**Phage Genome Analysis**

1. **Download and install Ubuntu** [**https://ubuntu.com/download/desktop**](https://ubuntu.com/download/desktop)
2. **Download and install Virtual box (optional)** [**https://www.virtualbox.org/wiki/Downloads**](https://www.virtualbox.org/wiki/Downloads)
3. **Go to the terminal**
4. **Check the current directory**

**ls**

1. **Create a new directory to store all your works**

**mkdir TU\_MDS\_Bioinformatics**

1. **Go to the directory**

**cd TU\_MDS\_Bioinformatics**

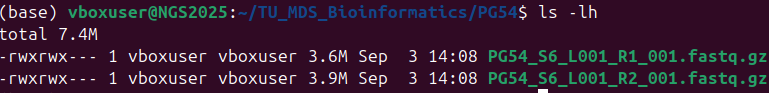
1. **Create another directory inside TU\_MDS\_Bioinformatics**

**mkdir PG54**

1. **Download or copy the raw fastq files into the folder PG54**
2. **List the contents in the folder using**

**ls -lh**

1. **The results should show two zipped files as shown in figure below:**

****

1. **Install Miniconda**

**wget** [**https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86\_64.sh -O ~/miniconda.sh**](https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh%20-O%20~/miniconda.sh)

**conda init**

1. **Create a new environement for Phage analysis**

**conda create –n phage\_env**

1. **Activate the environment**

**conda activate phage\_env**

****

1. **Install required conda channels**

**conda config --add channels defaults**

**conda config --add channels conda-forge**

**conda config --add channels bioconda**

**conda config --set channel\_priority strict**

1. **Setup and install necessary packages**

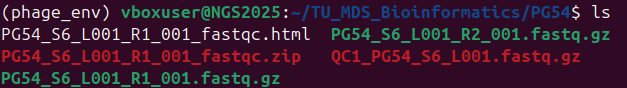
**conda install -y fastp fastqc multiqc spades quast prodigal bwa samtools seqtk**

1. **Verify installation for your packages**

**which fastp fastqc multiqc spades.py quast.py prodigal bwa samtools seqtk**

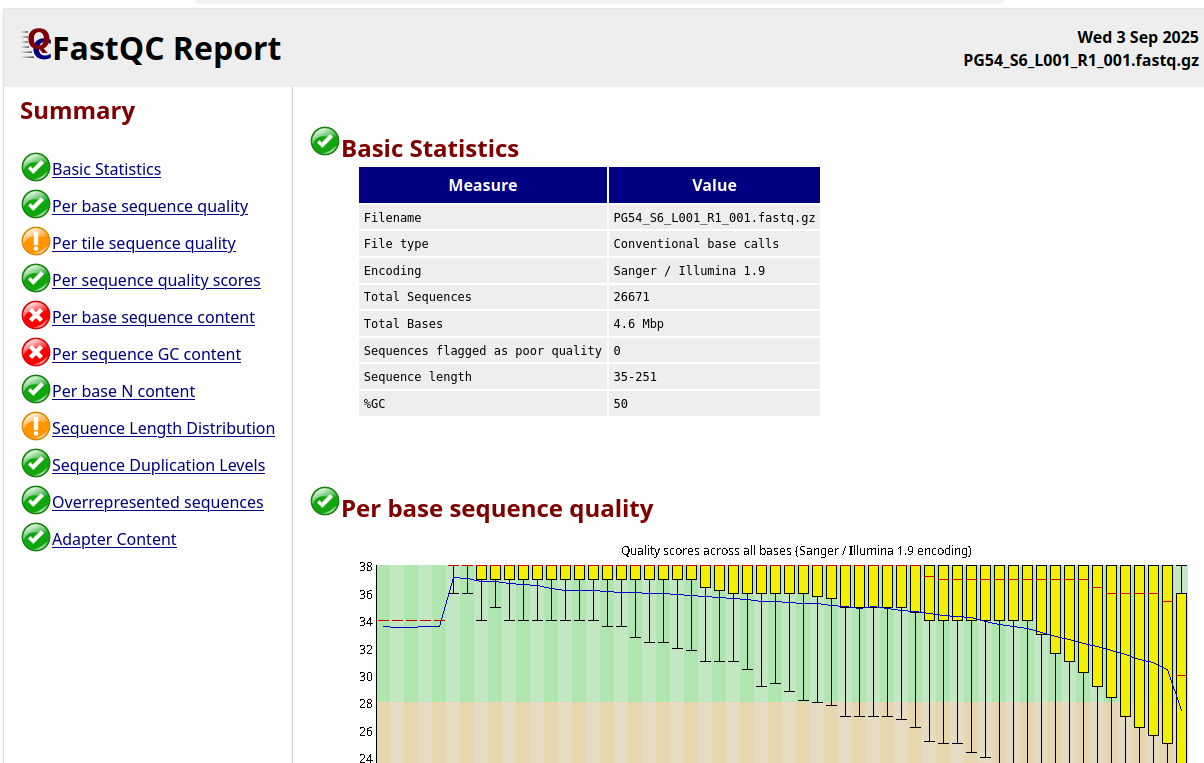
1. **Perform Quality check for your reads**

