, chr9: 15, chr10: 13, chr11: 7, chr12: 11, chf13: 7, chr14: 5, chr15: 3 chr16: 1, chr18: 7, chr19: 5, chr20: 9, chr21: 2, chr22: 3, chrX: 9 BB41C: 557173
Mapped reads to chromosome 17 & presence of YM tag	392542
After removal of duplicate read ids	390119
Filtering for reads that overlap multiple regions and NM score of -1	390071
Mapped reads without region 2	351163
Perfect reads – non overlap	312255
Non perfect reads – non overlap	38908
Overlap between all prediction read sets	292100
Perfect reads - overlap	259150
Non perfect reads – overlap	32950

8. Analysis of prediction of Genome Wide CyclomicsSeq (dataset C)

For each DNN model (DNN 3-5X, DNN 6-10X, DNN 11-15X, DNN 16-20X, DNN 3-20X) the resulting fastq files from the prediction of each CyclomicsSeq read of dataset C are mapped to the hs37d5 reference genome using minimap2. From the reads that map to the 23 chromosomes (not

including decoy contigs), the following information is stored in tabular format: region, read id, NM tag, alignment length, cigar string and the model that is used for prediction.

The consensus reads of the CyclomicsSeq pipeline for this dataset are retrieved in bam format and have been mapped to the hs37d5 reference genome using minimap2. From the reads that map to the 23 chromosomes (not including decoy contigs) and that have three or more repeats (based on presence of YM tag), the following information is stored in tabular format: read id, NM tag, alignment length, YM tag, YR tag, and the cigar string.

For each read set, predicted with one of the DNN models or with the Cycas Consensus method, only reads that do not overlap with a known SNP or indel of this individual are included. For each duplicate read id, only the read with the longest alignment length is kept. Finally, we selected the reads that occur in each dataset. This resulted in 34603 reads per model. The non-overlap dataset is analysed separately. The number of reads per step can be found in table 9 and table 10. We calculated the error rate per read adjusted for indels ($\frac{NM-\sum indels}{Alignment\ length}$) with $\sum indels$ being the number of inserted and deleted nucleotides based on the cigar string. Reads are considered perfect when no alignment mistakes are made besides indels. The per-read error rate is calculated at normal scale and at the Phred scale: $Q = -10 * \log_{10} P$ with P as the per-read error rate. The Phred Q-score of reads with no alignment mistakes (P = 0) is set at Q50 (P = 0.00001).

Table 9 – Number of reads for each step for the predicted consensus reads of CyclomicsSeq dataset C.

	DNN 3-5X	**DNN 6-10X**	**DNN 11-15X**	**DNN 16-20X**	**DNN 3-20X**
Split reads – raw input (MapQ NM filtered)	5426282	5426282	5426282	5426282	5426282
Nr. reads – Matrix input (>3x repeat)	75074	75074	75074	75074	75074
Fastq reads – Model output	77019	77026	77012	77036	77029
Unmapped reads	503	509	519	546	516
After removal of reads that overlap with known variants	46036	46040	46023	46012	46039
Mapped reads (normal contigs)	45921	45924	45910	45897	45922
After removal of duplicate read ids	45476	45481	45469	45461	45478
Perfect reads (non-overlap)	40764	39770	37178	35045	38877
Non-perfect reads (non-overlap)	4712	5771	8291	10416	6601
Overlap between all read sets (including Cycas Consensus)	34603	34603	34603	34603	34603
Perfect reads (overlap)	31149	30479	28742	27337	29875
Non-perfect reads (overlap)	3454	4124	5861	7266	4728

Table 10 – Number of reads for each step for the Cycas consensus reads of CyclomicsSeq dataset C.

	Cycas Consensus
Mapped reads to all chromosomes (no unmapped reads)	179464
After removal of reads that overlap with known variants	154047
Mapped reads (backbone)	113772
Mapped reads (normal contigs)	39115
After removal of duplicate read ids	36171
Perfect reads (non-overlap)	31916
Non-perfect reads (non-overlap)	4255
Overlap between all read sets	34603
Perfect reads (overlap)	30659
Non-perfect reads (overlap)	3944

9. Software

All code is written in Python (version 3.9.7) and bash. A complete list of used packages and tools can be found in table 11. Code can be found on GitHub (https://github.com/icdh99/cyclomics_consensus_models_inez).

Table 11 – Used software packages.

