QAA_Replication

Setup

Clone the repo, and run the following commands to install everything.
Ensure versions match.
python - 3.12.5
FastQC - v0.12.1
cutadapt - 4.9
trimmomatic - 0.39
matplotlib - 3.9.2
numpy - 1.26.4
star - 2.7.11b
htseq - 2.0.5

```
cd QAA
conda create --name QAA
conda activate QAA
conda install bioconda::fastqc
fastqc -v
conda install bioconda::cutadapt
cutadapt --version
conda install bioconda::trimmomatic
trimmomatic -version
conda install matplotlib
conda list -n QAA matplotlib
# matplotlib installed numpy for me, so this wasn't necessary. Still, check
version
conda install numpy
conda list -n QAA numpy
conda install star -c bioconda
conda list -n QAA star
conda install bioconda::htseq
conda list -n QAA htseq
```

Note: a list of versions used in this project can be found in QAA_redux_versions.md. QAA_versions.md is somewhat incorrect, but QAA_redux_versions.md is what you should get from following above directions.

Getting per-base quality distributions

Use the provided bash scripts to run plotting python script and fastqc.

```
sbatch run_demux_plotting.sh
sbatch run_fastqc.sh
```

Cutting adapters and quality trimming

Run cut adapt

```
# Cutting library 15
cutadapt -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA -A
AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT -o 15_R1_adapters_removed.fastq -p
15_R2_adapters_removed.fastq
/projects/bgmp/shared/2017_sequencing/demultiplexed/15_3C_mbnl_S11_L008_R1_001
.fastq.gz
/projects/bgmp/shared/2017_sequencing/demultiplexed/15_3C_mbnl_S11_L008_R2_001
.fastq.gz

# Cutting library 17
cutadapt -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCA -A
AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGT -o 17_R1_adapters_removed.fastq -p
17_R2_adapters_removed.fastq
/projects/bgmp/shared/2017_sequencing/demultiplexed/17_3E_fox_S13_L008_R1_001.
fastq.gz
/projects/bgmp/shared/2017_sequencing/demultiplexed/17_3E_fox_S13_L008_R2_001.
fastq.gz
```

Run trimmomatic

```
# trimming library 15
trimmomatic PE -phred33 15_R1_adapters_removed.fastq
15_R2_adapters_removed.fastq 15_R1_adapters_removed_paired.fq.gz
15_R1_adapters_removed_unpaired.fq.gz 15_R2_adapters_removed_paired.fq.gz
15_R2_adapters_removed_unpaired.fq.gz LEADING:3 TRAILING:3 SLIDINGWINDOW:5:15
MINLEN:35
```

```
# trimming library 17
trimmomatic PE -phred33 17_R1_adapters_removed.fastq
17_R2_adapters_removed.fastq 17_R1_adapters_removed_paired.fq.gz
17_R1_adapters_removed_unpaired.fq.gz 17_R2_adapters_removed_paired.fq.gz
17_R2_adapters_removed_unpaired.fq.gz LEADING:3 TRAILING:3 SLIDINGWINDOW:5:15
MINLEN:35
```

Generate length distribution plots

```
chmod 755 plot_length_dist.sh
./plot_length_dist.py
```

Alignment

Make folders, get mouse genome/gtf files:

```
mkdir gtf_primary_assembly
mkdir lib_17_ALIGNMENT
mkdir lib_15_ALIGNMENT
mkdir mouse_GENOME_INDEX
cd gtf_primary_assembly
wget https://ftp.ensembl.org/pub/release-
112/fasta/mus_musculus/dna/Mus_musculus.GRCm39.dna_sm.primary_assembly.fa.gz
wget https://ftp.ensembl.org/pub/release-
112/gtf/mus_musculus/Mus_musculus.GRCm39.112.gtf.gz
gunzip Mus_musculus.GRCm39.112.gtf.gz
Mus_musculus.GRCm39.dna_sm.primary_assembly.fa.gz
```

To generate genome index, run:

```
sbatch genGen.sh
```

When finished, do the alignments with:

```
sbatch alignReads.sh
```

When complete, run parse_sam.sh to determine reads mapped and unmapped:

```
chmod 755 parse_sam.sh
./parse_sam.sh
```

Finally, further parse alignments by running:

```
chmod 755 run_htseq.sh
./run_htseq.sh
```