Assignment 1: Next Generation Sequencing Technologies

Arturo Torres Ortiz

2026-01-13

Due date: February 12th 2026

DNA sequencing can be broadly defined as the determination of the identity and order of nucleic acid residues in biological samples. Bioinformatic analysis and sequencing results are greatly affected by the choice of sequencing technology. Sequencing technologies can be broadly categorized in three groups:

1. **First-Generation Sequencing**
    - First-generation sequencing methods, mostly notably Sanger sequencing, relied on using radio- or fluorescently-labelled dNTPs or oligonucleotides before electrophoretic analysis.
2. **Second-Generation Sequencing**
    - Second-generation sequencing is characterized by its high throughput due to the parallelisation of a large number of reactions, sequencing thousands to millions of DNA fragments. Second-Generation sequencing includes the current sequencing leader, Illumina.
3. **Third-Generation Sequencing**
    - Third-generation sequencing technologies include those capable of sequencing single molecules, so they don't require DNA amplification. The two most common third-generation sequencing technologies include PacBio (Pacific Biosciences) and ONT (Oxford Nanopore).

Generation	Sequencing Technology	Year	Company	Avg Read Length	Cost per Gigabase
First Generation	Sanger Sequencing	1977	Frederick Sanger	~800 bp	Very High (>$1000)
Second Generation	454 Sequencing	2005	Roche	~400 bp	High ($100 - $500)
	Illumina (Solexa)	2006	Illumina	~150-300 bp	Low ($1 - $10)

Generation	Sequencing Technology	Year	Company	Avg Read Length	Cost per Gigabase
	Ion Torrent	2010	Thermo Fisher	~200 bp	Medium ($10 - $50)
Third Generation	PacBio SMRT	2009	Pacific Biosciences	10,000+ bp	High ($50 - $200)
	Oxford Nanopore	2014	ONT	10,000+ bp	Medium-High ($10-$100)

We will use a human genome standard (HG002) that has been extremely well-characterized as our gold standard. The HG002 genome assembly is part of an effort hosted by NIST that includes The Telomere-to-Telomere Consortium, the Human Pangenome Reference Consortium and the Genome in a Bottle Consortium, to sequence, assemble and polish the HG002 (also known as GM24385 and huAA53E0) cell line.

Some reference papers regarding this dataset:

- Zook, J., Catoe, D., McDaniel, J. et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. Sci Data 3, 160025 (2016). https://doi.org/10.1038/sdata.2016.25
- Zook, J.M., McDaniel, J., Olson, N.D. et al. An open resource for accurately benchmarking small variant and reference calls. Nat Biotechnol 37, 561–566 (2019). https://doi.org/10.1038/s41587-019-0074-6

We will compare the characteristics and SNP calls for three sequencing technologies: Illumina, PacBio HiFi, and Nanopore.

Learning objectives

At the end of this week's assignment you will be able to:

1. Understand the principles behind the different sequencing technologies
2. Library preparation methods and their effect on the data
3. Understand the notion of a resequencing experiment and a reference genome
4. Perform quality check and control (QC) on Illumina short-read data
5. Perform quality check and control on long-read sequencing data
6. Align short-reads to a reference genome
7. Align long-reads to a reference genome
8. Understand the SAM and BAM format
9. Call SNPs against a reference genome in short-read data
10. Call SNPs against a reference genome in long-read sequencing data
11. Understand the VCF format
12. Analyze differences between sequencing technologies
13. Use common benchmarking metrics

Input and outputs

The input for this assignment are the fastq files generated by Illumina, PacBio HiFi and ONT of the standard human genome HG002. Each student will be given a chromosome to work with.

Illumina files: `/project2/msalomon_1816/trgn_515/1_seq_techs/illumina/fastq`

PacBio files: `/project2/msalomon_1816/trgn_515/1_seq_techs/pacbio/fastq`

Nanopore files: `/project2/msalomon_1816/trgn_515/1_seq_techs/ont/fastq`

Benchmarking file: `/project2/msalomon_1816/trgn_515/1_seq_techs/benchmark/HG002_GRCh38_1_22_v4.2`

Human Reference Genome GRCh38: `/project2/biodb/genomes/Homo_sapiens/NCBI/GRCh38/Sequence/BWAI`

The final output will be a list of SNPs and statistics for each sequencing technology.

