

HKU SPACE and Edinburgh Napier University
Msc Biomedical Science
Scientific Skills Module

Data handling on Bioinformatics II

GU Haogao
The University of Hong Kong
guhaogao@hku.hk

2024-11-08

The Learning Objective

Data handling on Bioinformatics

- Analyse and interpret DNA and protein sequence data using bioinformatics.
 - Knowing how DNA and protein sequences are stored as data
 - Knowing how to create and manipulate such data
 - How to use existing tools to boost your analysis
 - Standard practice and advices for bioinformatics projects
- In these two lessons, we will learn:
 - **General data skills** that applicable to a wide range of (bio)informatics projects.

Rundown

Lesson 2

- Storing sequencing data (50 mins):
 - Recap on sequencing
 - fasta/fastq format
 - Nucleotide and amino acid codes
- Break (10 mins)
- Working with sequence data (50 mins)
 - Example: reading sequences data and counting nucleotides
 - Example: trimming low-quality bases
- Break (10 mins)
- Phylogenetic analysis and application (30 mins)

Sequencing

DNA Sequencing

- Sanger sequencing
 - Chain termination method
 - Also known as Sequencing **after** elongation
 - Requires the construction of millions of sequences with different sizes



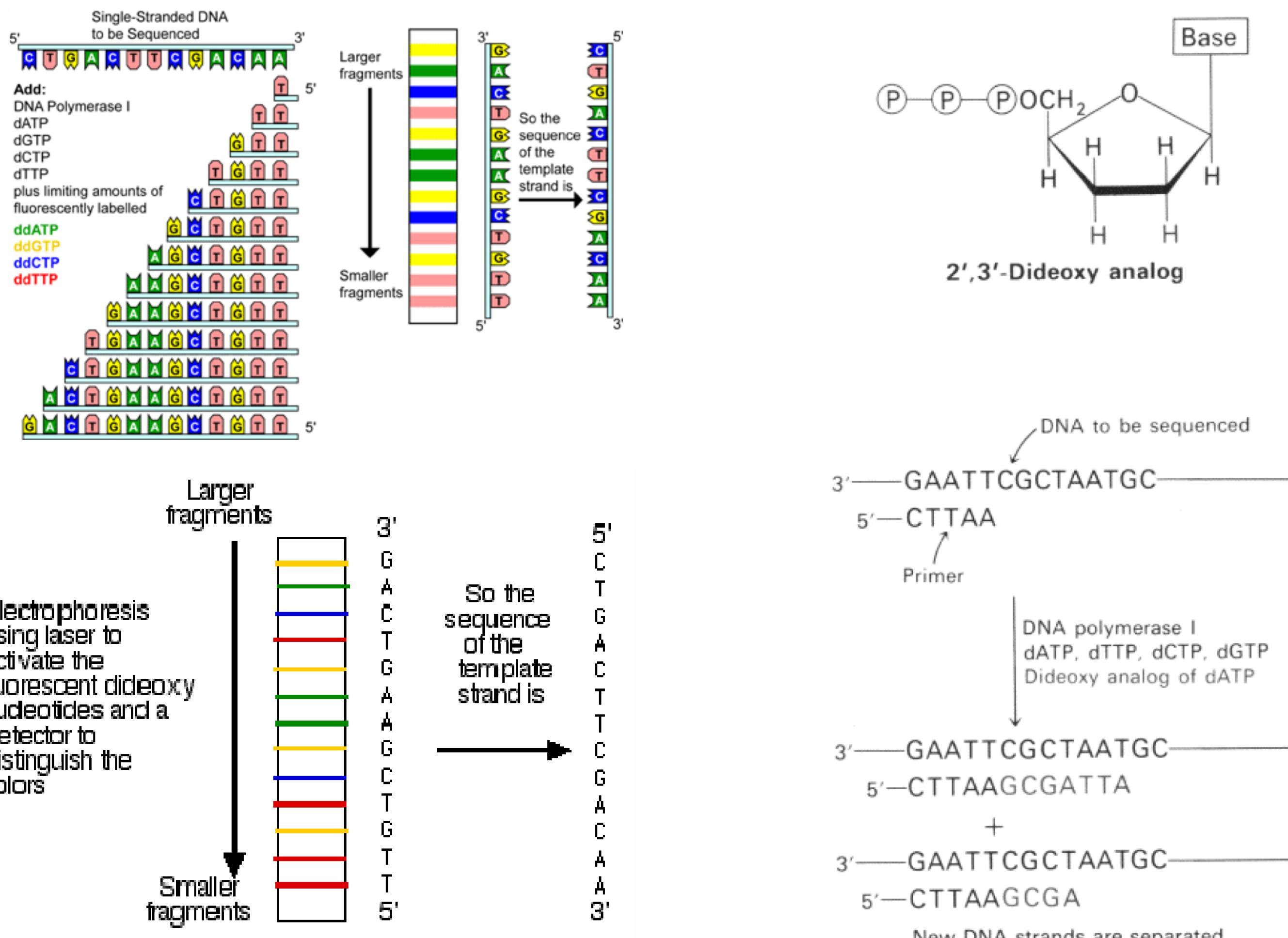
HKUSPACE
香港大學專業進修學院
HKU School of Professional and Continuing Education

Sanger Method

- Uses dideoxy-nucleotides (ddNTP) that inhibits DNA elongation after integration
- Requires primer, DNA polymerase, template, mixture of deoxyribonucleic acids (dNTP) and ddNTP
- Incorporation of ddNTP into growing strand to terminate DNA synthesis during **cycle sequencing**
- The synthesized strand sizes are determined using capillary electrophoresis

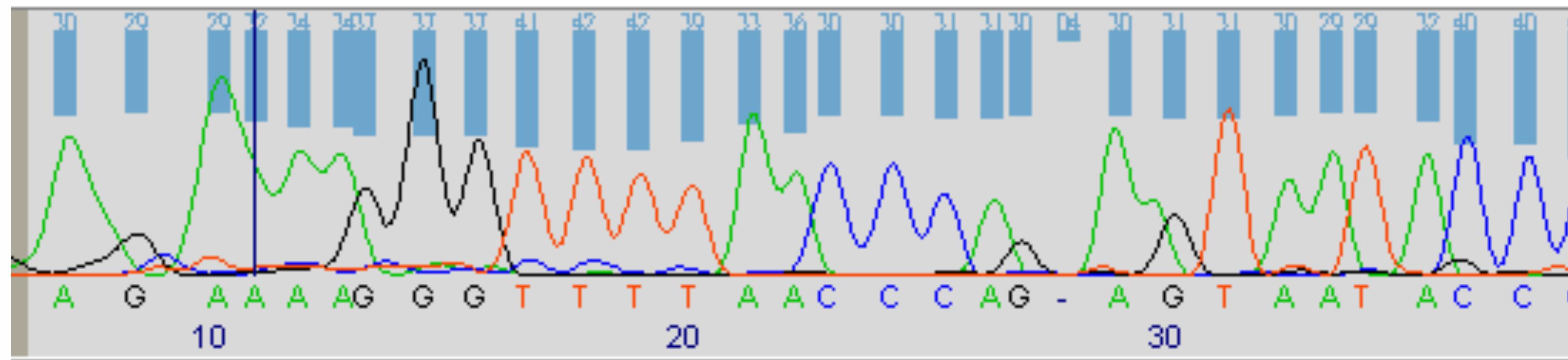


Sanger sequencing



Why do we have to do manual sequencing proof-reading and editing?

