Sequencing errors in Nanopore data

Dario Bassi

Supervisors: Prof.Dr.Richard Neher & Dr.Marco Molari

FS 2022

Project Applied research in bioinformatics and systems biology II

1 Introduction

Being able to sequence DNA/RNA has opened up great horizons in understanding a multitude of biological aspects of life. In this context, nanopore sequencing is one of the most popular recent technologies because it enables the sequencing of "long" DNA/RNA fragments. However, it is prone to sequencing errors[Delahaye Clara and Jacques, 2021], and in order to be able to draw reliable conclusions from sequencing data, it is important to characterise these errors and quantify their probabilities.

This work aims precisely to characterise sequencing error on bacterial genome data collected in the laboratory. Using reads generated by nanpore sequencing, we focused on three different aspects: the probability of errors on single nucleotides and whether it is related to the quality score assigned by the sequencing technology, whether the error probability depends on the local nucleotide context, and what is the probability of misestimating the length of a homopolymer.

Our results[Dario, 2022] show that the quality score is reliable because the error does not depend on the context, and nanopore tends to underestimate the length of homopolymers. The conclusions reached will be considered in subsequent analyses of these data.

2 Materials and Methods

2.1 Data Generation and Setup

In this section, we will explain how the data we analysed were generated, starting with the experiments performed to produce them, through to their sequencing with nanopore and finally the alignment of the sequences to the reference genome.

2.1.1 Experimental Procedure

The data we analysed came from an experiment conducted within the Neher research group that aimed to study the mechanisms of antibiotic resistance development. In this experiment, the bacterium E. coli strain k-12 [Riley et al., 2006] was used. Isolated for the first time in 1922, it has become the most widely used bacterium in laboratories and research around

the world since it was discovered to be capable of genetic recombination by conjugation [Lederberg and Tatum, 1946] and generalised transduction [Lennox, 1955]. In the experiment to cultivate the population of bacteria, some bacteria preserved at -80°C in solid form were taken. The bacteria were then left to grow overnight in an incubation machine. Due to the high density, a sample of 0.5% was diluted in a new medium in order to have only a small part of the bacterial population. The new small bacterial population was placed in the morbidostat[Toprak et al., 2013] and allowed to grow. Then the experiment involved the addition of an antibiotic with three possible different cases: in the first, a high concentration of antibiotic was added, in the second, a low concentration of antibiotic was added, and the last involved the addition of no antibiotic. The resistance reaction of these bacteria was observed for an arbitrary time t. At the end of each time interval t, a small fraction of the bacteria was extracted in order to sequence the genome, observing and analysing which mutations occurred from the starting genome.

In order to quantify the probability of error, we used the data measured at the start of the experiment at time 0, referred to as *time_1*.

2.1.2 Nanopore sequencing[1]

Genome sequencing is carried out using a technology developed in the late 1980s, called nanopore. Nanopore is a third-generation sequencing technology, produced and developed by the company Oxford Nanopore Technologies (ONT)[Wang et al., 2021]. It allows DNA fragments over 300 Kbp in length to be sequenced.

How it works. This technology is based on a nanometer-sized protein pore. It is contained in an electrically resistant polymer membrane and acts as a biosensor. From figure 1 we can see the components of this particular membrane: ① the nanopore, ② the electrified membrane, ③ the motor protein and ④ the DNA fragment to be analysed. The membrane is immersed in an electrolyte solution to which a constant current is applied from the *cis part* to the *trans part*. The fragment of the DNA double helix is split by the helicase enzyme, so that only a single string of DNA passes inside the nanopore. The motor protein on the nanopore pushes the single string of DNA inside the nanopore from the *cis part* to the *trans part*. Changes in ionic current[*Ion Current - an overview — ScienceDirect Topics* 2022] inside the pore are signals that will be used to identify one of the four ACGT nucleotides.

[1]The flow cell version was **FLO-MIN 106D R9 version**, and the Guppy version was: **v.6.1.2**

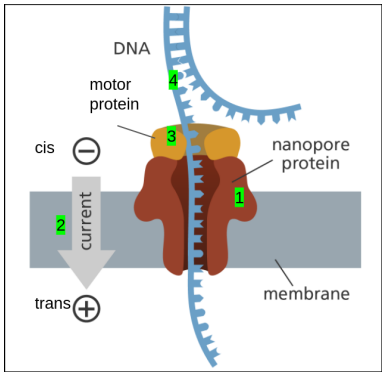


*Figure 1: Description of the Nanopore. We see the nanopore (**1**), the electrified membrane (**2**), the motor protein (**3**) and the fragment of DNA (**4**).*

Guppy and basecalling. The signals collected during the passage of the genome along the pore are decoded using Guppy. Guppy is a data processing toolkit that translates the received electric signal into nucleotides. This process is called *basecalling*. The results produced by the basecaller are written into a FASTQ file.

FASTQ file. The FASTQ file[*FASTQ format* 2022] format is used to store not only the DNA/RNA sequences but also the qualities values assigned by the basecaller. In figure 2 we

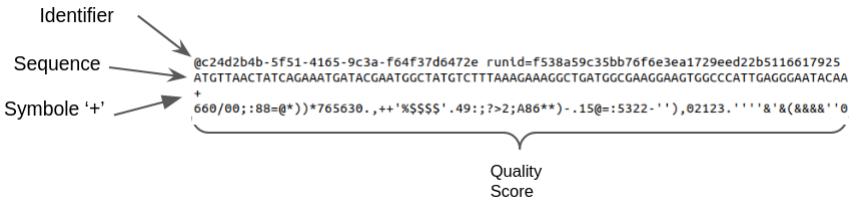


*Figure 2: Description of the content of a FASTQ file. On the first line there is **the identifier**, then **the sequences**, after that there is **Symbol "+"** and at the end **the qualities scores**.*

can see an example of a DNA fragment decoded by Guppy. The 4 standard elements written for each sequence are **the identifier**, **the sequence**, **the '+' sign** and **the quality value**.

- **Identifier:** This particular String always begins with the symbol @. It contains in addition to the code assigned by nanopore, an indication of the length of the DNA/RNA fragment and when the fragment was analysed.
- **Sequence:** Represents the sequence of the analysed DNA/RNA fragment

- **Symbol '+':** It is used to divide the biological sequence from the qualities values assigned by the basecaller
- **Quality Score:** The quality score represents the probability of the error associated with the DNA/RNA fragment read.

The quality score[*Phred quality score* 2022] is calculated with

$$Q = -10 \log_{10} P_{error} \tag{1}$$

and represents the numerical value of the probability that the identified nucleotide is incorrect. If the nucleotide has a low quality value, the basecaller is very uncertain of the nucleotide it has assigned to the signal and is very likely to make a mistake. In contrast, if the quality is high, the basecaller is very sure of the nucleotide it has assigned to the signal and therefore the probability of making a mistake is lower. In the FASTQ file this value is encoded in an ASCII character from 33 to 126, in order to save memory but at the same time be easily accessible.

2.1.3 Aligning the reads to a reference genome

Generating a reference genome with Trycycler. The reference genome is reconstructed using all the DNA/RNA fragments read with the help of the computational tool Trycycler[Wick et al., 2021]. From separate long DNA/RNA fragments, Trycycler forms one that at the end will be the longest, using four different assemblers: Canu[*Canu* 2022], Flye[Kolmogorov, 2022], Raven[*Raven* 2022] and Redbean[Ruan, 2022]. Each assembler reconstructs the genome, then Trycycler performs a multiple alignment of the genomes produced by taking all the parts of the genome that have the most similarities to the others and eventually obtaining a reference genome.

