pipeline

March 30, 2025

1 Project Pipline

1.1 Setting Up Environment

1.1.1 Create Environment

```
module load miniconda3
conda activate
conda create --name project -y python=3.12
```

python-3.12.9

conda-25.3.0

```
conda activate project
```

1.1.2 Load Tools

```
module load OpenJDK/22.0.2
module load fastqc/0.12.1
module load samtools/1.21
module load nextflow/24.10.3
conda install -c bioconda -c conda-forge cutadapt -y
conda install -c bioconda star/2.7.10a -y
conda install -c bioconda multiqc -y

STAR --version

cd /courses/BINF6310.202530/students/SEC04_Group_3
```

load_pipeline_modules.sh This is for future convenience.

```
#!/bin/bash

echo "Loading available modules for the transcriptome pipeline..."

# Load modules that exist on Explorer
module load OpenJDK/22.0.2
module load perl/5.40.0
module load fastqc/0.12.1
module load samtools/1.21
module load nextflow/24.10.3
```

```
echo "Modules loaded:"
module list
```

check_software.sh This is to doublecheck for the future.

```
#!/bin/bash

# Exit if any command fails
set -e

# Install tools via conda/mamba if not already installed
echo "Installing required tools via conda..."
# conda install -y -c bioconda cutadapt

# Load system modules
module load fastqc/0.12.1
module load star/2.7.11b
module load samtools/1.21

# Confirm installation
echo ""
echo "  Installed tool versions:"
echo "------------------------------"
fastqc --version
cutadapt --version
STAR --version
multiqc --version
featureCounts -v
echo "------------------------------"
echo "All tools installed and ready!"
```

I have not installed featureCounts yet, so you should get that one error.

1.2 Trim

1.2.1 fastq_processing.nf

This will cutadapt fastq files in parallel

```
nextflow.enable.dsl = 2

params.quality = 20

// Define the input channel
fastqFiles = Channel.fromPath('./*.fastq')


// Define the trimming process
process trimAndFilter {
```

```
    input:
    path fastq

    output:
    path "trimmed/trimmed_${fastq.baseName}.fastq"

    script:
    """
    mkdir -p trimmed
    cutadapt -q ${params.quality} \\
             -o trimmed/trimmed_${fastq.baseName}.fastq \\
             ${fastq} > trimmed/${fastq.baseName}_cutadapt.log
    """
}

// Define the workflow
workflow {
    fastqFiles | trimAndFilter
}
```

```
nextflow run fastq_processing.nf
```

You should now see a **work** directory with a separate folder for each processed fastq file. If you drill down each of them, you will find a **trimmed** directory with the trimmed fastq version, ready for QC and then aligning.

In the parent directory:

```
mkdir -p trimmed_outputs
find work -name "trimmed_*.fastq" -exec cp {} trimmed_outputs/ \;

mkdir -p trimmed_outputs/logs
find work -name "*_cutadapt.log" -exec cp {} trimmed_outputs/logs/ \;
```

1.3 QC

In fastq folder:

```
mkdir qc_reports
fastqc *.fastq -o qc_reports/
```

You should see a separate html and zip file for each fastq processed. Now, time to consolidate them.

```
multiqc qc_reports/ -o qc_reports/qc_summary/
```

scp qc_summary to local system

scp -r explorer:/courses/BINF6310.202530/students/SEC04_Group_3/files/test/trimmed_outputs/qc_reports/q
.

1.4 STAR

Before continuing, make sure you are on a compute node and in your env.

```
srun --pty --partition=courses --export=ALL --mem=16G -t 4:00:00 bash
conda activate project
```

1.4.1 Download Reference Genome and Gene Transfer Format (GTF)

```
mkdir -p genome && cd genome
wget ftp://ftp.ensembl.org/pub/release-109/fasta/homo_sapiens/dna/Homo_sapiens.GRCh38.dna.prima

wget ftp://ftp.ensembl.org/pub/release-109/gtf/homo_sapiens/Homo_sapiens.GRCh38.109.gtf.gz

gunzip *
```