Name	Version
bedtools	2.30.0
cudatoolkit	11.2.2
cudnn	8.1.0.77
freebayes	0.9.21.7
h5py	3.2.1
keras	2.7.0
keras-tuner	1.1.3
matplotlib	3.5.1
medaka	1.7.2
minimap2	2.24
natsort	8.2.0
numpy	1.22.3
Pandas	1.4.2
perbase	0.8.3
pysam	0.18.0
Python	3.9.7
samtools	1.15
scikit-learn	1.1.1
Scipy	1.8.1
Seaborn	0.11.2
statannot	0.2.3
statannotations	0.5.0
tensorflow	2.7.0
tensorflow-gpu	2.9.1

10. Storage location of datasets

CyclomicsSeq dataset A
/hpc/compgen/projects/gw_cfdna/snv_qs/raw/CYC000025-extended-data

CyclomicsSeq dataset B
/hpc/compgen/projects/gw_cfdna/snv_qs/raw/000025-sup-extended-data.

CyclomicsSeq dataset C
/hpc/compgen/projects/gw_cfdna/gw_cyclomics/analysis/lchen/cycloseq-output/HC02_CYC36-20ng/CycasConsensus/MapqAndNMFilters
/hpc/compgen/projects/gw_cfdna/gw_cyclomics/analysis/lchen/cycloseq-output/HC02_CYC36-20ng/bams/HC02_CYC36-20ng.tagged.bam
Known variants of HC02:

/hpc/compgen/projects/gw_cfdna/gw_cyclomics/raw/gw_cyclomics/mixing_reference/HC02_20211119/HC02_processed/VCFS/VCF/MAR6281_processed.vcf.filtered_variants_dbnsfp_CosmicCodingMuts_v80_gonl.snps_indels.r5.vcf

11. List of abbreviations

cfDNA – Cell-free DNA
ctDNA – Circulating tumour DNA
DNN – Dense neural network
CNN – Convolutional neural network
FPR – False positive rate
RNN – Recurrent neural network
CCS – Circular Consensus Sequencing
ONT – Oxford Nanopore Technologies
PacBio – Pacific BioSciences
PCR – Polymerase chain reaction
MSA – Multiple sequence alignment
T2T – Telomere-to-Telomere
VAF – Variant allele frequency