Aligning the reads to the reference genome. In order to compare the reads obtained with the reference genome and observe possible divergences, it is necessary to align the reads with the reference genome. We used Minimap2[Li, 2018] which is a tool that aligns several fragments to a reference genome; it has several modes but the main advantage it has over other aligners, such as BLASR, BWA-MEM, NGMLR and GMAP[Li, 2022], is that it is much faster and more accurate.
In order to map all reads to the found reference genome, we ran the following commands in the terminal after installing minimap2 and samtools[*Samtools* 2022]:

```
minimap2 -a -x map-ont -t {CPU} ref.fa query.fq > alignment.sam
samtools sort -@ {CPU} file.sam -> file.bam
samtools index -@ {CPU} file.bam -> file.bam.bai (Important for IGV Visualisation)
```

After mapping the fragments, we sorted and indexed the reads using samtools, and obtained a file of type *bam* and *bam.bai*. Using these two files, it was possible to visualise the result obtained with the IGV software. Interactively, we observed in detail the different reads at their positions on the genome with the different errors present on the reads. In figure 3 it is possible to observe the screenshot of our reads aligned with the genome. As one

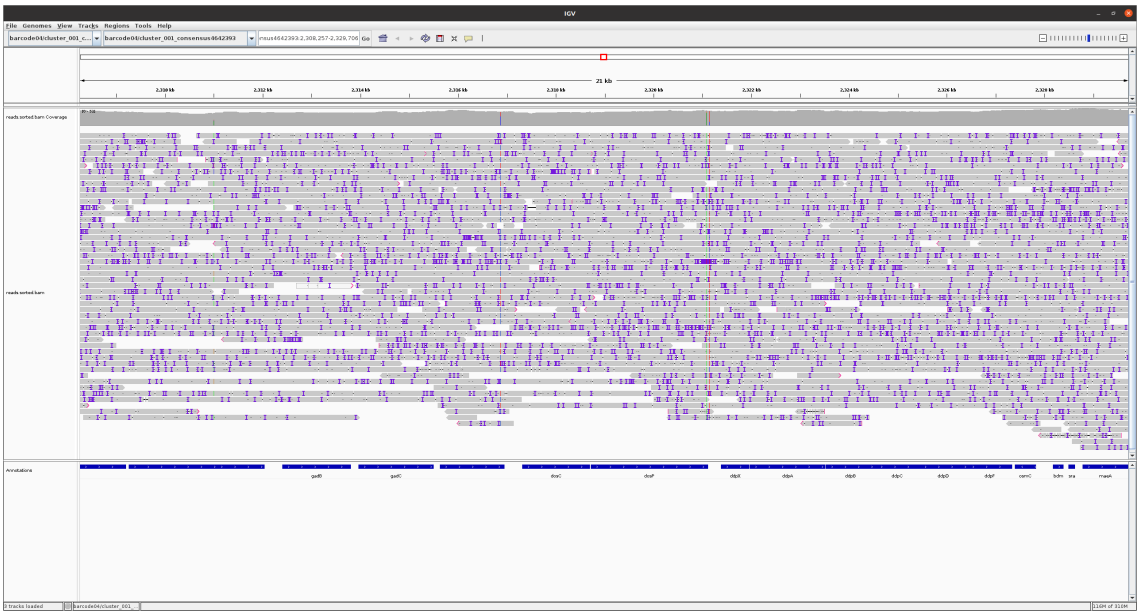


Figure 3: Screenshot of IGV visualisation software

can sees, there are different colours in the different fragments. They correspond to the errors in the fragments, i.e. insertions or deletions. Also, when there is no consensus on a particular nucleotide, the whole column is highlighted, so that all nucleotides of all reads that correspond to that position can be observed.

Sam/Bam file format. The bam file is the binary version of the sam file (Sequence Alignment Map)[*Samtools* 2022] and contains the biological information of sequences aligned with a reference genome. A sam file contains two sections: the first is called the Header section and the second the Alignments section. In the Header, which starts with the '@' symbol, there are the sample name, the sample length and the alignment method. Whereas in the Alignments section there are several fields[*Samtools* 2022]. The ones we made most

use of are POS(start position of the mapping on the reference), query_length(length of the mapping), CIGAR and is_forward(reads can be mapped forward or reverse).

CIGAR String. CIGAR string is an abbreviation for Concise Idiosyncratic Gapped Alignment Report and is a compressed representation of the alignment of a sequence to a reference sequence. The structure of a CIGAR string is as follows

$$<integer><desc>\ pairs \tag{2}$$

where *desc* contains the operations abbreviated to one character.

Example:

```
Ref:   AGCGGCCCTT--ATT
Read:  AGCGG--CTTGGATT
Cigar: 120H5M2D3M2I3M
```

At the beginning or end of each fragment it is possible to find a hard clipping that corresponds to the part of the fragment that is not alignable to the reference and is not saved, or a soft clipping that corresponds to the part of the fragment that is not alignable but is saved in the sequence, or no clipping at all. Then we can find one of the operations described in Table 1

desc	Description
M	Match; can be a sequence match or a mismatch
I	Insertion into the reference
D	Deletion from the reference
H	Hard clip; clipped sequence not present in SEQ
S	Soft clip; clipped sequence present in SEQ
N	Skipped region from the reference
P	Padding; padded area in the read not in the reference
=	Read Match
X	Read mismatch

Table 1: Content of a CIGAR string

In order to read the bam file, we used the Pysam library[*Samtools* 2022], which reads, manipulates and writes genomic datasets. Among the different functions used, we would

like to mention *pileup* and *fetch*. Pileup sorts the reads starting from the first position on which they appear on the reference genome: it therefore reads vertically. This way we have access to the observed nucleotide and its quality in each read for each of the positions on the reference genome. In contrast, fetch does not change the order in which the reads are saved in the bam file, but reads the various reads horizontally. This means that we read all the information contained in one read, before moving on to the next.

2.2 Error Analysis

In this section, we will introduce the concepts used in this analysis and clarify how they were calculated. More precisely, we will initially discuss the probability of errors in relation to the quality score assigned by nanopore. Next we will test whether the error probabilities depend on the local nucleotide context using the conditional entropy of the error in several different contexts, and finally we will measure the probability of incorrectly estimating the length of a homopolymer. The results obtained will be presented and discussed in the section 3.

2.2.1 Frequency of Error

Quality score distribution for all reads. Initially, we wanted to observe the distribution of the quality scores in our reads. To this end we parsed all the reads and saved in a dictionary the count of quality scores separately for each of the four nucleotides. The resulting distributions are discussed in section 3.1.1 figure 4.

Probability of errors in relation to the quality score assigned by nanopore. Then we wanted to look at the frequency of error for each quality value assigned to the nucleotides of all the reads. We only consider alignable regions, corresponding to matches (M) in the CIGAR string. To this end, we first created a 4x4x2 nested dictionary indexed by a tuple of three values: true nucleotide on the reference genome, nucleotide observed on the read, orientation of the read (forward/reverse). For each key, the corresponding value was a dictionary with the number of times a certain quality appeared when reading the fragments:[2]

$$\Gamma = \{(N_R, N_Q, s) \longrightarrow \{q \longrightarrow N_q\}\} \tag{3}$$

[2]For consistency when considering *reverse* reads we take as N_R the complementary nucleotide on the reference genome, i.e. the one on the reverse strand.

