#!/bin/bash

# Make sure you are in the correct Conda environment

# Your prompt should be (polony\_env)

# --- Step 0: Define the 50 samples to be analyzed ---

SAMPLES=(

"SRR27013147\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013245\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013246\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013247\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013248\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013249\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013250\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013251\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013252\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013253\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013254\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013255\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013256\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013257\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013258\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013259\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013260\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013261\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013262\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013263\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013264\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013265\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013266\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013267\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013268\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013269\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013270\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013271\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013272\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013273\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013274\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013275\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013276\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013277\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013278\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013279\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013280\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013281\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013282\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013283\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013284\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013285\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013286\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013287\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013288\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013289\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013290\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013291\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013292\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

"SRR27013293\_Genome\_Sequencing\_of\_Listeria\_monocytogenes\_SA\_outbreak\_2017"

)

# --- Step 1: Data Download & Quality Control ---

echo "--- Starting Data Download & QC ---"

# Download the dataset. This script is idempotent.

wget https://raw.githubusercontent.com/HackBio-Internship/2025\_project\_collection/refs/heads/main/SA\_Polony\_100\_download.sh

chmod +x SA\_Polony\_100\_download.sh

./SA\_Polony\_100\_download.sh

# Run FastQC on all samples

mkdir -p qc\_reports

fastqc \*.fastq.gz -o qc\_reports

# Trim low-quality reads and adapters with FastP

mkdir -p trimmed\_reads

for SAMPLE in "${SAMPLES[@]}"; do

R1="${SAMPLE}\_1.fastq.gz"

R2="${SAMPLE}\_2.fastq.gz"

echo "Trimming sample: $SAMPLE"

fastp -i $R1 -I $R2 -o trimmed\_reads/${SAMPLE}\_trimmed\_1.fastq.gz -O trimmed\_reads/${SAMPLE}\_trimmed\_2.fastq.gz

done

# --- Step 2: De-novo Genome Assembly & Quality Assessment ---

echo "--- Starting Genome Assembly with SPAdes & Quast ---"

mkdir -p assembled\_genomes quast\_reports

for SAMPLE in "${SAMPLES[@]}"; do

trimmed\_R1="trimmed\_reads/${SAMPLE}\_trimmed\_1.fastq.gz"

trimmed\_R2="trimmed\_reads/${SAMPLE}\_trimmed\_2.fastq.gz"

echo "Assembling genome for sample: $SAMPLE"

# Corrected SPAdes command with --phred-offset 33

spades.py -o assembled\_genomes/$SAMPLE --careful --phred-offset 33 -1 $trimmed\_R1 -2 $trimmed\_R2

# Check if assembly was successful before running Quast

assembled\_genome\_path=assembled\_genomes/$SAMPLE/scaffolds.fasta

if [ -f "$assembled\_genome\_path" ]; then

echo "Assembly successful for $SAMPLE. Running Quast."

quast.py -o quast\_reports/$SAMPLE $assembled\_genome\_path

else

echo "Assembly failed for $SAMPLE. Skipping Quast."

fi

done

# --- Step 3: AMR and Toxin Gene Detection ---

echo "--- Starting AMR and Toxin Gene Detection with ABRicate ---"

mkdir -p abricate\_results

for SAMPLE in "${SAMPLES[@]}"; do

assembled\_genome\_path=assembled\_genomes/$SAMPLE/scaffolds.fasta

# Check if assembly was successful before running ABRicate

if [ -f "$assembled\_genome\_path" ]; then

echo "Running ABRicate on sample: $SAMPLE"

abricate --db card $assembled\_genome\_path > abricate\_results/${SAMPLE}\_amr.tsv

abricate --db vfdb $assembled\_genome\_path > abricate\_results/${SAMPLE}\_toxin.tsv

else

echo "Assembly failed for $SAMPLE. Skipping ABRicate."

fi

done

# --- Step 4: Final Results Aggregation ---

echo "--- Aggregating results for final report ---"

# Combine all AMR results into one file

cat abricate\_results/\*\_amr.tsv > all\_amr\_results.tsv

# Combine all toxin results into one file

cat abricate\_results/\*\_toxin.tsv > all\_toxin\_results.tsv

echo "--- All analysis steps are complete! ---"