Handbook Seq Suite

Methylation Data Analysis

For Research Use Only.

Legal Notices

Document 1020783, Rev A, Feb 2024

3210 Merryfield Row
San Diego, CA 92121, United States
https://scale.bio/
support@scale.bio

Disclaimer of Warranty:
TO THE EXTENT PERMITTED BY APPLICABLE LAW, SCALEBIO PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL SCALEBIO BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENT, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF SCALEBIO IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE. Any warranties applicable to the ScaleBio products are set forth in the Terms and Conditions accompanying such product and such Terms and Conditions are not modified in any way by the terms of this notice.

Trademark Information:
ScaleBio may make reference to products or services provided by other companies using their brand names or company names solely for the purpose of clarity, and does not assert any ownership rights over those third-party marks or names. Images were created with BioRender.com

Patent Information:
ScaleBio products may be covered by one or more patents as indicated at: https://scale.bio/legal-notice/

Terms and Conditions:
The use of the ScaleBio products described herein is subject to ScaleBio's Terms and Conditions that accompany the product, or such other terms as have been agreed to in writing between ScaleBio and the user.

Intended Use:
All products and services described herein are intended FOR RESEARCH USE ONLY and NOT FOR USE IN DIAGNOSTIC PROCEDURES.

Table of Contents

Introduction

The ScaleBio™ Single Cell Methylation Sequencing Kit uses combinatorial indexing strategy to resolve 5mC methylation whole genome bisulfite sequencing data at single cell resolution. The ScaleBio Seq Suite: Methylation Data Analysis Pipeline ScaleMethyl is designed as an end-to-end workflow that takes users from raw sequencing output of their ScaleBio Single Cell Methylation Sequencing Kit library to a thorough assessment of that library's performance, cell calling, and ultimately methylation rate and coverage matrices.

This handbook serves as a high-level guide for setting up, running, and understanding the outputs of the ScaleBio Single Cell Methylation Data Analysis Pipeline. For specific step-by-step instructions on installing and running the pipeline please refer to our GitHub repository. The introductory readme markdown file (README.md) provides an overview of the workflow; additionally, there are a series of markdown files (*.md) within ScaleMethyl/docs to help guide users in more detail at each major step.

Quick Start Guide

- First install Nextflow (22.04 or later).
- Download this workflow to your machine.
- Install dependencies.
- Launch the small pipeline test run.
- Download / configure a reference genome for your samples.
- Create a samples.csv table for your samples.
- Create runParams.yml, specifying inputs and analysis options for your run.
- Launch the workflow for your run.

Chapter 1: Pipeline Setup, Installation and Testing

1.1. Requirements

The workflow can be launched on any POSIX compatible system (Linux, macOS, etc.). It requires Bash 3.2 (or later) and Java 11 (or later, up to 20). For running the pipeline on either system, the recommended configuration is 128 GB of RAM and 32 CPU cores, 1TB free SSD. It is strongly recommended to run on an HPC with a job scheduler or batch environment like AWS batch. In addition, the workflow requires temporary storage for intermediate files, ranging up to 5 TB for large (e.g. NovaSeq S4) sequencing runs.

1.2. Install Nextflow

To install the pipeline on a new system, first install Nextflow, following the instructions at https://www.nextflow.io/. The installation requires Java version 11 or higher (up to 20). Once downloaded to your system, you can confirm that the Nextflow executable works as expected by running the following command:

```
nextflow run hello
```

Note that Nextflow can be installed in your user directory without admin rights on the system. Once you have the Nextflow command running, it is recommended to add the executable as a path variable in the users bash startup file (i.e., .bash_profile, .bashrc, .profile).

1.3. Install the ScaleBio Seq Suite: Methylation Pipeline

The pipeline can be downloaded by two methods:

1. By going to the ScaleBio Seq Suite: Methylation GitHub page, clicking the green “Code” button and then “Download ZIP”. Unpack this file on your system directly in the directory in which you want to install the pipeline. To make sure the download is complete, make sure the executable commands (PY files) in the ScaleMethyl/bin directory have the appropriate read/write/execute privileges on your server.
2. The GitHub repository can be cloned to your machine:

   ```
   git clone https://github.com/ScaleBio/ScaleMethyl.git
   ```

Note this may require setting up a personal access token, instructions for which can be found here.

1.4. Install Dependencies

The workflow requires several dependencies (see *Appendix B: Software Dependencies*). These can be provided in one of three ways:

1. Using containers (‘docker’).
2. Using the conda package manager.
3. Installed manually.

1.4.1. Using Containers

Containers have all the necessary code, libraries, and configurations for the workflow to operate on most systems. If your system supports execution of a container (either using *Docker* or *Singularity*) this is likely the easiest way to handle dependencies, especially if you are familiar with running containerized workflows.

1.4.2. Docker vs. Singularity Container

Docker support is enabled with the Nextflow command-line option `-profile docker`. Note that setting up *Docker* support for the first time on a new system typically requires admin (root) access and some familiarity with the system.

If your system access does not support the use of *Docker*, then *Singularity* is an alternative for container execution that is available on many HPC clusters. We require Singularity 2.3 or newer. Setting `-profile docker,singularity` (no space) will use the Singularity engine for all dependencies.