N_R = Nucleotide on the reference (genome)
N_Q = Nucleotide on the query (read)
s = Orientation of the read, either Forward or Reverse
q = Quality score
N_q = Count of nucleotides with quality q
We then calculated the error rate defined as:

$$p_{error}(N_R, s, q) = \frac{\#errors(N_R, s, q)}{\#reads(N_R, s, q)} \tag{4}$$

where

$$\textbf{\#errors}(N_R, s, q) = \sum_{N_Q \neq N_R} \Gamma[(N_R, N_Q, s)][q]$$

$$\textbf{\#reads}(N_R, s, q) = \sum_{N_Q} \Gamma[(N_R, N_Q, s)][q]$$

Using the matrix created above, we caculated for each nucleotide and for each quality the percentage of error. Then for each probability of error we approximated the standard error of the mean (considering the number of errors as binomially-distributed) as:

$$\sigma_{error} = \sqrt{\frac{p_{error}(1 - p_{error})}{\#reads}} \tag{5}$$

Finally, we created a plot where the x-axis had the values of the qualities scores and the y-axis had the values of the probabilities of the errors for each nucleotide, with its standard deviation. In addition to this we added the line *Phred score quality* representing the expected error calculated with

$$P_{error} = 10^{-q/10} \tag{6}$$

to show whether the calculated error probability is better, worse or equal to the expected error. The results obtained will be discussed in Section 3.1.2 in figure 5.

Error Bias. After calculating the probability of error, we wondered if there was a bias in the error, i.e. if upon mis-reading a nucleotide some errors were more frequent than others. To do this, we initialised a pair of 4x4 matrices M (one for forward and one for reverse reads) whose first index was the nucleotide on the reference genome and second index the nucleotide on the read.

$$M_s(N_R, N_Q) = \frac{\sum_q \Gamma[(N_R, N_Q, s)][q]}{\sum_{q, N_Q} \Gamma[(N_R, N_Q, s)][q]} \tag{7}$$

This matrix was normalized on the second index, having on the diagonal all the cases where the read is correct ($N_R = N_Q$), instead all the others were all the cases that the machine got wrong ($N_R \neq N_Q$). We did this type of plot for both Forward and Reverse reads. We represented this graphically with *matshow* in the Section 3.1.3 in figure 6.

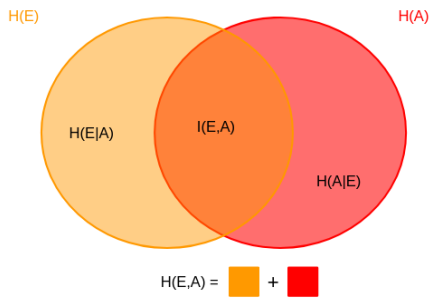
2.2.2 Does the contex matter?

The second part of our analysis concerned context. We asked ourselves whether context, i.e. the presence of a particular nucleotide or nucleotide sequence, markedly increases the possibility of an error. Are there nucleotides or nucleotide sequences that are more frequently wrong? To find this type of error, we used Shannon's Information Theory. In information theory[MacKay, n.d.], there is a key concept, namely entropy, which quantifies the level of "uncertainty" of an event. In our analysis, we will use the concept of conditional entropy[*Conditional entropy* 2022], which is defined as the amount of information required to know the value of a random variable while knowing already another one.
In our case we consider the process of reading a particular position on a nucleotide sequence and define two random variable:

$E \longrightarrow$ binary variable indicating whether the read has an error on the considered position

$A \longrightarrow$ nucleotide context around the considered position

We call e the value of the variable associated with the error, whereas a is associated with the context. Visualising our problem with a Venn diagram, we would like to find the value of $H(E|A)$ that reflects the amount of uncertainty related to the error, that is independent from the context.



Knowing the joint entropy

$$H(E, A) = -\sum_{e,a} P(e, a) \log_2 P(e, a) \tag{8}$$

we can find the conditional entropy with

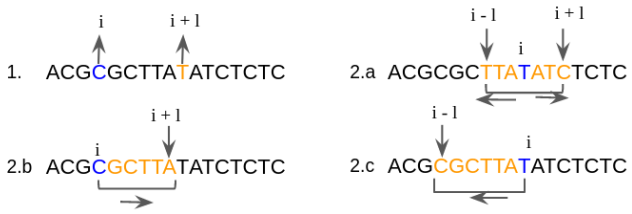
$$H(E|A) = H(E,A) - H(A) \tag{9}$$

$$H(E|A) = -\sum_{e,a} P(e,a)\log_2 P(e,a) + \sum_{a} P(a)\log_2 P(a) \tag{10}$$

$$H(E|A) = -\sum_{e,a} P(e,a)\log_2 P(e,a) + \sum_{ea} P(e,a)\log_2 P(a) \tag{11}$$

$$H(E|A) = -\sum_{e,a} P(e,a)\log_2 \frac{P(e,a)}{P(a)} \tag{12}$$

In this analysis we have considered four types of context. The first context (**1**) consists of the nucleotide at the considered position i plus another nucleotide at distance l. In this case the context always consists of two nucleotides (or one if $l = 0$) at variable distance. Instead, the second type of context concerns information contained in a sequence of nucleotides of variable size. For this second type, we have considered 3 types of sequences: the first sequence (**2.a**) starts from nucleotide i and is expanded by progressively adding all nucleotides up to distance l, consisting of the interval (*i-l;i+l*), the second sequence (**2.b**), on the other hand, starts from a nucleotide i and is expanded by adding the subsequent nucleotides with *positive distance l (i;i+l)*, and finally the third sequence (**2.c**) starts from a nucleotide i and is enlarged by adding the previous nucleotides with *negative distance l (i-l;i)*[3]



For each read r and at each position i on the reads, we can find the error $e_{r,i}$ and the context $a_{r,i}$. From the above definitions of contexts, we have extracted a context $a_{r,i}$ which is different depending on which case we have considered (**1, 2.a, 2.b and 2.c**) but which is still a sequence of nucleotides. Furthermore, knowing whether the nucleotide at position i was identical to the nucleotide on the reference genome, we found the error $e_{r,i}$.

At this point, thanks to these two variables, we were able to calculate the number of times we saw, or did not see, a given context associated with an error. We did this by constructing

[3]The positions and contexts are always relative to the reference genome.

a dictionary N(a,e)

$$\begin{aligned} N(a,e) &= \text{n. times context } a \text{ is observed with error } e \\ &= \sum_{r,i} \delta(a_{r,i} = a)\ \delta(e_{r,i} = e) \end{aligned}$$

The joint probability distribution [4] is simply given by:

$$P(a,e) = \frac{N(a,e)}{\sum_{a,e} N(a,e)} \tag{13}$$

and the marginal probabilities $P(a)$ and $P(e)$ are defined as

$$P(a) = \sum_{e} P(a,e)\,, \quad P(e) = \sum_{a} P(a,e) \tag{14}$$

where $P(a)$ represents the frequency with which we see a certain context and $P(e)$ indicates the probability of the error regardless of contexts. From the value $P(e)$ we were able to calculate the entropy of the error $H(E)$ as

$$H(E) = -\sum_{e=0,1} P(e)\ \log_2 P(e) \tag{15}$$

We calculated the conditional entropy in the following cases:

- Context (**1**) with the following distances l of (-15,-10,-5,-4,-3,-2,-1,0,1,2,3,4,5,10,15). The results can be seen in Section 3.2 in figure 7.
- Context (**2.a**) with the following distances l of (0,1,2,3,4,5,6). The results are in Section 5 in figure 10.
- Context (**2.b**) with the following distances l of (0,1,2,3,4,5,6,7,8). The results are in Section 5 in figure 11a.
- Context (**2.c**) with the following distances l of (0,1,2,3,4,5,6,7,8). The results are in Section 5 in figure 11b

2.2.3 Homopolymer Insertion/Deletion

The last part of our analysis concerned the error that occurs most frequently in homopolymers. Why are we interested in this type of error? Nanopore tends to recognise with more

[4] We calculated these probabilities only for the positions i on the reads that correspond to matches (M). In the context **2.c** we considered the context a as contexts of all read reverses

difficulty DNA/RNA fragments that have homopolymer regions, i.e. those regions that have chains of varying lengths of identical nucleotides. This happens because during homopolymer transition the ionic signal does not change, which is why nanopore will have more difficulty recognising homopolymer lengths. In general, it will differentiate but sometimes it may make errors. Because of this difficulty, we will probably find errors such as insertions or deletions.

