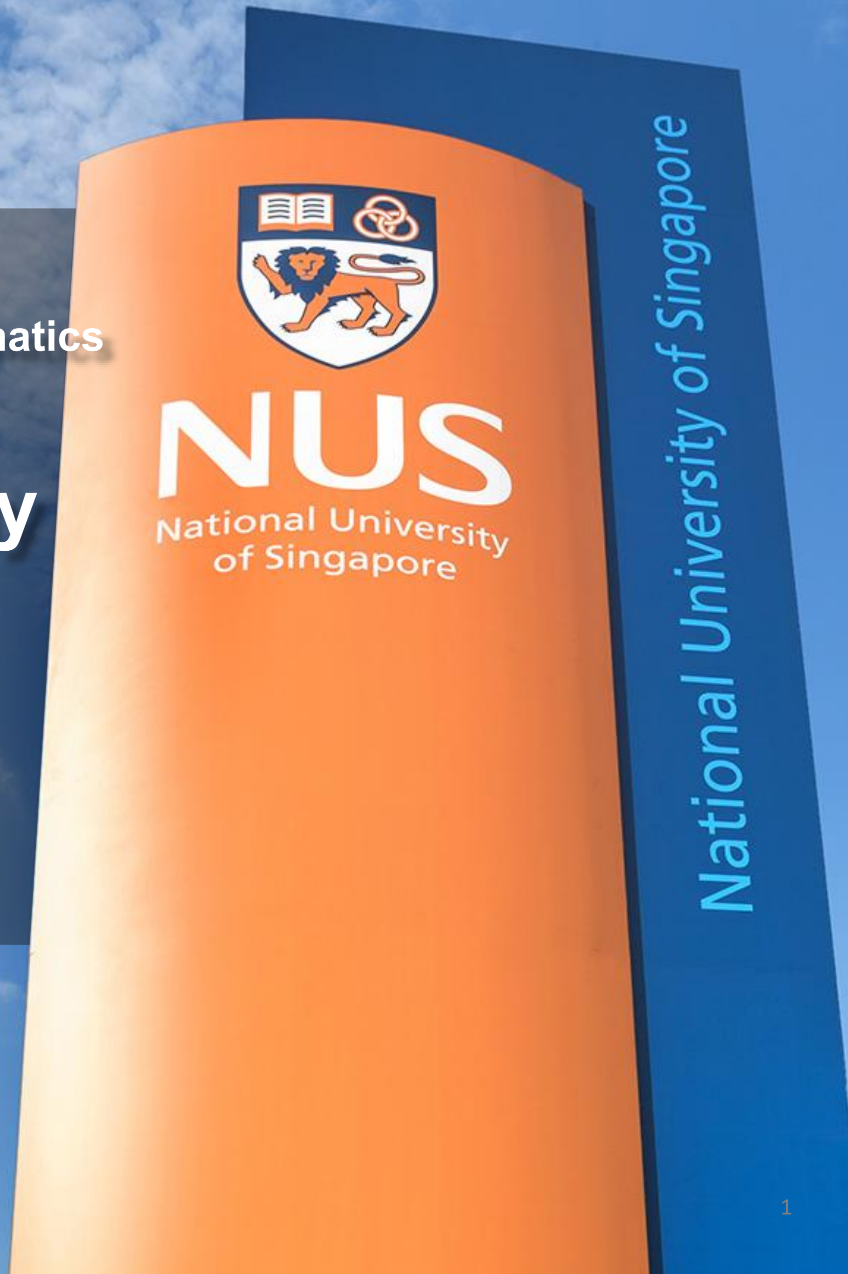
CS4330: Combinatorial Methods in Bioinformatics

Primer on referenced-based genome assembly

Wong Limsoon


NUS
National University of Singapore
National University of Singapore

Types of genome assembly

Reference-based genome assembly

Reconstruct a genome by aligning/mapping sequencing reads to a known well-annotated genome (the reference)

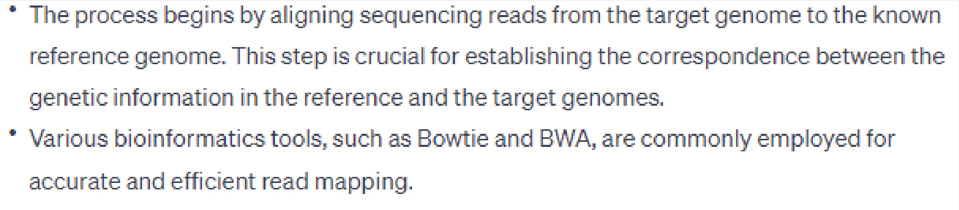
De novo genome assembly

Reconstruct a genome from sequencing reads w/o using any reference genome

Reference-guided assembly for long reads

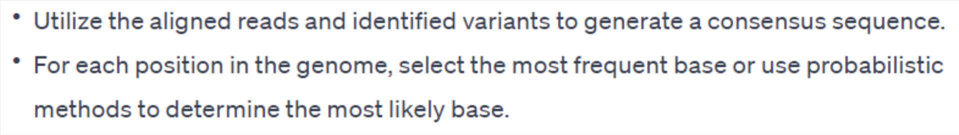
Key steps of reference-based assembly

Map reads to ref genome



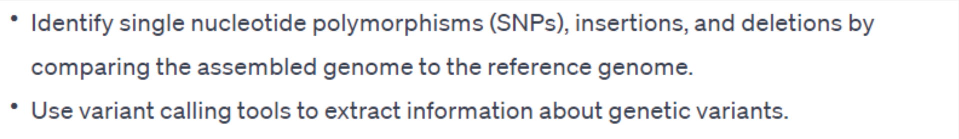
- The process begins by aligning sequencing reads from the target genome to the known reference genome. This step is crucial for establishing the correspondence between the genetic information in the reference and the target genomes.
- Various bioinformatics tools, such as Bowtie and BWA, are commonly employed for accurate and efficient read mapping.

