Handbook Seq Suite

Methylation Data Analysis

For Research Use Only.

Legal Notices

Document 1020783, Rev B, Oct 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 the documentation bundled with the software release on our GitHub repository. The introductory readme 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.

System Requirements:

- Linux system with GLIBC >= 2.17 (such as CentOS 7 or later)
- Java 11 or later
- 128GB of RAM and 32 CPU cores

Quick Start Checklist

- Install Nextflow (23.10 or later).
- Download the latest ScaleMethyl workflow to your machine.
- Determine your system's optimal method to install dependencies.
- Launch the ScaleBio down sampled pipeline test run.
- Download one of the ScaleBio pre-built genomes or generate a new custom genome reference
- Create a sample barcode table (samples.csv) for ScaleBio sample demultiplexing.
- Specify all analysis parameters in the runParams.yml.
    - To specify the fastq or bcl runFolder directory:
        - *fastqDir : path/to/fastqDir*
        - *runFolder : path/to/bcl/runFolder*
    - To specify the library structure:
        - *libStructure : lib.json*
- Setup your final nextflow command and launch the workflow!

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
   ```

1.4. Install Dependencies

The ScaleBio Seq Suite: Methylation workflow requires a number of dependencies to run. These include ScaleBio developed tools and third-party, open-source executables and python libraries. All of these can be provided in one of three ways:

1. Using software containers (*docker* or *singularity* / *apptainer*).
2. Using the *conda* package manager.
3. Installed manually.

1.4.1. Using Containers

ScaleBio provides pre-built software containers with all dependencies for the workflow to operate on most systems. If your system supports a container engine (*Docker* or *Singularity*) this is likely the easiest way to handle dependencies, especially if you are familiar with running containerized workflows. Installing a container engine for the first time on a new system typically requires administrator (root) access and some familiarity with system configuration.

1.4.2. Using Docker or Singularity

Docker support is enabled with the Nextflow command-line option `-profile docker`. With this option the workflow will automatically download and use the pre-built containers for all dependencies.
If your system does not support the use of *Docker*, then *Singularity or Apptainer* is an alternative for container execution that is available on many HPC clusters. This is enabled by setting `-profile singularity.`
One important point is that all input and output paths need to be available (bind) inside the containers. For docker, Nextflow will set the relevant options automatically at runtime; for singularity this requires user mounts to be enabled in the system-wide configuration
See ScaleMethyl/docs/dependencies.md at main · ScaleBio/ScaleMethyl (github.com) for further details and https://www.nextflow.io/docs/latest/container.html for background and additional configuration options for containers in Nextflow.

1.4.3. Using conda

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

- Install and update *conda*
  - *conda update -n base -c defaults conda*
  - The workflow should be launched from a clean 'base' environment to avoid any version conflicts.
- 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 the conda environment scaleMethylTools.conda.yml file. All these dependencies and the ScaleBioTools (see above) need to be available on `$PATH` or in `ScaleMethyl/bin` then the workflow can be run without any `-profile` option.

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 **Error! Reference source not found.**).

- 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.
-

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. The sequencing data from the Scale Bio Methylation library
   Both paired-end reads and both index reads (R1, R2, and I1, I2) are required!

   - **Note that the indexing reads, while not a standard output for many core facilities or sequencing providers, are required for the ScaleBio pipeline to 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
   - A BSBolt genome index and associated files; See ScaleMethyl/docs/genomes.md at main · ScaleBio/ScaleMethyl (github.com)

3. Sample Barcode Table (samples.csv)
   - Listing which input samples where loaded into which wells of the ScaleBio Single Cell Methylation kit; See ScaleMethyl/docs/samplesCsv.md at main · ScaleBio/ScaleMethyl (github.com)

2.1. Sequencing Reads

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, the simplest option is to start the analysis workflow directly from the sequencer RunFolder [with the `–runFolder option`] (this is the sequencer output folder containing the *RunInfo.xml* file). In this case, the ScaleBio Methylation 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 [Read 1, Read 2, Index 1 (i7), Index 2 (i5)]

If the raw sequencer output is not available, or the ScaleBio Methylation library was multiplexed with other libraries during sequencing, the analysis can be started from FASTQ files [`--fastqDir`]. ScaleBio provides pre-made samplesheets for both small and large kit formats here.

- INDEX READS: Generating index reads in fastq format can be done using the *CreateFastqForIndexReads* option in Illumina *bcl-convert* OR by using ScaleBio provided samplesheets (here)
- Each ScaleMethyl library has 192 unique i7/i5 index combinations. Using ScaleBio samplesheets users should expect 192 X 4 fastq files (768 R1, R2, I1 and I2). Operating on a large number of files is optimal for ScaleMethyl and will result in decreased runtimes. However, concatenation of these files into a single 4 file faste set is also acceptable as

input in case users would like to store less files per run. Simply use “SampleID” column to contain only “ScaleMethyl”. Repeating names in the Sample_ID column is acceptable.

- Premade samplesheets are included in the ScaleMethyl pipeline and can be used for FASTQ generation with *bcl-convert*. Using these will create the FASTQ files with names as shown below for a full ScaleBio run. The resulting files will have the libName prefix, the plate ID and PCR well position.