Distribution of homopolymer lengths on the reference genome. In the first part of this last exercise, we searched in the reference genome for all possible homopolymers. We created a dictionary:

$$\{(n, L) \longrightarrow [(p_{start}, p_{end})]\} \tag{16}$$

n = Nucleotide of the homopolymer (A,C,T,or G)
L = Length of the homopolymer
p_{start} = Start position of the homo-polymer on the reference genome
p_{end} = End position of the homo-polymer on the reference genome

which saved for each length and type of homopolymer, a list containing tuples with the start and end position of the homopolymers. Using this dictionary, we were able to calculate the number of homopolymers of each length, creating a plot with an x-axis "length of homo-polymer" and a y-axis "number of homo-polymers per nucleotide". The results obtained can be seen in Section 3.3 in figure 8.

Errors in homopolymer length. After estimating the distribution of homopolymer lengths in the reference genome, we turned to the analysis of the errors that Nanopore makes in estimating the correct length of the homopolymer. In practice, for every read including an homopolymer of nucleotide n and real length L, we can evaluate the discrepancy Δ between the real length (assumed to be the one on the reference genome) and the one observed on the read. Three different cases are possible:

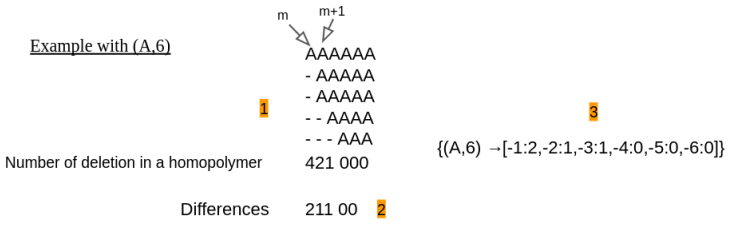
$$\begin{cases} \Delta > 0 & \text{homopolymer on query is longer than on reference genome } (\textit{insertion}) \\ \Delta = 0 & \text{homopolymer on query and reference genome have the same length} \\ L \leq \Delta < 0 & \text{homopolymer on query is shorter than on reference genome } (\textit{deletion}) \end{cases}$$

For every value of n, L we are interested in knowing the distribution of values of Δ. We call $N(n, L, \Delta)$ the number of times we observe a particular value of Δ for homopolymers of type n, L. Values of $\Delta < 0$ correspond to *deletions* in the mapped read, and $\Delta > 0$ to *insertions*. We therefore treat these two cases separately.

`Insertions.` To evaluate $N(n, L, \Delta)$ for $\Delta > 0$ we inspected reads containing insertions

on the edges of homopolymers. For every such read, we check that the insertion adds nucleotides of the same type to the homopolymer. We evaluated Δ as the number of extra inserted nucleotides of the same kind.

`Deletions.` To evaluate $N(n, L, \Delta)$ for $\Delta < 0$ we used the number of deletions in the pileup of homopolymer sequences (**1**). Exploiting the fact that deletions tend to be mapped all on the same side of the homopolymer by minimap2, we can get the number of reads by looking at the difference between the number of deletions between any two subsequent position in the homopolymer (**2**). In general, the number of reads with $\Delta = -m$ is equal to the difference between the number of gaps at position m and position $m + 1$ (**3**).



Finally, given the total number of reads $N(n, L)$ of homopolymers of type n, L, the value of $N(n, L, \Delta = 0)$ is simply given by:

$$N(n, L, \Delta = 0) = N(n, L) - \sum_{\Delta \neq 0} N(n, L, \Delta)$$

The distribution of values of Δ is discussed in Section 3.3 in figure 9.

3 Results

In this section, we will show and discuss the results obtained from the 2.2 section. We will start with the error frequency which includes the distribution of the assigned quality values, the error probability of assigning the quality of a nucleotide and the error matrix for Forward and Reverse reads. Furthermore we will then discuss the results obtained by analysing the contexts and finally show the distribution of the lengths of the homopolymers as well as their actual lengths, showing which errors appear most frequently.

3.1 Frequency of Error

3.1.1 Quality Score Distribution

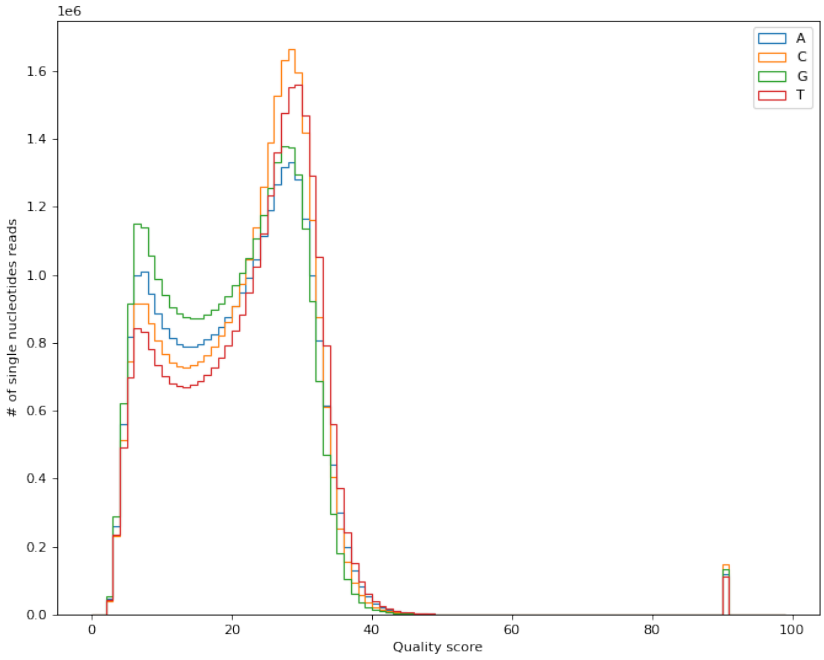


Figure 4: Distribution of the assigned qualities scores by the basecaller for the nucleotides ACGT in all reads

The quality score distribution calculated for all readings is shown in fig. 4. Two interesting aspects can be seen in this figure. This distribution is bimodal, in fact we can see two distinct peaks around quality 10 and around quality 30. In the first peak we see that nucleotides A and G are much more numerous than nucleotides C and T, but in the second peak there is a change and we find that the most numerous nucleotide pair is formed by nucleotides C and T. We also find a small signal with quality values of 93 that we were unable to identify. Finding this marked difference between these two peaks gives us a first indication of the quality of the basecaller's recognition of nucleotides: at low quality values, the probability of making mistakes is very high, so having many nucleotides of type A and G means that they are likely to make mistakes more frequently. This behaviour is in line with the result found in the second peak, where fewer nucleotides of quality 30 are found than in the other two nucleotides. Secondly, this distribution can be helpful in setting a threshold value for

quality filtering, if the quality score is correlated with error probability. In the next section we verified that this indeed is the case.

