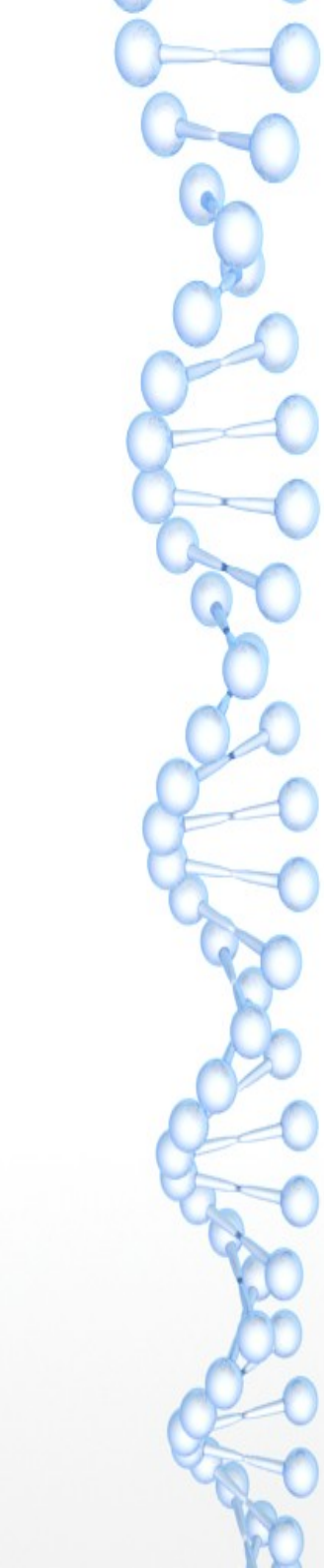


# Applied Project in Computational Biology

FS 2022

# Summary

- Frequency of error
- Entropy
- Homo-polymer Insertion/Deletion

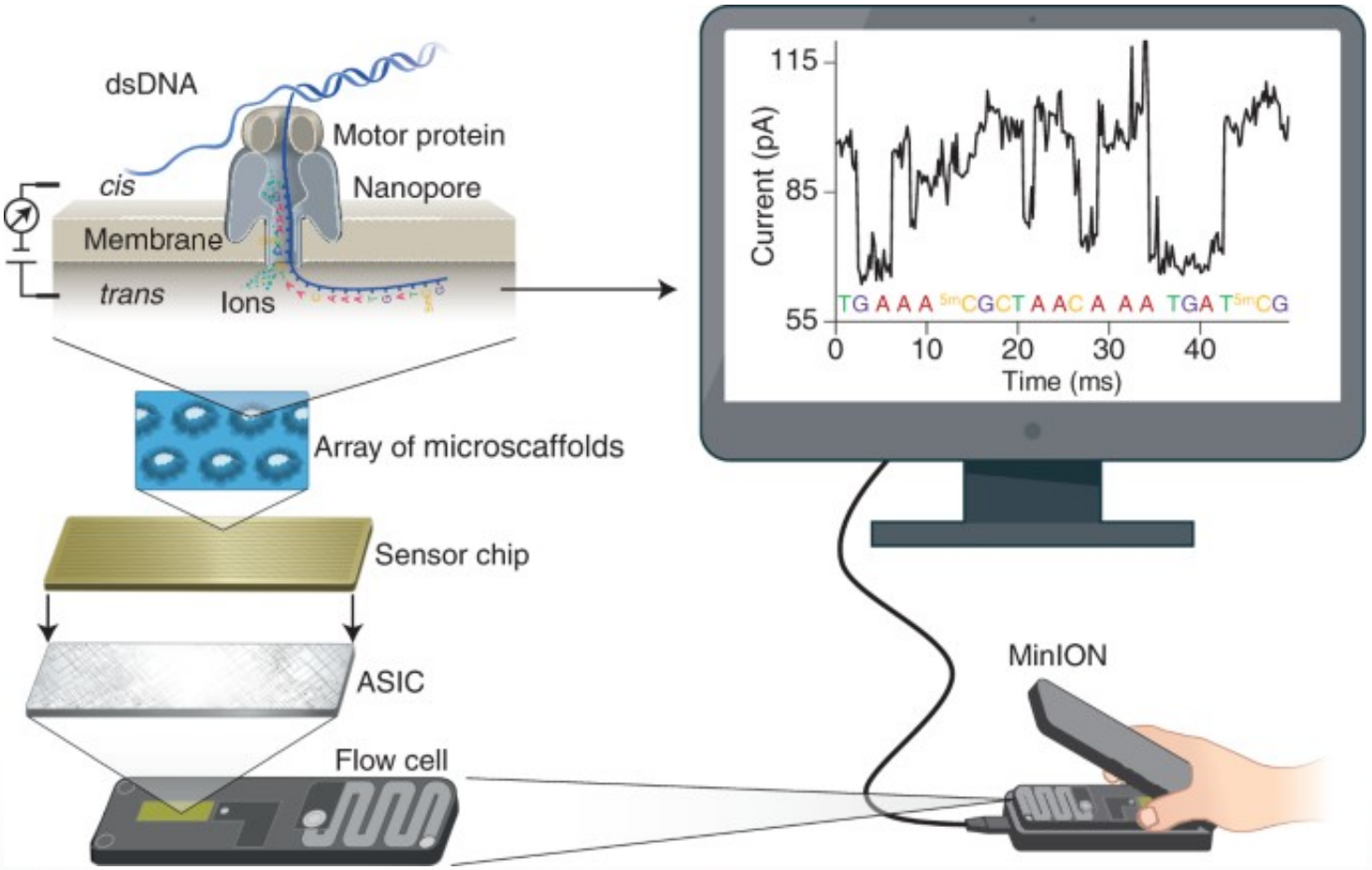
# Introduction

- Typically, a bacterial population is left to grow for a time t. Then their genome is broke by a protein to be analyze.
- One use **Nanopore** to analyze the genome. How does it work?