Container usage can require some extra setup to ensure all relevant file paths are accessible from inside the containers:

- For *Docker*, Nextflow will set the relevant options automatically at runtime to mount (`bind`) input and output paths to the container.
- For *Singularity*, this requires `USER BIND CONTROL` to be enabled in the system-wide configuration (see The Singularity Config File and the notes in the Nextflow singularity documentation).
- The environment variable `NXF_SINGULARITY_CACHEDIR` can be used to control where *Singularity* images are stored. This should be a writable location that is available on all compute nodes.
- Similarly, `TMPDIR` should be changed from the default `/tmp` to a location writable from the container if necessary.

1.4.3. Using conda

Another option is using the conda package manager. Nextflow can automatically create conda environments with most of the needed dependencies. This mode is selected by setting `-profile conda`. In this case, the following additional steps need to be completed:

- Install ScaleBio Tools
  - These are ScaleBio specific programs that are currently not available through conda.
  - Run `/PATH/TO/ScaleMethyl/envs/download-scale-tools.sh`
  - This will install the pre-compiled binaries in `ScaleMethyl/bin` (inside the Nextflow workflow directory), from where they will be available during workflow execution.
- If running from a sequencer runFolder (BCL) Illumina BCL Convert v3.9.3 is required to be installed (and available on `$PATH`).

See the Nextflow documentation for additional details of conda support in Nextflow.

1.4.4. Manual Dependency Installation

As a final alternative, the required dependencies can be installed directly, either by hand or using conda. A list of all requirements can be found in *Appendix B: Software Dependencies* or in the environment conda.yml files in `ScaleMethyl/envs`. Once you have installed the dependencies, and made sure they are available on `$PATH`, the workflow can be run without any `-profile`.

1.4.5. Run the Workflow Test

A sample dataset including usage instructions can be found here. Running this small sample dataset will allow you to rapidly test that you have installed the workflow correctly and your system requirements are met. This may also give you the opportunity to check compatibility of processed files with any downstream/secondary analysis.

Please note that the raw sample data is stored on Amazon Web Services (AWS S3) and will be downloaded to your system automatically once you execute the Nextflow command. Review your systems documentation for any additional steps that are required to download from AWS, or alternatively follow the instructions in the README to download the data directly to your system and run the test from the local copy.

1.4.6. Workflow Parallelism

Individual steps of the pipeline are run as separate Nextflow tasks. Nextflow will launch as many parallel tasks concurrently as possible given available resources. Additionally, for many tasks, such as BSBolt alignment, the number of parallel threads inside a single task is configured dynamically to match available resources.

The dynamic allocation of resources in Nextflow depends on which "executor" is selected (see Executors — Nextflow 23.04.1 documentation). By default ('*local*' executor), all tasks are launched on the same machine that Nextflow itself was started on, limiting the parallelism to the resources available on that computer (CPU cores and memory).

However, Nextflow also offers a wide range of other executors. On HPC systems, it can use SGE, slurm, or similar to submit individual tasks to different compute nodes. On AWS, Azure, or Google Cloud it can use *Batch* to achieve massive parallelism. The Nextflow documentation contains details about how each of these use-cases can be configured. We recommend using a batch mutli compute node system for the ScaleBio Methylation Nextflow workflow due to the level of parallelization and time required without splitting alignments by tagmentation barcode.

Other than the parallelism that Nextflow supports, the ScaleMethyl workflow provides two flags that control different levels of additional parallelism in the workflow:

- The `--splitFastq` flag enables bcl-convert to produce FASTQ files split by i5 barcode (when starting from a run folder). Since this produces smaller FASTQ files, the workflow can run multiple alignment, deduplication and extraction steps in parallel (see Figure 5).

    - This is beneficial on the full library sequencing runs for both the small and large kits by breaking up the computationally intense demultiplexing jobs more than normal lane lane splitting allows.
    - This option is on by default and must also be set with `--bclConvertParams` = "--no-lane-splitting true".
    - To run with normal lane splitting use `--splitFastq true --bclConvertParams` = "`--no-lane-splitting false`". For more details please see our examples using **pre-generated** FASTQ files as input.
    -

- The `--splitBarcodeParsing` flag enables bcParser, which is a ScaleBio developed tool for barcode demultiplexing and correction, to produce demultiplexed FASTQ files split by the tagmentation barcode. This produces as many output files as there are tagmentation barcodes. This has the same benefit as using the –*splitFastq* flag since it produces smaller FASTQ files which can go through alignment, deduplication and extraction in parallel, thus shortening the total execution time of the workflow.

Additional information about these types of parallelism can be found here.