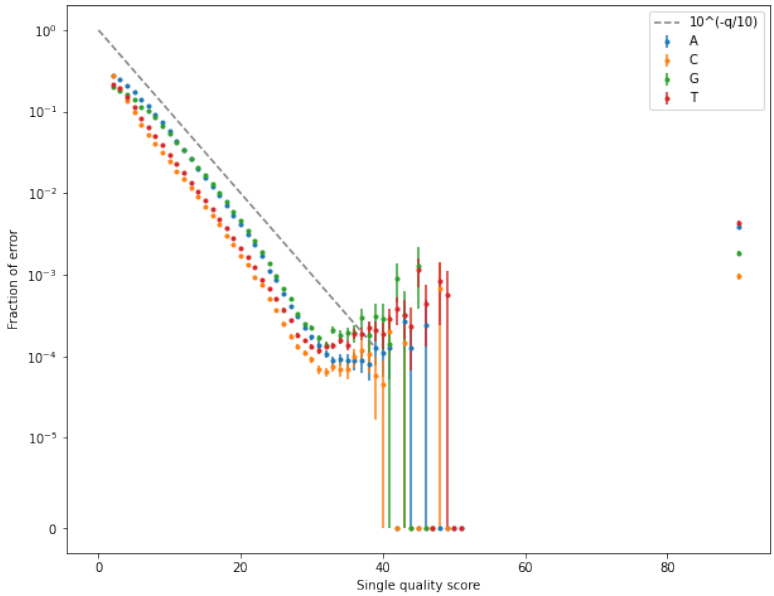
3.1.2 Error probability vs quality score

After the analysis of the quality distribution, we focused on the probability of finding an error for each quality value assigned to the nucleotides. For this second analysis, we considered two cases: the first, where there was only *Forward* reads and the second with only *Reverse* reads. We wanted to verify that there was no differences between the two types of readings. In figure 5a, we see the error probabilities with error bars for each quality value. The grey line represents the probability of expected error (eq.6). As shown in Table 2 the expected error

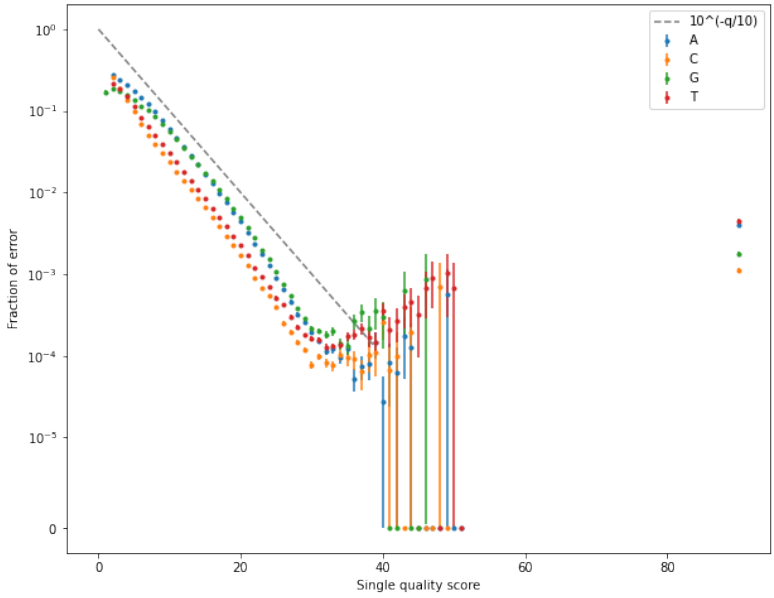
Phred Quality Score	Incorrect base call
10	1 in 10
20	1 in 100
30	1 in 1000
40	1 in 10000
50	1 in 100000
60	1 in 1000000

Table 2: Table of probability of incorrect base call giving the Phred Quality Score

for a nucleotide with a quality of 20, would be 1 wrong nucleotide per 100 nucleotides. If you get an error probability above the Phred Quality Score line, it means that the basecaller got it wrong more frequently, but if it is below it means that the basecaller got it wrong less than the expected error. In our two plots, we can clearly see that the error probabilities for nucleotides are below the grey line and this implies that (at least on alignable basepairs) the basecaller makes fewer mistakes than expected. In addition to this, we can see in the two graphs that the probability of errors for nucleotides A and G are higher than for nucleotides C and T; this allows us to confirm that C and T pairs are called more accurately. When a quality value of 40 is reached, the error bars are longer, and this is due to the fact that there are very few nucleotides with qualities above 40, so the error range is greater. Thanks to the results of the quality distribution and the results obtained by calculating the probability of error, a reasonable quality filter strategy would be to discard reads with quality score lower than a threshold value of around 20. It is reasonable to think of creating this limit because the probability of a nucloetide being named incorrectly is 1 in every $\simeq 500$.



(a) probability of error for Forward reads for each quality score



(b) probability of error for Reverse reads for each quality score

Figure 5: Probability of error on single nucleotides at each quality score for forward and reverse reads

3.1.3 Matrix of error

We concluded our analysis of error frequency by looking for biases in the sequencing error, i.e. whether a particular nucleotide was more likely to be read when an error was made. Figure

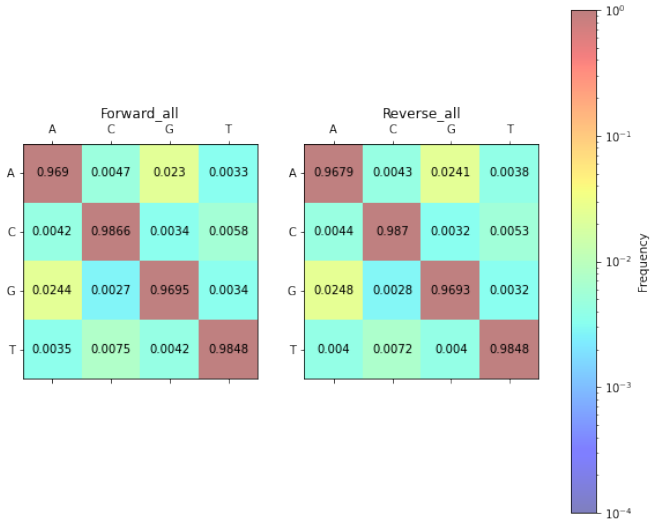


Figure 6: The figure shows the probability matrix of errors. In the indices on the y-axis we have the true nucleotide on the reference genome and on the x-axis we have the nucleotide read on the nanopore reads. The errors of the probabilities are normalised on the x-axis. On the diagonal are the probability of correctly recognizing nucleotides, and error probabilities are off-diagonal.

6 shows the results obtained for the reads *Forward* and *Reverse*. Both matrices have very similar values. Thanks to the colours, we can clearly see that the largest errors are around 2.3% - 2.4%; they occur when instead of reading the true nucleotide A the nucleotide G is read and instead of reading the true nucleotide G the nucleotide A is read. In all other cases, the errors are of the order of magnitude of 0.3%-0.75%. On the diagonal, we observe that the correct nucleotides have a high probability of success. These data give us confirmation that a bias in the error is present; if an error does occur then it is most likely to be AG and GA. This is consistent with what was also found for an older version of Guppy [Delahaye Clara and Jacques, 2021].

3.2 Information of the Contex

Our analysis shifted to contexts and we asked whether having a specific context increased the likelihood of errors. The contexts we studied can be found in Section 2.2.2.

