Read Cleaning Tutorial

This document provides a step-by-step guide to carry out read cleaning on your local computer. Read cleaning is crucial because it enhances the quality of sequencing data by removing low-quality bases, adapters, and contaminants that may interfere with downstream analyses like genome assembly or gene prediction.

Technologies Used

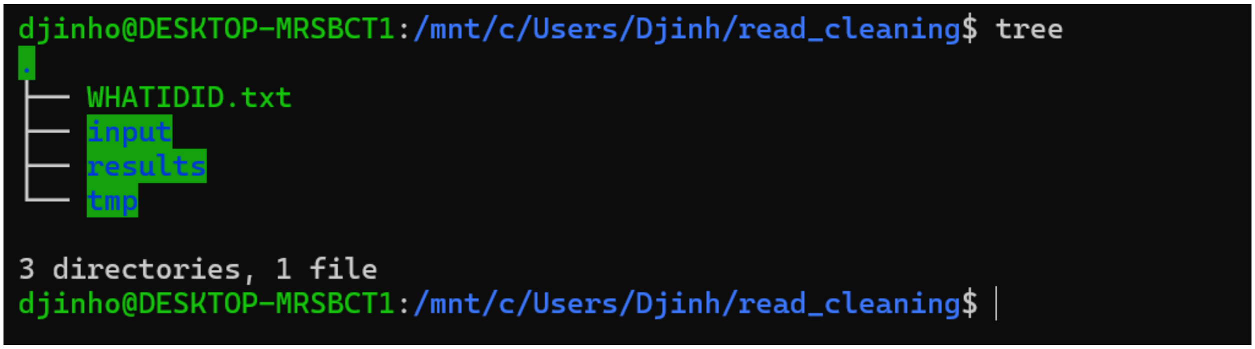
- **SRA Toolkit**: For downloading sequencing data from the NCBI Sequence Read Archive.
- **Seqtk**: A toolkit for processing sequences in FASTA/Q formats.
- **FastQC**: A tool for quality control checks on raw sequence data.
- **Cutadapt**: Trims adapters, primers, poly-A tails, and other unwanted sequences from high-throughput sequencing reads.
- **KMC**: A k-mer counting tool optimized for large sequencing datasets.
- **Firefox**: A web browser for viewing HTML reports.

Directory Setup

Create the necessary directories to structure and store the inputs and outputs:

```
mkdir -p read_cleaning/input
mkdir read_cleaning/tmp
mkdir read_cleaning/results
```

Your directory structure should look like this:



```
djinho@DESKTOP-MRSBCT1:/mnt/c/Users/Djinh/read_cleaning$ tree
.
├── WHATIDID.txt
├── input
├── results
└── tmp

3 directories, 1 file
djinho@DESKTOP-MRSBCT1:/mnt/c/Users/Djinh/read_cleaning$
```

Software Installation

Install the required software:

```
sudo apt-get install sra-toolkit
sudo apt-get install seqtk
sudo apt-get install fastqc
sudo apt-get install cutadapt
sudo apt-get install kmc
sudo apt-get install firefox
```

Data Selection and Download

We will use a small dataset that doesn't require much memory:

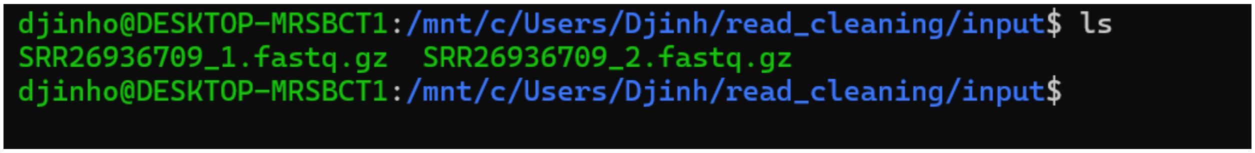
- **Sample**: **SRX22630588** - Native Miscanthus microbiome rhizosphere soil sample.
- **Description**: Part of a study on the metagenomic analysis of microbial communities in Miscanthus rhizosphere soil, using 16S V4 and fungal ITS amplicon sequencing on an Illumina HiSeq 2500 platform.
- Details on how to find the dataset will be in the accompanying pdf in the read_cleaning directory called Dataset.