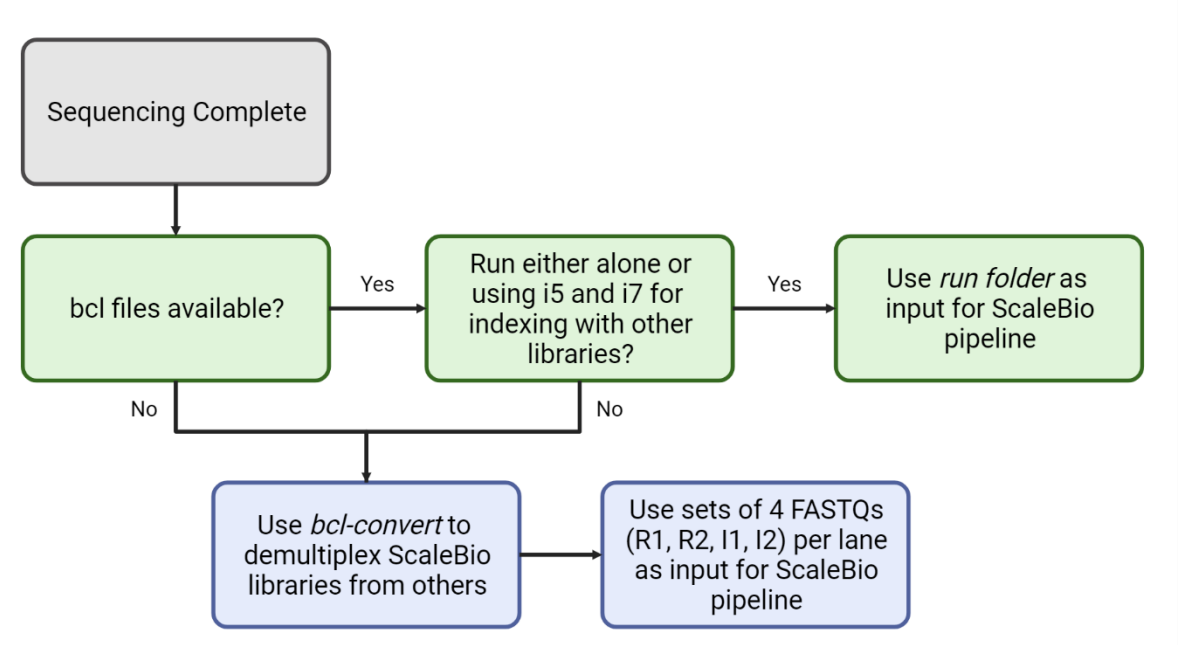
Chapter 2: Input Files

To run the pipeline the user will need to provide 3 types of input files:

1. Sequencing reads from both reads and both indices (R1, R2, and I1, I2)
    - **Note that the indexing reads, while not a standard output for many core facilities or sequencing providers, are required for the ScaleBio pipeline to effectively identify cell barcodes**. Please contact your local Field Application Scientist or email *support@scale.bio* to get instructions that can be shared with your core facility or sequencing provider to ensure you obtain the correct output files.
2. Reference genome (BSBolt index)
3. Sample table (samples.csv)

2.1. Sequencing Reads

Figure 1: Workflow Decision Tree for Data Processing



There are two ways to input the reads for the pipeline (ScaleMethyl/Fastq Generation):

2.1.1. BCL files as input – ***preferred pathway***

If the ScaleBio Methylation library was sequenced alone in a sequencing run or on a flow cell lane, or sequenced with other libraries *using only the i7 index for demultiplexing*, the easiest way to run the analysis is to start directly from the sequencer RunFolder [`--runFolder`] (this is the sequencer output folder containing the *RunInfo.xml* file). In this case, the ScaleBio RNA workflow will internally generate FASTQ files appropriate for pipeline input using Illumina *bcl-convert* (BCL Convert Support (illumina.com)).

2.1.2. FASTQ files as input

If the raw sequencer output is not available as BCL files, or the ScaleBio Methylation library was multiplexed with other libraries during sequencing and requires the i5 index read for demultiplexing, the analysis can be started from FASTQ files [`--fastqDir`] generated ahead of time. In this case, please note the following:

- For ScaleBio Methylation libraries, the Tagmentation Barcodes are included in read 2, while the Met i5 Index Barcodes and Met i7 Index Barcodes are in index reads 1 and 2, (`*_I[1,2]_*.fastq.gz`). The structure of the Methylation library is shown in *Appendix A: Methylation Library Structure and List of Barcode Sequences*. Since the index reads are needed for demultiplexing and barcode correction, we need to tell bcl-convert to generate index read FASTQs using the samplesheet.csv setting: CreateFastqForIndexReads,1
- ScaleMethyl by default generates FASTQ files split by the Met i5 Index Barcodes rather than by lane. This is beneficial for large sequencing runs as it enables better breaking up of the demultiplexing and barcode correction steps into smaller jobs than lane splitting would allow. When splitting by the Met i5 Index Barcodes, lane splitting must be disabled. This can be done using the option:

  ```
  bclConvertParams : "--no-lane-splitting true"
  ```