**fastqc PG54\_S6\_L001\_R1\_001.fastq.gz > QC1\_PG54\_S6\_L001\_R1\_001.fastq.gz**

****

1. **Check the html file from folder or**

**xdg-open PG54\_S6\_L001\_R1\_001\_fastqc.html**

****

1. **Do the same for another read as well**

**fastqc PG54\_S6\_L001\_R2\_001.fastq.gz > QC1\_PG54\_S6\_L001\_R2\_001.fastq.gz**

**xdg-open PG54\_S6\_L001\_R2\_001\_fastqc.html**

1. **Create new directory**

**mkdir trimmed**

**fastp \**

**-i PG54\_S6\_L001\_R1\_001.fastq.gz \**

**-I PG54\_S6\_L001\_R2\_001.fastq.gz \**

**-o trimmed/PG54\_R1\_trimmed.fastq.gz \**

**-O trimmed/PG54\_R2\_trimmed.fastq.gz \**

**-h trimmed/fastp\_report.html \**

**-j trimmed/fastp\_report.json \**

**-w 8 \**

**--detect\_adapter\_for\_pe \**

**--length\_required 50 \**

**--cut\_front --cut\_tail --cut\_window\_size 4 --cut\_mean\_quality 20**

|  |  |
| --- | --- |
| **Option** | **Meaning** |
| -i PG54\_S6\_L001\_R1\_001.fastq.gz | Input file for Read 1 |
| -I PG54\_S6\_L001\_R2\_001.fastq.gz | Input file for Read 2 (paired-end) |
| -o trimmed/PG54\_R1\_trimmed.fastq.gz | Output trimmed Read 1 |
| -O trimmed/PG54\_R2\_trimmed.fastq.gz | Output trimmed Read 2 |
| -h trimmed/fastp\_report.html | Generates an interactive HTML report of QC |
| -j trimmed/fastp\_report.json | Generates a JSON report for programmatic parsing |
| -w 8 | Number of CPU threads to use (faster processing) |
| --detect\_adapter\_for\_pe | Automatically detects adapter sequences for paired-end reads |
| --length\_required 50 | Discards reads shorter than 50 bp after trimming |
| --cut\_front | Trims low-quality bases from the start of the read |
| --cut\_tail | Trims low-quality bases from the end of the read |
| --cut\_window\_size 4 | Sliding window size for trimming |
| --cut\_mean\_quality 20 | Minimum average Phred score within sliding window; bases below are trimmed |