Download the FASTQ files into the input directory:

```
cd read_cleaning
fastq-dump --split-files --gzip SRR26936709 -O input
```

```
fastq-dump --split-files --gzip SRR26936709 -O input
```

When you run ls command in the input directory you should get this output:



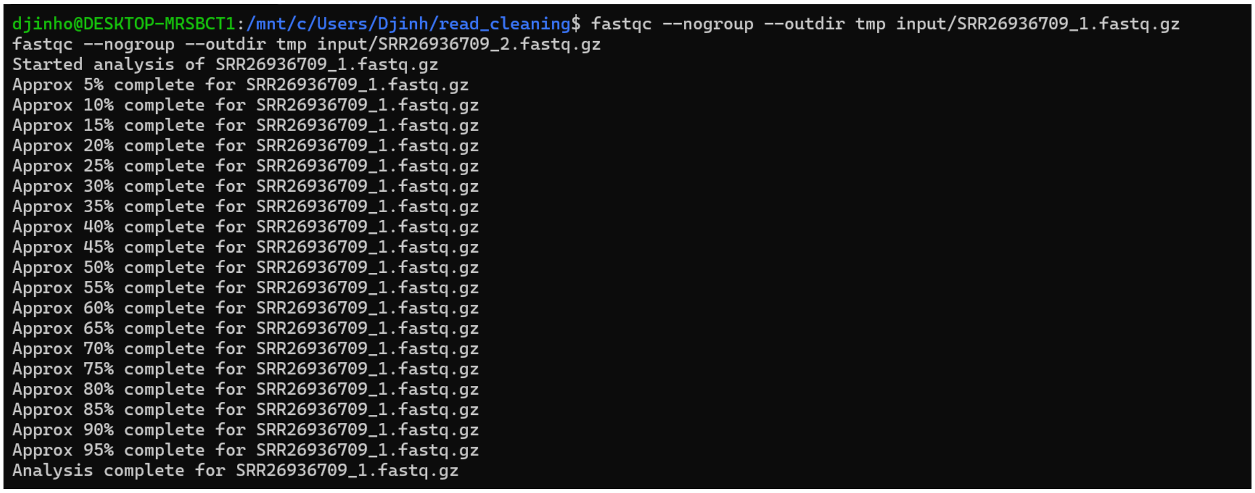
```
djinho@DESKTOP-MRSBCT1:/mnt/c/Users/Djinh/read_cleaning/input$ ls
SRR26936709_1.fastq.gz  SRR26936709_2.fastq.gz
djinho@DESKTOP-MRSBCT1:/mnt/c/Users/Djinh/read_cleaning/input$
```

Initial Quality Check

Run FastQC on the downloaded FASTQ files and save the output to the tmp directory:

```
fastqc --nogroup --outdir tmp input/SRR26936709_1.fastq.gz
fastqc --nogroup --outdir tmp input/SRR26936709_2.fastq.gz
```

After running the above commands in the read_cleaning directory you should get this output:



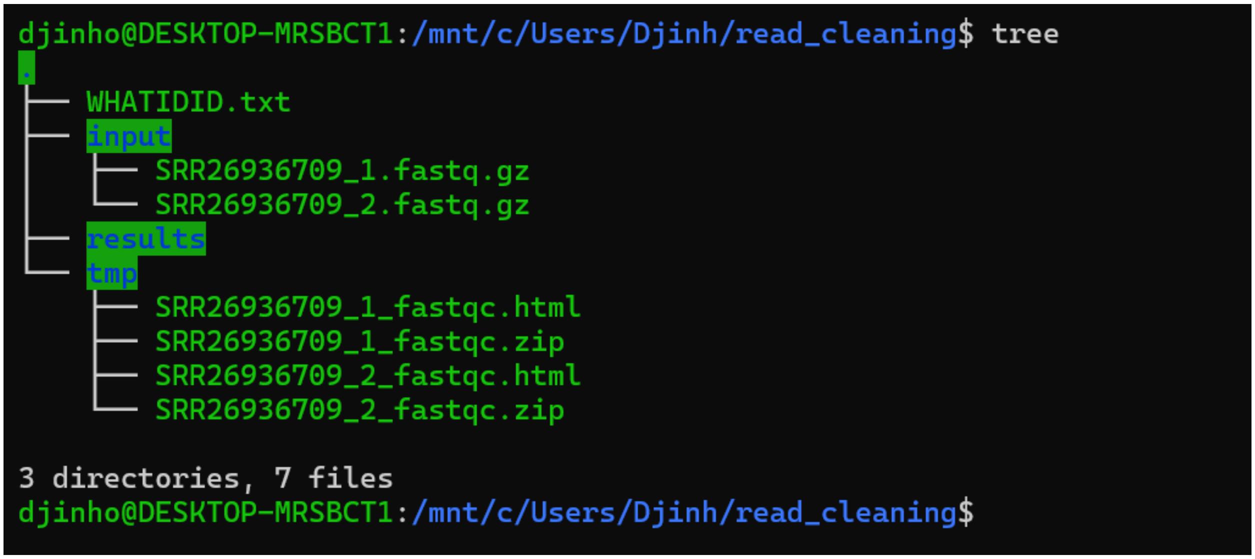
```
djinho@DESKTOP-MRSBCT1:/mnt/c/Users/Djinh/read_cleaning$ fastqc --nogroup --outdir tmp input/SRR26936709_1.fastq.gz
fastqc --nogroup --outdir tmp input/SRR26936709_2.fastq.gz
Started analysis of SRR26936709_1.fastq.gz
Approx 5% complete for SRR26936709_1.fastq.gz
Approx 10% complete for SRR26936709_1.fastq.gz
Approx 15% complete for SRR26936709_1.fastq.gz
Approx 20% complete for SRR26936709_1.fastq.gz
Approx 25% complete for SRR26936709_1.fastq.gz
Approx 30% complete for SRR26936709_1.fastq.gz
Approx 35% complete for SRR26936709_1.fastq.gz
Approx 40% complete for SRR26936709_1.fastq.gz
Approx 45% complete for SRR26936709_1.fastq.gz
Approx 50% complete for SRR26936709_1.fastq.gz
Approx 55% complete for SRR26936709_1.fastq.gz
Approx 60% complete for SRR26936709_1.fastq.gz
Approx 65% complete for SRR26936709_1.fastq.gz
Approx 70% complete for SRR26936709_1.fastq.gz
Approx 75% complete for SRR26936709_1.fastq.gz
Approx 80% complete for SRR26936709_1.fastq.gz
Approx 85% complete for SRR26936709_1.fastq.gz
Approx 90% complete for SRR26936709_1.fastq.gz
Approx 95% complete for SRR26936709_1.fastq.gz
Analysis complete for SRR26936709_1.fastq.gz
```

1. **Directory Structure Verification:**

- Verify that the directory structure is correct:

Check the directory structure, you should have .html files also and it should look like this:

Run this command to double check: tree read_cleaning:



```
djinho@DESKTOP-MRSBCT1:/mnt/c/Users/Djinh/read_cleaning$ tree
.
├── WHATIDID.txt
├── input
│   ├── SRR26936709_1.fastq.gz
│   └── SRR26936709_2.fastq.gz
├── results
└── tmp
    ├── SRR26936709_1_fastqc.html
    ├── SRR26936709_1_fastqc.zip
    ├── SRR26936709_2_fastqc.html
    └── SRR26936709_2_fastqc.zip

3 directories, 7 files
djinho@DESKTOP-MRSBCT1:/mnt/c/Users/Djinh/read_cleaning$
```

Viewing FastQC Reports

Open the FastQC HTML files in Firefox:

```
firefox tmp/SRR26936709_1_fastqc.html
firefox tmp/SRR26936709_2_fastqc.html
```