Generate consensus & assemble the genome



- Utilize the aligned reads and identified variants to generate a consensus sequence.
- For each position in the genome, select the most frequent base or use probabilistic methods to determine the most likely base.

Call variant



- Identify single nucleotide polymorphisms (SNPs), insertions, and deletions by comparing the assembled genome to the reference genome.
- Use variant calling tools to extract information about genetic variants.

Mapping reads to reference genome

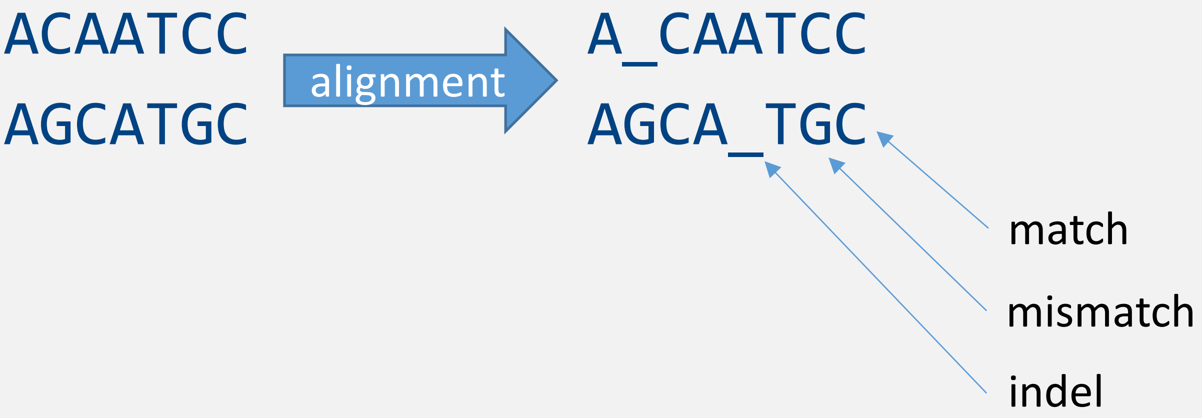
BWA (a popular read-mapping tool) creates index of reference genome using **Burrows-Wheeler transform**

During read alignment, BWA breaks a read into overlapping **k-mers**, and uses k-mers as “seeds” to search the index for match positions

BWA then extends the alignment from seed locations to find best alignment

Alignment

Alignment is the process of comparing DNA sequences to identify similarities, differences, or regions of interest



ACAATCC

AGCATGC

alignment

A_CAATCC

AGCA_TGC

match

mismatch

indel

Computational complexity is cubic wrt length of sequences ⇒ K-mer matching is used to speed this up

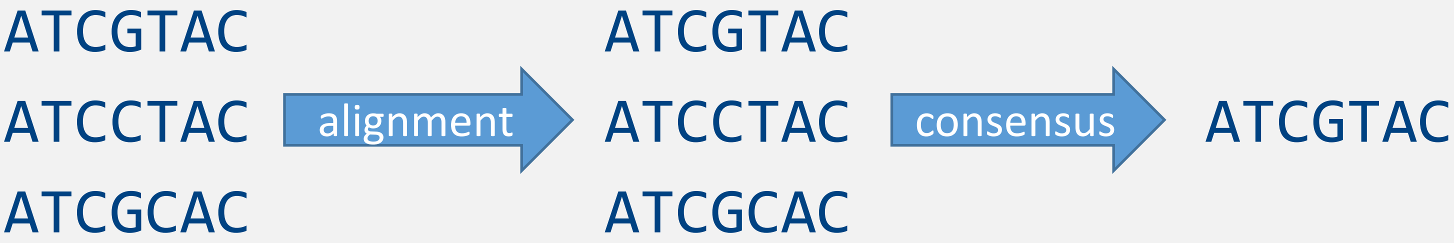
K-mers

A k-mer is a contiguous sequence of k nucleotide bases

K-mers provide a balance between sensitivity and efficiency in many bioinformatics applications

Referenced-based assembly methods like BWA use k-mers extensively for mapping reads to the reference genome

Consensus generation



ATCGTAC
ATCCTAC
ATCGCAC

alignment

ATCGTAC
ATCCTAC
ATCGCAC

consensus

ATCGTAC

Usually, a consensus sequence represents the most common base at each position. If there is a tie, any of the tied bases can be chosen

But other representations are often used as well. E.g., for a diploid genome, we might use `ATC[G/C][T/C]AC` as the consensus

Advantages compared to de novo assembly

Faster and less resource intensive

More accurate in regions with high similarity to the ref

Suitable for species with well-characterized genomes

Using the same ref provides a standardized framework for genome analysis & comparison across studies

Key limitation

A suitable well-annotated reference genome is a pre-requisite for reference-based genome assembly

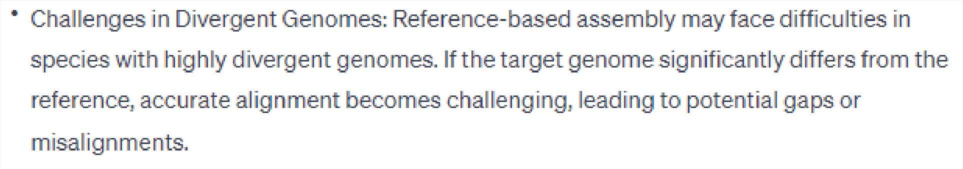
Works well for model organisms and humans

Inapplicable for assembling the genome of an organism that is too different from the above

