Introduction

This document outlines a step-by-step workflow to process Cellecta Clonetracker Barcode single-cell data and assign clonetracker barcodes to each cell using shell and Python scripts. The aim is to ensure reproducibility and clarity in sequencing data processing and barcode analysis.

Input Data

The input data for this workflow includes:

- **FASTQ files from Gene Expression Profiling NGS data:**
  - `/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/AD_GEX/AD_GEX_S3_R1_001.fastq.gz`
  - `/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/AD_GEX/AD_GEX_S3_R2_001.fastq.gz`
- **FASTQ files from the Cellecta Clonetracker barcode NGS library:**
  - `/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/AD_FBP1_S7_R1_001.fastq.gz`
  - `/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/AD_FBP1_S7_R2_001.fastq.gz`
- **Reference barcode sequences provided by Cellecta:**
  - `Cellecta-CloneTrackerXP-5M-Pool1-BC14-LNGS-300-Library-Design.txt`
  - `Cellecta-CloneTrackerXP-50M-BC30-LNGS-300-Library-Design.txt`

Step 1: Load Reference Barcode Sequences

Analyze Gene Expression Profiling NGS data and extract cell barcodes using `cellranger count`:

```
/mnt/project/Pipeline/Software/cellranger-8.0.0/cellranger count \
    --id=AD_GEX \
    --fastqs=/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/AD_GEX/ \
    --transcriptome=/mnt/project/Pipeline/Reference/10XGenomics/refdata-gex-GRCh38-2024-A \
    --create-bam true \
    --include-introns false

zcat /mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/AD_GEX/outs/filtered_feature_bc_matrix/barcodes.tsv.gz > AD_barcode.xls

sed 's/-.*//' AD_barcode.xls > AD_barcode_cleaned.tsv
```

Step 2: Process Barcode NGS Library Using UMI-Tools

Use `umi_tools` to extract and match barcodes from FASTQ files:

Shell Script

```
#!/bin/bash

FILEIN1=/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/AD_FBP1_S7_R1_001.fastq.gz
FILEOUT1=/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/`basename ${FILEIN1} .fastq.gz`_extracted.fastq.gz
FILEIN2=/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/AD_FBP1_S7_R2_001.fastq.gz
FILEOUT2=/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/`basename ${FILEIN2} .fastq.gz`_extracted.fastq.gz

WHITELIST=/mnt/project/Cellecta_scCRISPR/VCU_scCloneTracker/data/AD_barcode_cleaned.tsv

umi_tools extract \
    --bc-pattern=CCCCCCCCCCCCCCCCNNNNNNNNNNNN \
    --stdin $FILEIN1 \
    --stdout $FILEOUT1 \
    --read2-in $FILEIN2 \
    --read2-out=$FILEOUT2 \
    --whitelist=$WHITELIST
```

Step 3: Identify the Best Sequence for Each UMI

Run the following Python script to select the best-quality sequence for each UMI:

```
python best_sequence_umi.py
```

Step 4: Assign Clonetracker Barcodes to 10X Cell Barcodes

Use this script to map Clonetracker barcodes to cell barcodes:

```
python barcode_process_umi_5.py
```

Step 5: Final Barcode Assignment Using UMI Distribution

Perform the final barcode assignment based on UMI distribution and criteria:

```
python umi_distribution_analysis_umi_5.py
```

Conclusion

This workflow provides a systematic approach to process Cellecta Clonetracker single-cell data, assign barcodes, and analyze barcode distributions. Scripts and parameters can be adjusted based on dataset-specific requirements. By following these steps, you can ensure accurate and reproducible results for your analysis.