User’s Manual for scATACpipe

**Overview**

The goal of scATACpipe is to provide an end-to-end and easy-to-use pipeline to facilitate single-cell ATAC-seq (scATAC-seq) data analysis. This pipeline takes minimal input from user and applies various processing steps including raw fastq sequence preprocessing, cell clustering, single cell embedding, motif enrichment, and more (**Fig.1**).

Timeline

Description automatically generated with medium confidence

As illustrated in **Fig.1**, the workflow consists of 3 sub-workflows, namely, PREPROCESS\_DEFAULT, PREPROCESS\_10XGENOMICS, and DOWNSTREAM. Users can choose to skip PREPROCESS and run DOWNSTREAM by directly providing fragment files as input.

Guided by the best practices for building pipelines with Nextflow, scATACpipe was built using the TEMPLATE provided by nf-core (<https://github.com/nf-core/tools/tree/master/nf_core/pipeline-template>) as the backbone. To ensure the best reusability and portability, each analytical process is wrapped into an individual module that comes with a Docker image containing the software dependencies (**Table.1**, attached to the end). This makes it easy for advanced users to extend/modify the functionality of the pipeline.

Add a table summarizing the softwires used in this pipeline, also the references.

Add Docker Dockerfile GitHub links.

Add samplesheet.csv example.

For ArchR, natively supported genomes, and other supported genomes with already built annotations and other custom genomes.

Prerequisite:

Input: samplesheet structure;

Config:

Note: for cellranger option, the fastq name must not contain special characters other than letter, digit, dash, number, and dot is not allowed per cellranger-atac count requirement.

References:

1. Nextflow
2. Nf-core
3. Software used.

In order to maximize the reusability, each process is wrapped into

Each of the

Backed by Nextflow engine, scATACpipe integrates numerous commonly practiced analytical software and is designed to work compatibly across different sequencing platforms including 10xgenomics, biorad, *etc.*.

The workflow is summarized as below

Integrated steps:

1. Preprocessing:

The goal includes:

* 1. Prepare alignment bam file and fragment file from raw fastq files.
  2. Perform FastqQC and bam QC.

Available methods:

1. default: this will perform the following steps: barcode correction; debarcoding, trimming, mapping, QC.
2. 10xgenomics: leverages its official cellranger-atac count software
3. biorad: leverages its official ATAC-seq Analysis Toolkit, which includes: FASTQ Quality Control, FASTQ Debarcoding, Read Trimming, Alignment, Alignment QC, (under development: Bead Filtration, Bead Deconvolution, Cell Filtration, Peak Calling, Atac-Seq QC, and Report).
4. Downstream analysis:

Clustering, motif enrichment, …

**Usage**

**Prerequisite**

This pipeline requires Nextflow pipeline engine and one container engine (either Docker or Singularity) to function. Follow the instructions below to install all dependencies before continuing.

1. Download and install Nextflow engine: <https://www.nextflow.io/docs/latest/getstarted.html>

After installation, upgrade Nextflow to 20.06.0-edge by typing the following command:

NXF\_VER=20.06.0-edge nextflow self-update

1. Download and install Singularty or Docker engine:

Singularity: <https://sylabs.io/guides/3.5/user-guide/quick_start.html>

Docker: <https://docs.docker.com/get-docker/>

Note that for Docker to function properly on MacOS with Nextflow, must mount /Users and /tmp folder to the docker virtual machine so that Docker can access the host file system. This can be achieved by setting the docker.runOptions under nextflow.config. By default, it is configured to ‘-u \$(id -u):\$(id -g) --rm -v /Users:/Users -v /tmp:/tmp’.

Also note that when running on MacOS, Nextflow will automatically leverage all cores and spawn parallel threads for different task instances to minimize the running time. By default, the max\_memory allowed for a single individual instance is set to 128.GB meaning that if your Mac has 16.GB, 1 instance would request 128.GB, greater than 16.GB, therefore, an error message will pop out. To reduce memory usage, you can use ‘-profile local’ that preset the max\_memory to 8.GB so that for a 16.GB Mac with 4 cores, 2 jobs can be run in parallel.