```
ScaleMethyl_plate1-1A_L001_R1_001.fastq.gz
ScaleMethyl_plate1-1A_L001_R2_001.fastq.gz
ScaleMethyl_plate1-1A_L001_I1_001.fastq.gz
ScaleMethyl_plate1-1A_L001_I2_001.fastq.gz
```

- The ScaleMethyl i5/i7 Index Barcodes are ScaleMethyl library barcodes; they DO NOT contain sample level information. ScaleMethyl tagmentation barcodes are located in Read 2 and represent sample level data, which is demultiplexed within the ScaleMethyl pipeline. Index reads 1 and 2, (`*_I[1,2]_*.fastq.gz`) are required as they are one of the many levels of parallel indexing for cell barcode generation. The structure of the Methylation library is shown in *Appendix A: Methylation Library Structure and List of Barcode Sequences*.
- FASTQ GENERATION FOR LARGE KIT: We recommend using our samplesheets for demultiplexing in order to demultiplex all combinatorial indexes and per final pcr plate. Analysis of fastqs can be performed either on a subset (i.e. per plate) or for all plates. Each final pcr plate has a unique i5 index, and therefore subsequent methylation coverage files can be easily combined into one directory for subsequent analysis.

2.2. Reference Genome

The workflow uses a genome reference and a gene annotation for analysis. Reads are aligned to the full genome sequence, accounting for bisulfite-conversion, using bsbolt. Hence specifically a bsbolt (v1.5 or higher) genome index is required. Gene annotation is optionally used for methylation feature extraction and QC.

2.2.1. Pre-built Genomes

ScaleBio pre-built reference 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.

2.2.2. Creating a new Genome Index

To build a new BSBolt reference index for a different genome requires the genome sequence in fasta format. Then install bsbolt (v1.5 or later) and run:

```
bsbolt Index-G {genome.fasta} -DB {indexOutDir}
```

The analysis workflow uses a genome.json file, the specify the genome index location and other related parameters. To use a new reference genome, create a new json file with the path to the bsbolt index and annotation filse and specify for the workflow with the `--genome`. See Scale Reference Genomes for additional details on the `genome.json` file format.

2.2.3. Creating genome bed files with a custom genome

For every customer genome, the following files need to be generated:

[1] 100kb non-overlapping windows for CG; genomeTiles
[2] 250kb non-overlapping windows for CH; genomeTilesCh
[3] Transcription Start Sites (TSS); tssWin
[4] 1kb upstream of TSS; backgroundWin
[5] bsbolt_chrs; list of chromosomes to filter

The chrom.sizes file is required for generating [1] and [2]. The species chromosome sizes are listed in the chrom.sizes file and can be used as input for bedtools.