Supplementary figures

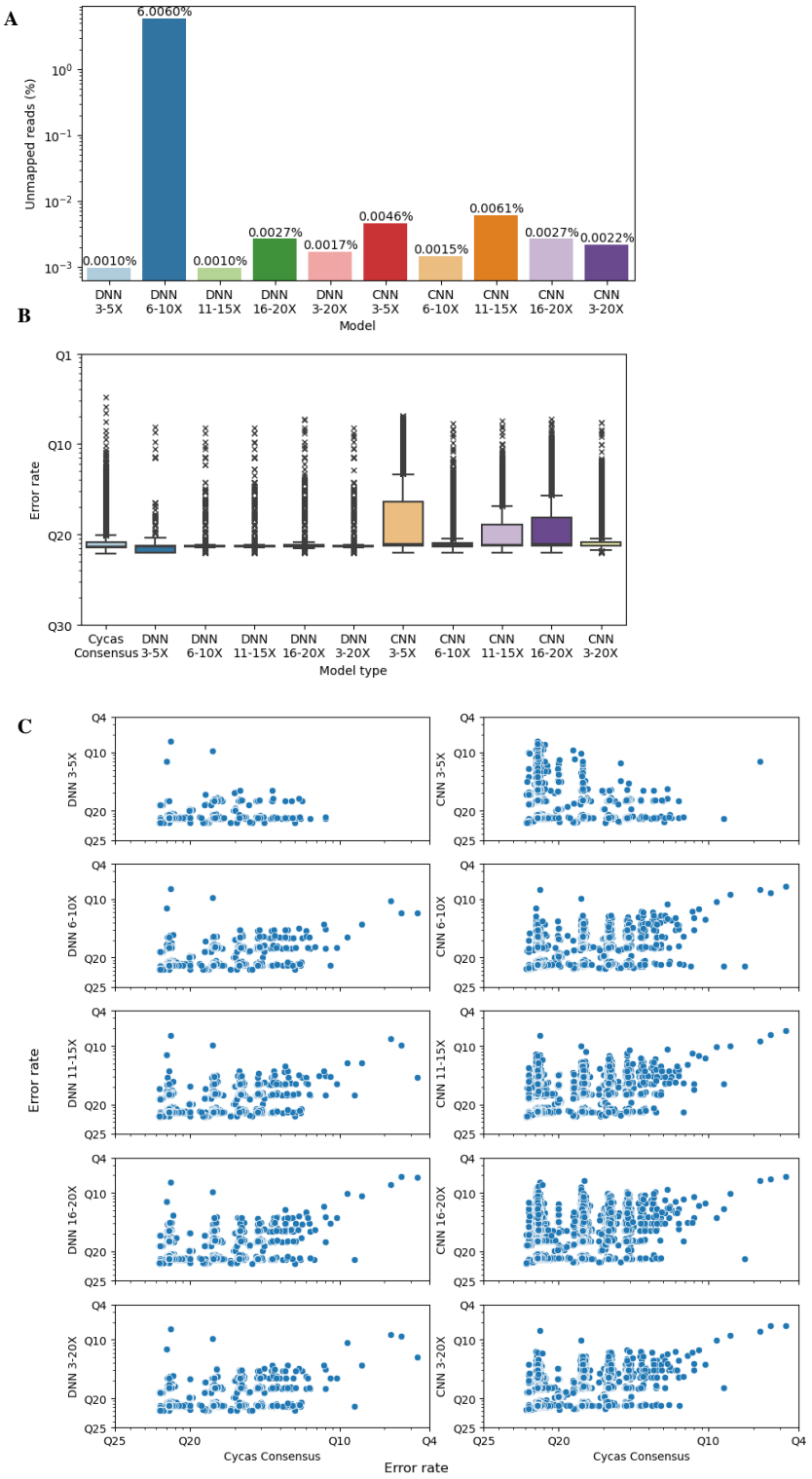


Figure S1 - Analysis of the consensus reads of CyclomicsSeq dataset B predicted with DNN or CNN T2T models trained on different coverages and with the Cycas Consensus pipeline. All error rates are corrected for presence of known mutations and only the overlapping subset of reads is analysed. A: percentage of unmapped reads for each DNN T2T model calculated relative to the number of fastq sequences predicted by each model. Y-axis is plotted logarithmically. B: Median per-read quality score for reads with at least one alignment mistake. Y-axis is plotted logarithmically. C: Per-read quality score comparison for each model versus the Cycas Consensus method. Only reads with at least one single nucleotide alignment mistake are shown. Both axes are plotted logarithmically.