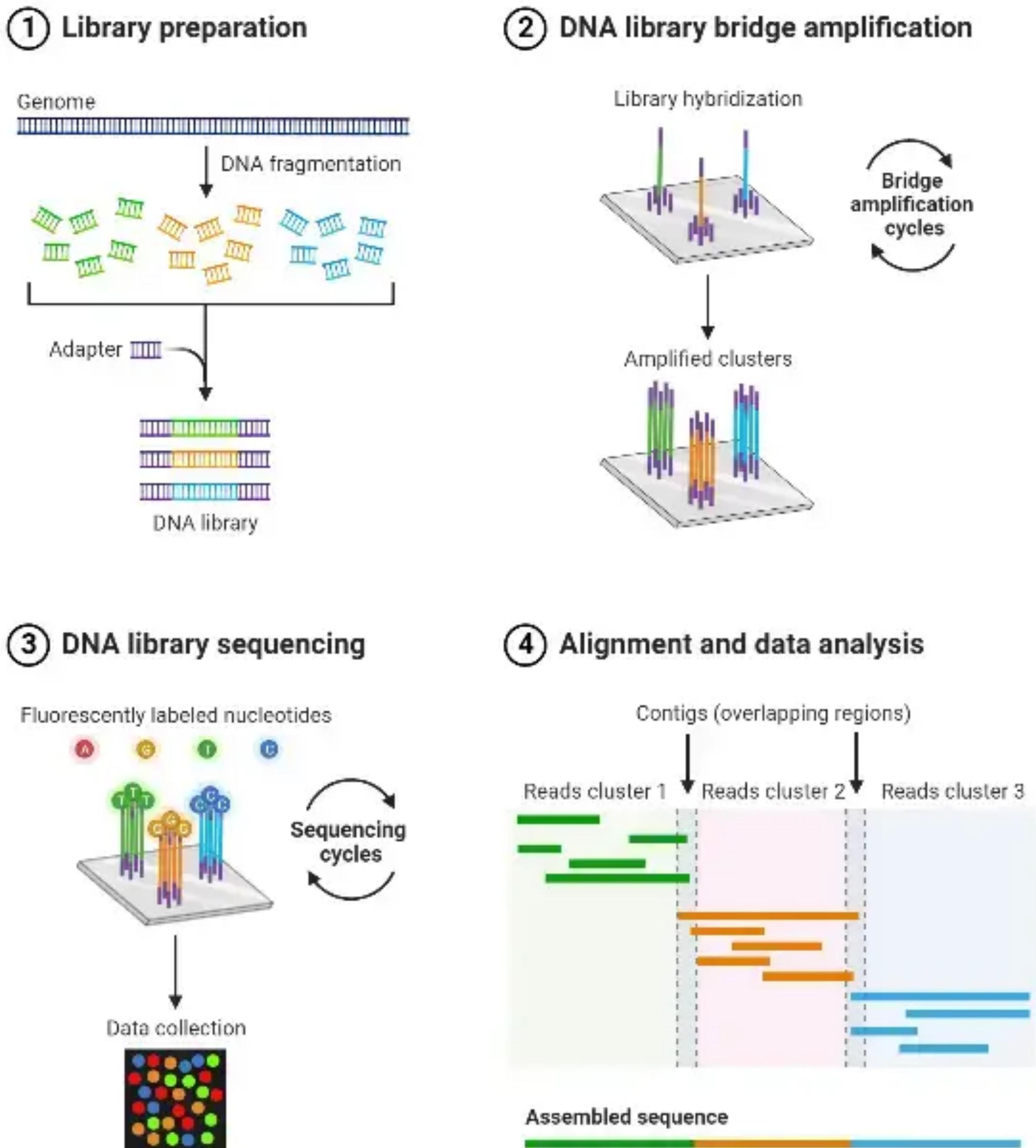
What is the right sequence at this particular position???????



NGS

High-throughput sequencing

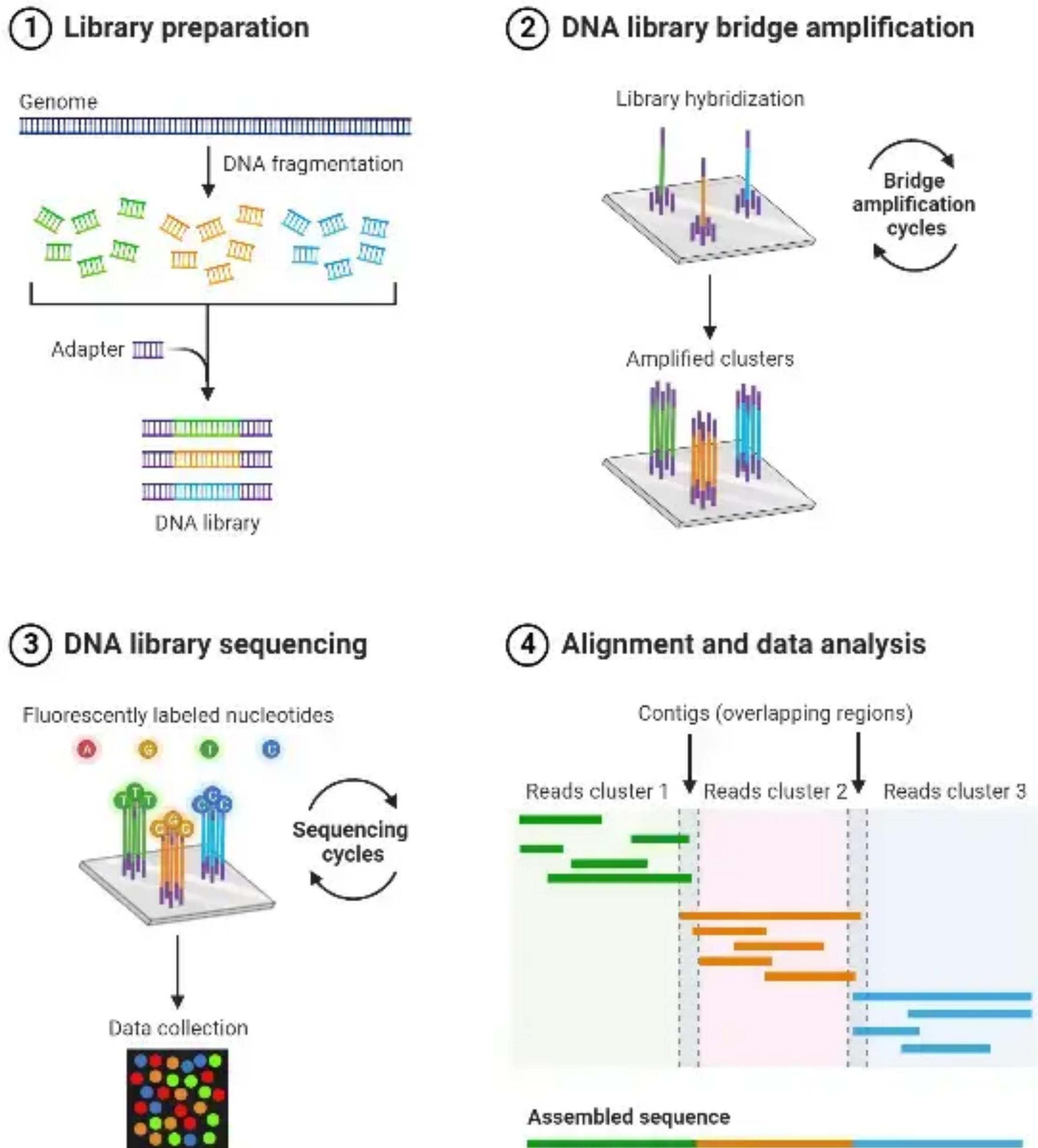
- Sequencing Process:
 - NGS: Massively parallel, short reads, needs assembly.
 - Sanger: Chain termination, longer reads, direct sequence.
- Read Lengths:
 - NGS: Short reads (50–300 bp), high volume.
 - Sanger: Long reads (up to 1000 bp), single fragments.