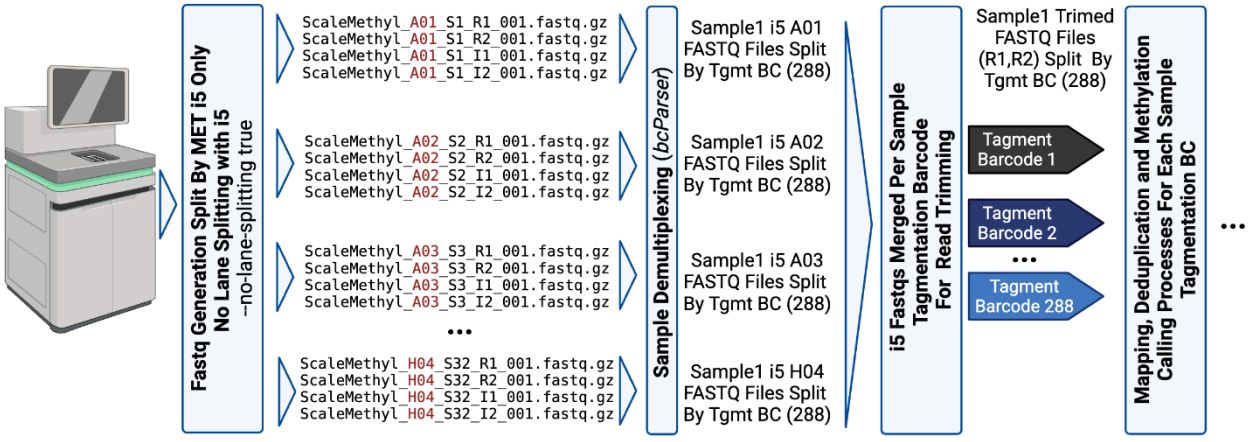
  in addition to

  ```
  splitFastq : true
  ```

  Both of these options are set by default in the workflow when starting from BCL files.

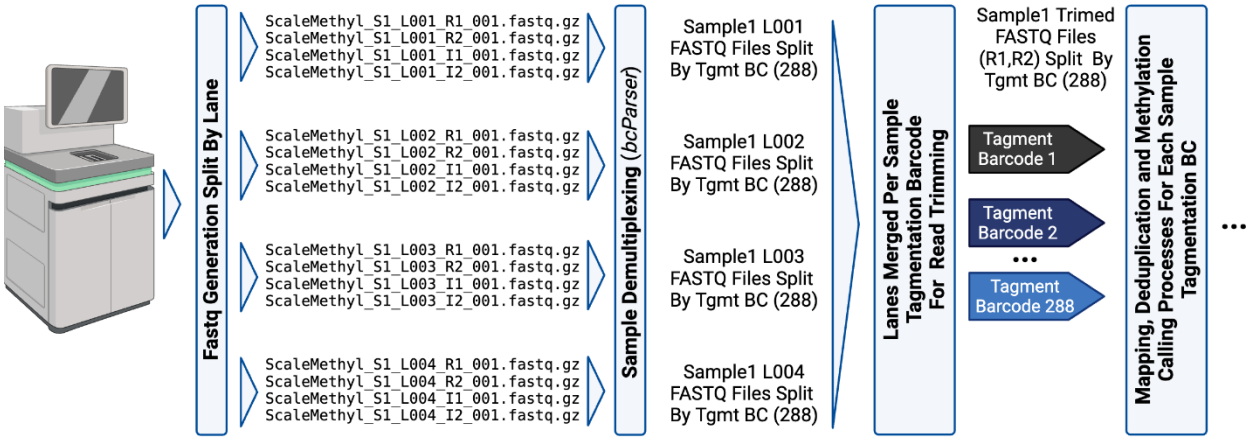
  The example file samplesheet_pcr_split.csv (which includes Met i5 Index Barcode information) can be used in this case. Please see parallel execution and fastq generation documentation for more details.

Figure 2: FASTQ Files Split by Met i5 Index Barcodes



- The example samplesheet.csv included in the ScaleBio pipeline can be used for FASTQ generation with bcl-convert and would create the FASTQ files below for a full ScaleBio run split by lane. These files can be fed directly into the ScaleBio Pipeline demultiplexing step which will run a demultiplexing job for each lane per library.

Figure 3: FASTQ Files Split by Lane



2.2. Reference Genome

The workflow requires a reference genome file and gene annotation file (note: these must be built in BSBolt v1.5.0 or newer). These are specified using a genome.json file, which contains the file paths to the reference files and other parameters. Further information can be found here: Scale Reference Genomes
Pre-built genomes for human (hg38), mouse (mm39), and a mixed human/mouse barnyard genome are available here:

- http://scale.pub.s3.amazonaws.com/genomes/methyl/grch38.tgz
- http://scale.pub.s3.amazonaws.com/genomes/methyl/mm39.tgz
- http://scale.pub.s3.amazonaws.com/genomes/methyl/grch38_mm39.tgz

Download the appropriate reference file to your system, unpack (`tar -xzf GENOME.tgz`), and then specify the JSON file inside the directory (e.g. grch38/grch38.json) for the analysis [`--genome`].

Note: You must download and unpack these files locally first. Do not specify the URLs to these TGZ files directly in the `--genome` option. Similarly, do not use the example genome.json (`docs/examples/genome.json`) for real analysis beyond the test run. This file refers to a genome index stored online (AWS S3), which would be downloaded anew for every analysis run, slowing down the analysis significantly.