# Introduction

- The technology relies on a nanoscale protein pore, or ‘nanopore’, that serves as a biosensor and is embedded in an electrically resistant polymer membrane 1,3.
- In an electrolytic solution, a constant voltage is applied to produce an ionic current through the nanopore such that negatively charged single-stranded DNA or RNA molecules are driven through the nanopore from the negatively charged ‘cis’ side to the positively charged ‘trans’ side.
- Changes in the ionic current during translocation correspond to the nucleotide sequence present in the sensing region.

# Introduction



# Introduction

- The collected signals are decoded by **Guppy** producing a **FASTQ** file containing all different reads
- From this read the genome is assembled by **Trycycler**
- In my analysis I used the genome's data of Reto at time 1 (24h)
- We aligned all the reads against the reference genome using Minimap2 and then we ordered it using samtools.
- The final output was a **.bam** file

# Introduction

- In the .bam file there are lot of information. The most relevant for my analysis are:
  - CIGAR string (2348H32M34D3I….)
  - Quality Score (encode ASCII characters from 33 to 126 (0 to 93))
- To access the information in the .bam file we used the pysam library

# Introduction

- We want to observe variations of some kind within a population that can be snips, genomic reassembly.
- These measurements are affected by noise because the Nanopore is not perfect and we want to quantify this noise so that in the future it will be possible to separate signal from noise

# Frequency of error

**Aim:** We want to observe for each quality assigned by Nanopore what the error frequency is:

- Having only **Matches**

# Frequency of error

Using pileup we created a matrix that collect for each pair of nucleotides the count of the observed quality.

$$[(N_i, N_j, \text{True/False}) \rightarrow \{q:N_q\}]$$

$N_i$ = Nucleotide read on the genome

$N_j$ = Nucleotide read on the query

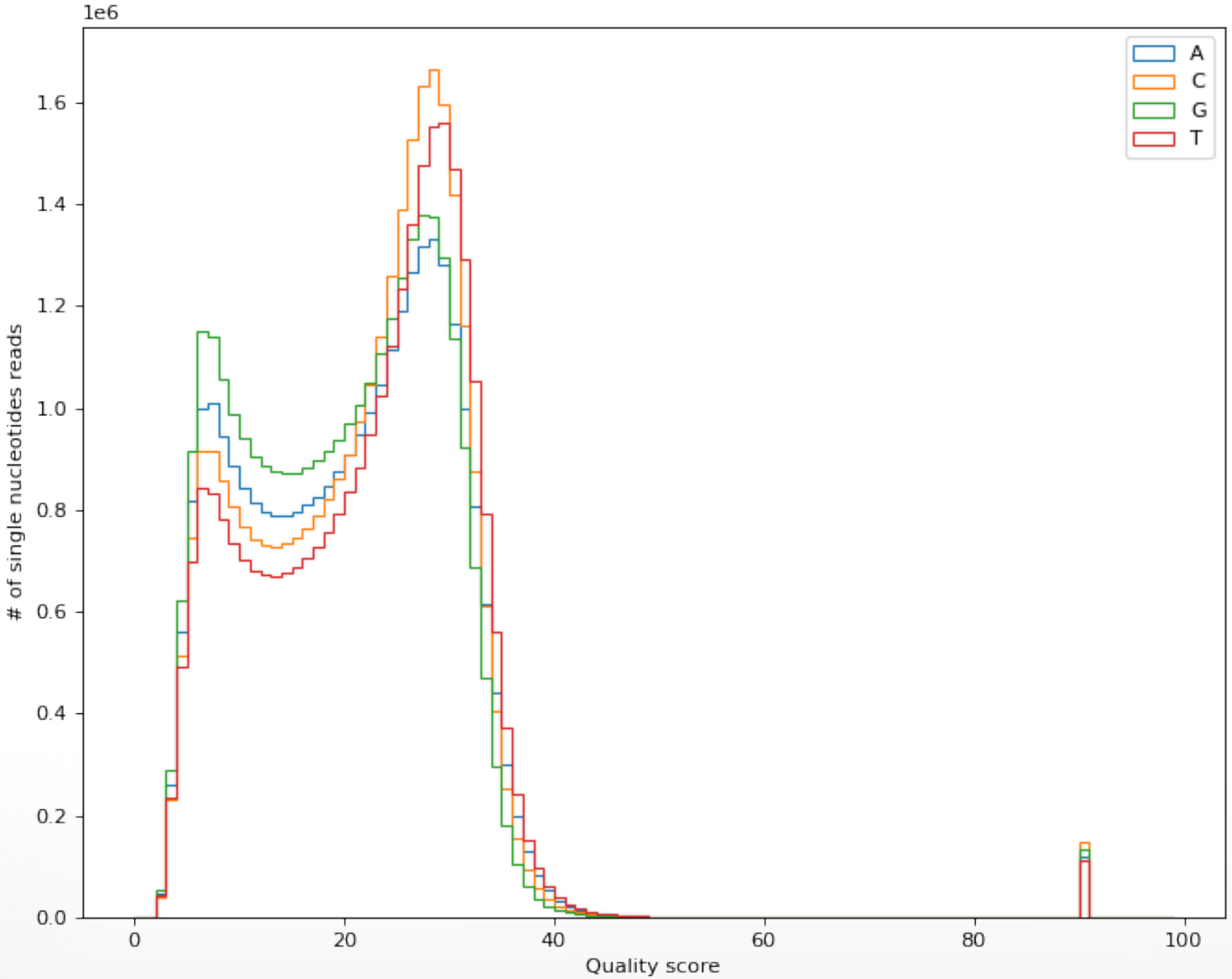
True/False = Forward or Reverse

q = Quality score

$N_q$ = count of the observed quality

# Frequency of the error

- To start off we decided to see the distribution of the qualities of the nucleotides  in all reads
- We wanted to find a quality threshold