NGS

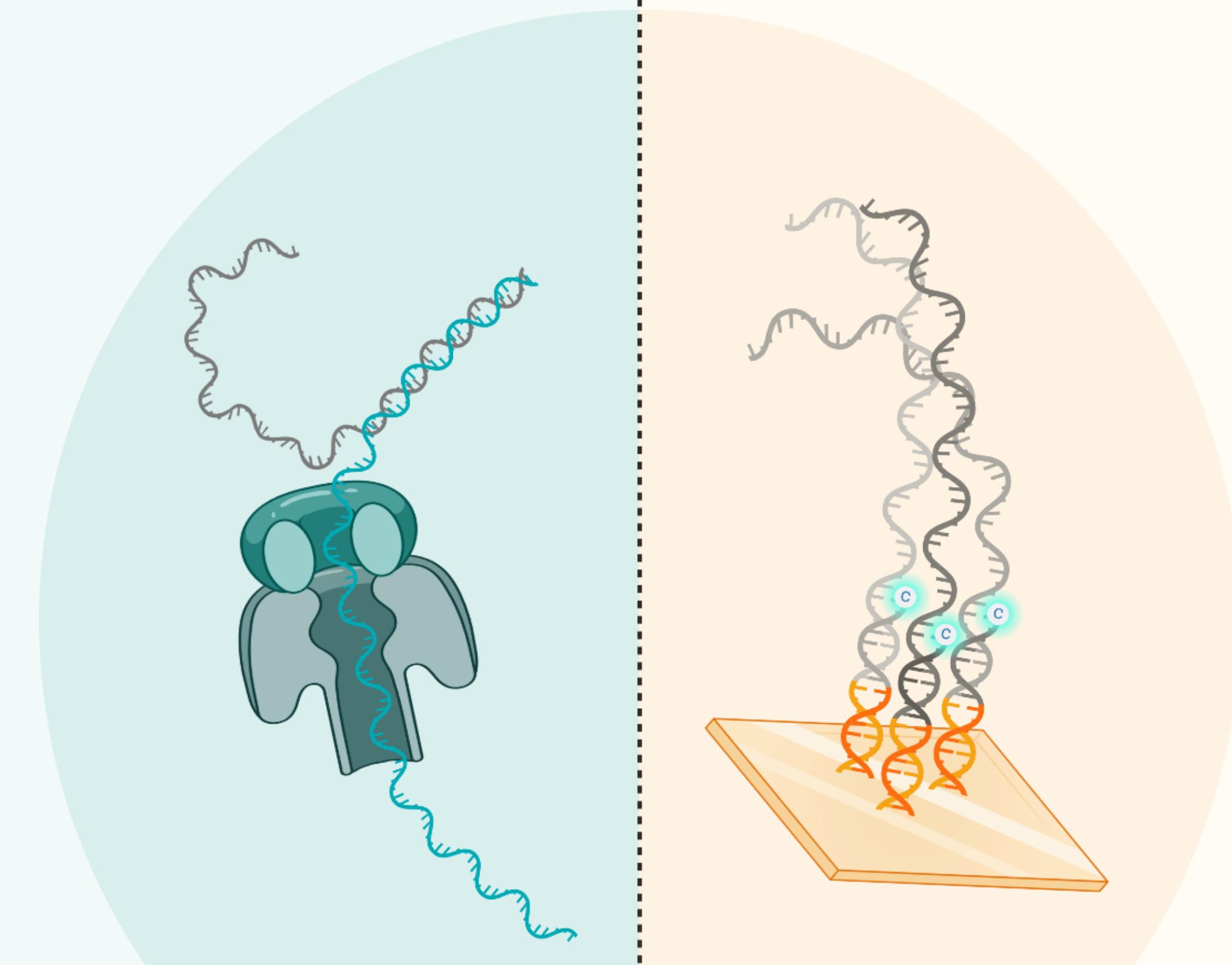
High-throughput sequencing

- Data Detection:
 - NGS: Real-time base detection, needs bioinformatics for assembly.
 - Sanger: Fluorescent chain-termination, minimal assembly.
- Amplification:
 - NGS: Cluster amplification (bridge or emulsion PCR).
 - Sanger: Traditional PCR, no clustering needed.



Oxford
NANOPORE
Technologies

illumina



Fasta and Fastq

Storing sequences

In Fasta format

- When we don't need to record the quality information.
- Only need to record the sequence.
- Two fields:

- Header

- Sequence

Header	>VIT_201s0011g03530.1
Sequence	AATTAAGCATAAAATACTCACTCTTACCCCTTATTTCTTATCTCTCATCACTTTGGTGCAG GACCATGAGAACAAAGCTGCAATGGGTGTAGGGTTCTCGCAAGGCATGCAGCCAAGACTGCATCA
Header	>VIT_201s0011g03540.1
Sequence	CAGGTAGCGTGAAGTTAACCCCTAGCGCTTAGACAAACAGCTGTAGTCACCGCCCACAAACACC AGCCTCTGAGACACCACCTCAAACCTTCCACTAAATACACATCCCTCACACCCTTTCAATT
Header	>VIT_201s0011g03550.1
Sequence	CATGCAAAGCTGAACCGCGATGCTGTGATTGGTGGTAAGTGGTAGTTGAGTAAATTGACAGTGAA GCCGAAATGGTAAAAGACTAAGGCTAGAAGTAGAATACCACTGTTCTCATCACGTGGGCCA

Storing sequences

In Fasta format

- Pros:
 - Simple
- Cons:
 - No universal specification for the format of an header
 - Use a strict naming convention and be consistent.

```
>ENSMUSG00000020122|ENSMUST00000138518
>ENSMUSG00000020122|ENSMUST00000125984
>ENSMUSG00000020122|ENSMUST00000125984|epidermal growth factor receptor
>ENSMUSG00000020122|ENSMUST00000125984|Egfr
>ENSMUSG00000020122|ENSMUST00000125984|11|ENSF00410000138465
```