Building a new BSBolt reference index for a different species requires the genome sequence (FASTA). Also, if you require a different genome annotation for the pre-built genomes listed above, a new reference index still needs to be created with the same reference genome. Please

see below the commands for generating a BSBolt genome index (v1.5.0 or higher) and new JSON file (see the BSBolt documentation for additional options):

```
bsbolt Index-G {fasta reference} -DB {database output}
```

Then create a genome.json file with the following as a template:

```
{
    "name": "hg38",
    "speciesName": "Homo sapiens",
    "bsbolt_index": "bsbolt.ref",
    "genomeTiles": "50kbp.bed",
    "genomeTilesCh": "250kbp.bed",
    "bsbolt_chrs": "bsbolt_chrs.tsv",
    "tssWin": "tss100UpDn.bed",
    "backgroundWin": "backgroundTssUpstream1200to1000.bed"
}
```

Table 1: Example genome.json File Structure

Column	Description	Example
name	The name of the species / genome-version	Hg38
speciesName	Name of the species	Homo sapiens
bsbolt_index	Path to the BSBolt index directory	`/PATH/TO/bsbolt.ref`
genomeTiles	Path genomic non-overlapping bins (default 50kb) sorted BED file used as features for CG methylation matrices.	`/PATH/TO/genomeTiles`
genomeTilesCh	Path genomic non-overlapping bins (default 250kb) sorted BED file used as features for CH methylation matrices.	`/PATH/TO/genomeTilesCh`
bsbolt_chrs example	Path to tsv labeling the mitochondrial chromosome and non-primary chromosomes (unlocalized, unplaced and alt contigs) to filter from the bam in the deduplication step.	`/PATH/TO/bsbolt_chrs.tsv`
tssWin	BED file with 200bp windows around the TSS (transcription start site) to calculate TSS enrichment (quality control for the nucleosome depletion step).	`/PATH/TO/tss.bed`
backgroundWin	BED file with 200bp windows 1kb upstream (background regions) of the TSS to calculate TSS enrichment (quality control for the nucleosome depletion step).	`PATH/TO/background.bed`

2.3. Sample Table (samples.csv)

A sample table (e.g. samples.csv) file is used to list the samples included in an analysis run, their sample barcode (Tagmentation Barcode) sequences and optional sample-specific analysis parameters.

It is a comma separated file (CSV), with a header line (column names), followed by one sample per line. The first column ("sample") contains the name of each sample. All other columns are optional.

Table 2: Example Sample Table

Column	Description	Example
sample	Sample Name	Foobar-2
barcodes	Tagmentation Barcode Plate wells used for this sample	1A01-1H02
libName	Name for the sequencing library/fastqfiles	ScaleMethyl
libIndex (optional)	Subset of Met i7 Index Barcodes	TTGATATGAA;CCTAAGCGGT
libIndex2 (optional)	Subset of Met i5 Index Barcodes	TATCATGATC;GAGCATATGG
threshold (optional)	Minimal number of uniquely mapped reads to call a passing cell (overrides the threshold called in *Cell Filtering*	

- Column names are case sensitive.
- "barcodes" are sample tagmentation barcodes for 3 plates (plate-row-column). Please check our whitelist to be sure that you have them in the correct order to take a range.
- "sample" and "libName" should consist only of letters, numbers, dash (-) and dot (.). Underscores are not valid.
- When running from pre-existing FASTQ file input, "libName" should match the first part of the FASTQ file name for this sample, e.g.: `Foobar-2` for `Foobar-2_*.fastq.gz`.
- When providing sequences in the "libIndex" and "libIndex2" columns, separate them using
- If "libIndex" and "libIndex2" are not provided, all Met i5 Index Barcodes and Met i7 Index Barcodes will be used in the samplesheet.csv for bcl-convert.

Notes on the "barcodes" column used for demultiplexing samples

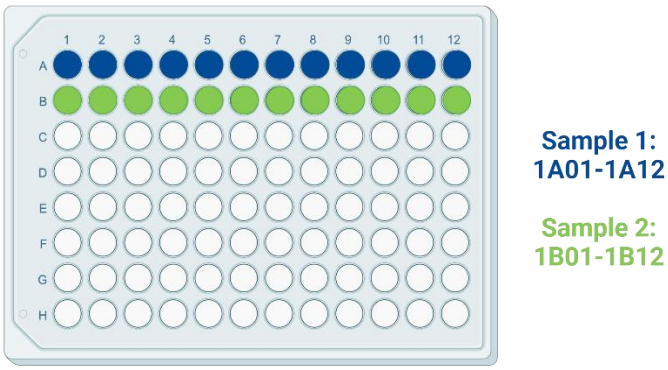
During analysis, sequencing data is first converted into library FASTQ files (libName column). If multiple samples were included in one sequencing library, these are then demultiplexed based on the sample (Tagmentation) barcodes, for example:

sample	Barcodes (wells in Tagmentation Plate)
Foo	1A01-1H06
Bar	1A07-1H12

- The tagmentation wells used for each sample are given in barcodes as either:
    - An individual value (**1A01)**
    - A range of wells (**A01-H06**)*
    - A list of values or ranges, separated by semicolon (;) (1A01;1H06)
- Wells are sorted first by plate, row number then column number i.e., 1A01-1H06.

*Note that all ranges are read in row-wise order, e.g. **1A01-1C12**, refers to the first 3 rows (A-C) of plate 1.

Figure 4: Example of Sample Plate Layout and Sample Sheet Annotation



Chapter 3: Step-by-Step Overview of the Pipeline

In this chapter, we list the steps performed by the sequencing analysis pipeline, showing the order in which they are performed, and providing high-level information about the analyses performed at each step.