1e6
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
# of single nucleotides reads
0
20
40
60
80
100
Quality score
A
C
G
T

# Frequency of error

And then we found the frequency of error

$$P_{error} = \frac{\#e(q)}{\#reads(q)}$$

**For example**:

A,A → {10:2397}

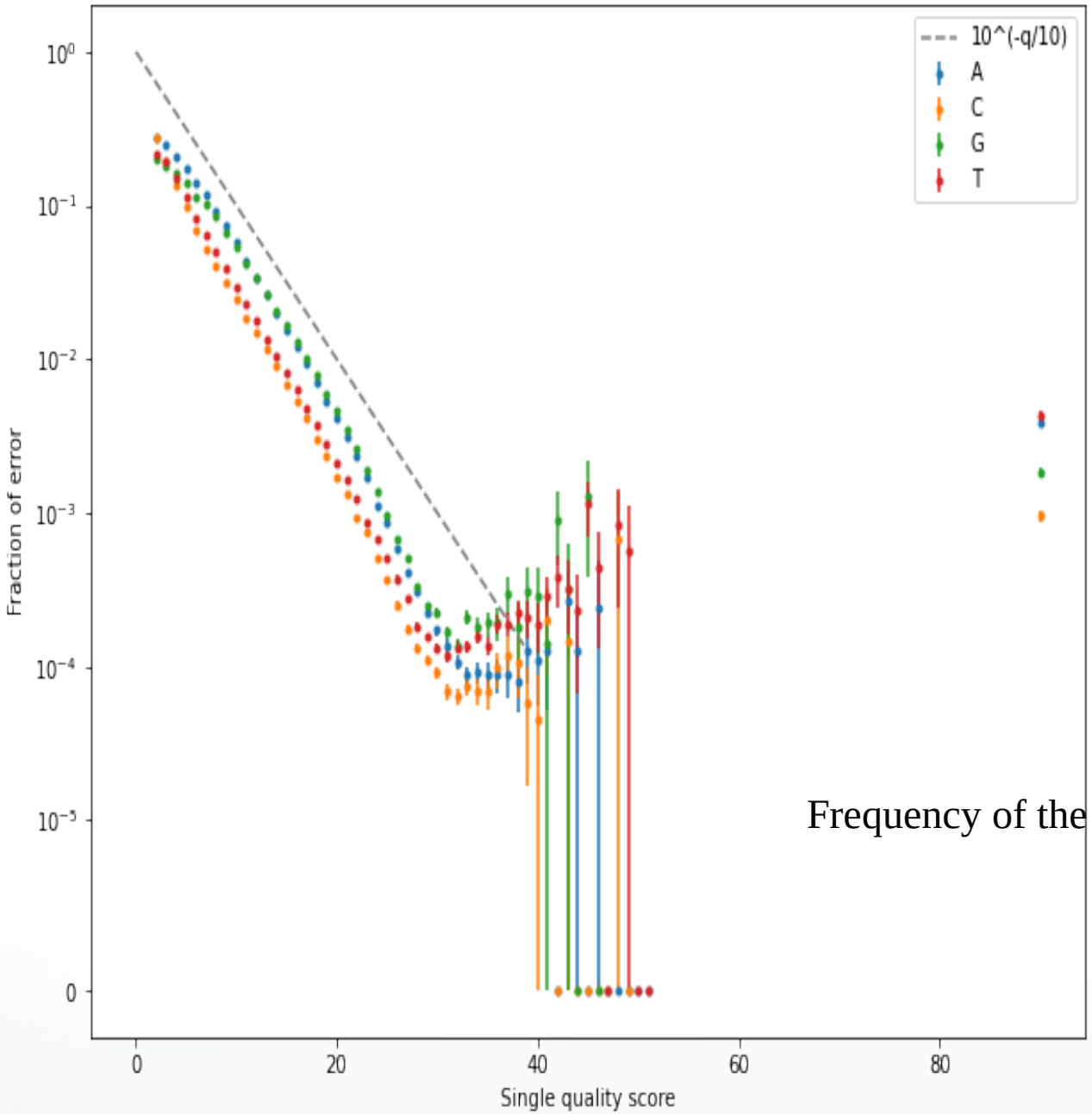
A,C → {10:101}

A,G → {10:23}

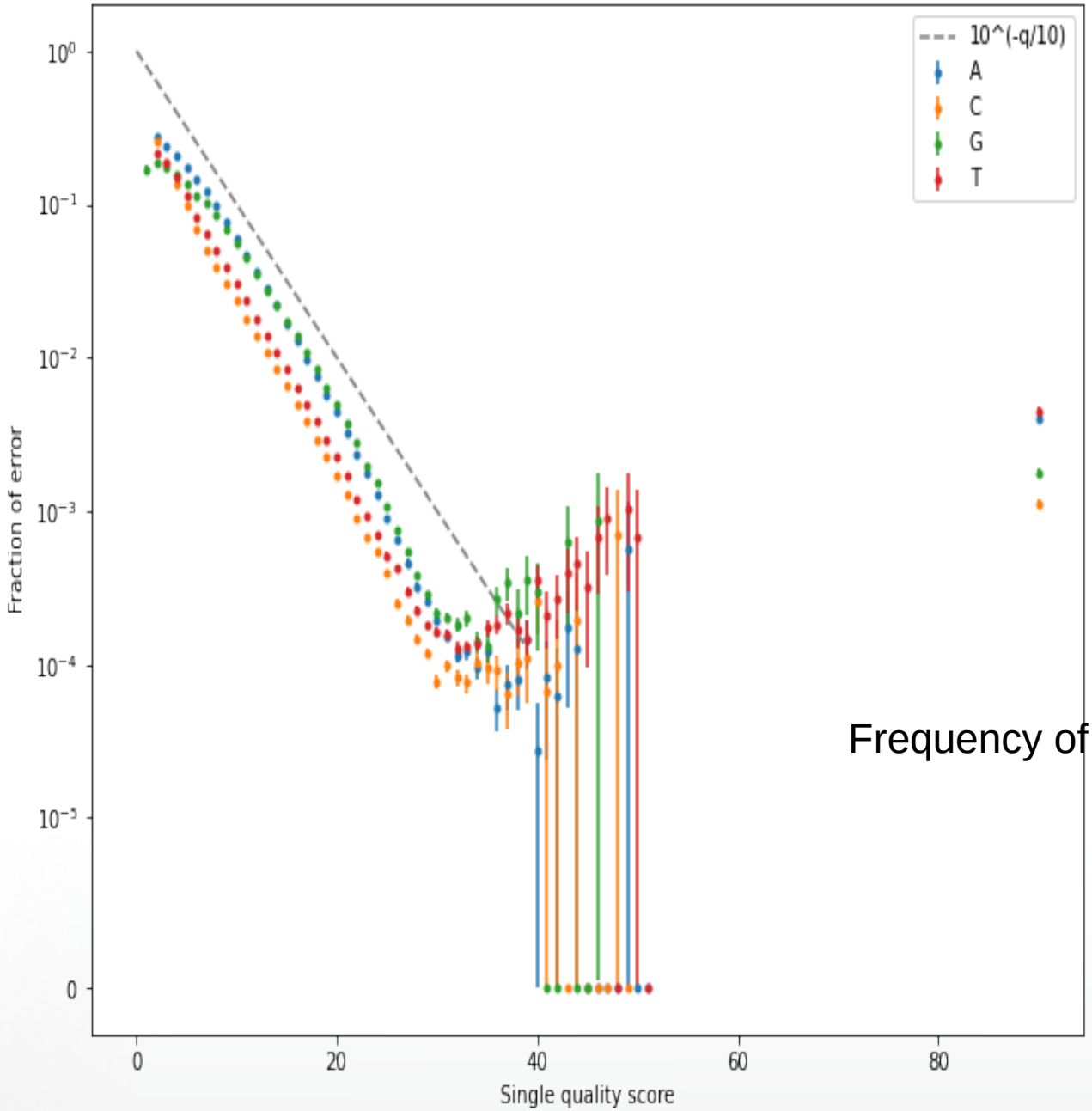
A,T → {10:1}

# e (A,10) = A,C + A,G + A,T

# reads(A,10) = A,A + A,C +A,G + A,T




Frequency of the error Forward



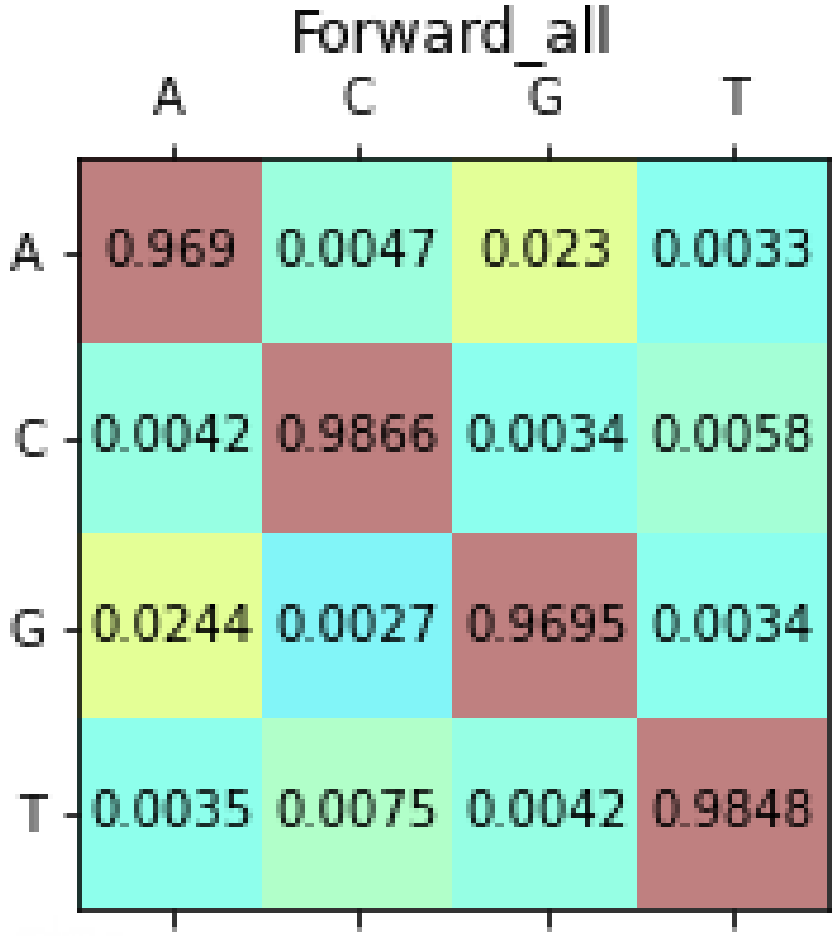

Frequency of the error Reverse

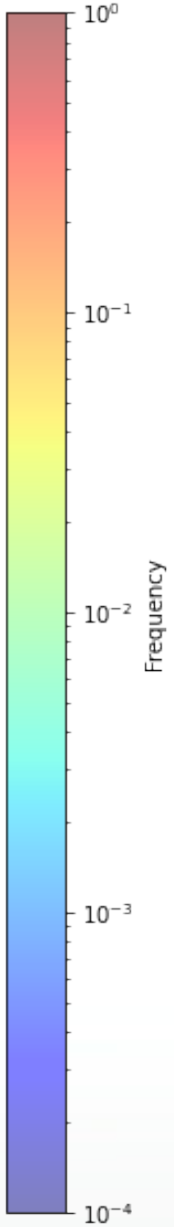
# Frequency of the error

- The distribution of the qualities is a bimodal distribution
- Eventually, we can take as quality threshold all nucleotides with quality >= 20