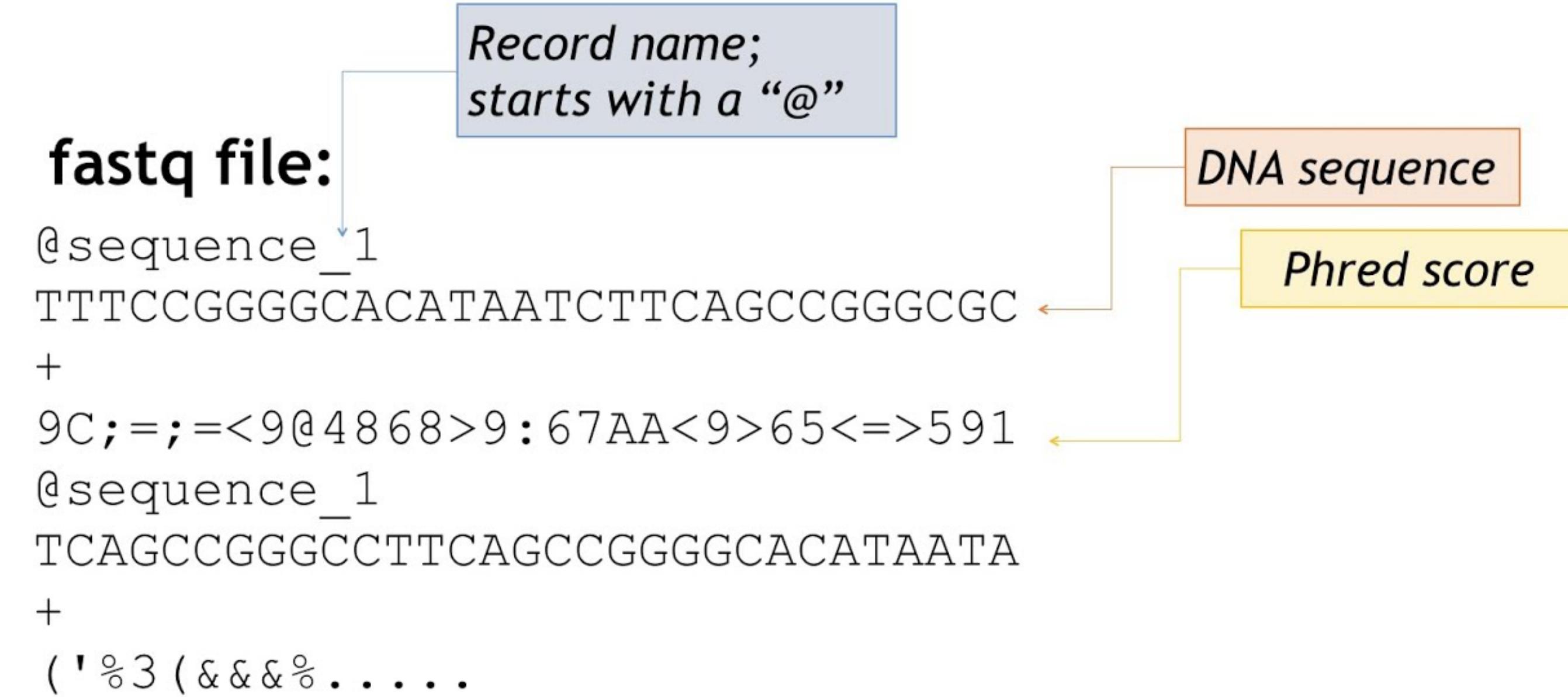
```
>gene_00284728 length=231;type=dna
GAGAACTGATTCTGTTACCGCAGGGCATTGGATGTGCTAAGGTAGTAATCCATTATAAGTAACATGCGCGGAATATCCG
GAGGTCATAGTCGTAATGCATAATTATTCCCTCCCTCAGAAGGACTCCCTGGCAGACGCCAACCAAGACTTCTGTA
GCTGGAACGATTGGACGGCCAACCGGGGGGAGTCGGCTACGTCTGATTGCTACGCCGGACTTCTCTT
```

Storing sequences

In Fastq format

- The FASTQ format extends FASTA by including a numeric quality score to each base in the sequence.
- The FASTQ format is widely used to store high-throughput sequencing data, which is reported with a per-base quality score indicating the confidence of each base call.
- Unfortunately like FASTA, FASTQ has variants and pitfalls that can make the seemingly simple format frustrating to work with.

Sequence File Formats



Storing sequences

In Fastq format

1. The description line, beginning with @. This contains the record identifier and other information.
2. Sequence data, which can be on one or many lines.
3. The line beginning with +, following the sequence line(s) indicates the end of the sequence. In older FASTQ files, it was common to repeat the description line here, but this is redundant and leads to unnecessarily large FASTQ files.
4. The quality data, which can also be on one or many lines, but must be the same length as the sequence. Each numeric base quality is encoded with ASCII characters using a particular scheme.

The diagram illustrates two FASTQ records. Each record consists of four lines: an identifier line starting with '@', a sequence line, a line starting with '+', and quality scores. The quality scores line uses ASCII characters to represent base qualities. Lines are color-coded: pink for labels ('Identifier', 'Sequence', etc.) and grey for the actual data lines. Pink arrows point from the labels to their corresponding lines.

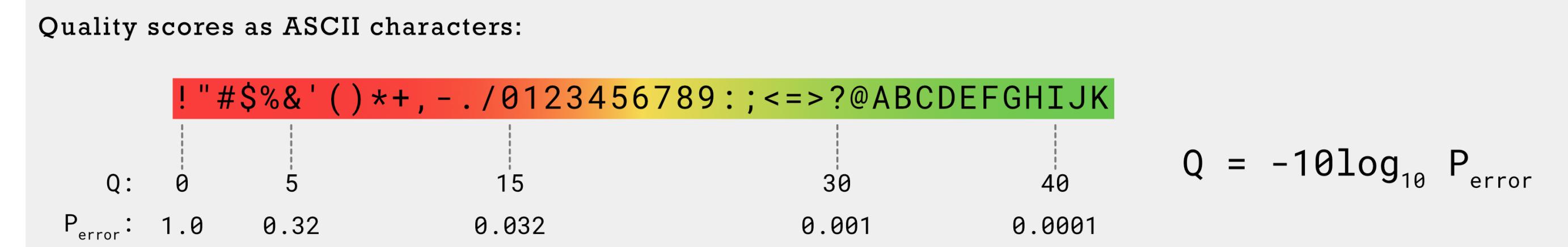
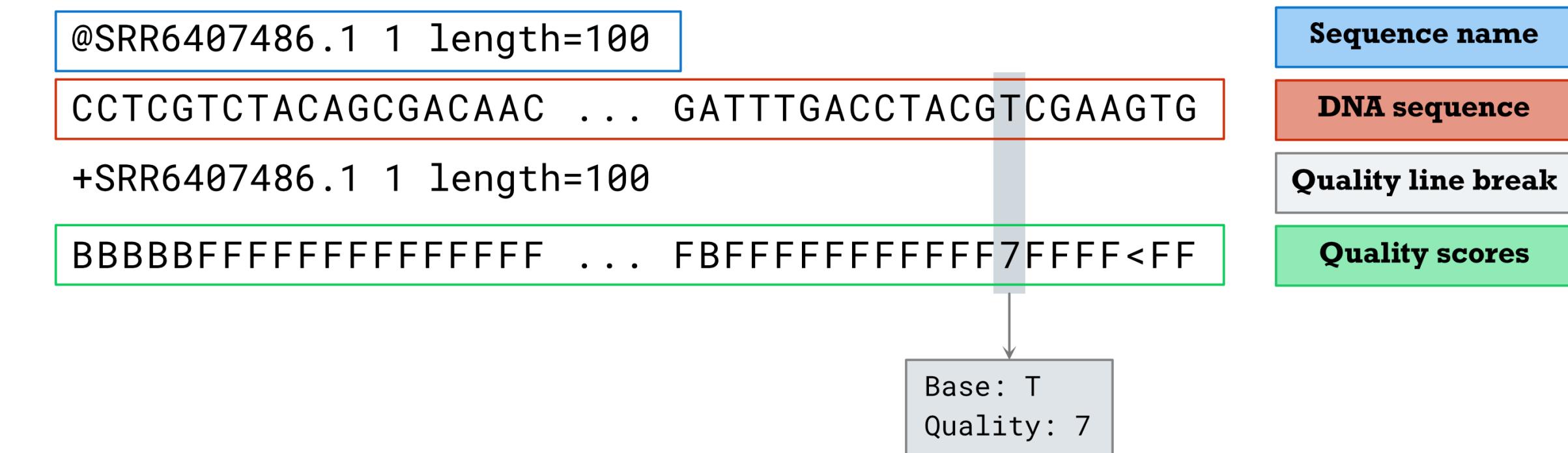
Identifier	@SRR566546.970 HWUSI-EAS1673_11067_FC7070M:4:1:2299:1109 length=50
Sequence	TTGCCTGCCTATCATTAGTGCCTGTGAGGTGGAGATGTGAGGATCAGT
'+' sign	+
Quality scores	hhhhhhhhghhhhhfffffe'ee['X]b[d[ed' [Y[^Y
Identifier	@SRR566546.971 HWUSI-EAS1673_11067_FC7070M:4:1:2374:1108 length=50
Sequence	GATTGTATGAAAGTATAACAACTAAAACTGCAGGTGGATCAGAGTAAGTC
'+' sign	+
Quality scores	hhhhgfhhcghghggfcffdhfehhhhcehdchhdhahehffffde'bVd