De novo assembly has to be used in such a case

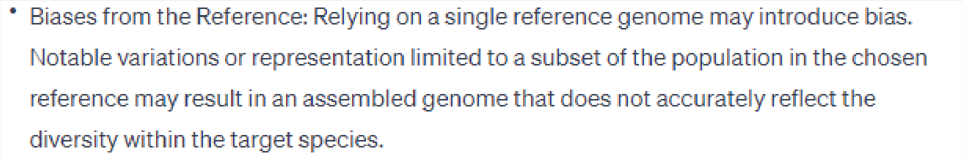
Other challenges to reference-based assembly

Genome variability



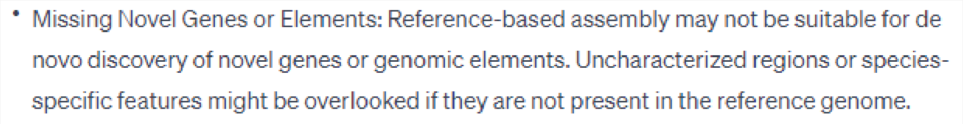
- Challenges in Divergent Genomes: Reference-based assembly may face difficulties in species with highly divergent genomes. If the target genome significantly differs from the reference, accurate alignment becomes challenging, leading to potential gaps or misalignments.

Introduction of bias

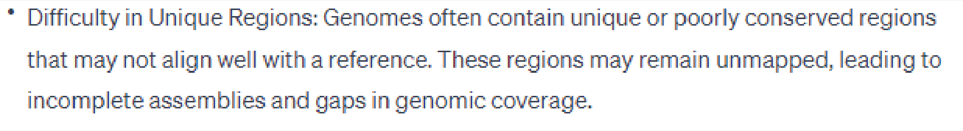


- Biases from the Reference: Relying on a single reference genome may introduce bias. Notable variations or representation limited to a subset of the population in the chosen reference may result in an assembled genome that does not accurately reflect the diversity within the target species.

Incomplete or unmapped regions

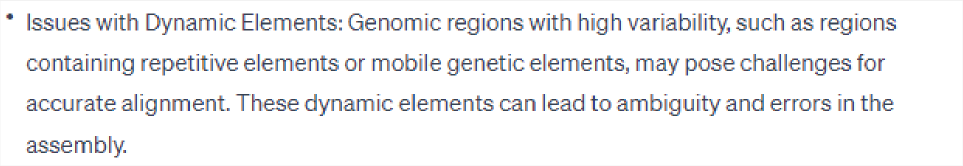


- Missing Novel Genes or Elements: Reference-based assembly may not be suitable for de novo discovery of novel genes or genomic elements. Uncharacterized regions or species-specific features might be overlooked if they are not present in the reference genome.



- Difficulty in Unique Regions: Genomes often contain unique or poorly conserved regions that may not align well with a reference. These regions may remain unmapped, leading to incomplete assemblies and gaps in genomic coverage.

Dynamic genomic regions



- Issues with Dynamic Elements: Genomic regions with high variability, such as regions containing repetitive elements or mobile genetic elements, may pose challenges for accurate alignment. These dynamic elements can lead to ambiguity and errors in the assembly.

Best practices

Quality control

Thoroughly check quality of raw sequencing data

Trim low-quality bases, remove adapters, and filter reads that do not meet predefined quality standards

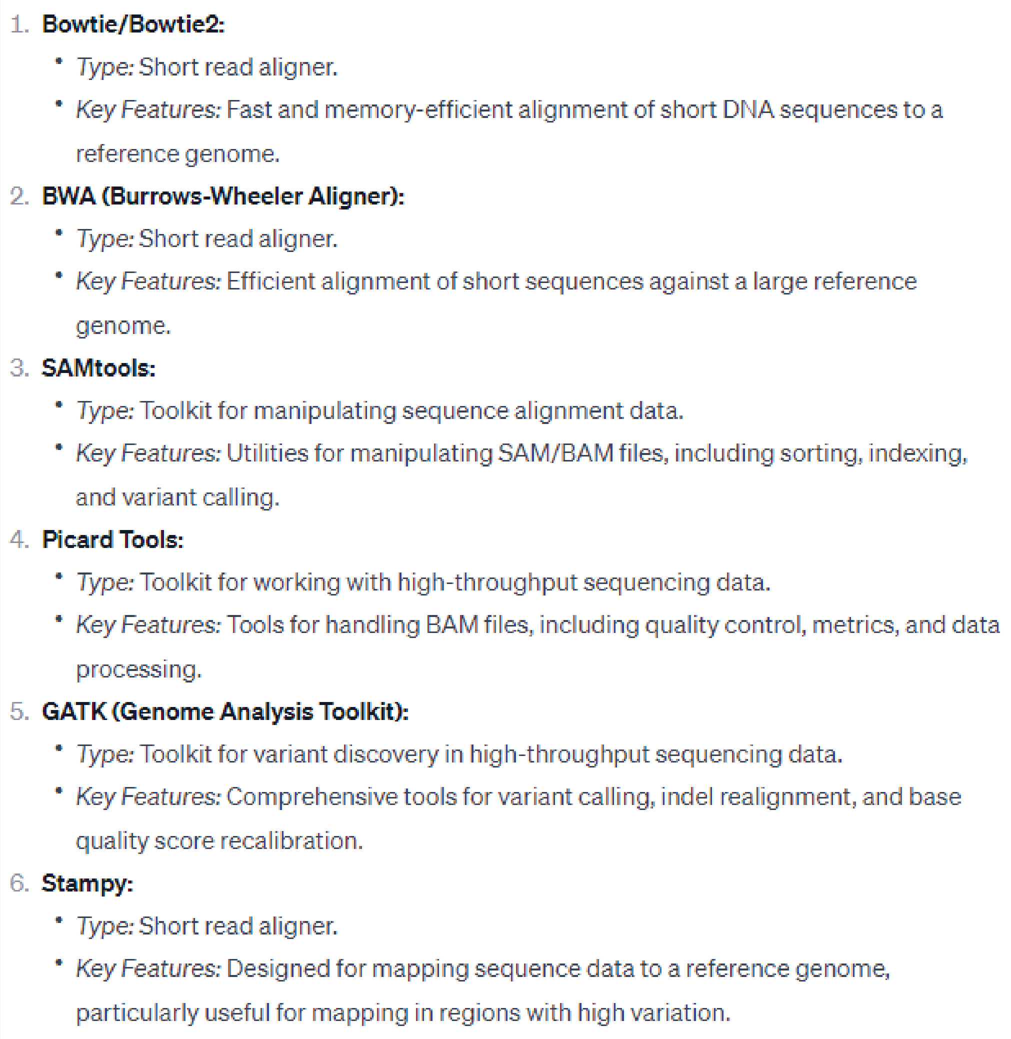
Reference genome selection

Choose a ref genome closely related to the target species

Be mindful of the quality and representativeness of the ref; biases may be introduced if ref is not suited to the target