  - Because on the first pick the error for low qualities is more commune, instead from nucleotides with quality of 20, the error is only 1%.

# Frequency of the error

- To understand in detail which pair of nucleotides it is possible to make mistakes with more frequency, we have visualized the matrix found previously


Forward_all
A
C
G
T
A
C
G
T
0.969
0.0047
0.023
0.0033
0.0042
0.9866
0.0034
0.0058
0.0244
0.0027
0.9695
0.0034
0.0035
0.0075
0.0042
0.9848


$10^{0}$
$10^{-1}$
$10^{-2}$
$10^{-3}$
$10^{-4}$
Frequency

# Frequency of the error

- From this graph we see that the pairs **AG** and **GA** are the most common error.

# Entropy

- **Aim:** How the presence of one or more nucleotides before or after affects the correct identification of the nucleotide in the query
- We used the Shannon Theory information

> "The fundamental problem of communication is that of reproducing at one point, either exactly or approximately, a message selected at another point." (Shannon,1948)

modem → phone lines → modem

Galileo → radio waves → Earth

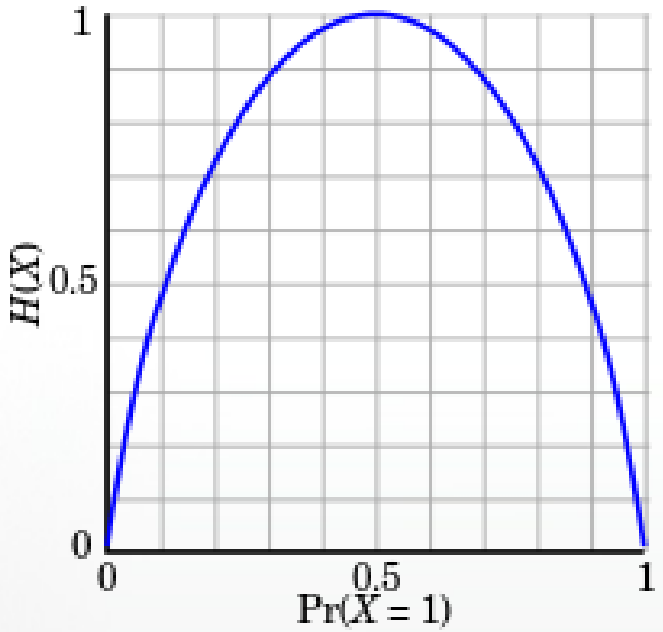
parent cell → 2x daughter cell

computer memory → disk drive → computer memory

# Entropy

- Entropy in the theory of information is define as:

  *“The entropy of a random variable is the average level of "information", "surprise", or "uncertainty" inherent to the variable's possible outcomes.”*



# Entropy

- The context as much uncertainty takes away from us?
- We are interested in quantifying how much context I have to remove to get information

# Entropy

We define two random variable:

E → e = { $\begin{matrix} 1\, error\,(p) \\ 0\, no\, error\,(1-p) \end{matrix}$

A → a (contex) = { $n_i + n_j$

$H(E,A) = -\sum_{ea} P(e,a)\log P(e,a)$

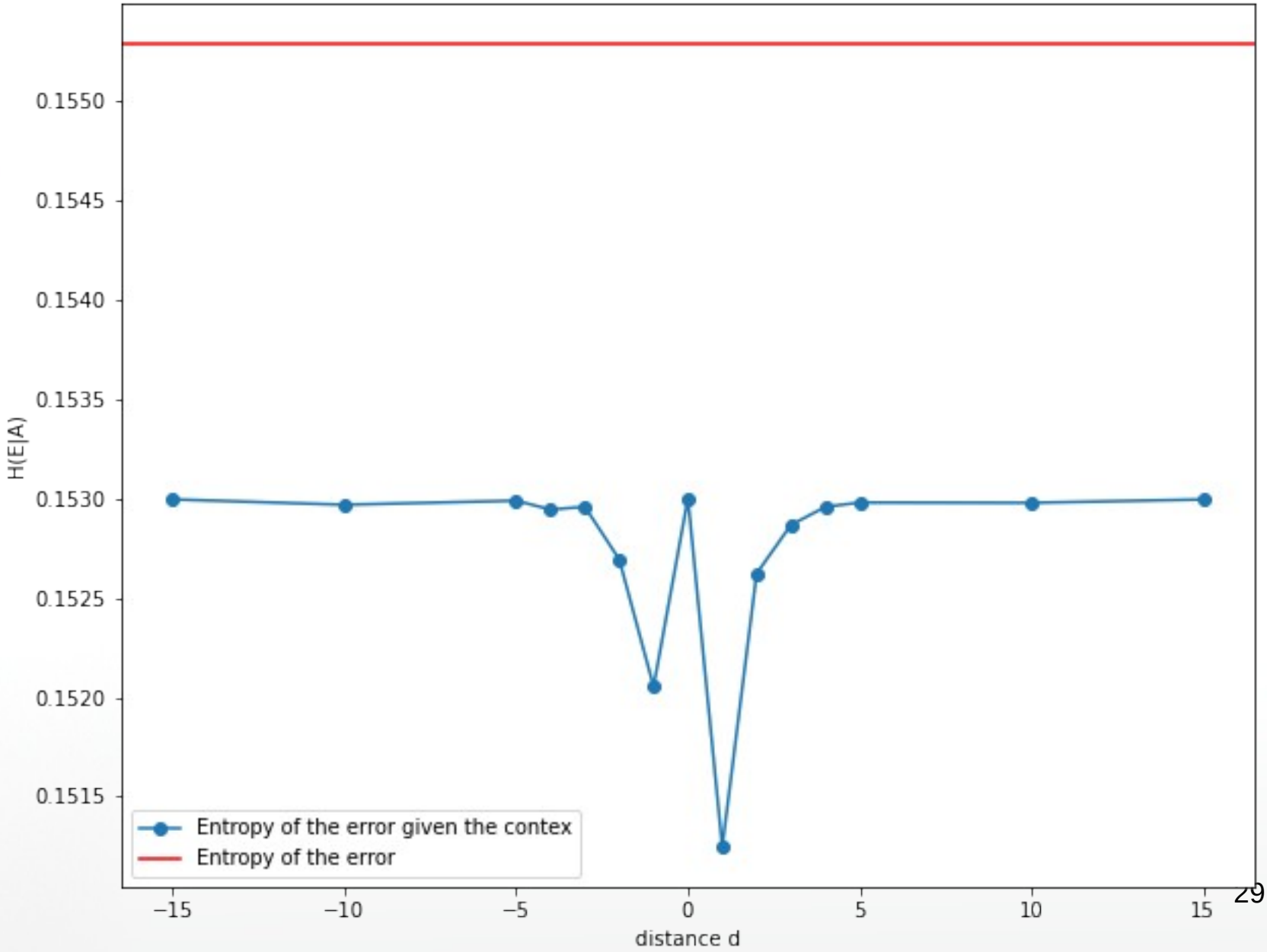
$H(E|A) = H(E,A) - H(A)$

$$=> = -\sum_{ea} P(e,a)\log(P(e,a)) + \sum_{ea} P(a)\log(P(a))$$

$$= -\sum_{ea} P(e,a)\log(P(e,a)) + \sum_{e,a} P(e,a)\log P(a)$$

$$= -\sum_{e,a} P(e,a)\log\frac{P(e,a)}{P(a)}$$

# Entropy


0.1550
0.1545
0.1540
0.1535
0.1530
0.1525
0.1520
0.1515
H(E|A)
−15
−10
−5
0
5
10
15
distance d
Entropy of the error given the contex
Entropy of the error

# Entropy

- After this we wondered how nucleotide sequences had an influence on entropy
- Ex. ACGATTTAAGCGATAAAA (reading in one direction)

L(0) = “A”

L(1) = “AC”

L(2) = “ACG”

L(3) = “ACGA”

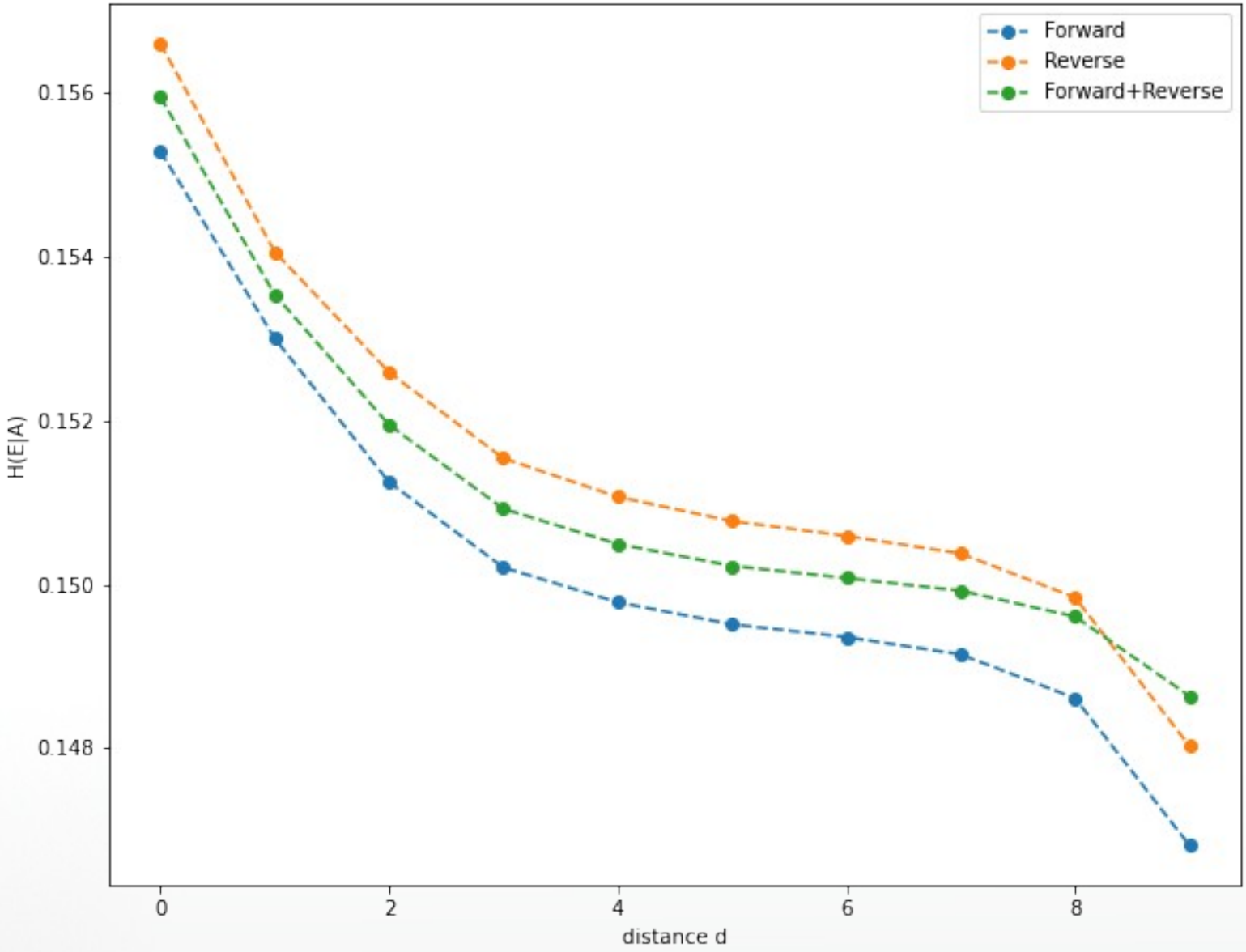
L(4) = “ACGAT”