Parsing FASTQ Files Correctly

- Common Pitfall: Treating every line starting with @ as a header line is incorrect.
- Quality Scores: @ is a valid quality character, so it may appear in quality lines.
- Line Wrapping: Sequence and quality lines can wrap, so position alone is unreliable.
- Solution: Ensure the number of quality score characters matches sequence characters for accurate parsing.

FASTQ file sample:

```
@SRR6407486.1 1 length=100
CCTCGTCTACAGCGACAACGTCCAGACCCGCGAACGGGTGATGCGGGCCCTGGCAAACGGTTGCACCCGGATCTGCCGATTGACCTACGTCGAAGTG
+SRR6407486.1 1 length=100
BBBBBFFFFFFFFFFFFFFFFFFF...<FFFFFFFFFFFBFFFFFFFBFFFFFFFBFFFFFFFB7FFFF<FF
```



Nucleotide codes

IUPAC scheme

- A, T, C, G represent the nucleotides adenine, thymine, cytosine, and guanine.
- Degenerate (or ambiguous) nucleotide codes are used to represent two or more bases.
- For example, N is used to represent any base. The International Union of Pure and Applied Chemistry (**IUPAC**) has a standardized set of nucleotides, both unambiguous and ambiguous.

IUPAC nucleotide code	Base
A	Adenine
C	Cytosine
G	Guanine
T (or U)	Thymine (or Uracil)
R	A or G
Y	C or T
S	G or C
W	A or T
K	G or T
M	A or C
B	C or G or T
D	A or G or T
H	A or C or T
V	A or C or G
N	any base
. or -	gap

Codon table

coding for amino acids

- Codon:** A sequence of three nucleotides coding for a specific amino acid or stop signal.
- Codon Table:** Maps 64 codons to amino acids or stop signals, essential for interpreting the genetic code.
- Redundancy:** Multiple codons can code for the same amino acid (e.g., leucine has six codons).
- Start Codon:** AUG, signaling the beginning of translation (codes for methionine).
- Stop Codons:** UAA, UAG, UGA, marking the end of protein synthesis.
- Universal Code:** Nearly universal across all organisms, with few exceptions in mitochondria and some microorganisms.

Standard DNA codon table [edit]

Amino-acid biochemical properties	Nonpolar (np)	Polar (p)	Basic (b)	Acidic (a)	Termination: stop codon *	Initiation: possible start codon ⇒
-----------------------------------	---------------	-----------	-----------	------------	---------------------------	------------------------------------

Standard genetic code^{[17][note 3]}

1st base	2nd base				3rd base	
	T	C	A	G		
T	TTT (Phe/F) Phenylalanine (np)	TCT TCC TCA TCG ⇒	TAT TAC (Ser/S) Serine (p) TAA TAG	TAT (Tyr/Y) Tyrosine (p) TAA Stop (Ochre) * ^[note 2] TAG Stop (Amber) * ^[note 2]	TGT (Cys/C) Cysteine (p)	T C
	CTT (Leu/L) Leucine (np)	CCT CCC CCA CCG	CAT CAC (Pro/P) Proline (np) CAA CAG	CGT (His/H) Histidine (b) CGC CGA CGG	TGC TGA TGG Stop (Opal) * ^[note 2] (Trp/W) Tryptophan (np)	A G
	ATC (Ile/I) Isoleucine (np)	ACT ACC ACA ACG	AAT AAC (Thr/T) Threonine (p) AAA AAG	AGT (Asn/N) Asparagine (p) AGC AGA AGG	CGT CGC CGA CGG (Arg/R) Arginine (b)	T C A G
	ATG ⇒ (Met/M) Methionine (np)					
G	GTT (Val/V) Valine (np)	GCT GCC GCA GCG ⇒	GAT GAC (Ala/A) Alanine (np) GAA GAG	GAT (Asp/D) Aspartic acid (a) GAC GAA (Glu/E) Glutamic acid (a)	GGT GGC GGA GGG	T C A G
	GTC					
	GTA					
	GTG ⇒					

Base qualities

Schemes

- Phred+33 (Sanger): Commonly used in Illumina and modern NGS data. Scores range from 0 to 93, with ASCII characters ! to ~.
- Phred+64 (Illumina 1.3–1.7): Older Illumina format, now mostly obsolete. Scores range from 0 to 62, with ASCII characters @ to h.
- Solexa/Illumina: Used in early Solexa sequencing. Negative scores are possible, less common today.
- Ensure compatibility with software by knowing the scheme used.
- Quality Interpretation: Higher scores = better quality (probability of correct base call).

Table 10-2. FASTQ quality schemes (adapted from Cock et al., 2010 with permission)

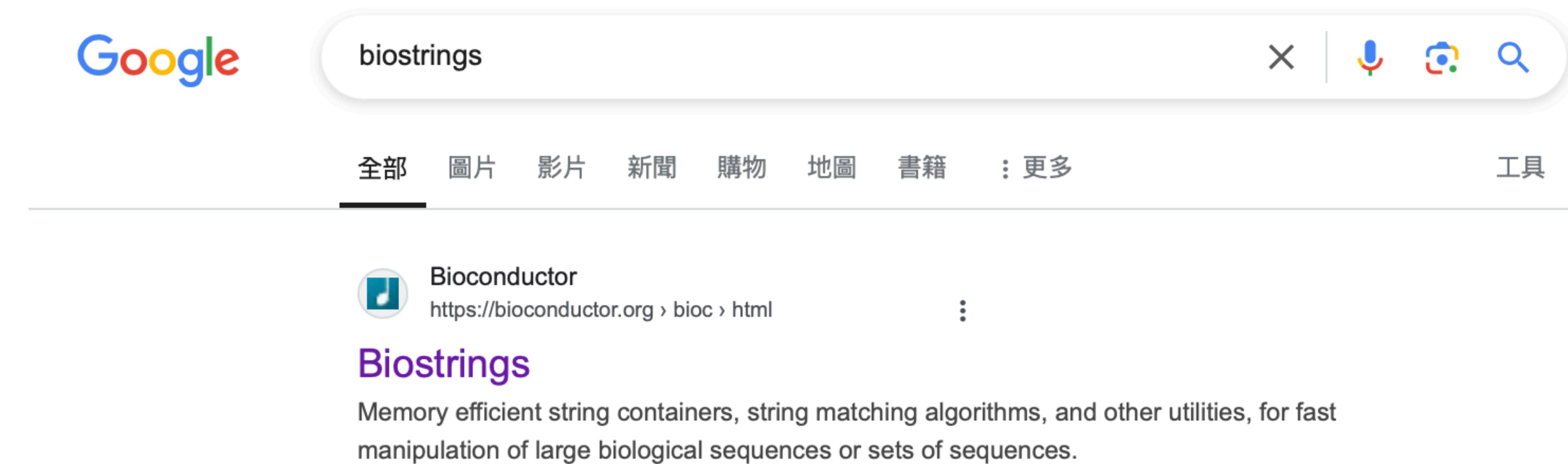
Name	ASCII character range	Offset	Quality score type	Quality score range
Sanger, Illumina (versions 1.8 onward)	33–126	33	PHRED	0–93
Solexa, early Illumina (before 1.3)	59–126	64	Solexa	5–62
Illumina (versions 1.3–1.7)	64–126	64	PHRED	0–62