Required software

In order to complete the assignment, the following tools need to be installed:

```
mamba install bioconda::filtlong
mamba install bioconda::minimap2
mamba install bioconda::seqtk
mamba install bioconda::seqkit

module load bwa
module load htslib
module load bcftools
module load samtools
module load fastqc
module load gcc/13.3.0
module load bedtools2/2.31.1
module load gatk/4.5.0.0
```

You can also use a conda environment I provide here:

```
conda activate /scratch1/atortiz/1_seq_techs/conda_env/1_seq_techs
```

Manipulating Fastq files

Fastq files are simply text files with a specific and constant format, so you can always parse them using `bash` or any scripting language. For instance, since you know the read sequence is on the 2nd line and the quality sequence is on the 4th line, to get read length for every read you can simply do:

```
awk 'NR==4 {print length}' <fastq_file.fq>
```

However, for reproducibility and consistency (and to save us time), there are already well-established software to parse and manipulate fastq files efficiently. Some widely used tools are:

1. `seqtk` - https://github.com/lh3/seqtk
2. `seqkit` - https://bioinf.shenwei.me/seqkit/
3. `bioawk` - https://github.com/lh3/bioawk

Task 1: Compare raw read statistics between sequencing technologies

For this first task, let's extract some basic metrics from the raw fastq files and compare between sequencing technologies. To be efficient, let's take a random sample of 200,000 reads per sequencing technology.
The metrics we will report are:

- Read length
- Average Base quality per read
- GC content per read

One way to do it using one of the previously shown software is:

```
seqtk sample -s 100 # random seed
<fastq_file> # Input fastq file
<n_reads> # Number of reads to downselect
| seqkit fx2tab # Main seqkit function to get read stats
-q # Get average read quality
-l # Get read length
-g # Get GC content
-n # Only print read name
- > <output_file.txt>
```

1. Plot a histogram of each of the three metrics. Plot all illumina, ont and pacbio together. **(10pts)**
2. Describe briefly the differences in read length, base quality and GC content per sequencing platform **(10pts)**

Oxford Nanopore output files

All sequencing technologies reviewed during this assignment output reads either in Fastq or BAM format (unaligned BAM or aligned if a reference genome is provided). However, ONT also output a special type of file. The output of Nanopore sequencing is in the Pod5 file format.

You can use `dorado` to basecall and demultiplex:

```
dorado basecaller # Basecalling algorithm
-r # Recursively scan through folders and files
--kit-name # ONT Nanopore kit (eg: SQK-NBD114-24)
<model> # Model for basecalling: fast, hac, sup
<input_directory> # Input directory with Pod5 files
| dorado demux # Demultiplexing algorithm
--threads <n> # Number of threads
--no-classify # Skip barcode classification
--emit-fastq # Output fastq
--output-dir <out_dir>
```

Quality Check

Most quality control depends on base qualities. Sequencers attempt to read the "sequence" of DNA letters. For each base they read, they assign a quality score that serves as a measure of confidence in that base, called a Phred quality score.

Phred Quality Score	Probability of incorrect basecall	Basecall accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%

Quality check for short-reads - FastQC

The most widely used tool for visually evaluating fastq data is FastQC.

```
fastqc # The main command call
--extract # Extract files from output
-o <output_dir> # Directory for output files
-d <temporary_dir> # Directory for temporary files
<input.fq> # Input fasta file
```

Quality check for long-reads

You can visualize the quality of your long reads using FastQC, but there's also specific software built for long-reads like `NanoPlot`.

```
NanoPlot
--fastq <input.fq> # Input in fastq file
-o <output_dir> # Output directory
--loglength # Set up for log read length in the plots
```

Task 2: Run FastQC on Illumina reads

Run FastqQC on the Illumina reads and answer the questions. Focus only on the following figures:

- Per base sequence quality
- Per sequence quality scores
- Per base sequence content
- Per sequence GC content
- Per base N content
- Sequence Length Distribution
- Overrepresented sequences
- Adapter Content

Answer the following questions for each plot:

1. What is the plot representing? **(10pts)**
2. What quality issue (if any) is the plot showing? **(10pts)**
3. How would you solve the issue (if any) represented in the plot? **(10pts)**

Answer the questions for the forward reads only.