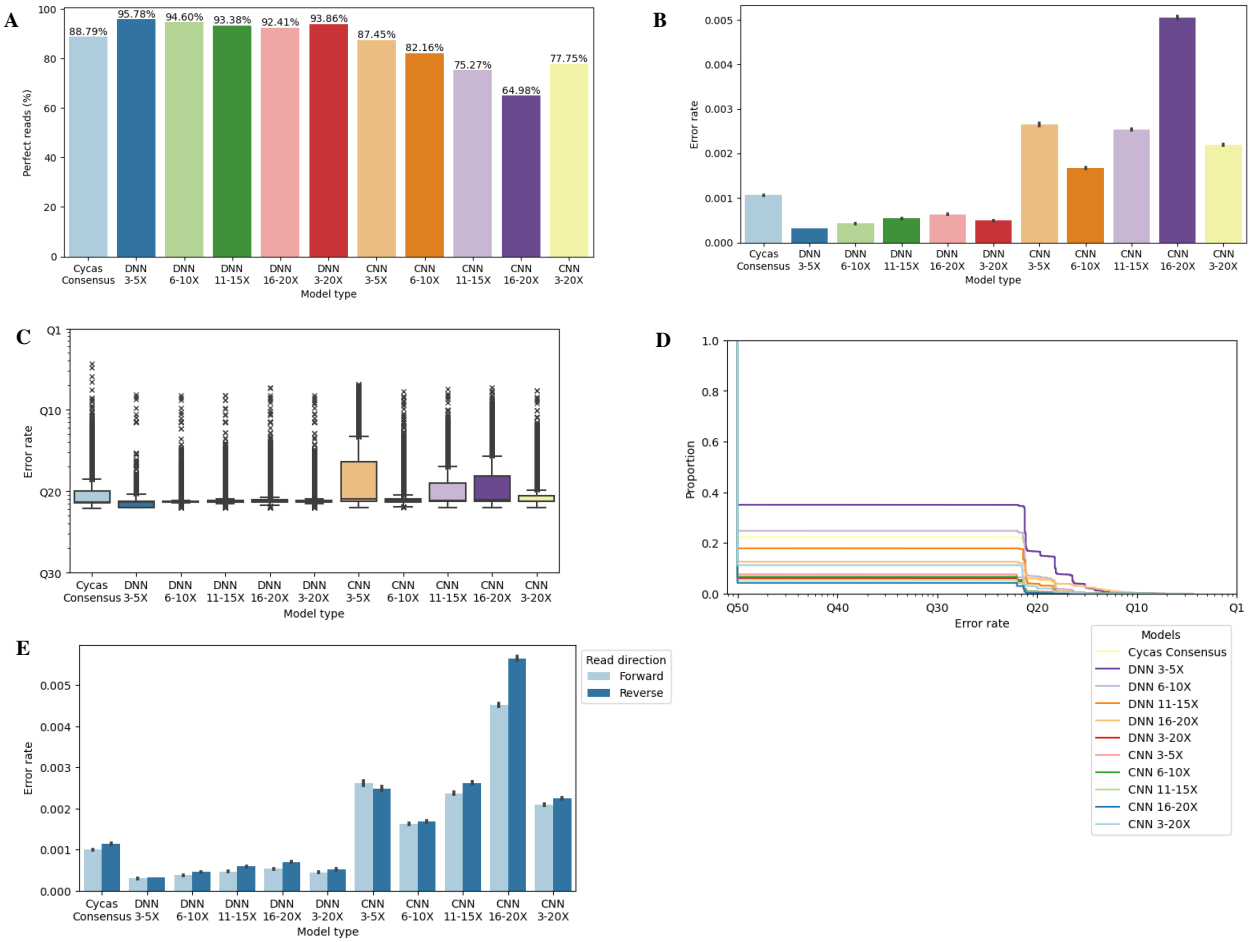


Figure S2 - Analysis of the consensus reads of CyclomicsSeq dataset B predicted with DNN or CNN T2T models trained on different coverages and with the Cycas Consensus pipeline. All error rates are corrected for presence of known mutations and all reads are analysed. A: Percentage of predicted reads with no alignment mistakes. B: Mean error rate for prediction of CyclomicsSeq dataset C with DNN and CNN T2T models trained on different coverages and with the Cycas Consensus pipeline. Error bars show 95% confidence interval. C: Median per-read quality score for reads with at least one alignment mistake. Y-axis is plotted logarithmically. D: Accumulative proportion of reads based on quality score. X-axis is plotted logarithmically. E: Median per-read error rate split for reads mapping to the forward or reverse strand.