Working with sequencing data in R

Google colab

New notebook

- Create a new notebook with proper name.
- Change the runtime to R.
- Install the “Biostrings” library.



<https://bit.ly/40DmtJm>



Github repository

for this lesson

- **Access Course Resources:** All materials, including scripts and datasets, are hosted on GitHub.
- **Repository Structure:** Organized for easy navigation, with separate folders for code, data, and documentation.
- **Hands-On Practice:** Follow examples and exercises directly from the provided resources.
- **Version Control:** Using GitHub ensures that you always have access to the latest updates and materials.
- **Contribute and Collaborate:** Students are encouraged to explore and even contribute if they have ideas or improvements.

The screenshot shows a GitHub repository page for 'HKU_Space_Data_handling_bioinformatics_lessons_2024'. The repository is public and has 1 branch and 0 tags. A recent commit by 'Koohoko' updated the content of 'coursework', 'scripts', 'sequence_data', '.gitignore', and 'README.md' one hour ago. The README file contains the following text:

```
HKU Space - Data handling on bioinformatics lessons (2024)

This repository contains the scripts for generating the data for the bioinformatics lessons at HKU Space. As well as the slides and other materials for the lessons.

For the coursework, please refer to the coursework repository
```

Github repository

Important resources

- Coursework requirements.
- Input data.
- Reference scripts.

The screenshot shows a GitHub repository page for 'HKU_Space_Data_handling_bioinformatics_lessons_2024'. The 'Code' tab is selected. The repository structure is shown under 'coursework':

- main (branch)
- HKU_Space_Data_handling_bioinformatics_lessons_2024 / coursework /

Commit history:

Name	Last commit message	Last commit date
...		
readme.md	update content	1 hour ago

File content for 'readme.md':

Scientific Skills Bioinformatics Exercise (30%)

Module Title: Scientific Skills
Program: MSc Biomedical Science
Assessment Weighting: 30% of Module Marks

Exercise Overview

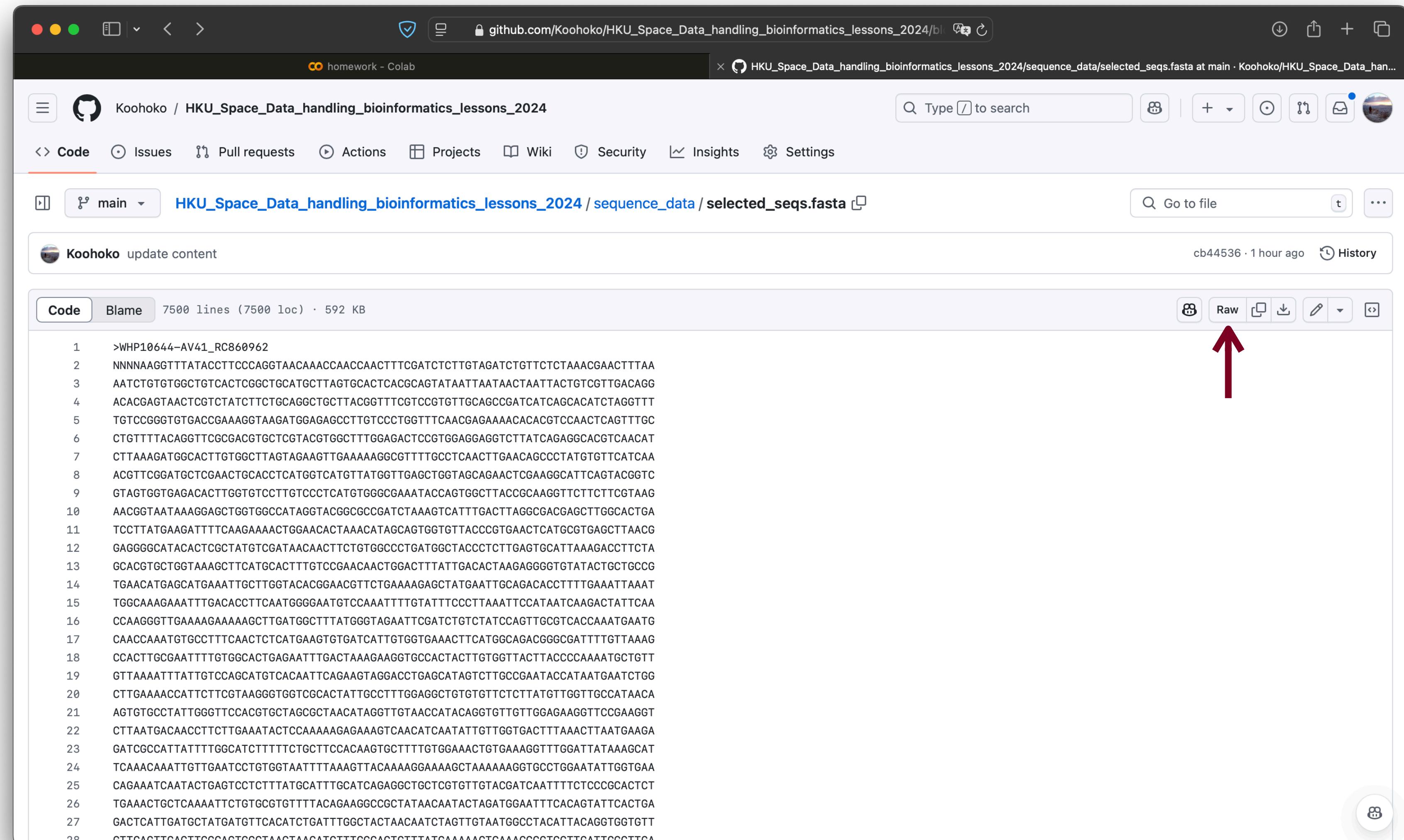
This exercise introduces students to bioinformatics data handling, focusing on data cleaning, sequence analysis, and reproducibility. Students will work in a pre-configured Google Colab notebook with guided sections to facilitate learning without requiring prior programming experience.

Learning Objectives

Github repository

Important resources

- Coursework requirements.
- Input data.
- Reference scripts.