1. **Check the .html and .json files**

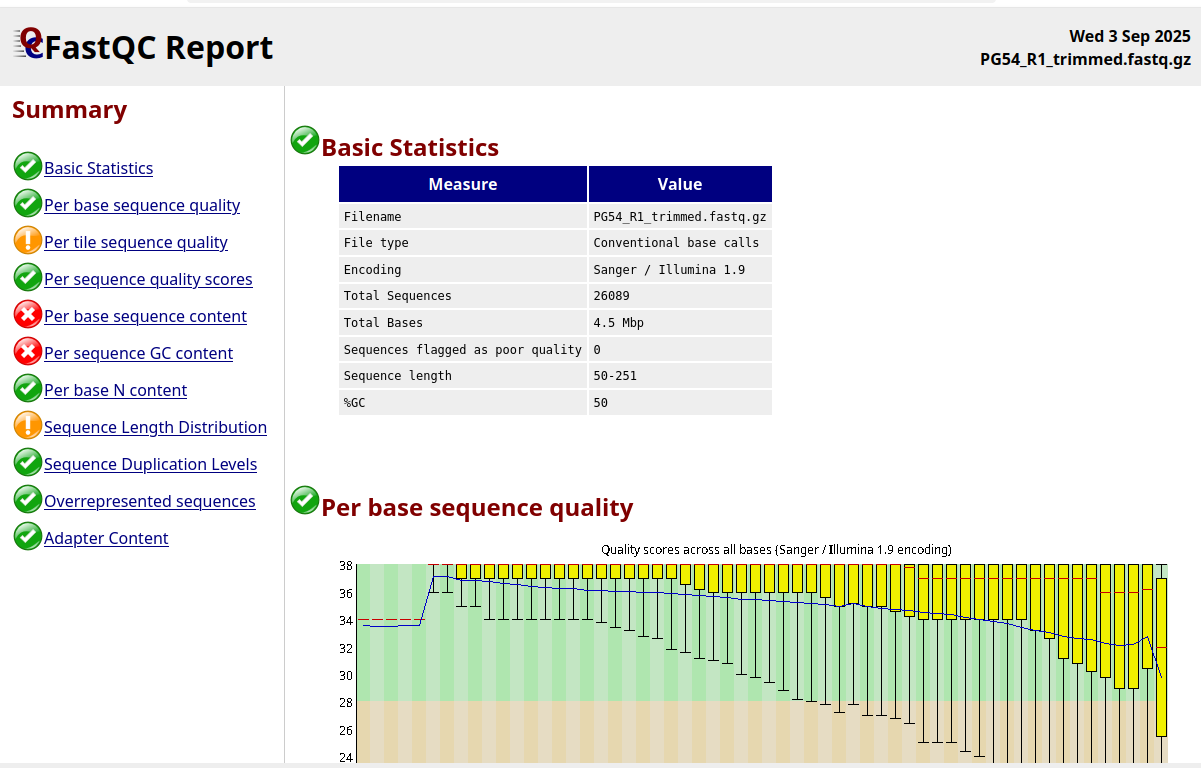
**xdg-open fastp\_report.html**

**nano fastp\_report.json**

1. **Recheck the sequences after trimming**

**mkdir -p fastqc\_trimmed**

**fastqc PG54\_R1\_trimmed.fastq.gz trimmed -o fastqc\_trimmed**

****

**fastqc PG54\_R2\_trimmed.fastq.gz trimmed -o fastqc\_trimmed**

1. **Install multiqc**

**sudo apt install multiqc**

**multiqc fastqc\_trimmed -o multiqc\_trimmed**

**xdg-open multiqc\_report.html**

1. **Assemble the trimmed reads using SPAdes**

**spades.py -1 PG54\_R1\_trimmed.fastq.gz -2 PG54\_R2\_trimmed.fastq.gz -o spades\_output --careful -t 2 -m 32**

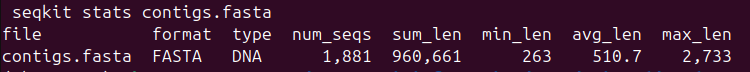
|  |  |
| --- | --- |
| **Part** | **Meaning** |
| spades.py | Runs the SPAdes assembler (Python wrapper) |
| -1 PG54\_R1\_trimmed.fastq.gz | Specifies the forward reads (R1) for paired-end assembly |
| -2 PG54\_R2\_trimmed.fastq.gz | Specifies the reverse reads (R2) for paired-end assembly |
| -o spades\_output | Output directory where SPAdes will store assembly results |
| --careful | Tries to reduce mismatches and small indels in the assembly by correcting errors; recommended for high-quality reads like phage |
| -t 8 | Use 8 CPU threads → speeds up assembly |
| -m 32 | Allocate 32 GB RAM for the assembly process |

1. **Check the sequences in your contigs**

**Install seqkit**

**sudo apt install seqkit**

**seqkit stats contigs.fasta**

****

1. **Install QUAST**

**conda install -c bioconda quast**

**quast.py spades\_output/contigs.fasta -o quast\_report -t -2**

1. **Install blast**

**conda install –c bioconda blast**

1. **Download the phage\_refseq.fasta files from NCBI database and create a phage database to blast the contigs**

**makeblastdb -in refseq\_phage.fasta -dbtype nucl -out refseq\_phage\_db**

1. **Blast the contigs with the created database**

**blastn -query contigs.fasta \**

**-db refseq\_phage\_db \**

**-out blast\_results.txt \**

**-outfmt "6 qseqid sseqid pident length mismatch gapopen qstart qend sstart send evalue bitscore stitle" \**

**-max\_target\_seqs 5**

1. **Sort the best hits in blast results**

**sort -k1,1 -k12,12nr blast\_results.txt | awk '!seen[$1]++' > blast\_best\_hits.txt**

**-k1,1 → sort by contig name**

**-k12,12nr → then by bitscore (numeric, reverse = highest first)**

**awk '!seen[$1]++' → keep only the first (best) hit for each contig**

1. **Summarize the coverage of the contigs**

**conda activate phage\_env**

**conda install –c conda-forge pandas**

1. **Save the python script for summarizing the blast with coverage**

**Run the script with python (make sure the script and blast\_results.txt are in same folder)**

**python3 summarize\_blast\_with\_coverage.py**

**Assemble with Refseq sequence**

**spades.py \**

**-1 PG54\_R1\_trimmed.fastq.gz\**

**-2 PG54\_R2\_trimmed.fastq.gz\**

**--careful \**

**--trusted-contigs salmonellaphage\_reference\_genome.fasta \**

**-o spades\_ref\_assembly/**

**BLAST AS ABOVE**

**Compare**

**Assemble with Refseq bacteria**

**spades.py \**

**-1 PG54\_R1\_trimmed.fastq.gz\**

**-2 PG54\_R2\_trimmed.fastq.gz\**

**--careful \**

**--trusted-contigs salmonellabact\_reference\_genome.fasta \**

**-o spadesbact\_ref\_assembly/**