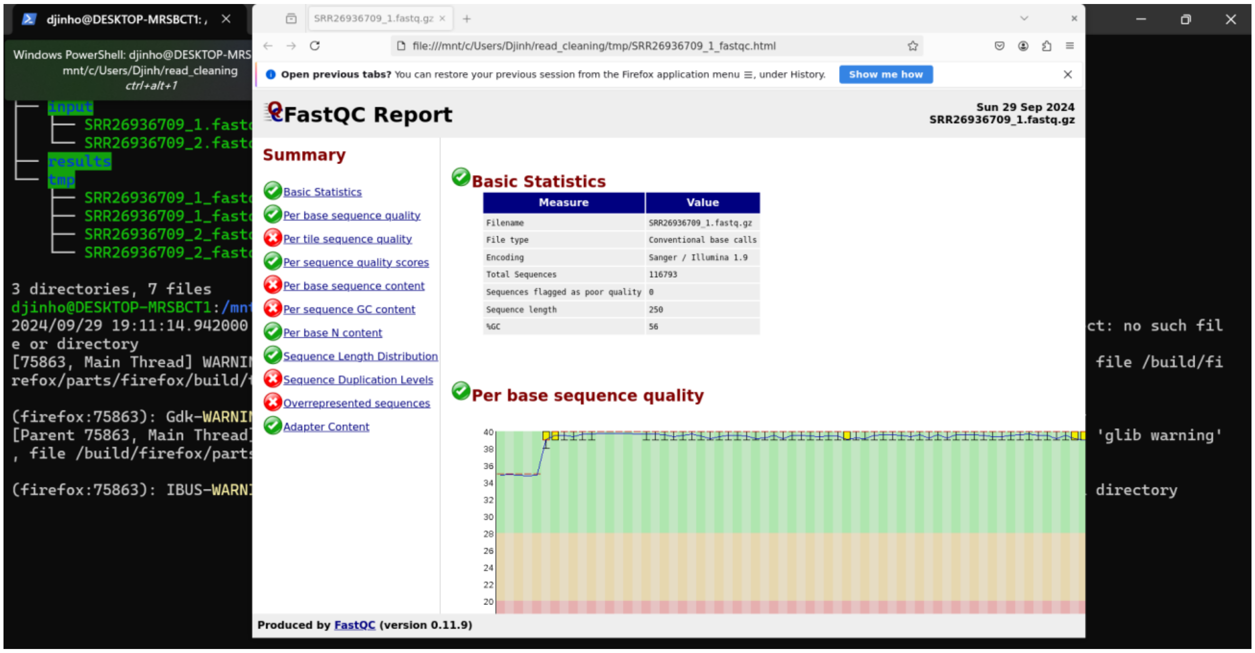
Note: You may encounter warnings, but the files will display correctly.

Review the reports to assess the quality of your sequencing data. Pay attention to:

- **Per base sequence quality**
- **Per sequence quality scores**
- **Per base sequence content**
- **Per sequence GC content**

For detailed guidance on interpreting FastQC reports, refer to the FastQC documentation.

This should pop on up on your screen:



```
djinho@DESKTOP-MRSBCT1: ,
Windows PowerShell: djinho@DESKTOP-MRS
mnt/c/Users/Djinh/read_cleaning
ctrl+alt+1
├── input
│   ├── SRR26936709_1.fastc
│   └── SRR26936709_2.fastc
├── results
└── tmp
    ├── SRR26936709_1_fastc
    ├── SRR26936709_1_fastc
    ├── SRR26936709_2_fastc
    └── SRR26936709_2_fastc

3 directories, 7 files
djinho@DESKTOP-MRSBCT1:/mnt
2024/09/29 19:11:14.942000                                   ct: no such fil
e or directory
[75863, Main Thread] WARNIN                                   file /build/fi
refox/parts/firefox/build/t

(firefox:75863): Gdk-WARNIN
[Parent 75863, Main Thread]                                   'glib warning'
, file /build/firefox/parts

(firefox:75863): IBUS-WARNI                                   directory
```