Figure 5: Overview of Workflow Parallelism

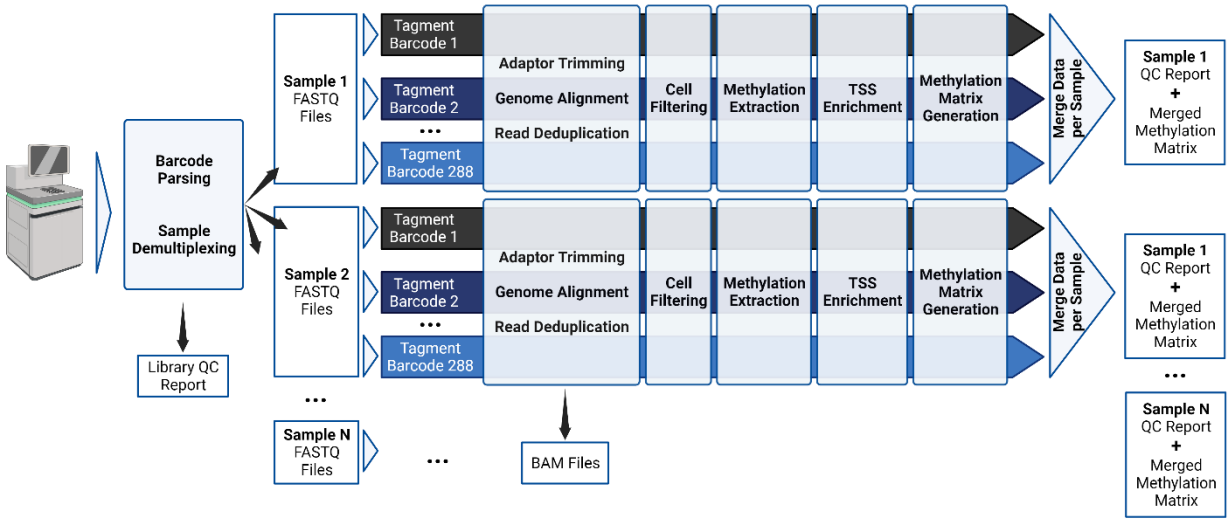
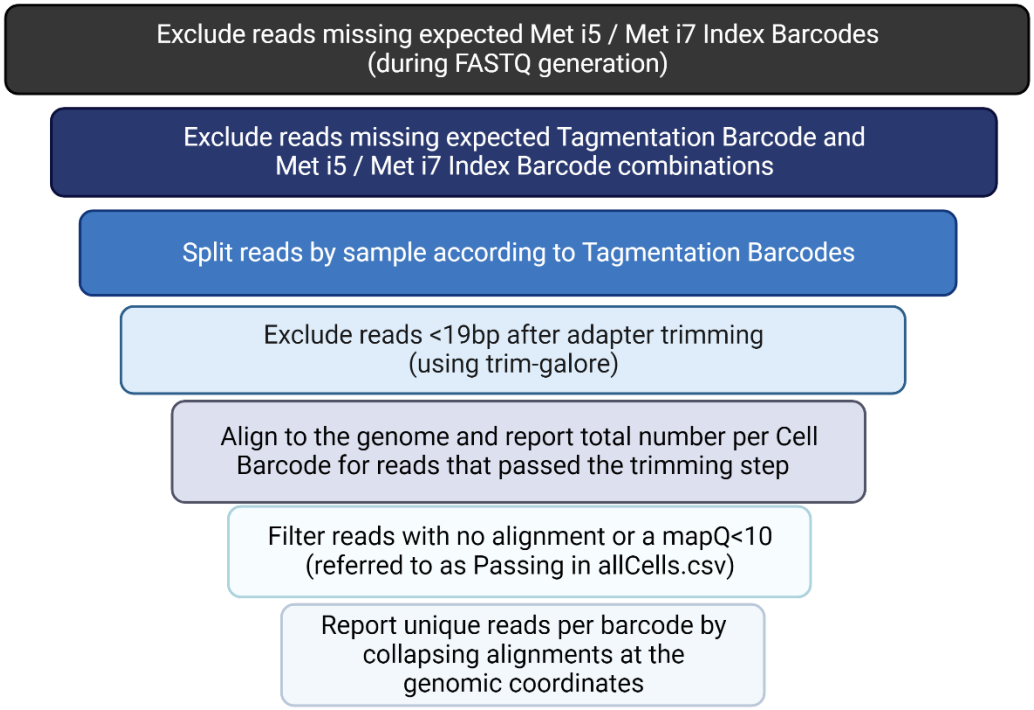


Figure 6: Overview of Pipeline Filter Steps



3.1. FASTQ Generation

- This is performed only if a RunFolder (BCL files) is provided as input, rather than FASTQ files that were pre-generated using Illumina *bcl-convert*.
- All necessary FASTQ files are generated, including the reads 1 and 2, i7 and i5 index reads (R1, R2, I1 and I2, respectively)
    - One set of FASTQ files, containing the raw unaligned reads, is generated for the entire sequencing run/i5 FASTQ.
    - Reads not matching the expected ScaleBio Met i5 Index Barcode or Met i7 Index Barcode sequences are filtered (into the *Undetermined* FASTQ files).
- FASTQ samplesheet.csv can be generated from the samples.csv with optional subsets of i5,i7 indexes, or can be passed to the workflow using the **fastqSamplesheet** config option. Here is an example samplesheet.csv
- If --splitFastq is enabled, a separate FASTQ file is produced for each i5 index resulting multiple files, that can be processed in parallel for the barcode demultiplexing and correction step with bcParser. Here is an example samplesheet_pcr_split.csv with the additional i5 splitting from –-splitFastq.
- When using this option, bcl-convert default for splitting by lane must be shut off with the option bclConvertParams = "--no-lane-splitting true", as we do not support splitting by lane and i5.
    - These i5 split or lane split FASTQs will be joined during the trimming step.
- The *bcl-convert* step produces the standard Illumina reports on the number of reads per library and related metrics: Output Files (illumina.com)

3.2. FastQC

- This is an optional step to run QC reports on the input FASTQ files [--fastqc] using *FastQC (*Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data.)
- The FastQC reports include information on output such as Q30 scores, adapter content, and base distribution for Read 1 & 2.
- The index reads are excluded from the report.