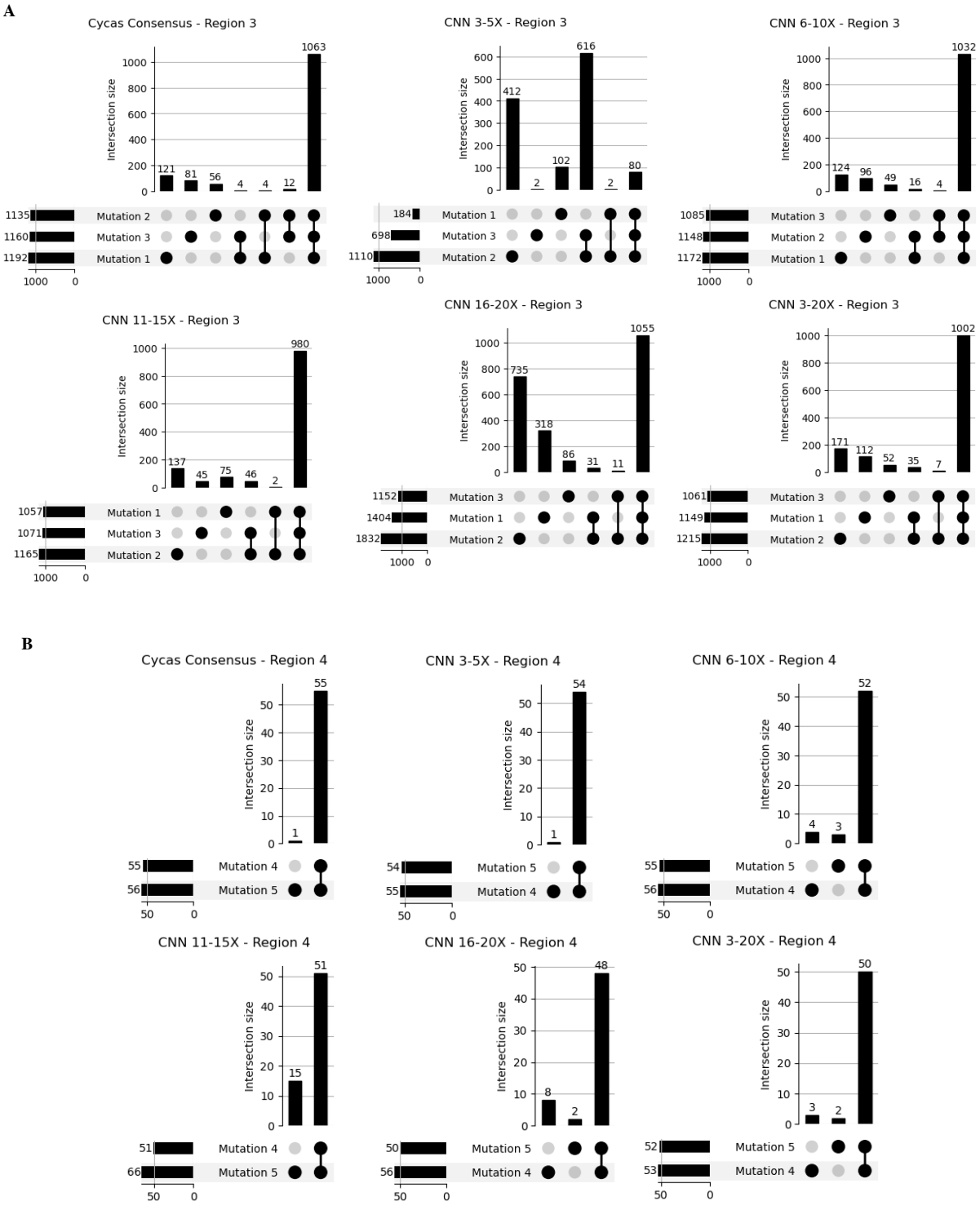


Figure S3 – Mutation analysis of CyclomicsSeq dataset C predicted with CNN T2T models trained on different coverage bins or with the Cycas Consensus method. Analysis is based only on the overlap between all datasets. C: Frequency of each combination of mutation 1, 2, and 3 in consensus reads predicted with the CNN T2T models or the Cycas Consensus method. Based on 117923 reads mapping to region 3. D: Frequency of each combination of mutation 4 and 5 in consensus reads predicted with the CNN T2T models or the Cycas Consensus method. Based on 65582 reads mapping to region 4.