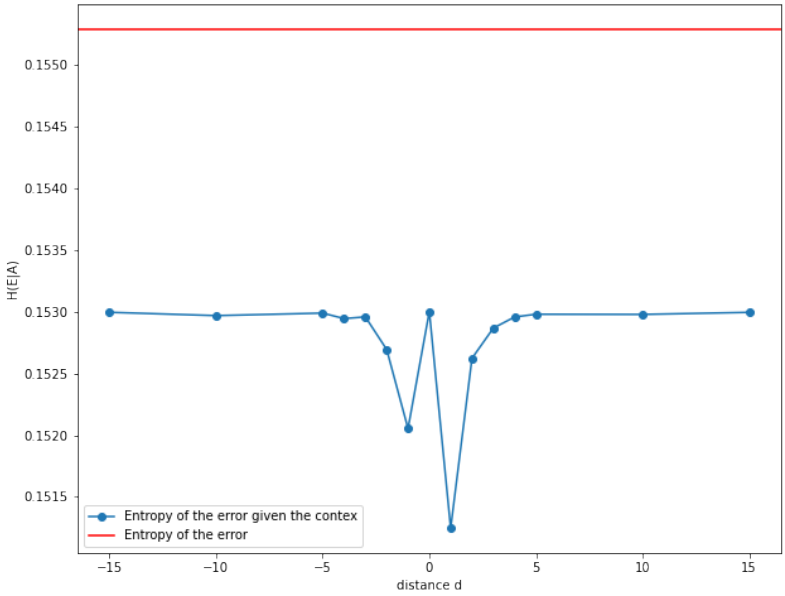


*Figure 7: Conditional Entropy of the error given the nucleotide context. Context **1** is considered, as defined in section 2.2.2, which consist of the nucleotide on the position on which the potential error occurs, plus a second nucleotide at distance d. To generate the figure only for forward reads are considered.*

`Context 1`[5] For the context (**1**) we obtained the figure 7. The red line represents the information contained in the error *H(E)*, i.e. the surprise of the error itself. The blue line represents the information of the conditional entropy between two nucleotides at distance *d*. For distances *0* the context only consists of the nucleotide on which we see the error, whereas for $d \neq 0$ the context also includes the nucleotide at distance *d*. From the analysis in the 2.2.1 section, we know that when faced with an error, the probability is greater if it is either nucleotide A or nucleotide G than if it is the other two nucleotides. If we had obtained the result in which context has no influence at distance *0*, we would have seen the value of the blue line at the same level as the red line, and this would have meant that we had no

[5]In the figure 7, conditional entropy and error entropy are only for **forward reads**.

effect, i.e. we were unrelated to the error. Knowing that the two nucleotides A and G have a higher probability of being wrong decreases our "uncertainty" and for this reason, the value at distance *0* of the conditional entropy is lower than the value of the entropy of the error itself. At distance 1 from nucleotide i, the surprise of the error knowing the nucleotide at distance 1 is about 2% less than when the context consists of the nucleotide on which we see the error. At distance 2 the surprise contained in the error is less than at distance 1 and in fact the gap has narrowed to 0.5 and finally to distance 5 when the gap is practically null and the surprise contained in the error has practically disappeared. At negative distances we observe the same behaviour with slightly higher values and even in this second case at distance -5 the surprise of the error given a context disappears. For distances greater than 5, conditional entropy does not add any new surprise to the error, so we can observe that the values are identical to those at distance 0. This behaviour is also motivated by the fact that in a pore there are at most 5 nucleotides at a time, and this can be clearly seen for positive and negative distances.

This plot allows us to conclude that having a certain context over another is not very informative, in fact the error is not influenced in a major way. In the section 5, one can see the results of the analysis of the other contexts that confirmed our conclusion.

3.3 Homopolymer error in Nanopore

The last part of our analysis concerned homopolymers and their errors. After identifying homopolymers on the reference genome, we analysed their distribution according to their length. From the figure 8, we generally observe a decreasing exponential trend of homopolymers, i.e. many homopolymers of lengths 2 and 3 and few homopolymers of lengths 8 and 9. For homopolymers of length ≥ 4 the pairs of nucleotides that have the most homopolymers are **A** and **T**. The same result was also observed in the article[Delahaye Clara and Jacques, 2021] in Figure 7.

Next, we focused on the length of homopolymers. Knowing that nanopore has great difficulty in accurately identifying the length of homopolymers, we wanted to observe the trend in the actual length of homopolymers to identify which error, insertion or deletion, occurs most frequently. The figure 9 represents the trend of the true homopolymer lengths for nucleotides $ACGT$. The figure confirms that there is no major nucleotide bias as they all seem to behave in the same way. However, what is important, is the actual length of the homopolymers. If up to homopolymer length 4, the real length corresponded to the

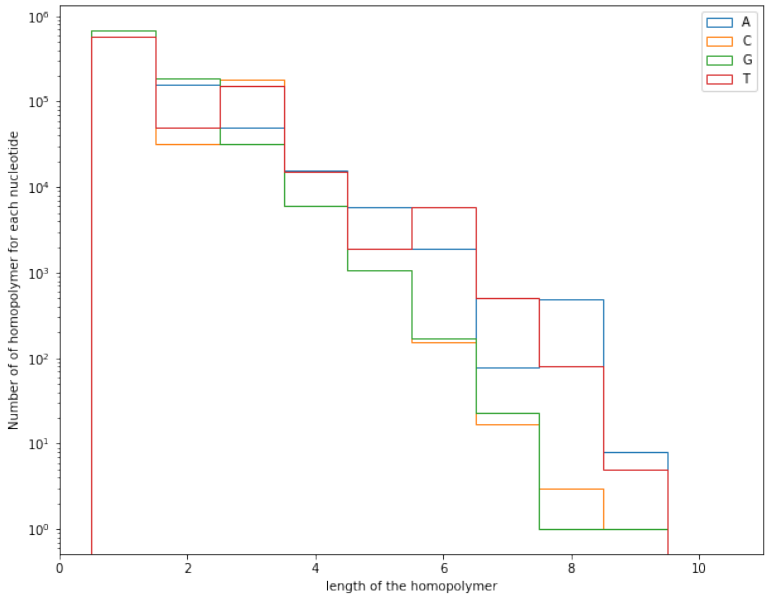


Figure 8: The figure shows the amount of homopolymers according to their length on our reference genome.

measured length, with long insertions rarely appearing, from homopolymer length 5 onwards, we see that there are many more deletions than insertions, which means that the real length is strongly underestimated for long homopolymers. This confirms that nanopore cannot accurately identify long homopolymers. A similar result to ours was observed by the article [Delahaye Clara and Jacques, 2021] in Figure 9. The authors of this study used Guppy version 3.3.3, while our data used Guppy version 6.1.2. Therefore, it appears that the identification of homopolymers remains somewhat problematic even for the new version.

4 Conclusion

The analysis we carried out, allowed us to investigate certain aspects of Nanopore sequencing technology, which together with Illumina represents the main means used to sequence longer or shorter DNA/RNA fragments. Initially, we tried to establish whether there was any correlation between the nucleotide and the assigned quality value and asked ourselves how often we would encounter errors. We obtained that the assigned quality is reliable and the error is smaller than the expected error. Then we investigated the degree of dependence of the

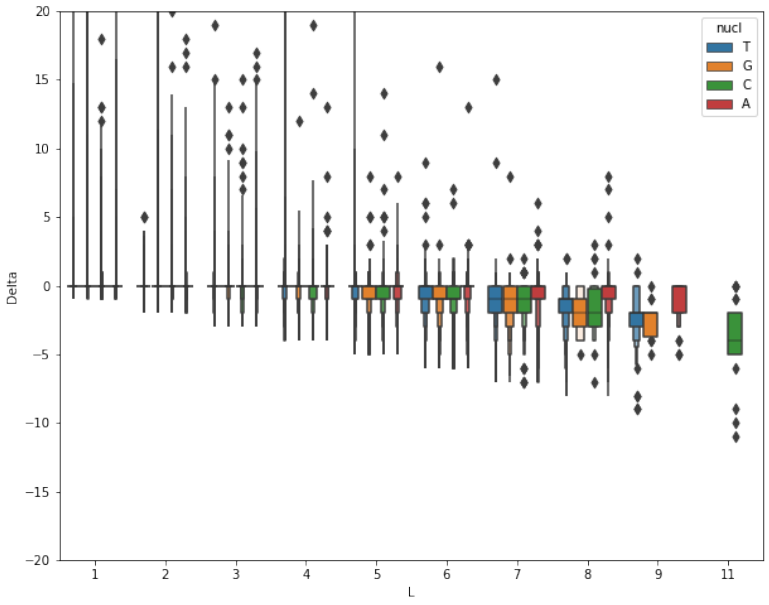


Figure 9: Distribution of mis-estimation of homopolymer length Δ, as a function of homopolymer lenght L and nucleotide. It can be seen that on average the length of long homopolymers tend to be underestimated.

error with the context by determining, whether a certain context is more prone to error; in this case we found that the context is not relevant to the error. Finally, we questioned the probability of incorrectly estimating the length of homopolymers and found that homopolymer lengths are underestimated. Due to time constraints for this project, we were not able to investigate the errors in nanopore sequencing in more detail, but there are certainly other aspects that would be worth investigating.