3.2. Barcode Parsing

- In this step, the three single-cell barcodes (Tagmentation Barcode, Met i5 and Met i7 Index Barcodes) are extracted from the reads and error corrected using ScaleBio *bcParser* executable.
- This is included in the Docker/Singularity container or can be downloaded with `ScaleMethyl/env/download-scale-tools.sh`
- Error correction is done against the list of expected barcode sequences, allowing up to 1 mismatch.
- If --splitFastq is enabled, each set of FASTQ files (e.g. each sequencing lane) is processed in parallel, otherwise they are pooled before this step.

3.3. Sample Demultiplexing

- If different samples were loaded into different wells of the RT Barcode plate during the ScaleBio Single Cell Methylation Sequencing workflow, they are separated at this point for independent analysis.
- Implemented in *bcParser* together with Barcode Parsing (see above).
- Uses the *samples.csv* sheet (see Chapter 2: Inputs for more information) to define RT barcodes for each sample.
- Outputs a pair of FASTQ files (barcode read and transcript read) for each sample.
- If --splitFastq is enabled and the reads were not already split on PCR barcode during bcl-convert, reads within each input FASTQ will be split based on RT barcode here. These subsets can then also be processed in parallel. RT barcodes corresponding to the same sample will be merged at the end of the workflow.

3.4. Read Trimming

- After the barcode demultiplexing and correction that removes set barcode locations, we must trim from the other end to get any additional adapter sequences. trim-galore v0.6.7 that utilizes cut-adapt trims additional adapter sequences and 10 additional bases from the 3' end of read 2, which is the Hmer 10 location. trim-galore is employed in paired-end mode.
- Read pairs with reads shorter than 19 bp after trimming are discarded.

3.5. Genome Alignment

- Aligns adapter trimmed reads to the genome using BiSuflite Bolt (BSBolt)
- BSBolt performs local sequence alignment, which utilizes forked versions of BWA.
- Optionally outputs a sorted BAM file with alignments for custom downstream analysis or visualization [*--bamOuf*]

3.6. Alignment Filtering and Deduplication

- `sc_dedup` is a ScaleBio developed tool for removing duplicate reads from an aligned BAM file. The tool is barcode-aware where reads from different barcodes aren't considered duplicates of one another.
- The number of passing reads is reported for properly paired reads that meet a minimal mapping quality cutoff [`--minMapq` default: 10].
- `sc_dedup` uses the leftmost position of the leftmost aligned fragment in a mate-pair and compares it to the corresponding position of previously encountered reads in the <IN.BAM> file. For singletons and mate-pairs with one unmapped fragment, the leftmost alignment position of the mapped fragment will be used. The rightmost fragment in a mate-pair is not considered. After the first read at a position is encountered, future reads whose alignments start at that position will be discarded as duplicates.
- These passing reads are used to calculate, and the number of unique reads collapsed by genome coordinates. Total, Passing and Unique reads are all reported in the allCells.csv.

- Optionally outputs a sorted deduplicated BAM file with alignments for custom downstream analysis or visualization [`--bamDedupOut true`].

3.7. Cell Filtering

- The default cell thresholding algorithm used by the ScaleMethyl workflow imposes a one-dimensional threshold on the deduplicated coordinate collapsed uniquely mapped reads per cell barcode.
- The cell barcode corresponds to the combination of the Tagmentation Barcode, the Met i5 Index Barcode and Met i7 Index Barcode.
- The cell threshold is determined using the following parameters:
    - A preliminary list of possible cells is set (either based on the number of expected cells set in samples.csv, or the number of cell-barcodes with over 1000 [`--minUniqCount`] unique reads.
    - The read-count of top cells, topCount, is estimated as the `--topCellPercentile` [99] of read-counts of cell-barcodes above `--minUniqCount`.
    - The cell threshold is set at a fixed `--minCellRatio` [20] of the TopCount (topCount/20 [*--minCellRatio*]).
- You can adjust `--minUniqCount`, `--topCellPercentile`, `--minCellRatio` in the runParams.yml see. These can be overridden by providing the threshold column in the samples.csv.

3.8. Methylation extraction

- Extracts methylation calls from a BSBolt deduplicated BAM and outputs bismark annotated methylation calls in the chrom folder in sorted BED format
    - z unmethylated C in CpG context
    - Z methylated C in CpG context
    - x unmethylated C in CHG context
    - X methylated C in CHG context
    - h unmethylated C in CHH context
    - H methylated C in CHH context
- Optionally outputs sorted BED raw methylation calls for custom downstream analysis [`--covOut default false`].