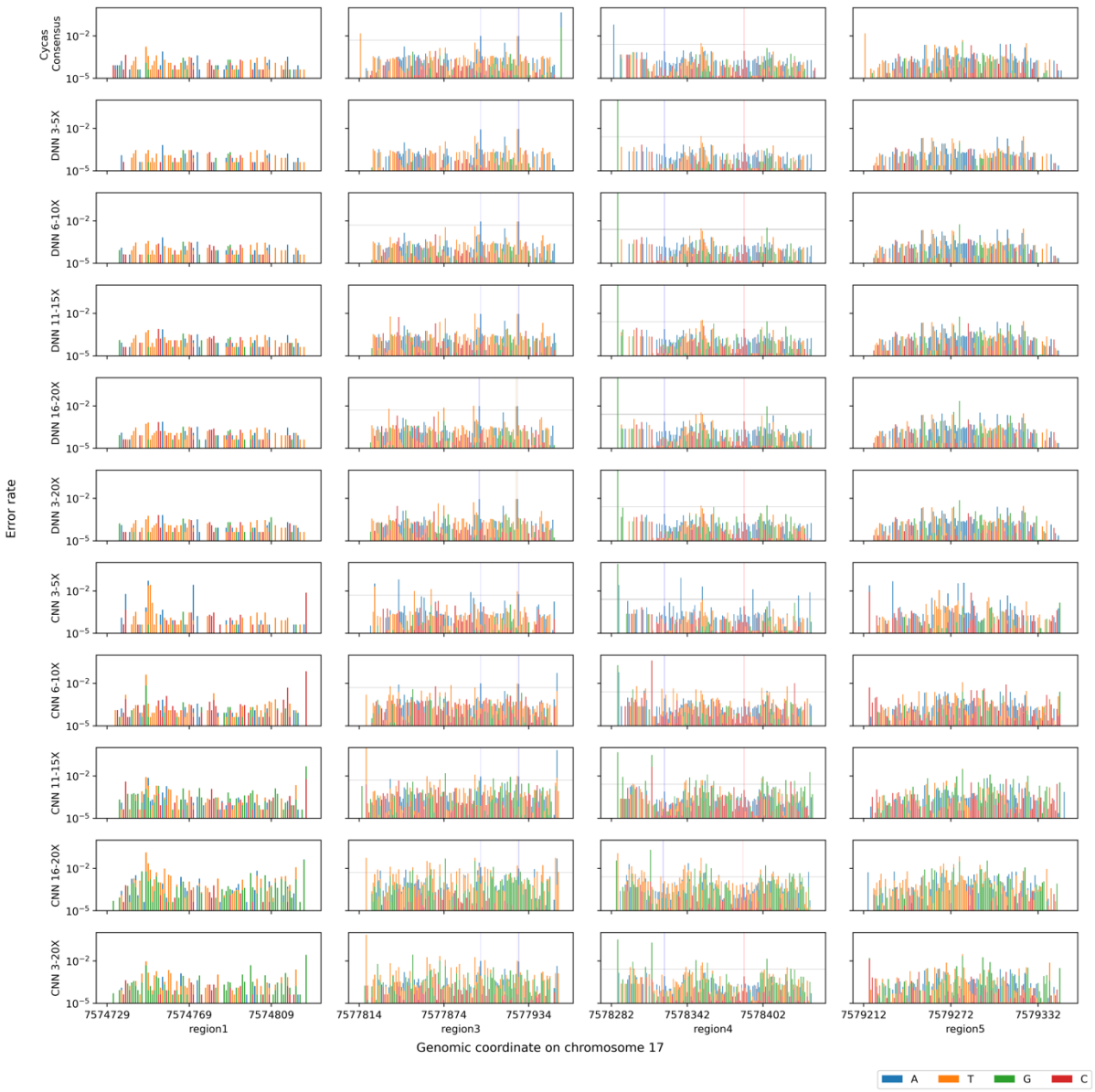


Figure S4 – Error rate per position per base for consensus reads predicted with the Cycas Consensus method or with the DNN and CNN T2T models. Only the overlapping subsets of reads between all models and the Cycas Consensus method are used. Vertical lines indicate mutation 1-4 coloured according to the alternative base. Horizontal grey lines indicate the expected variant allele frequency (0.5% for region 3, 0.25% for region 4.