Popular tools

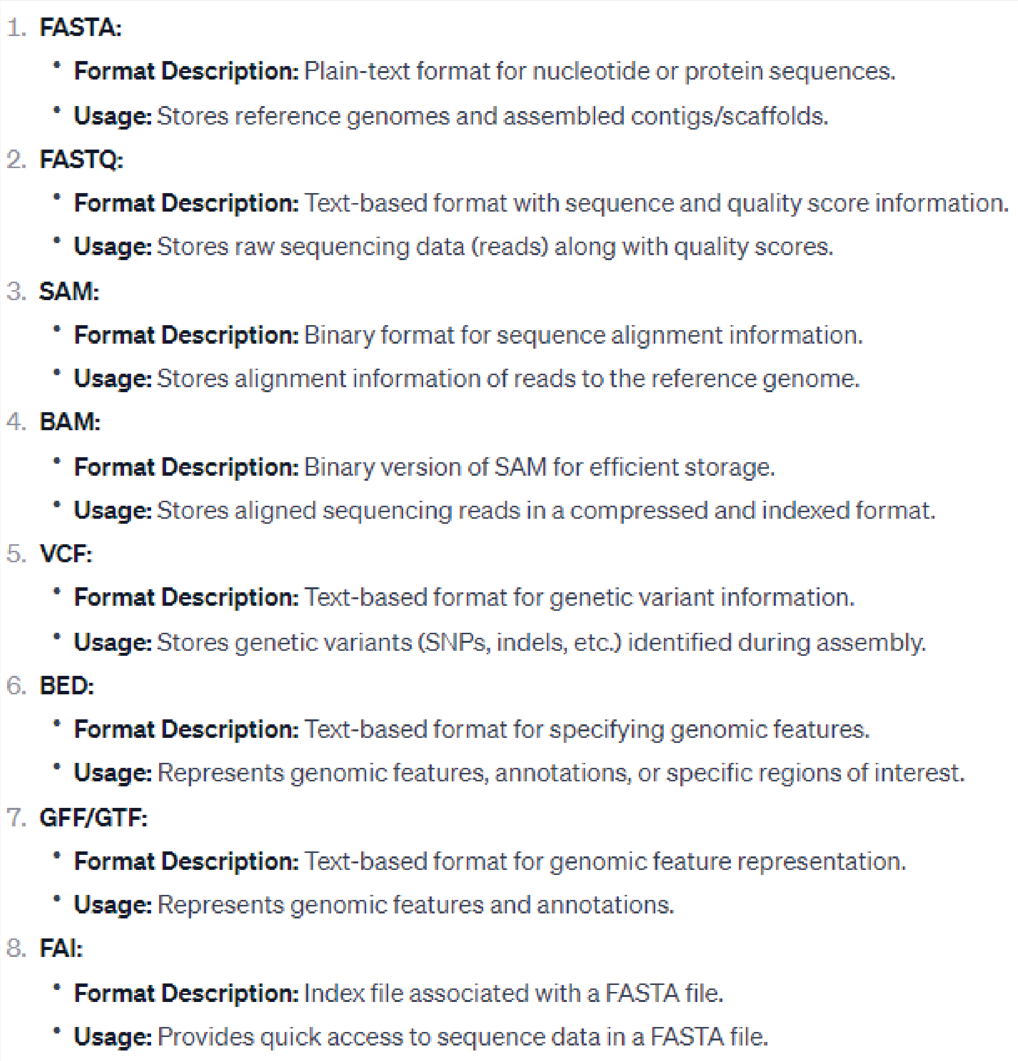
Check these out yourself



1. **Bowtie/Bowtie2:**
   - *Type:* Short read aligner.
   - *Key Features:* Fast and memory-efficient alignment of short DNA sequences to a reference genome.
2. **BWA (Burrows-Wheeler Aligner):**
   - *Type:* Short read aligner.
   - *Key Features:* Efficient alignment of short sequences against a large reference genome.
3. **SAMtools:**
   - *Type:* Toolkit for manipulating sequence alignment data.
   - *Key Features:* Utilities for manipulating SAM/BAM files, including sorting, indexing, and variant calling.
4. **Picard Tools:**
   - *Type:* Toolkit for working with high-throughput sequencing data.
   - *Key Features:* Tools for handling BAM files, including quality control, metrics, and data processing.
5. **GATK (Genome Analysis Toolkit):**
   - *Type:* Toolkit for variant discovery in high-throughput sequencing data.
   - *Key Features:* Comprehensive tools for variant calling, indel realignment, and base quality score recalibration.
6. **Stampy:**
   - *Type:* Short read aligner.
   - *Key Features:* Designed for mapping sequence data to a reference genome, particularly useful for mapping in regions with high variation.