**Basics**

This section covers the most fundamentals to get the pipeline running after installation of all dependencies.

1. First, clone the pipeline by:

git clone https://github.com/hukai916/scatacseqflow.git

1. Ensure Nextflow version > 20.06.0:

nextflow -v

1. Show help:

cd scatacseqflow

nextflow run main.nf –help

1. Specifying configurations:

There are two types of options in Nextflow: pipeline option and parameter option. While the pipeline option (supplied with single dash) regulates the behavior of the workflow (e.g. by specifying -resume, the previously analyzed results will be cached.), the parameter option (supplied with double dash) is specific to certain processes (e.g. by --input path\_to\_sample, the path\_to\_sample will be used as input). You can also set the default parameter options in the nextflow.config file, under the params scope.

1. Specifying profiles:

Nextflow profiles are sets of predefined pipeline parameters, which can be set by -profile profile\_name. For example, to run pipeline using singularity containers: nextflow run main.nf -profile singularity; to run pipeline with lsf executor (default is local, which will run on your local machine): nextflow run main.nf -profile lsf. Note that if multiple profiles, separate them by comma.

For complete Nextflow tutorial, see documentation: <https://www.nextflow.io/docs/latest/index.html>

**Preprocessing**

**Method1**: default

Invoked by: “-preprocess default”

Input:

1. A samplesheet.csv: supplied by “--input full\_path\_to\_samplesheet.csv”

A comma separated file containing data meta info, the header must be “sample\_name,path\_fastq\_1,path\_fastq\_2,path\_barcode”, please use assets/ samplesheet\_test.csv as template. Note that you may specify multiple samples, each with a new row. For each sample, multiple fastq files (e.g. multiple sequencing lanes) can be listed under each column, and they must be separated by semicolon (;), those files will be merged into one single fastq file during the preprocessing step.

1. Read1 adapter sequence supplied by “--read1\_adapter”

Default to Illumina universal adapter: “AGATCGGAAGAGC”. This will be used for trimming with cutadapt

1. Read2 adapter sequence supplied by “--read2\_adapter”

Default to Illumina universal adapter: “AGATCGGAAGAGC”. This will be used for trimming with cutadapt

1. A mapper software to perform read alignment, which should be supplied by “--mapper”

Default to bwa, choose from “bwa” and “minimap2”.

1. A reference genome to map reads to, which should be supplied by “--ref\_fasta\_ucsc/--ref\_fasta\_ensembl”

You can choose from “hg19”, “hg38”, “mm10”, *etc.* if using --ref\_fasta\_ucsc.Must follow UCSC naming convention. Or chose from “homo\_sapiens”, “mus\_musculus”, if using --ref\_fasta\_ensembl. Alternatively, you can choose to use customized fasta file as reference genome by “--ref\_fasta”. If you would like to use already built index for faster processing, you can supplied it with either “--ref\_bwa\_index” or --ref\_minimap2\_index”. Note that when using bwa, must pass in the full path to the folder containing only the intended bwa indices; when using minimap2, must pass in the full path to the intended minimap2 index file.

Note that the parameter “--ref\_bwa\_index/--ref\_minimap2\_index” will overwrite “--ref\_fasta”, which in turn overwrites “--ref\_fasta\_ucsc/--ref\_fasta\_ensembl”.

Output:

**Method2**: 10xgenomics

Invoked by: “-preprocess 10xgenomics”

Input:

1. A samplesheet.csv: supplied by “--input full\_path\_to\_samplesheet.csv”

A comma separated file containing data meta info, the header must be “sample\_name,path\_fastq\_1,path\_fastq\_2,path\_barcode”, please use assets/ samplesheet\_test.csv as template. Note that you may specify multiple samples, each with a new row. For each sample, multiple fastq files (e.g. multiple sequencing lanes) can be listed under each column, and they must be separated by semicolon (;), those files will be merged into one single fastq file during the preprocessing step.