L(5) = ”ACGATT”

L(6) = “ACGATTT”

….

# Entropy


Forward
Reverse
Forward+Reverse
0.156
0.154
0.152
0.150
0.148
H(E|A)
0
2
4
6
8
distance d

# Entropy

- After this we wondered how nucleotide sequences had an influence on entropy
- Ex. ACGATTTAAGCGATAAAA (reading in one direction)

L(0) = “A”

L(1) = “AG”

L(2) = “AGC”

L(3) = “AGCG”

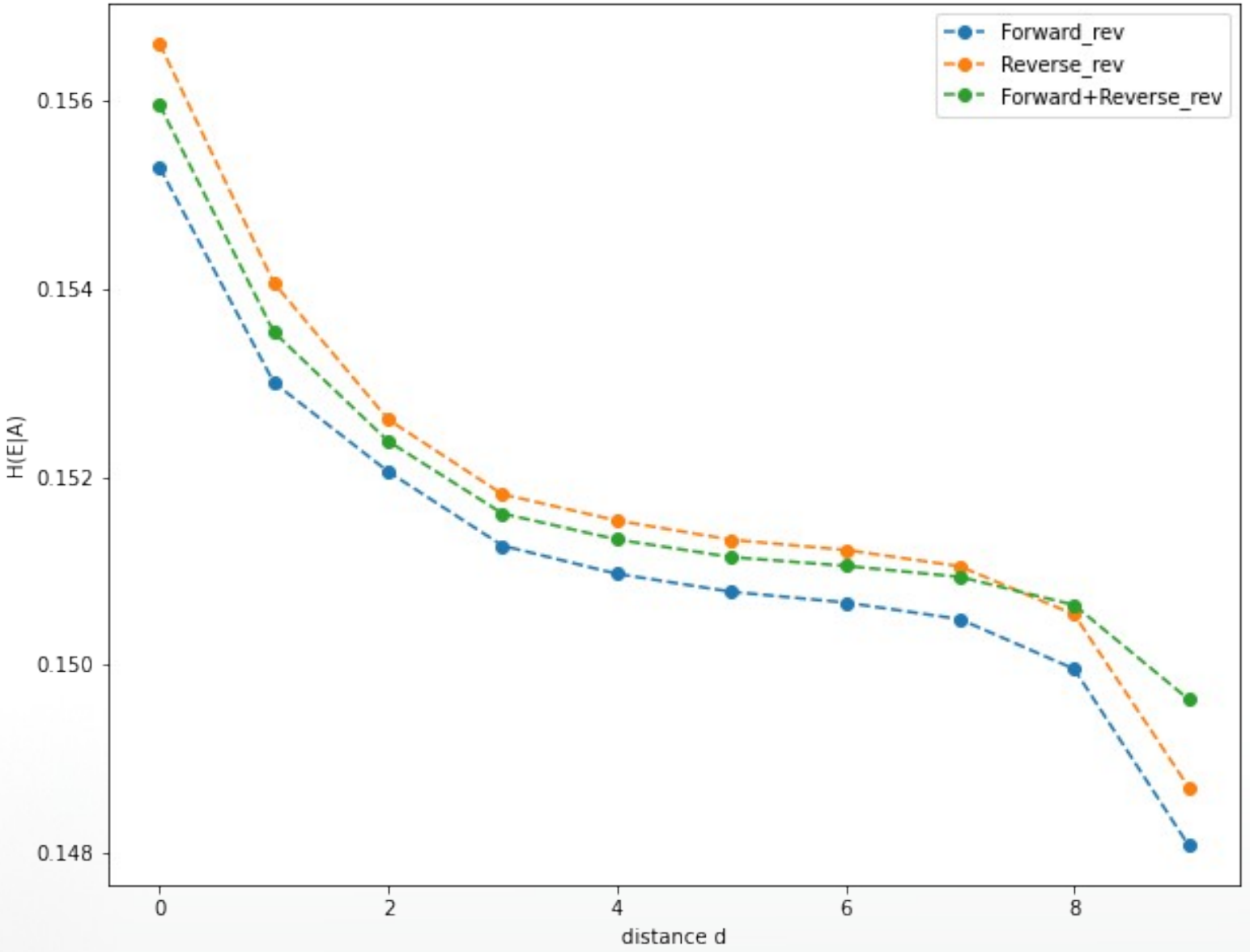
L(4) = “AGCGA”

L(5) = ”AGCGAA”

L(6) = “AGCGAAT”

….