Popular data formats

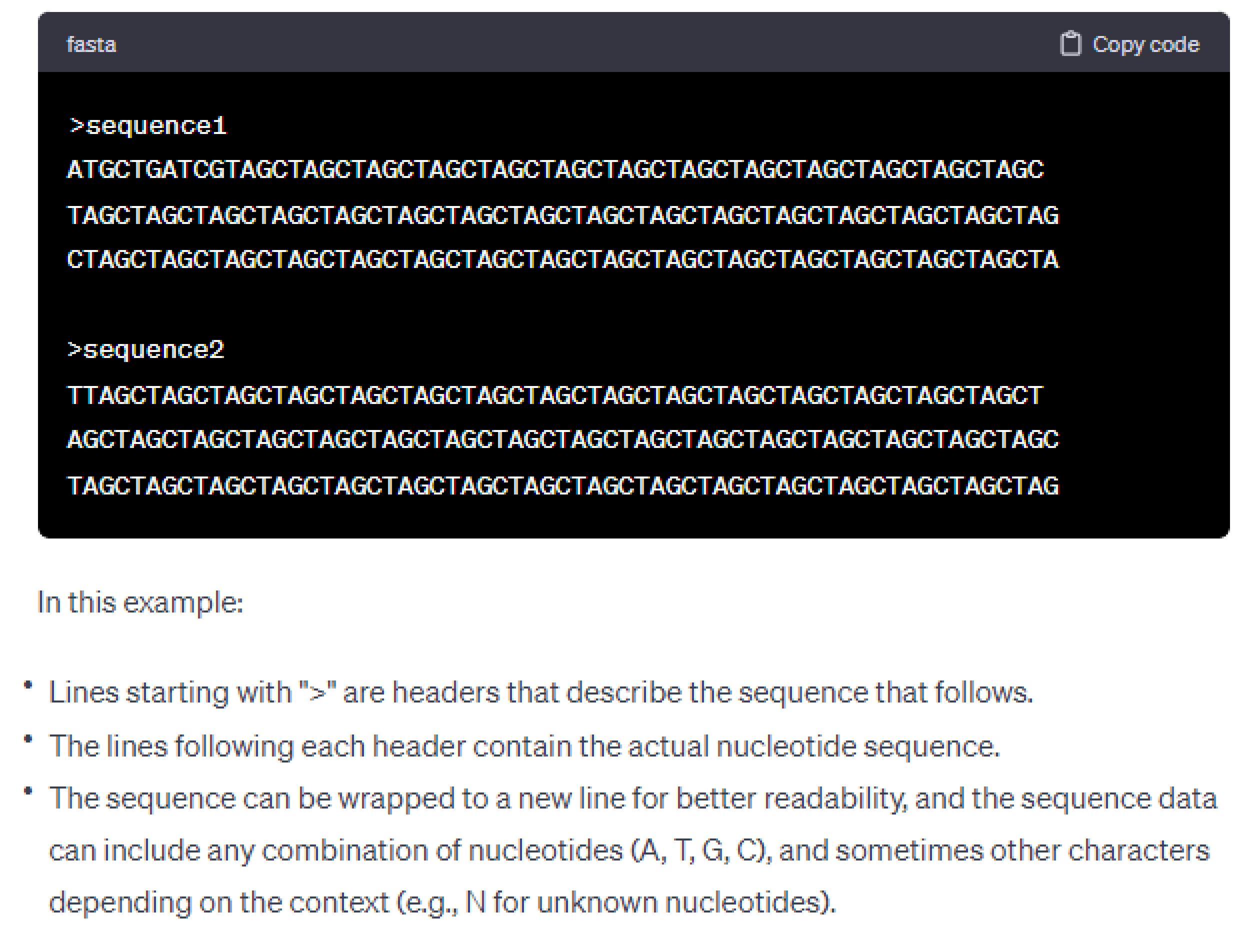
Check these out yourself



1. **FASTA:**
   - **Format Description:** Plain-text format for nucleotide or protein sequences.
   - **Usage:** Stores reference genomes and assembled contigs/scaffolds.
2. **FASTQ:**
   - **Format Description:** Text-based format with sequence and quality score information.
   - **Usage:** Stores raw sequencing data (reads) along with quality scores.
3. **SAM:**
   - **Format Description:** Binary format for sequence alignment information.
   - **Usage:** Stores alignment information of reads to the reference genome.
4. **BAM:**
   - **Format Description:** Binary version of SAM for efficient storage.
   - **Usage:** Stores aligned sequencing reads in a compressed and indexed format.
5. **VCF:**
   - **Format Description:** Text-based format for genetic variant information.
   - **Usage:** Stores genetic variants (SNPs, indels, etc.) identified during assembly.
6. **BED:**
   - **Format Description:** Text-based format for specifying genomic features.
   - **Usage:** Represents genomic features, annotations, or specific regions of interest.
7. **GFF/GTF:**
   - **Format Description:** Text-based format for genomic feature representation.
   - **Usage:** Represents genomic features and annotations.
8. **FAI:**
   - **Format Description:** Index file associated with a FASTA file.
   - **Usage:** Provides quick access to sequence data in a FASTA file.