1. A folder containing 10xgenomics provided reference: supplied by “--ref\_cellranger full\_path\_to\_reference\_folder”

A 10xgenomics provided reference that is designed to work with cellranger-atac, follow the followings to get one: <https://support.10xgenomics.com/single-cell-gene-expression/software/pipelines/latest/advanced/references>

Output: bam, fragment file, clustering, *etc.*

**Method3**: biorad (under development)

Note that this option is wrapped around the ***Bio-Rad ATAC-Seq Analysis*** ***Toolkit*** (<https://www.bio-rad.com/webroot/web/pdf/lsr/literature/Bulletin_7191.pdf>) and must be used with compatible sequencing data.

Invoked by: “-preprocess biorad”

Input:

1. A samplesheet.csv: supplied by “--input full\_path\_to\_samplesheet.csv”

A comma separated file containing data meta info, the header must be “sample\_name,path\_fastq\_1,path\_fastq\_2,path\_barcode”, please use assets/ samplesheet\_test.csv as template. Note that you may specify multiple samples, each with a new row. For each sample, multiple fastq files (e.g. multiple sequencing lanes) can be listed under each column, and they must be separated by semicolon (;), those files will be merged into one single fastq file during the preprocessing step.

1. A folder containing BWA index file: supplied by “--ref\_bwa\_index full\_path\_to\_bwa\_index\_folder”
2. A fasta file containing reference genome: supplied by “--ref\_bwa\_fasta”
3. A pipeline parameter supplied by “--biorad\_genome”, which takes the value of either “hg19”, “hg38”, “mm10”, or “hg19-mm10”.

Output: bam, fragment file, clustering, *etc.*

For complete Nextflow tutorial, see documentation:

1. Clone hukai916/Samplesheet.csv:

Raw sequencing

**Preprocessing**

Samplesheet.csv:

Raw sequencing fastq files including barcode sequences.

Some example commands:

Test of bwa:

nextflow run main.nf --input /home/kh45w/workflow/scatacseqflow/assets/samplesheet\_test\_hpc\_mini.csv --preprocess default --outdir test\_run\_default\_bwa -profile singularity,lsf --ref\_fasta\_ucsc mm10 --mapper bwa

nextflow run main.nf --input /home/kh45w/workflow/scatacseqflow/assets/samplesheet\_test\_hpc\_mini.csv --preprocess default --outdir test\_run\_default\_bwa\_fa -profile singularity,lsf --mapper bwa --ref\_fasta /home/kh45w/workflow/test\_data/biorad/ref/mm10/fasta/mm10.fa

nextflow run main.nf --input /home/kh45w/workflow/scatacseqflow/assets/samplesheet\_test\_hpc\_mini.csv --preprocess default --outdir test\_run\_default\_bwa\_index -profile singularity,lsf --mapper bwa --ref\_bwa\_index /home/kh45w/workflow/test\_data/biorad/ref/mm10/bwa

Test of minimap2:

nextflow run main.nf --input /home/kh45w/workflow/scatacseqflow/assets/samplesheet\_test\_hpc\_mini.csv --preprocess default --outdir test\_run\_default\_minimap -profile singularity,lsf --ref\_fasta\_ucsc mm10 --mapper minimap2

nextflow run main.nf --input /home/kh45w/workflow/scatacseqflow/assets/samplesheet\_test\_hpc\_mini.csv --preprocess default --outdir test\_run\_default\_minimap\_fa -profile singularity,lsf --mapper minimap2 --ref\_fasta /home/kh45w/workflow/test\_data/biorad/ref/mm10/fasta/mm10.fa

nextflow run main.nf --input /home/kh45w/workflow/scatacseqflow/assets/samplesheet\_test\_hpc\_mini.csv --preprocess default --outdir test\_run\_default\_minimap\_index -profile singularity,lsf --mapper minimap2 --ref\_minimap2\_index /home/kh45w/workflow/scatacseqflow/work/ac/278c8937706e6131573743536ed640/mm10.fa.mmi