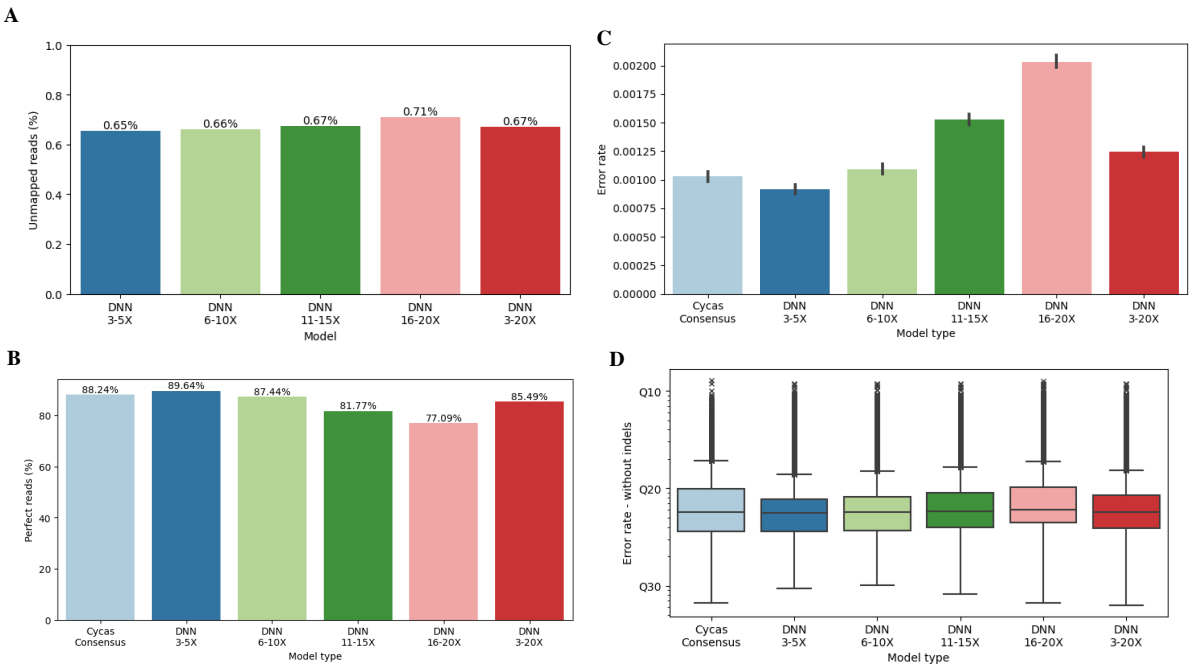


Figure S5 – Analysis of the consensus reads of genome wide cell-free DNA sequenced with CyclomicsSeq (dataset C). The consensus reads are predicted with DNN models trained on different coverages or with the Cycas Consensus method. The per-read error rate is corrected for the number of insertions and deletions per read. All reads output by the model and the Cycas Consensus methods are analysed. A: Percentage of unmapped reads for each of the DNN models. B: percentage of reads with no single nucleotide alignment mistakes. C: Mean per-read error rate. Error bars indicate 95% confidence interval. D: Median per-read quality score for reads with at least one single nucleotide alignment mistake. Y-axis is plotted logarithmically.

References

1. Corcoran RB, Chabner BA. Application of Cell-free DNA Analysis to Cancer Treatment. N Engl J Med. 2018 Nov 1;379(18):1754–65.

2. Marcozzi A, Jager M, Elferink M, Straver R, van Ginkel JH, Peltenburg B, et al. Accurate detection of circulating tumor DNA using nanopore consensus sequencing. Npj Genomic Med 2021 61. 2021 Dec;6(1):1–11.

3. Pessoa LS, Heringer M, Ferrer VP. ctDNA as a cancer biomarker: A broad overview. Crit Rev Oncol Hematol. 2020 Nov 1;155:103109.

4. Wan JCM, Massie C, Garcia-Corbacho J, Mouliere F, Brenton JD, Caldas C, et al. Liquid biopsies come of age: towards implementation of circulating tumour DNA. Nat Rev Cancer 2017 174. 2017 Feb;17(4):223–38.

5. Miga KH, Koren S, Rhie A, Vollger MR, Gershman A, Bzikadze A, et al. Telomere-to-telomere assembly of a complete human X chromosome. Nat 2020 5857823. 2020 Jul;585(7823):79–84.

6. Nurk S, Koren S, Rhie A, Rautiainen M, Bzikadze AV, Mikheenko A, et al. The complete sequence of a human genome. Science. 2022 Apr;376(6588):44–53.

7. Logsdon GA, Vollger MR, Eichler EE. Long-read human genome sequencing and its applications. Nat Rev Genet. 2020 Oct;21(10):597.

8. Leggett RM, Clark MD. A world of opportunities with nanopore sequencing. J Exp Bot. 2017 Nov;68(20):5419–29.

9. Ardui S, Ameur A, Vermeesch JR, Hestand MS. Single molecule real-time (SMRT) sequencing comes of age: applications and utilities for medical diagnostics. Nucleic Acids Res. 2018 Mar 16;46(5):2159–68.

10. Petersen LM, Martin IW, Moschetti WE, Kershaw CM, Tsongalis GJ. Third-Generation Sequencing in the Clinical Laboratory: Exploring the Advantages and Challenges of Nanopore Sequencing. J Clin Microbiol [Internet]. 2019 Oct;58(1). Available from: https://journals.asm.org/doi/abs/10.1128/JCM.01315-19

11. Pollard MO, Gurdasani D, Mentzer AJ, Porter T, Sandhu MS. Long reads: their purpose and place. Hum Mol Genet. 2018 Aug 1;27(R2):R234–41.

12. Rhoads A, Au KF. PacBio Sequencing and Its Applications. Genomics Proteomics Bioinformatics. 2015 Oct 1;13(5):278–89.

13. Wang Y, Zhao Y, Bollas A, Wang Y, Au KF. Nanopore sequencing technology, bioinformatics and applications. Nat Biotechnol 2021 3911. 2021 Nov;39(11):1348–65.

14. Xie S, Leung AWS, Zheng Z, Zhang D, Xiao C, Luo R, et al. Applications and potentials of nanopore sequencing in the (epi)genome and (epi)transcriptome era. The Innovation. 2021 Nov 28;2(4):100153.

15. Kono N, Arakawa K. Nanopore sequencing: Review of potential applications in functional genomics. Dev Growth Differ. 2019 Jun;61(5):316–26.

16. Rang FJ, Kloosterman WP, de Ridder J. From squiggle to basepair: Computational approaches for improving nanopore sequencing read accuracy. Genome Biol. 2018 Jul;19(1):1–11.