I got ChatGPT to generate some example data …

Example: FASTA



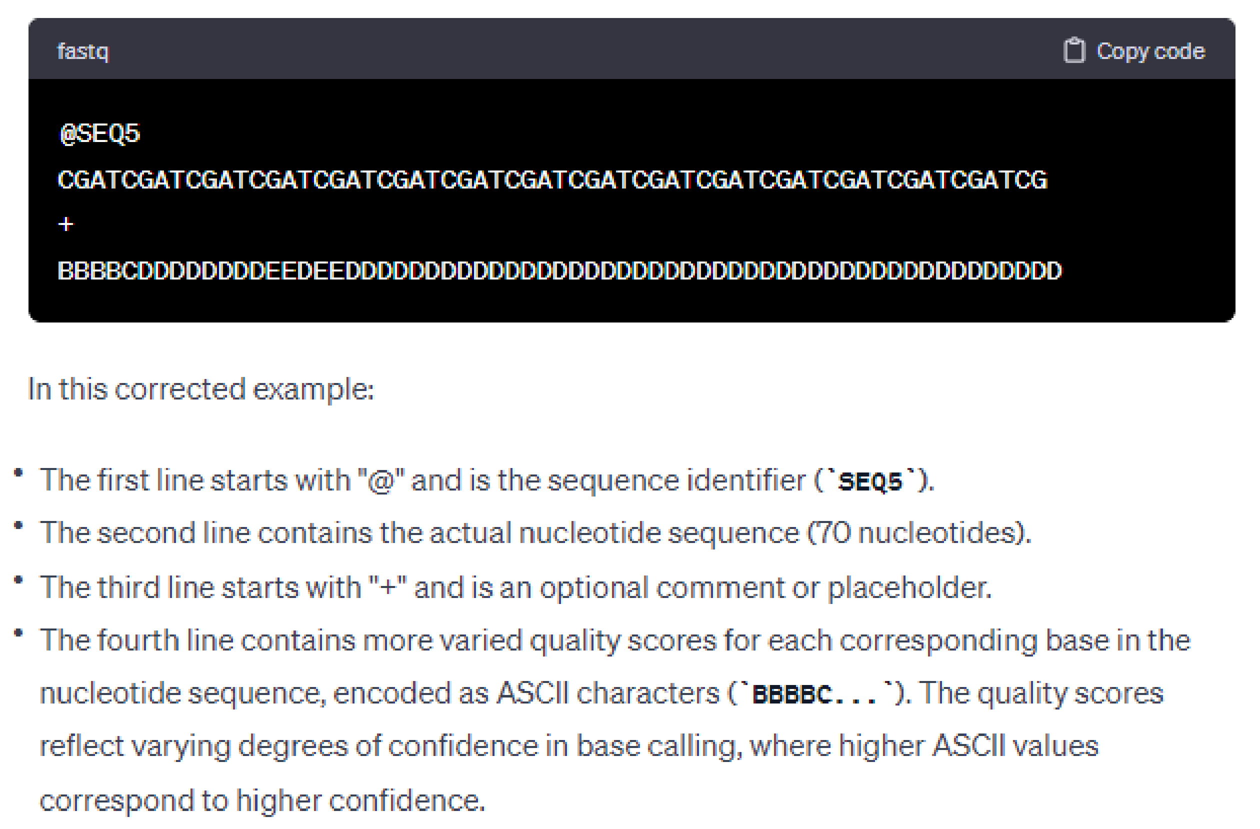
```fasta
>sequence1
ATGCTGATCGTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGC
TAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAG
CTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTA

>sequence2
TTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCT
AGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGC
TAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAGCTAG
```

In this example:

- Lines starting with ">" are headers that describe the sequence that follows.
- The lines following each header contain the actual nucleotide sequence.
- The sequence can be wrapped to a new line for better readability, and the sequence data can include any combination of nucleotides (A, T, G, C), and sometimes other characters depending on the context (e.g., N for unknown nucleotides).

Example: FASTQ

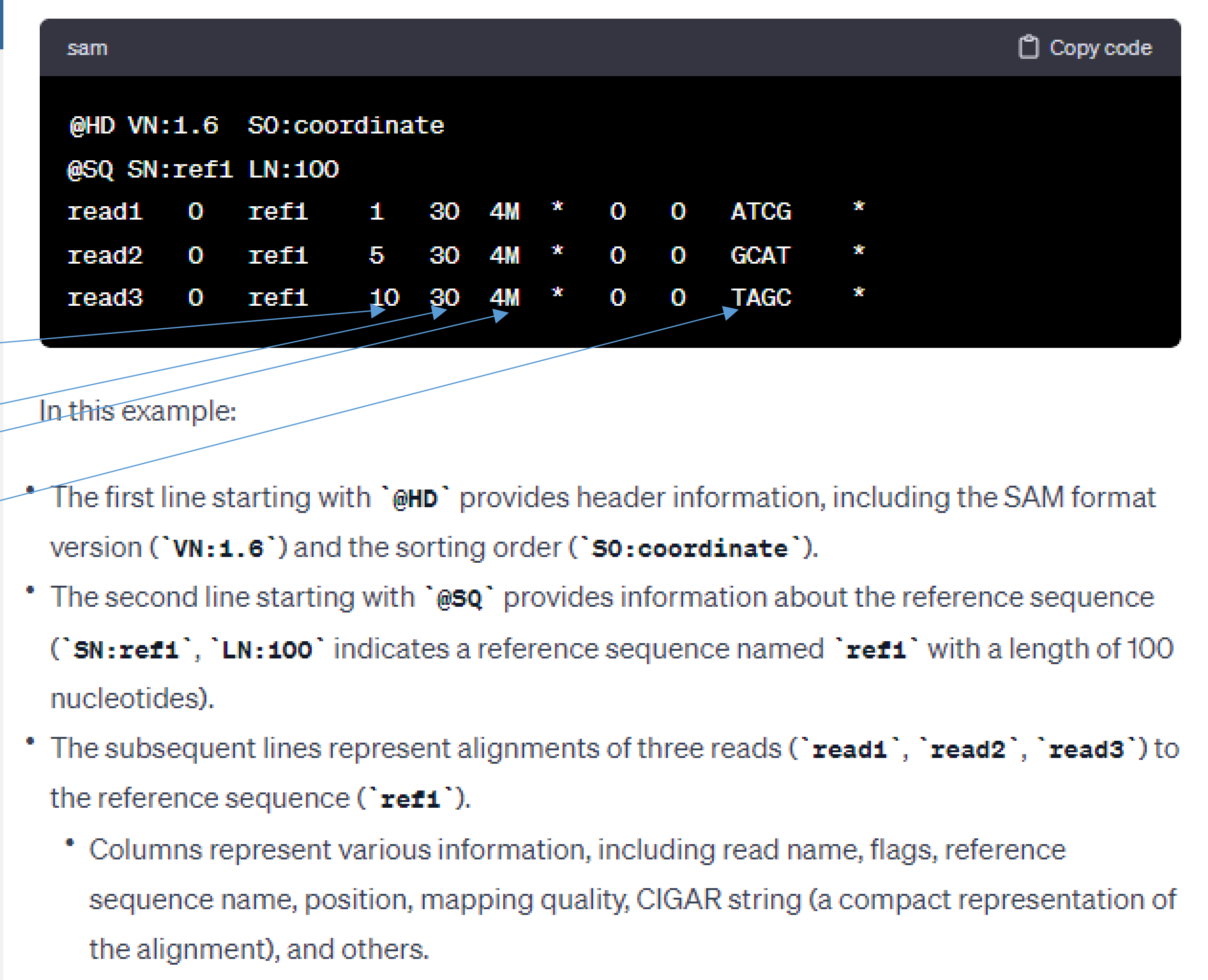


```
@SEQ5
CGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCGATCG
+
BBBBCDDDDDDDDEEDEEDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD
```