SRR26936709_1.fastq.gz

file:///mnt/c/Users/Djinh/read_cleaning/tmp/SRR26936709_1_fastqc.html

Open previous tabs? You can restore your previous session from the Firefox application menu ≡, under History. Show me how

FastQC Report

Sun 29 Sep 2024
SRR26936709_1.fastq.gz

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✗ Per tile sequence quality
- ✓ Per sequence quality scores
- ✗ Per base sequence content
- ✗ Per sequence GC content
- ✓ Per base N content
- ✓ Sequence Length Distribution
- ✗ Sequence Duplication Levels
- ✗ Overrepresented sequences
- ✓ Adapter Content

✓ **Basic Statistics**

Measure	Value
Filename	SRR26936709_1.fastq.gz
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	116793
Sequences flagged as poor quality	0
Sequence length	250
%GC	56

✓ **Per base sequence quality**

Produced by FastQC (version 0.11.9)

Read Trimming with Cutadapt

Based on the FastQC reports, trim low-quality bases from the reads using Cutadapt.

Run Cutadapt to trim the first 4 bases from the beginning of each read and perform quality trimming with a cutoff of 8:

```
cutadapt --cut 4 --quality-cutoff 8 input/SRR26936709_1.fastq.gz -o
tmp/SRR26936709_1.trimmed.fq
cutadapt --cut 4 --quality-cutoff 8 input/SRR26936709_2.fastq.gz -o
tmp/SRR26936709_2.trimmed.fq
```

K-mer Filtering with KMC

Create a list of trimmed files for KMC:

```
ls tmp/SRR26936709_1.trimmed.fq tmp/SRR26936709_2.trimmed.fq >
tmp/file_list_for_kmc
```

Run KMC to count k-mers (k=21) with limited memory (2 GB):

```
kmc -m2 -k21 @tmp/file_list_for_kmc tmp/21-mers tmp
```

Filter out rare k-mers (occurring less than twice):

```
kmc_tools -t1 filter -hm tmp/21-mers tmp/SRR26936709_1.trimmed.fq -ci2
tmp/SRR26936709_1.trimmed.norare.fq
kmc_tools -t1 filter -hm tmp/21-mers tmp/SRR26936709_2.trimmed.fq -ci2
tmp/SRR26936709_2.trimmed.norare.fq
```

Final Trimming and Cleaning

Check for unpaired reads:

```
cutadapt -o /dev/null -p /dev/null tmp/SRR26936709_1.trimmed.norare.fq
tmp/SRR26936709_2.trimmed.norare.fq
```

Trim 'N's from the ends and remove reads shorter than 21 bases:

```
cutadapt --trim-n --minimum-length 21 -o tmp/SRR26936709_1.clean.fq -p
tmp/SRR26936709_2.clean.fq tmp/SRR26936709_1.trimmed.norare.fq
tmp/SRR26936709_2.trimmed.norare.fq
```

Saving the Cleaned Reads

Copy the cleaned FASTQ files to the results directory:

```
cp tmp/SRR26936709_1.clean.fq tmp/SRR26936709_2.clean.fq results/
```

Conclusion

In this practical, we successfully downloaded paired-end FASTQ files, assessed their quality using FastQC, trimmed low-quality bases with Cutadapt, filtered out rare k-mers using KMC, and obtained clean reads ready for downstream analysis. This process is foundational for the rest of the practicals, so be sure to save this cleaned data on your local machine for future use.

Synopsis

This practical provides a comprehensive guide to read cleaning, a crucial first step in bioinformatics workflows that ensures high-quality sequencing data for accurate downstream analyses. By following the outlined steps—setting up directories, installing software, downloading data, assessing quality, trimming reads, and filtering k-mers—you have prepared a clean dataset that will serve as the foundation for subsequent practicals in this series. It's essential to save this cleaned data on your local machine, as it will be used in upcoming exercises involving genome assembly and gene prediction.