17. Wick RR, Judd LM, Holt KE. Performance of neural network basecalling tools for Oxford Nanopore sequencing. Genome Biol. 2019 Jun;20(1):1–10.

18. Senol Cali D, Kim JS, Ghose S, Alkan C, Mutlu O. Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions. Brief Bioinform. 2019 Jul 19;20(4):1542–59.

19. Lin B, Hui J, Mao H. Nanopore Technology and Its Applications in Gene Sequencing. Biosensors. 2021 Jul;11(7):214.

20. R10.3: the newest nanopore for high accuracy nanopore sequencing – now available in store [Internet]. [cited 2022 Dec 2]. Available from: https://nanoporetech.com/about-us/news/r103-newest-nanopore-high-accuracy-nanopore-sequencing-now-available-store

21. Dohm JC, Peters P, Stralis-Pavese N, Himmelbauer H. Benchmarking of long-read correction methods. NAR Genomics Bioinforma [Internet]. 2020 Jun;2(2). Available from: https://academic.oup.com/nargab/article/2/2/lqaa037/5843804

22. Espada R, Zarevski N, Dramé-Maigné A, Rondelez Y. Accurate gene consensus at low nanopore coverage. GigaScience. 2022 Jan 1;11:giac102.

23. Krishnakumar R, Sinha A, Bird SW, Jayamohan H, Edwards HS, Schoeniger JS, et al. Systematic and stochastic influences on the performance of the MinION nanopore sequencer across a range of nucleotide bias. Sci Rep 2018 81. 2018 Feb;8(1):1–13.

24. Nasrin S, Rahman A. Exploring systematic errors in sequencing technologies. Proc - 2019 IEEE 19th Int Conf Bioinforma Bioeng BIBE 2019. 2019 Oct;132–7.

25. Huang YT, Liu PY, Shih PW. Homopolish: a method for the removal of systematic errors in nanopore sequencing by homologous polishing. Genome Biol. 2021 Dec;22(1):1–17.

26. Monaco A, Pantaleo E, Amoroso N, Lacalamita A, Lo Giudice C, Fonzino A, et al. A primer on machine learning techniques for genomic applications. Comput Struct Biotechnol J. 2021 Jan;19:4345–59.

27. Dias R, Torkamani A. Artificial intelligence in clinical and genomic diagnostics. Genome Med. 2019 Nov;11(1):1–12.

28. Medaka [Internet]. Oxford Nanopore Technologies; 2022 [cited 2022 Dec 2]. Available from: https://github.com/nanoporetech/medaka

29. Baid G, Cook DE, Shafin K, Yun T, Llinares-López F, Berthet Q, et al. DeepConsensus improves the accuracy of sequences with a gap-aware sequence transformer. Nat Biotechnol. 2022 Sep 1;1–7.

30. Avsec Ž, Agarwal V, Visentin D, Ledsam JR, Grabska-Barwinska A, Taylor KR, et al. Effective gene expression prediction from sequence by integrating long-range interactions. Nat Methods 2021 1810. 2021 Oct;18(10):1196–203.

31. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, et al. Highly accurate protein structure prediction with AlphaFold. Nature. 2021 Aug;596(7873):583–9.

32. Ribeiro MT, Singh S, Guestrin C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier [Internet]. arXiv; 2016 [cited 2022 Dec 3]. Available from: http://arxiv.org/abs/1602.04938

33. Simonyan K, Vedaldi A, Zisserman A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps [Internet]. arXiv; 2014 [cited 2022 Dec 16]. Available from: http://arxiv.org/abs/1312.6034

34. Medaka [Internet]. Oxford Nanopore Technologies; 2022 [cited 2022 Dec 2]. Available from: https://github.com/nanoporetech/medaka/blob/481a2839f019ad0dce032b9a655b5e309968ca18/medaka/smolecule.py

35. medaka smolecule method versus consensus? · Issue #226 · nanoporetech/medaka [Internet]. GitHub. [cited 2022 Dec 2]. Available from: https://github.com/nanoporetech/medaka/issues/226

36. Loman NJ, Quick J, Simpson JT. A complete bacterial genome assembled de novo using only nanopore sequencing data. Nat Methods. 2015 Aug;12(8):733–5.

37. Telomere-to-telomere consortium CHM13 project [Internet]. MarBL; 2022 [cited 2022 Dec 11]. Available from: https://github.com/marbl/CHM13

38. Delahaye C, Nicolas J. Sequencing DNA with nanopores: Troubles and biases. PLoS ONE. 2021 Oct;16(10 October).