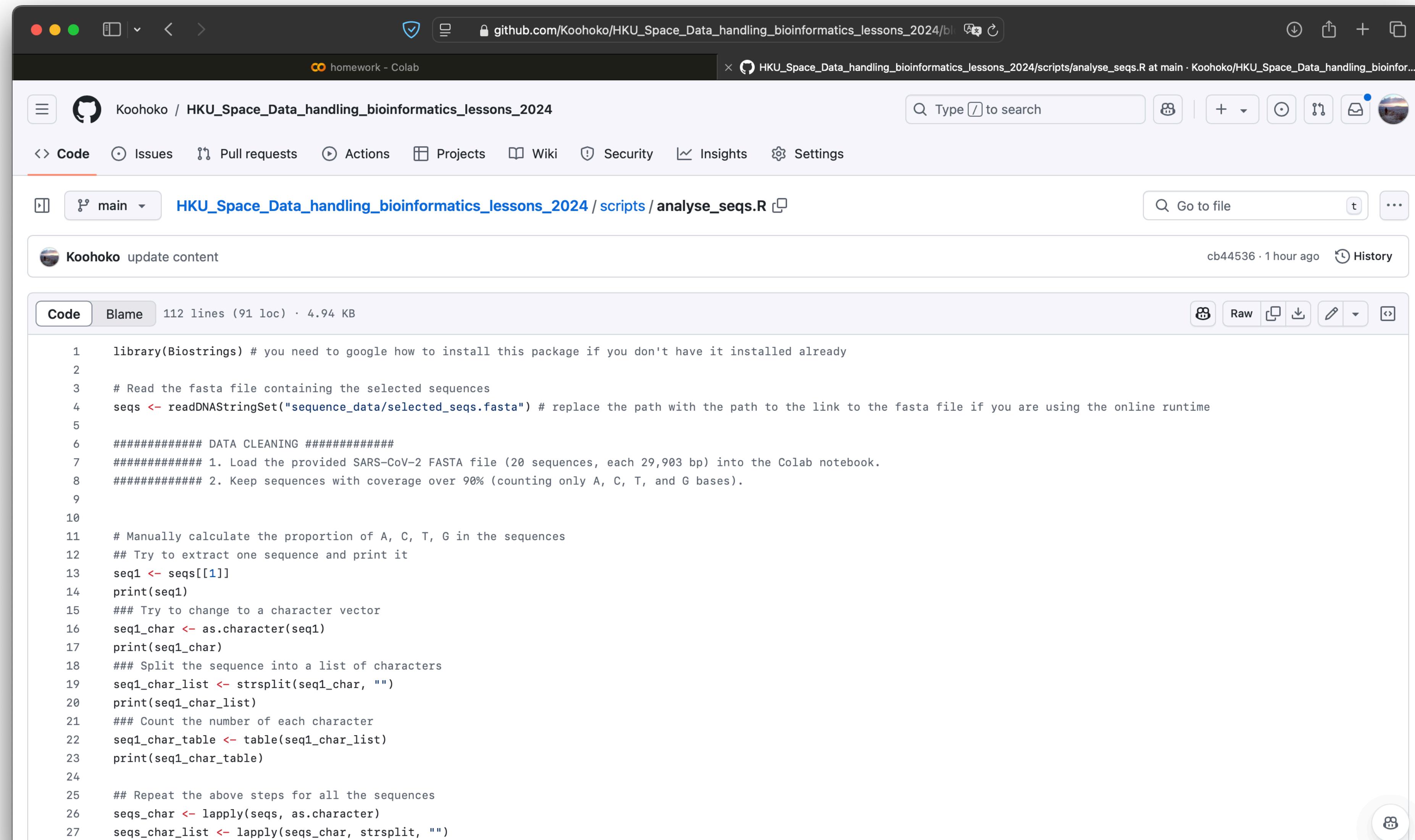
A screenshot of a GitHub repository page for 'Koohoko / HKU_Space_Data_handling_bioinformatics_lessons_2024'. The repository has 7500 lines (7500 loc) and 592 KB. A file named 'selected_seqs.fasta' is displayed. The file content is a sequence of DNA bases (A, T, C, G) starting with '1 >WHP10644-AV41_RC860962'. At the top right of the code editor, there is a red arrow pointing to the 'Raw' button. The URL in the browser bar is 'github.com/Koohoko/HKU_Space_Data_handling_bioinformatics_lessons_2024/b...'. The GitHub interface includes a navigation bar with 'Code', 'Issues', 'Pull requests', 'Actions', 'Projects', 'Wiki', 'Security', 'Insights', and 'Settings'.

```
1 >WHP10644-AV41_RC860962
2 NNNNAAGGTTTACCTCCCAAGTAACAAACCAACCAACTTCGATCTTGAGATCTGTTCTAAACGAACCTTAA
3 AATCTGTGGCTGTCACTCGGCTGCATGCTTAGTCACGCACTCAGCAGTATAATAACTAATTACTGTCGTTGACAGG
4 ACACGAGTAACCTGCTCATCTCTGCAGGCTGCTACGGTTCTGCCGTGTTGCAGCCGATCATCAGCACATCTAGTTG
5 TGTCGGGTGACCGAAAGGTAAGATGGAGAGCCTGTCCTGGTTCAACGAGAAAACACAGTCCAACCTCAGTTG
6 CTGTTTACAGGTCGCGACGTGCTGACGTGGCTTGGAGACTCCGTGGAGGAGTCTTACAGAGGCACGTCAACAT
7 CTTAAAGATGGCACTGTGGCTTAGTAGAAAGTTGAAAAAGGCCTTGCCTAACTTGAACAGCCCTATGTGTTCATCAA
8 ACGTTGGATGCTGCAACTGCACCTCATGGTCATGTTATGGTGAGCTGGTAGCAGAACTCGAAGGCATTCACTGGTC
9 GTAGTGGTGAGACACTTGGTGCCTTGTCCCTCATGTGGCGAAATACCAGTGGTACCGCAAGGTTCTTCGTAAG
10 AACGGTAATAAAGGGACTGGTGGCCATAGGTACGGGCCGATCTAAAGTCATTGACTTAGGGCAGAGCTGGCACTGA
11 TCCTTATGAAGATTTCAAGAAAACGAAACTAAACATAGCAGTGGTGTACCCGTGAACCTATGCGTGAGCTTAACG
12 GAGGGCATACTCGCTATGTCGATAACAACCTCTGTCCTGATGGCTACCCCTTGTGAGTCATTAAAGACCTTCTA
13 GCACGTGCTGGTAAAGCTTCATGCACTTGTCCGAACAACACTGGACTTATTGACACTAAGAGGGGTGTACTGCTGCC
14 TGAACATGAGCATGAAATTGCTGGTACACGGAACGCTTGAAAGAGCTATGAATTGCGACAGCCTTGAATTAAAT
15 TGGCAAAAGAATTGACACCTCAATGGGAATGCAAATTGTATTCCCTAAATCCATAATCAAGACTATTCAA
16 CCAAGGGTTGAAAGAAAAAGCTGTGATGGCTTATGGTGAATTGATCTGTCTATCCAGTTGCGTACCAATGAATG
17 CAACCAAATGTGCCCTTCAACTCTCATGAAAGTGTGATCATGGTGGAAACTCTGGCAGACGGCGATTGTTAAAG
18 CCACTTGCATGAAATTGCTGGACTGAGAATTGACTAAAGAAGGTGCCACTACTTGTTACTTACCCCAAATGCTGTT
19 GTTAAATTATTGTCCAGCATGTCACAATTGAGCTAGTGGACCTGAGCATAGTCTGGCGAATACCATATAATGTTG
20 CTTGAAAACCATCTTCGTAAGGGTGGTCCACTATTGCTTGGAGGCTGTGTTCTTATGTTGGTCCATAACA
21 AGTGTCCCTATTGGTCCACGTGCTAGCGCTAACATAGTTGTAACCATACAGGTGTTGGAGAAGGTTCCGAAGGT
22 CTTAATGACAACCTTCTGAAATCTCCAAAAGAGAAAGTCAACATCAATATTGTTGGTACTTAAACTTAATGAAGA
23 GATGCCATTATTTGGCATCTTCTGCTCCACAAGTGCTTGTGAAACTGTGAAAGGTTGGATTATAAGCAT
24 TCAAACAAATTGTTGAATCTGTGTAATTAAAGTACAAAGGAAAGCTAAAAAGGTGCTGGAAATTGGTGA
25 CAGAAATCAACTGAGTCTCTTATGTCATTTGCTACAGGGCTGCTGTTGACGATCAATTCTCCGCACACT
26 TGAAACTGCTCAAATTGCTGCGTGTGTTACAGAAGGGCGCTATAACATAACTAGTGGATTTACAGTATTCACTGA
27 GACTCATTGATGCTATGATGTTCACATCTGATTGGCTACTAACATCTAGTGTAAATGGCTACATTACAGGTGGTGT
28 GTTCAGTTGACTTCGAGTGGCTAACTAACATCTTGGACTTTATGAAAAACTCAAACCCCTCCTGATTGGCTGA
```