**Table.1** Nextflow modules and software dependencies

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Sub-workflow** | **Module** | **Software** | **Functionality** | **Docker image** | **References** |
| PREPROCESS\_DEFAULT | GET\_10XGENOMICS\_FASTQ | pure bash | concatenate fastq files that belong to the same sample | ubuntu:xenial |  |
| FASTQC | FastQC [1] | perform fastq sequence quality assessment with FastQC | hukai916/fastqc\_0.11.9:0.1 | [1] Andrews, S. (2010). FastQC: A Quality Control Tool for High Throughput Sequence Data [Online]. Available online at: http://www.bioinformatics.babraham.ac.uk/projects/fastqc/ |
| ADD\_BARCODE\_TO\_READS | sinto [2] | add barcode to the front of fastq name line | hukai916/sinto\_xenial:0.1 | [2] https://timoast.github.io/sinto/ |
| CORRECT\_BARCODE | custom R script | correct barcodes that are with 1 mismatch | hukai916/r\_sc\_atac:0.1 |  |
| CORRECT\_BARCODE\_PHENIQS | pheniqs [3], custom Python scripts | correct barcodes and add barcode to fastq | hukai916/pheniqs\_xenial:0.1 | [3] https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-021-04267-5 |
| MATCH\_READS | seqkit [4] | re-pair fastq files | hukai916/seqkit\_0.16.1:0.1 | [4] Shen, W., Le, S., Li, Y., & Hu, F. (2016). SeqKit: A Cross-Platform and Ultrafast Toolkit for FASTA/Q File Manipulation. PLOS ONE, 11(10), e0163962. https://doi.org/10.1371/journal.pone.0163962 |
| CUTADAPT | cutadapt [5] | trim off adapter sequences | hukai916/cutadapt\_xenial:0.1 | [5] Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.Journal, 17(1), 10. https://doi.org/10.14806/ej.17.1.200 |
| MATCH\_READS\_TRIMMED | seqkit | re-pair fastq files (the same to MATCH\_READS) | hukai916/seqkit\_0.16.1:0.1 |  |
| BWA\_INDEX | bwa [6] | index genome file | hukai916/bwa\_xenial:0.1 | [6] Li, H., & Durbin, R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics, 25(14), 1754–1760. https://doi.org/10.1093/bioinformatics/btp324 |
| DOWNLOAD\_FROM\_UCSC\* | wget | download genome fasta from UCSC website | hukai916/miniconda3\_xenial:0.1 |  |
| GET\_PRIMARY\_GENOME | samtools [7] and bash | retrieve the primary genome from downloaded UCSC genome | hukai916/bwa\_xenial:0.1 | [7] Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., & Durbin, R. (2009). The Sequence Alignment/Map format and SAMtools. Bioinformatics, 25(16), 2078–2079. https://doi.org/10.1093/bioinformatics/btp352 |
| BWA\_MAP | bwa | map reads to genome with BWA | hukai916/bwa\_xenial:0.1 |  |
| BAM\_FILTER | samtools and bash (awk) | filter out unqualified reads from bam file | hukai916/bwa\_xenial:0.1 |  |
| REMOVE\_DUPLICATE | custom Python code, samtools | remove PCR duplicated fragments | hukai916/pysam\_xenial:0.1 |  |
| QUALIMAP | qualimap [8] | perform bam QC analysis | hukai916/qualimap\_xenial:0.1 | [8] García-Alcalde, F., Okonechnikov, K., Carbonell, J., Cruz, L. M., Götz, S., Tarazona, S., Dopazo, J., Meyer, T. F., & Conesa, A. (2012). Qualimap: evaluating next-generation sequencing alignment data. Bioinformatics, 28(20), 2678–2679. https://doi.org/10.1093/bioinformatics/bts503 |
| GET\_FRAGMENTS | sinto | get fragment file from bam | hukai916/sinto\_xenial:0.2 |  |

The Dockerfiles that are used for creating the containers can be found: <https://github.com/hukai916/Dockerfile_collection>