3.9. Generation of Matrix

- This output files are dense matrices (.mtx file format) where columns represent single cell barcodes, and rows are non-overlapping genomic bins from `genomeTiles` in the genome.json.
- The CG or CH methylation rate matrix (mtx format) can be read into Seurat, ScanPy and similar tools for visualization or downstream analysis.
- Martrix generation can be turned off for small QC runs using `--matrixGenerationCG --matrixGenerationCH` [default true for both options].
- For more information on the specific martrix formats, see the links in *Chapter 4: Overview of Analysis Output Files* and our Matrix Detailed Descriptions in ScaleMethyl.

3.10. TSS enrichment

- TSS Enrichment is a QC metric to access signal-to-background. The goal of nucleosome disruption is to ablate this signal and achieve a value at or below 1. The ratio is calculated using averaged 200 bp centered on the TSS as signal and 200 bp windows 1000 bp upstream (-) from TSSs as background.
- Deduplicated bam files are used to calculate these average coverages to compute this signal over background ratio using the TSS BED file and background file in the genome.json.

3.11. Generation of Sample QC Report

- A summary report with metrics for each sample
- Includes mapping metrics, cell-counts, and sensitivity metrics.
- Includes distribution of RT barcodes for the specific sample
- Produces a HTML document and a CSV file with metrics in text format
- Note: for more information on the specific metrics found in this report please see the links in *Chapter 4: Overview of Analysis Output Files*.

3.12. Generation of Library QC Report

- Produces an HTML report with QC metrics for the whole ScaleBio Methylation library. This report focuses on barcode matching rates, read distribution across samples, and data quality across all barcodes (Tagmentation Barcode and Met i5/Met i7 Index Barcodes).
- You can also find combined read and methylation summary plots for all samples in the library.

Chapter 4: Overview of Analysis Output Files

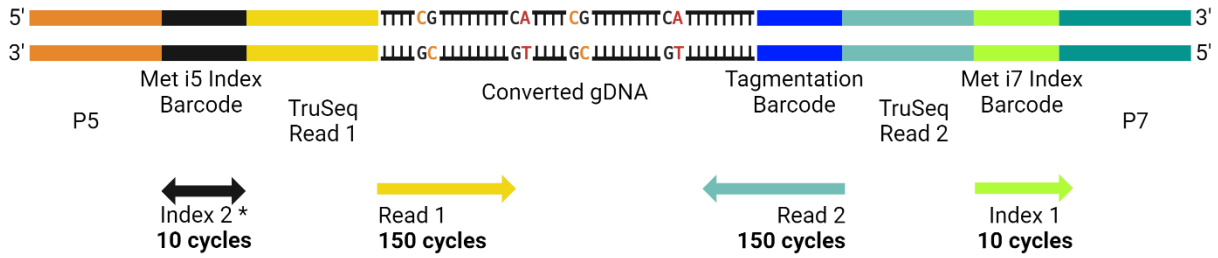
For more information about the specific metrics found in any of the output files please see the links below which contain the most up-to-date information.

- Outputs Overview
    - QC summaries, outputs reports and optional raw outputs see advanced options for more details on optional file outputs
    - Matrices
- QC Report Overview
    - Sample report
    - Library report

Appendix A: Methylation Library Structure and List of Barcode Sequences

The overall library structure for Single Cell Methylation Kit v1.0 is shown in Figure 7.

Figure 7: Library Structure for the Single Cell Methylation Kit v1.0 Assay



* orientation depends on sequencer and sequencing chemistry

Component	Size	Description
P5	–	Illumina P5 sequence (AATGATACGGCGACCACCGAGATCTACAC)
Met i5 Index Barcode	10 bp	Cell Barcode from Combinatorial Indexing
TruSeq Read 1	–	Illumina sequencing primer
Tagmentation Barcode	8 bp	Cell Barcode from Combinatorial Indexing
TruSeq Read 2	–	Illumina sequencing primer
Met i7 Index Barcode	10 bp	Cell Barcode from Combinatorial Indexing
P7	–	Illumina P7 sequence (ATCTCGTATGCCGTCTTCTGCTTG)

The full list of all barcode sequences can be found in the references directory of the workflow.

Appendix B: Software Dependencies

Nextflow Dependencies

- Java (11 or later)
- Nextflow (22.04 or later)

ScaleMethyl Dependencies

- Fastq generation:
  - nfcore/bcl-convert 3.9.3
- Main Analysis steps: (demultiplexing, trimming, alignment, deduplication, methylation extraction, matrix generation)
- TSS enrichment
- Report generation

ScaleBio Tools

In addition to third-party and open-source software the workflow also uses executable tools developed by ScaleBio:

- bc_parser
  - Extracts and error corrects cell-barcodes and UMIs from the original (input) FASTQ files
  - Splits (demultiplexes) the input FASTQ files into sample FASTQ files based on cell-barcodes (Tagmentation Barcode)
  - Barcode and read-level metrics
- sc_dedup
  - BAM deduplication aware of the multi-level combinatorial cell-barcodes.
  - Barcode and read-level metrics.
  - BAM filtering of non-primary chromosome contigs and low mapQ

Document Revision History

Revision	Revision Date	Document ID	Changes
Rev A	Feb 2024	1020783	Initial release.