In our analysis, we worked with only one dataset; the next step we could have taken, would have been to find one or more datasets in the literature whose solutions are known and carry out the same type of analysis we did. If similar results were obtained, it would be a further confirmation for our work.

In the third part of our analysis, we considered homopolymer length estimation errors rather than actual insertions. A further aspect we could have investigated is "genome rearrangement" phenomena, i.e. where even potentially large pieces of the genome change position, or are lost or are acquired. These at the level of our alignments could cause large insertions/deletions ($> 1Kbp$) and especially long hard/soft clips ($> 1Kbp$), i.e. before a match region we would find a long sequence that does not match the reference genome. So we

could investigate whether these parts that do not match have a match somewhere else on the genome. If we found these matches, we could prove that these parts have moved.
A third aspect that would be interesting to investigate would be to establish how well the longer homopolymers are reconstructed by Trycler. Trycler is the tool we used to reconstruct the reference genome from the signals read by Guppy. If it turns out that Trycler makes mistakes when it has to reconstruct the longer homopolymers, it would be a big problem because it could cause a frame-shift in the read and could destroy genes that are actually present. To check if this happens, one could compare the reference genome obtained with other reference sequences.
This project allowed us to quantify the errors in the specific case of our data, which was actually very useful to be sure of their "reability". In the future we could use this information to separate the actual signals from the errors made by nanopore.

5 Supplementary

Next, we looked for the influence of nucleotide sequences on errors. In these cases, sequences of increasing size were examined to see how much information is removed by knowing the context compared to the initial value where there is no context.
Our hypothesis assumed that at first we knew the nucleotide on which the error was, i.e. at a distance *1*, and we knew we had information about it, since we had found two nucleotides that were most likely to be wrong (i.e. **A or G**). By expanding our sequence by a few nucleotides, we would have some additional information. Then, however, as the nucleotide sequence would have become larger, the conditional entropy would have decreased until the curve would have flattened out and reached a "saturation", i.e. a point at which knowing further nucleotides would have brought no new information to the error.

`Context 2.a` Figure 10 shows the results obtained for context (**2.a**). At distance *0* there is no context and the entropy is the one of the error itself. One can compare the values of the entropy of the error on the red line, obtained in the figure 7 with the value at distance *0*, observing that they are identical. At distance *1* we find the nucleotide on which there is an error and it corresponds to the values at distance *0* in the figure 7 on the blue line, where only one nucleotide is taken. Subsequently, the nucleotide sequence gradually increases in size, increasing for each distance both one nucleotide to the right and one nucleotide to the

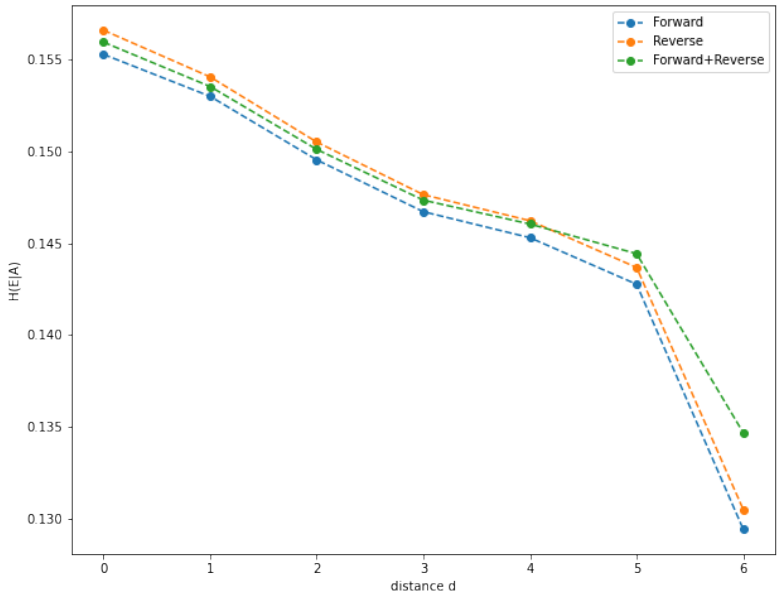


Figure 10: The figure shows the conditional error entropy calculated for context ***2.a*** *for forwards, reverses and forwards+reverses readings. It shows that at first the information of neighbouring nucleotides influences the error, but then due to the insufficient number of sequences we are no longer able to calculate the probability correctly for large values of d.*

left from nucleotide i. The length of these nucleotides can be expressed as a sequence:

$$Lengths(n) : 1 - 3 - 5 - 7 - 9 - 11 - 13 \tag{17}$$

Although unclear, a downward trend can be seen in our figure. However, we do not see the curve flatten out, and indeed at some point it drops further. This is due to the fact that as context length increases, we encounter fewer and fewer examples of sequences on our genome. This is important because it gives us an indication of a threshold where we can no longer estimate the probability of the contexts, i.e. some contexts will never be seen and in our model would be assigned a false probability of 0. The length of our reference genome was 4'642'393 nucleotides and already at distance 6 we found sequences longer than our reference genome, as shown in the table 3. In this case we found the conditional entropy for both Forwards and Reverse reads and considering all reads. Up to distance 5, the conditional entropy curve goes down, but without "saturating".

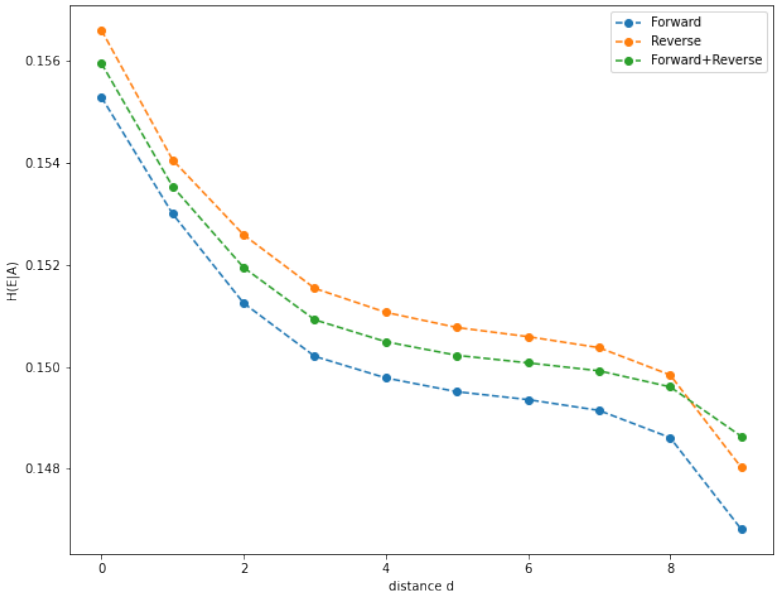
`Context 2.b` In figure 11a there are the results we obtained for the context (**2.b**). At distance *0* there is no context and represents the error itself, and at distance *1* there is the

distance	0	1	2	3	4	5	6	7
Possibilities	0	4^1	4^3	4^5	$4^7 \sim 2 \cdot 10^4$	$4^9 \sim 3 \cdot 10^5$	$4^{11} \sim 4 \cdot 10^6$	$4^{13} \sim 7 \cdot 10^7$