In this corrected example:

- The first line starts with "@" and is the sequence identifier (`SEQ5`).
- The second line contains the actual nucleotide sequence (70 nucleotides).
- The third line starts with "+" and is an optional comment or placeholder.
- The fourth line contains more varied quality scores for each corresponding base in the nucleotide sequence, encoded as ASCII characters (`BBBBC...`). The quality scores reflect varying degrees of confidence in base calling, where higher ASCII values correspond to higher confidence.

Example: SAM (simplified!)



```sam
@HD VN:1.6  SO:coordinate
@SQ SN:ref1 LN:100
read1    0    ref1    1    30   4M   *    0    0    ATCG     *
read2    0    ref1    5    30   4M   *    0    0    GCAT     *
read3    0    ref1    10   30   4M   *    0    0    TAGC     *
```

In this example:

- The first line starting with `@HD` provides header information, including the SAM format version (`VN:1.6`) and the sorting order (`SO:coordinate`).
- The second line starting with `@SQ` provides information about the reference sequence (`SN:ref1`, `LN:100` indicates a reference sequence named `ref1` with a length of 100 nucleotides).
- The subsequent lines represent alignments of three reads (`read1`, `read2`, `read3`) to the reference sequence (`ref1`).
  - Columns represent various information, including read name, flags, reference sequence name, position, mapping quality, CIGAR string (a compact representation of the alignment), and others.