1.4.2 Build STAR Index

```
STAR --runThreadN 8 \
     --runMode genomeGenerate \
     --genomeDir ./genome_index \
     --genomeFastaFiles Homo_sapiens.GRCh38.dna.primary_assembly.fa \
     --sjdbGTFfile Homo_sapiens.GRCh38.109.gtf \
     --sjdbOverhang 100
```

or use build_star_index.sh This is what I ended up doing. It took a couple hours.

```
#!/bin/bash
#SBATCH --job-name=star_index             # Name of the job
#SBATCH --partition=courses               # Who to bill for the job
#SBATCH -N 1                              # How many nodes do you need. When in doubt, use 1
#SBATCH -c 16                             # How many "threads" do you need for your job
#SBATCH --mem 32G                         # How much memory
#SBATCH -t 8:00:00                        # How long
#SBATCH --mail-type=END,FAIL
#SBATCH --mail-user=goodier.r@northeastern.edu
#SBATCH --out=/courses/BINF6310.202530/students/SEC04_Group_3/files/genome/genome_index/logs/%x
#SBATCH --error=/courses/BINF6310.202530/students/SEC04_Group_3/files/genome/genome_index/logs/

# Setup environment
conda activate project

# Create logs directory if it doesn't exist
mkdir -p /courses/BINF6310.202530/students/SEC04_Group_3/files/genome/genome_index/logs

# Run STAR index generation
STAR --runThreadN 16 \
     --runMode genomeGenerate \
     --genomeDir ./genome_index \
     --genomeFastaFiles Homo_sapiens.GRCh38.dna.primary_assembly.fa \
```

```
     --sjdbGTFfile Homo_sapiens.GRCh38.109.gtf \
     --sjdbOverhang 100

sbatch build_star_index.sh
```

To monitor progress:

```
squeue -u $USER
cat genome_index/logs/star_index_113745.log
watch -n 300 'ls -lh genome_index1 | tail'
```

1.4.3 Nextflow Script

star_align.nf

```
nextflow.enable.dsl = 2

params.reads = '/scratch/goodier.r/project_files/trimmed_t2d/*.fastq'
// params.reads = '/scratch/goodier.r/project_files/trimmed_healthy/*.fastq'

params.outdir = '/scratch/goodier.r/project_files/trimmed_t2d/aligned_t2d'
// params.outdir = '/scratch/goodier.r/project_files/trimmed_healthy/aligned_healthy'

params.genomeDir = '/scratch/goodier.r/project_files/genome/genome_index'
params.threads = 8

Channel.fromPath(params.reads)
    .set { trimmed_fastqs }

process STAR_Align {
    input:
        tuple path(fastq), val(genomeDir)

    output:
        path "*.bam"

    script:
        sample_id = fastq.getBaseName().replaceFirst(/^trimmed_/, "").replaceFirst(/\.fastq$/,
        """
        STAR --genomeDir $genomeDir \
            --readFilesIn $fastq \
            --runThreadN ${params.threads} \
            --outTmpDir ${sample_id}_STARtmp \
            --outSAMtype BAM SortedByCoordinate \
            --outFileNamePrefix ${sample_id}_

        # Rename STAR output to something simpler
        mv ${sample_id}_Aligned.sortedByCoord.out.bam ${sample_id}.bam

        # Copy results to final output dir
```

```
        mkdir -p ${params.outdir}
        cp ${sample_id}.bam ${params.outdir}/

        cp -f ${sample_id}.Log.final.out ${params.outdir}/ || true
        """
}

workflow {
    trimmed_fastqs
        .combine(Channel.value(params.genomeDir))
        | STAR_Align
}
```

```
nextflow info
```

```
  Version: 24.10.5 build 5935
  Created: 04-03-2025 17:55 UTC (12:55 GMT-04:00)
  System: Linux 5.14.0-362.13.1.el9_3.x86_64
  Runtime: Groovy 4.0.23 on OpenJDK 64-Bit Server VM 21.0.6+7-LTS
  Encoding: UTF-8 (UTF-8)
```

```
nextflow run star_align.nf
```

This took about 3:30 hours for all 149 fastq files.

1.5 featureCounts

1.6 Clean Count Matrix

1.6.1 Remove Low Gene Expression Cells

1.6.2 Remove Suspected Doublets

1.7 Analyze Count Matrix