Quality Control (QC)

Now that we have visualized the raw fastq files, we may want to filter some reads.

QC for short-reads - Trimmomatic

We will use `Trimmomatic` for this.

```
trimmomatic PE # Main trimmomatic function for paired-ends reads
<input_1.fastq.gz> # Forward reads
<input_2.fastq.gz> # Reverse reads
<output_1.fastq.gz> # Output forward reads
<output_unclassified_1.fastq.gz> # Output unclassified reads from forward
 ↪  file
<output_2.fastq.gz> # Output reverse reads
<output_unclassified_2.fastq.gz> # Output unclassified reads from reverse
 ↪  file
SLIDINGWINDOW:<window_size>:<quality>
LEADING:<quality>
TRAILING:<quality>
AVGQUAL:<quality>
MINLEN:<length>
ILLUMINACLIP:</path/to/adapter_file.fa>:<seed_mismatches>
 ↪  :palindrome_clip_threshold>:<simple_clip_threshold>
```

Task 3: Run FastQC after running Trimmomatic

Compare the FastqQC plots before and after.
Answer the following questions for each plot:

1. What has changed after running trimmomatic? **(10pts)**
2. Why did the change happen? **(10pts)**

QC of long reads

The QC for long reads is very similar to that of short-reads. `filtlong` is an interesting tool where you can keep the best x% reads, rather than hard filtering.

```
filtlong
--min_length <n> # Minimum read length
--min_mean_q <n> # Minimum mean base quality
```

```
--keep_percent <n> # Keep the best n% of reads
<input_reads> |
gzip > <output_reads>.gz
```

Task 4: Long read QC

Filter raw reads for the nanopore and pacbio datasets. Use `filtlong`. Keep the best 95% of reads.
Choose minimum read length and base quality based on the data from the first time you ran NanoPlot.

1. What changes in the statistics and plots can you observe after running filtlong? **(10pts)**

Read mapping to a reference genome

Short-read mapping

BWA

To run BWA:

```
bwa mem # Main call for the mem algorithm within BWA
<reference_genome> # Reference genome
<fastq1> # Forward reads
<fastq2> # Reverse reads
```

To add read groups on `bwa mem`, use the `-R` flag.

```
bwa mem -R "@RG\tID:$rg_id\tPU:$pu\tPL:$pl\tSM:$sample"
```

We can combine everything into one command using bash pipes:

```
bwa mem <reference_genome> <fastq1_path> <fastq2_path> | samtools view -bS -
 ↪  | samtools sort -@ <n> -T <path> -O bam -o <out_bam_path>

samtools index <out_bam_path>
```

Task 5: Complete the for loop to do the alignment

```
for _ in _; do
    # code
done
```

1) Attach the completed loop into your assignment. **(5pts)**

Long-read mapping

The most widely used long-read mapper is `minimap2`.

```
minimap2
<reference_genome>
<long_reads.fastq.gz>
```

Task 6: Map reads to the reference genome

Use `bwa mem` for short reads and `minimap2` for long-reads. For `minimap2`, choose the appropriate platform in the presets.
In both cases, create a pipe with the output of the mapper and samtools for coordinate sorting.
For the final output, any read that maps to chromosomes other than the chromosome you are working with should be considered ambiguous. Use the right command to keep only reads that mapped to your chromosome of interest.

1. Create a slurm array job to run the previous code. Attach your code to the assignment **(10pts)**

Variant calling

Variant calling in short-reads - BCFtools

For Illumina reads, we will use BCFtools for SNP calling.

```
bcftools mpileup # Pileup function
-f <reference_genome>
-a AD,INFO/AD,ADF,INFO/ADF,ADR,INFO/ADR,DP,SP # Add tags useful for filtering
-q <n> # Skip reads with read quality below n
```

```
-Q <n> # Skip bases with base quality below n
-Ou # Uncompressed output
<input_bam>
<output_vcf>
```

```
bcftools call # Main bcftools call function
-m # Multiallelic calling method
-Oz # Output in compressed format
<output.vcf.gz>
```