Position in ref

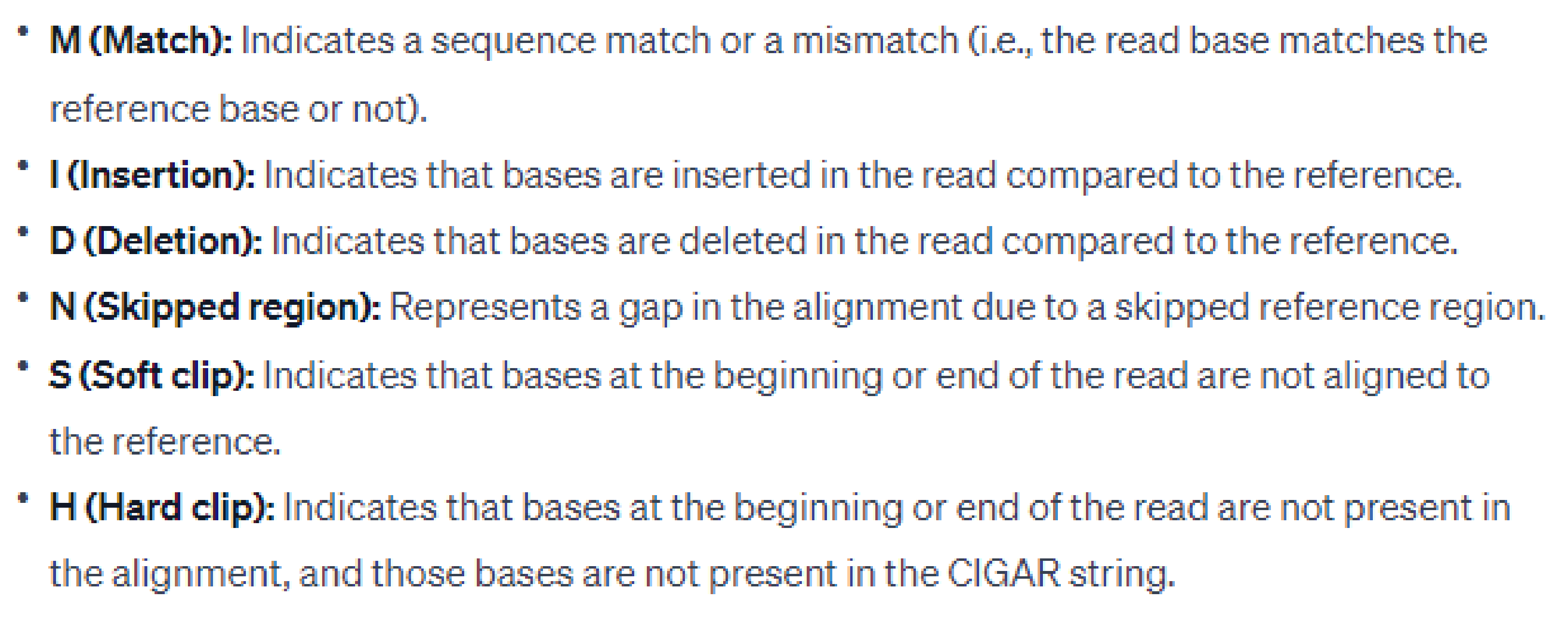
Mapping quality

CIGAR

The read

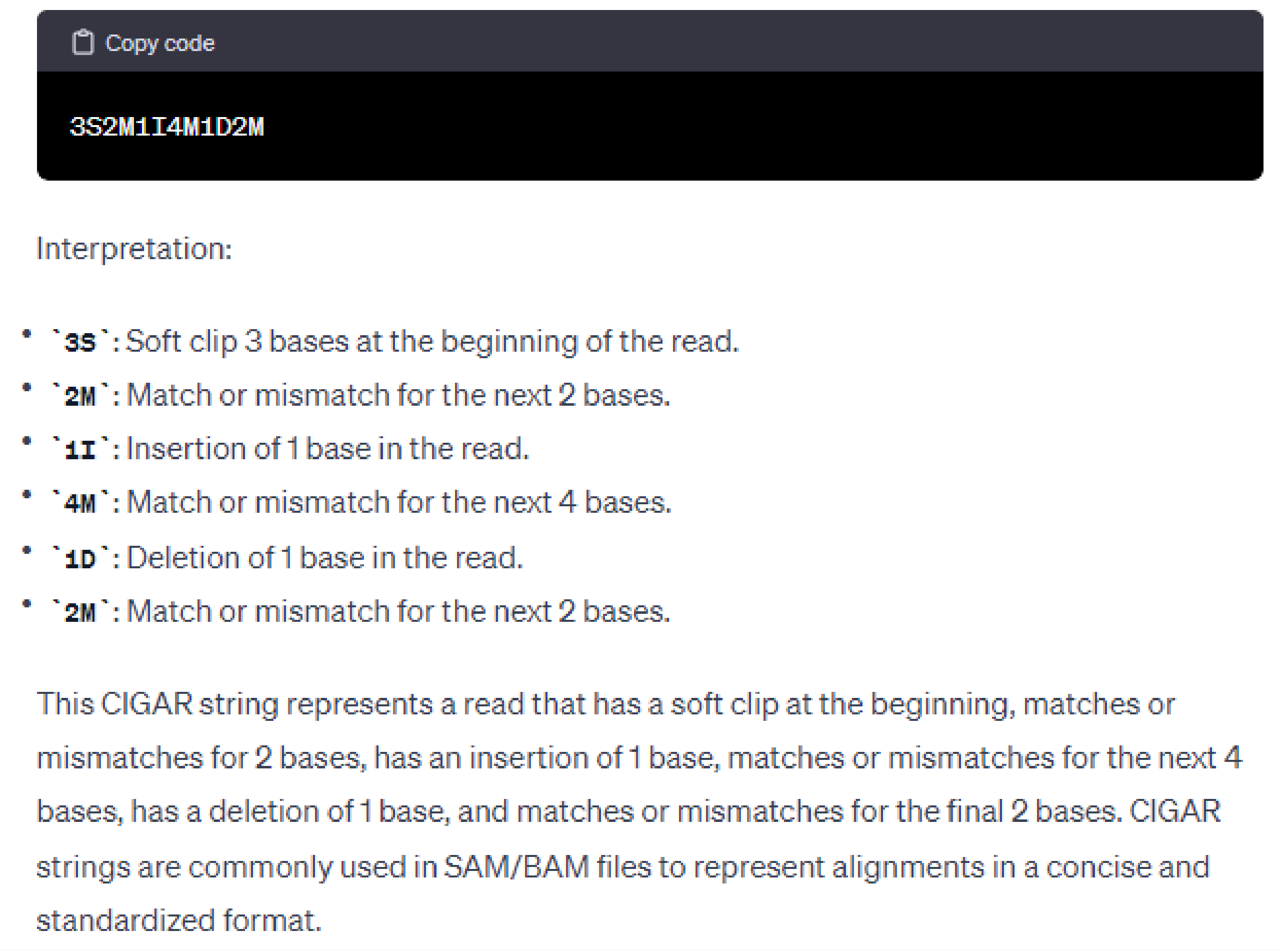
CIGAR string

Provide info about how a sequencing read aligns to the ref



- **M (Match):** Indicates a sequence match or a mismatch (i.e., the read base matches the reference base or not).
- **I (Insertion):** Indicates that bases are inserted in the read compared to the reference.
- **D (Deletion):** Indicates that bases are deleted in the read compared to the reference.
- **N (Skipped region):** Represents a gap in the alignment due to a skipped reference region.
- **S (Soft clip):** Indicates that bases at the beginning or end of the read are not aligned to the reference.
- **H (Hard clip):** Indicates that bases at the beginning or end of the read are not present in the alignment, and those bases are not present in the CIGAR string.

Example: CIGAR string



```
3S2M1I4M1D2M
```

Interpretation:

- `3S`: Soft clip 3 bases at the beginning of the read.
- `2M`: Match or mismatch for the next 2 bases.
- `1I`: Insertion of 1 base in the read.
- `4M`: Match or mismatch for the next 4 bases.
- `1D`: Deletion of 1 base in the read.
- `2M`: Match or mismatch for the next 2 bases.

This CIGAR string represents a read that has a soft clip at the beginning, matches or mismatches for 2 bases, has an insertion of 1 base, matches or mismatches for the next 4 bases, has a deletion of 1 base, and matches or mismatches for the final 2 bases. CIGAR strings are commonly used in SAM/BAM files to represent alignments in a concise and standardized format.

Good to read

[Bowtie] B. Langmead et al, “Ultrafast and memory-efficient alignment of short DNA sequences to the human genome”, Genome Biology 10:R25, 2009

[BWA] H. Li & R. Durbin, “Fast and accurate short read alignment with Burrows-Wheeler transform”, Bioinformatics 25(14):1754-1760, 2009

[Stampy] G. Lunter & M. Goodson, “Stampy: A statistical algorithm for sensitive and fast mapping of Illumina sequence reads”, Genome Research 21:936-939, 2011