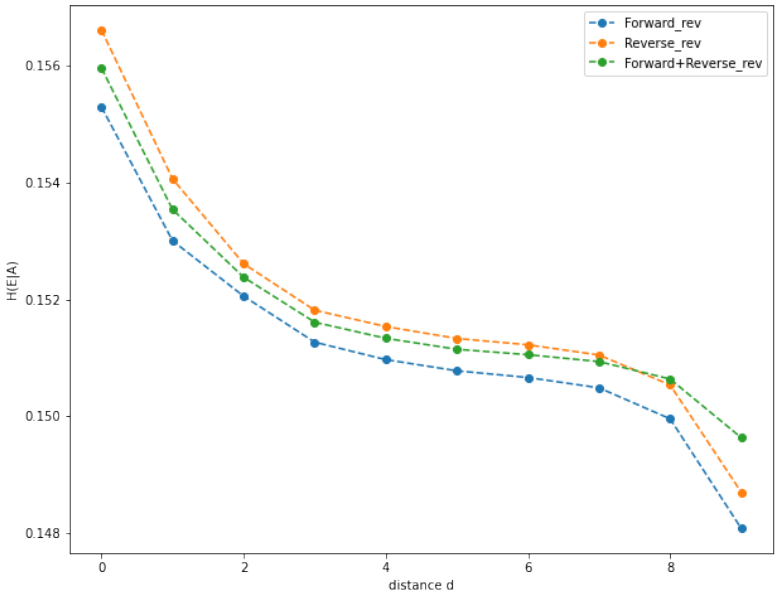
Table 3: Number of possible different contexts vs. context distance.

nucleotide on which there is the error, also seen for `Context 2.a`. Then the sequence is only expanded in one direction, in this case to the right as described in 2.2.2 section. The values are very similar to those obtained previously, and it is certainly clearer to see the curve of conditional entropy dropping down to sequences with a length of 4, and then flattening out to a length of 7. After passing this value, the entropy drops again, indicating that we no longer have enough examples of sequences in our genome. This result makes it clear that for the figure 10, we can only trust the results for distances up to 2 and then some sequences probably never appear.

`Context 2.c` Figure 11b shows the results obtained for context (**2.c**). It is structured in the same way as `Context 2.b`, only the direction in which the sequences are formed has changed, i.e. in this case they will grow to the left. Also in this context at distance 4 the curve "saturates" and after distance 7, the conditional entropy decreases again reaching the threshold where there are not enough sequences. A further indication of this behaviour can be seen in the green line on the graph. At first it lies between the forward and reverse reads, but at some point this line decreases less than the other two, until it remains above the two lines as seen at distance 8. Since there are not many reads for either forward or reverse, we will find a conditional entropy for all reads (forward+reverse) that is higher than for single reads because in total we would have more examples of sequences than single cases. Even considering the contexts of the sequences, the surprise in the error always remains around 1%. This is why we can conclude that the error does not depend on the context.



(a) Conditional Entropy calculated from context ***2.b***



(b) Conditional Entropy calculated from context ***2.c***

Figure 11: The figure shows the conditional error entropy calculated for context ***2.b*** *and* ***2.c*** *for forwards, reverses and forwards+reverses readings. It shows that at first the information of the neighbouring nucleotides influences the error, then as the length of the sequences become longer this influence becomes non-informative as the curve flattens. Towards the end, there will not be enough examples of sequences and you will no longer be able to calculate these probabilities correctly.*

References

Canu (2022). original-date: 2015-08-21T03:10:42Z. URL: https://github.com/marbl/canu.

Conditional entropy (2022). Page Version ID: 1102205732. URL: https://en.wikipedia.org/w/index.php?title=Conditional_entropy&oldid=1102205732.

Dario B. (2022). *ddd42-star/Applied-Project-in-Computational-Biology-2022: Project Computational Biology-FS 2022*. URL: https://github.com/ddd42-star/Applied-Project-in-Computational-Biology-2022.

Delahaye Clara and Jacques N. (2021). "Sequencing DNA with nanopores: Troubles and biases". URL: https://journals.plos.org/plosone/article/file?id=10.1371/journal.pone.0257521&type=printable#:~:text=Nanopore%20quality%20score.&text=As%20for%20Illumina%20data%2C%20Nano,Phred%20scores%200%20to%2093.

FASTQ format (2022). Page Version ID: 1094286335. URL: https://en.wikipedia.org/w/index.php?title=FASTQ_format&oldid=1094286335.

Ion Current - an overview — ScienceDirect Topics (2022). URL: https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/ion-current.

Kolmogorov M. (2022). *Flye assembler*. original-date: 2016-03-17T22:47:39Z. URL: https://github.com/fenderglass/Flye.

Lederberg J. and Tatum E. L. (1946). "Gene Recombination in Escherichia Coli". *Nature* 158.4016. Number: 4016 Publisher: Nature Publishing Group, pp. 558–558. DOI: 10.1038/158558a0. URL: https://www.nature.com/articles/158558a0.

Lennox E. S. (1955). "Transduction of linked genetic characters of the host by bacteriophage P1". *Virology* 1.2, pp. 190–206. DOI: 10.1016/0042-6822(55)90016-7.

Li H. (2018). "Minimap2: pairwise alignment for nucleotide sequences". *Bioinformatics* 34.18, pp. 3094–3100. DOI: 10.1093/bioinformatics/bty191. URL: https://doi.org/10.1093/bioinformatics/bty191.

– (2022). *lh3/minimap2*. original-date: 2017-07-18T15:04:53Z. URL: https://github.com/lh3/minimap2.

MacKay D. J. C. (n.d.). "Information Theory, Inference, and Learning Algorithms" (), p. 640.

Phred quality score (2022). Page Version ID: 1092452979. URL: https://en.wikipedia.org/w/index.php?title=Phred_quality_score&oldid=1092452979.

Raven (2022). original-date: 2019-07-16T12:47:24Z. URL: https://github.com/lbcb-sci/raven.

Riley M., Abe T., Arnaud M. B., Berlyn M. K., Blattner F. R., Chaudhuri R. R., Glasner J. D., Horiuchi T., Keseler I. M., Kosuge T., Mori H., Perna N. T., Plunkett III G., Rudd K. E., Serres M. H., Thomas G. H., Thomson N. R., Wishart D., and Wanner B. L.

(2006). “Escherichia coli K-12: a cooperatively developed annotation snapshot—2005”. *Nucleic Acids Research* 34.1, pp. 1–9. DOI: 10.1093/nar/gkj405. URL: https://doi.org/10.1093/nar/gkj405.

Ruan J. (2022). *ruanjue/wtdbg2*. original-date: 2017-09-29T10:24:08Z. URL: https://github.com/ruanjue/wtdbg2.

Samtools (2022). URL: http://www.htslib.org/.

Toprak E., Veres A., Yildiz S., Pedraza J. M., Chait R., Paulsson J., and Kishony R. (2013). “Building a morbidostat: an automated continuous-culture device for studying bacterial drug resistance under dynamically sustained drug inhibition”. *Nature Protocols* 8.3. Number: 3 Publisher: Nature Publishing Group, pp. 555–567. DOI: 10.1038/nprot.2013.021. URL: https://www.nature.com/articles/nprot.2013.021.

Wang Y., Zhao Y., Bollas A., Wang Y., and Au K. F. (2021). “Nanopore sequencing technology, bioinformatics and applications”. *Nature Biotechnology* 39.11, pp. 1348–1365. DOI: 10.1038/s41587-021-01108-x. URL: https://www.nature.com/articles/s41587-021-01108-x.

Wick R. R., Judd L. M., Cerdeira L. T., Hawkey J., Méric G., Vezina B., Wyres K. L., and Holt K. E. (2021). “Trycycler: consensus long-read assemblies for bacterial genomes”. *Genome Biology* 22.1, p. 266. DOI: 10.1186/s13059-021-02483-z. URL: https://doi.org/10.1186/s13059-021-02483-z.