Task 7: Complete the Variant Calling command

Pipe the commands `mpileup`, `call`, `norm` and `filter` into one command to stream from the input bam file to the final VCF output.

```
bcftools mpileup | ...
```

Use the commands following these instructions. You will have to look at the help for each command:

- For `bcftools mpileup`:
  - Use all reads regardless of mapping quality
  - Keep only bases with base quality higher or equal than 20
- For `bcftools call`:
  - Print variant sites only
  - Do not report indels
  - Keep all alternate alleles

Apply the following filter tags in `bcftools filter`:

- 'MinMQ' for MQ lower than 20
- 'QUAL' for QUAL lower than 20
- 'minAD' for allele depth lower than 20
- 'minADF' for allele forward depth lower than 5
- 'minADR' for allele reverse depth lower than 5
- 'MinDP' for total depth lower than 50

Long-read variant calling

Task 8: Call SNPs against the human reference genome

To be consistent between sequencing platforms, use a variant caller that's platform agnostic (i.e: it's not optimized for any specific sequencing technology or read type). This includes: BCFtools, GATK, FreeBayes or DeepVariant.
You can choose any variant caller you prefer, but run the same caller for all three sequencing technologies (Illumina, PacBio HiFi, ONT). Use any filters you see necessary.
For this task, report your variant calling command and your choice of options and filters.

1. Create a slurm array job to run the SNP calling code. **Call only SNPs.** Attach your code to the assignment **(5pts)**

Benchmarking

For our benchmarking, we are mostly interested in TP, FP and FN.

$$TPR = \frac{TP}{\text{Actual Positive}} = \frac{TP}{TP + FN}$$
$$FNR = \frac{FN}{\text{Actual Positive}} = \frac{FN}{TP + FN}$$

Another common metric used is the F1 score.

$$F_1 = \frac{2}{\text{recall}^{-1} + \text{precision}^{-1}} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2TP}{2TP + FP + FN}$$

VCF comparison

Task 9: Benchmarking metrics from VCF file

Match the definitions in the intersections column with the possible benchmarking metrics **(5pts)**.

- The intersections are between your sample VCF (Va) and the benchmarking VCF (Vb).
- The possible metrics are False Positive, False Negative, True Positive, and True Negative

VCF intersection	Benchmark metric
Va and Vb	

not Va and not Vb
Va and not Vb
not Va and Vb

Genome-wide assessment using genomic windows

A common way to represent genomic events is using genomic windows.

1. Get your FP or FN into bed format.
2. Make genomic windows.
3. Counting the overlaps.

Task 10: Benchmarking of different sequencing technologies for SNP calling

The latest task consists of a comparison between the variant calls you obtained and the gold standard.

1) How many variants are called using each sequencing technology and how many are in the gold standard? **(5pts)**
2) How many true positive, false positive and false negative variants are present from using the different sequencing technology? **(5pts)**
3) Calculate the precision and recall for each sequencing technology **(5pts)**
4) Calculate the F1 score for each sequencing technology **(5pts)**
5) Make a grouped barplot of each metric and each sequencing technology. Group bars by sequencing technology. Make one plot for TP, FP, FN; and another plot for precision, recall and F1 score (in percentage) **(10pts)**
6) Look at the distribution of false negatives and false positives along the genome per sequencing technology using a genomic window of 10kbp. Make a density plot or a histogram of FP and FN along the genome. X-axis is genome position as determined by the beginning of the window. Y-axis is the number of FP or FN. Is there any pattern in the distribution? Are there any areas with high FP and FN? Calculate recall, precision and F1 for each genomic window. Are there any areas with low recall/precision/F1? Give approximate genomic coordinates **(20pts)**
7) Repeat the analysis using genomic windows of 10kpb and a 5000bp window overlap. You can find this in the `bedtools makewindows` command help. Repeat the plot of the previous question. How has the plot change? Is there any pattern in the distribution? Are there any areas with high FP and FN? Calculate recall, precision and F1 for each genomic window. Are there any areas with low recall/precision/F1? Give approximate genomic coordinates **(20pts)**