# Entropy


Forward_rev
Reverse_rev
Forward+Reverse_rev
H(E|A)
0.156
0.154
0.152
0.150
0.148
0
2
4
6
8
distance d

# Homo-polymer insertion/deletion

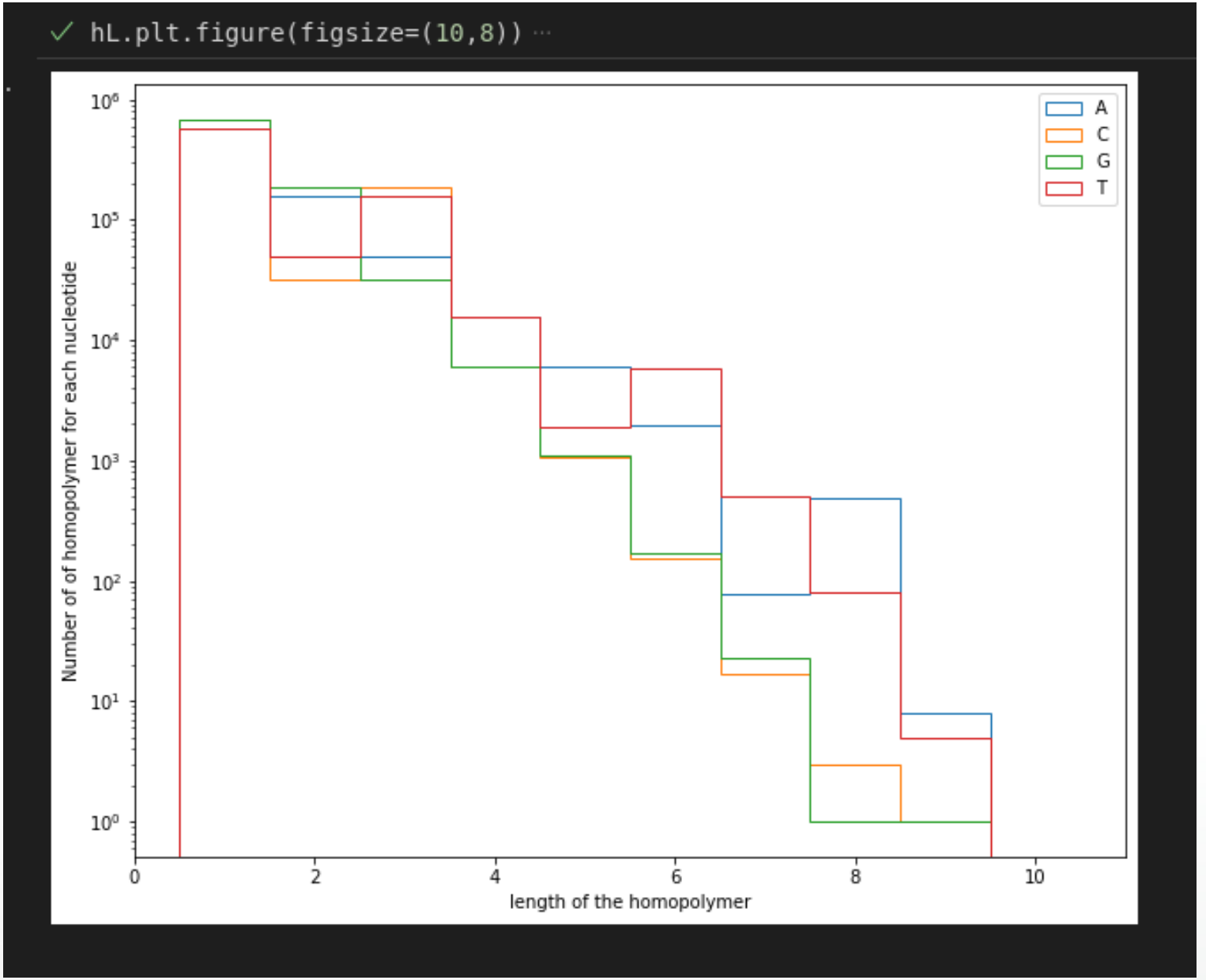
**Aim:** In the homopolymers on the genome, we want to find what kind of errors are more common.

- Nanopore sequencers tend to struggle to sequence low complexity regions accurately (minor variation in the electrical signal of the pore when the base does not change)

# Homo-polymer insertion/deletion

1) We found on the genome the nucleotide (i,i+1) where there was an insertion and we saved the insertions sequence
2) In a different dictionary we saved the **start** and **end position** of the homo-polymer and their length

# Homo-polymer insertion/deletion



# Homo-polymer insertion/deletion

3) We saved the length of the homo-polymer inserted starting from the nucleotide (i+1) so we obtained the positive delta for the insertion

4) We created a matrix containing in each position of the genome the count of A,C,G,T and “-“

5) We counted the number of deletion in each position of different homo-polymer. Then starting from the nucleotide (i+1), we calculated the delta of deletion

6) Delta = 0

Total number of reads – sum of (insertion+deletion)

# Homo-polymer insertion/deletion