Github repository

Important resources

- Coursework requirements.
- Input data.
- Reference scripts.



The screenshot shows a GitHub Colab notebook interface. The top bar indicates the URL is `github.com/Koohoko/HKU_Space_Data_handling_bioinformatics_lessons_2024`. The main navigation bar includes links for Code, Issues, Pull requests, Actions, Projects, Wiki, Security, Insights, and Settings. The current view is on the 'Code' tab of the file `HKU_Space_Data_handling_bioinformatics_lessons_2024 / scripts / analyse_seqs.R`. A commit by 'Koohoko' titled 'update content' is visible, made 1 hour ago. The code itself is an R script for sequence analysis:

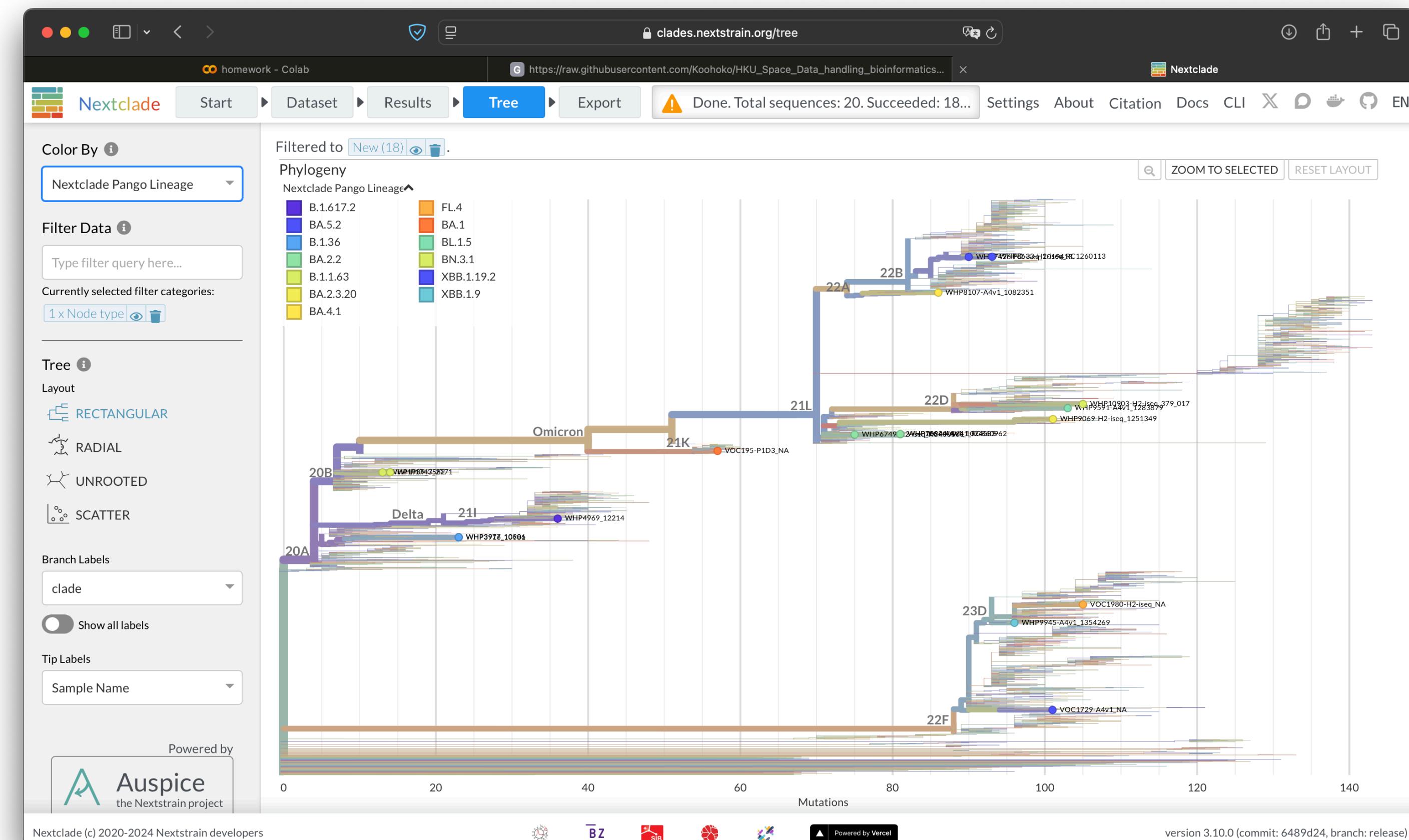
```
library(Biostrings) # you need to google how to install this package if you don't have it installed already
# Read the fasta file containing the selected sequences
seqs <- readDNAStringSet("sequence_data/selected_seqs.fasta") # replace the path with the link to the fasta file if you are using the online runtime
#####
## DATA CLEANING #####
### 1. Load the provided SARS-CoV-2 FASTA file (20 sequences, each 29,903 bp) into the Colab notebook.
### 2. Keep sequences with coverage over 90% (counting only A, C, T, and G bases).
#
# Manually calculate the proportion of A, C, T, G in the sequences
## Try to extract one sequence and print it
seq1 <- seqs[[1]]
print(seq1)
### Try to change to a character vector
seq1_char <- as.character(seq1)
print(seq1_char)
### Split the sequence into a list of characters
seq1_char_list <- strsplit(seq1_char, "")
print(seq1_char_list)
### Count the number of each character
seq1_char_table <- table(seq1_char_list)
print(seq1_char_table)
#
## Repeat the above steps for all the sequences
seqs_char <- lapply(seqs, as.character)
seqs_char_list <- lapply(seqs_char, strsplit, "")
```

**Checking the sequences
using nextclade**

Nextclade

online tool

- **Purpose:** Nextclade is a tool for the analysis of SARS-CoV-2 genetic sequences, focusing on the assignment of variants and detection of mutations.
- **Variant Classification:** It classifies viral genomes into clades and assigns them to specific SARS-CoV-2 lineages based on mutation patterns.
- **Mutation Detection:** Nextclade identifies both synonymous and non-synonymous mutations, helping to track the evolution of the virus.
- **Alignment-Based:** The tool compares input sequences to a reference genome, providing insights into the sequence variation.



End of lesson II