```
bedtools makeWindows –g chrom.sizes -w 100000 > my.genome.100kb.CG.bed
```

The purpose of the [3] tssWin and [4] backgrounWin is to assess the relative enrichment for read coverage at TSS regions, with the expectation that less than 1% of reads align to TSS regions. The TSS of genes can be obtained from the annotation (.gtf) file. The only data required for extraction is which chromosome and the gene start position. From this, the tssWin and backgroundWin can be generated. Here is an example of how to generate TSS regions:

```
grep "\tgene\t" my-custom.gtf > genes.bed
cut -f1,4,5 genes.bed > gene-regions.bed
awk -F $'\t' 'BEGIN { OFS=FS } { print $1,$2-100,$2+100 }' gene-regions.bed >
tssWin.bed
awk -F $'\t' 'BEGIN { OFS=FS } { print $1,$2-1100,$2-900 }' gene-regions.bed >
backgroundWin.bed
```

For the chromosome filter list [5] list the chromosome name followed by either “filter” or “mito” in tab-separated format.

2.3. Sample Barcode Table

The ScaleMethyl library is pooled and needs to be further separated into sample specific files for analysis. A sample barcode table (e.g. samples.csv) file is used to list the samples included in an analysis run, their sample barcode (Tagmentation) 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. Please see below the minimum requirements:

Table 2: Contents of the sample barcode table entries

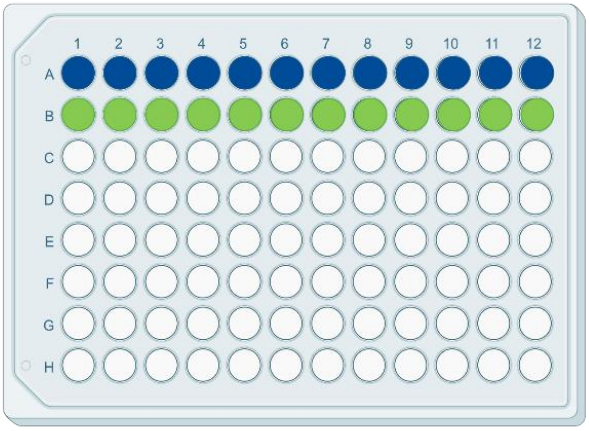
sample	barcodes	libName
Enter your desired sample name here	Enter the ScaleMethyl Tagmentation well IDs for the sample row	Fastq file prefix in order for ScaleMethyl to identify fastq files

Notes on the “barcodes” column used for demultiplexing samples

During analysis, sequencing data is first converted into library FASTQ files (libName column). When multiple samples are included in one sequencing library, the pipeline will further demultiplex based on the sample (Tagmentation) barcodes. These sample barcodes are defined by the wells on the RT Barcode Plate, for example see the table below.

Table 3: Example sample barcode table with 2 samples

sample	barcodes	libName
Sample1	1A01-1A12	ScaleMethyl
Sample2	1B01-1B12	ScaleMethyl



Rules to abide by for sample naming

- "sample" and "libName" should consist ONLY of letters, numbers, dash (-) and dot (.).
- *Underscores are not valid*
- "barcodes" is required if more than one sample is processed on the ScaleMethyl tagmentation plate
- When running from pre-existing FASTQ file input, "libName" should match the first part of the FASTQ file name for this sample, e.g.: `ScaleMethyl` for `ScaleMethyl_*.fastq.gz`.

Rules to abide by for sample barcodes

1. The Tagmentation wells used for each sample are given as either:
   - An individual value: 1A01
   - A range of wells: 1A01-1H12
   - A list of values or ranges, separated by semicolon (;): 1A01-1A12;1E01-1E12
2. All ranges are read in row-wise order,
   a. e.g. (1A01-1A12) refers to the first row of the tagmentation plate
   b. e.g. (1A01-1A06; 1B01-1B06; 1C01-1C06; 1D01-1D06) referse to the first 6 columns of plate 1, rows A-D
3. Samples plated by columns should input each Tagmentation well separated by a semi-colon
   a. e.g. Tag Plate 1 - Column 1 = 1A01;1B01; 1C01;1D01; 1E01;1F01; 1G01;1H01
   b. e.g. Tag Plate 2 - Column 5 = 2A05;2B05; 2C05;2D05; 2E05;2F05; 2G05;2H05

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. All raw read coverage and methylation are single nucleotide and single cell resolution, however there are outputs where either methylation coverage is accumulated into large genomic bins, and when metrics an average over a particle sample or the entire library.

1) ScaleMethyl Library fastq generation
    a. Using Illumina bcl-convert
    b. Using index reads (i5/i7 indexes)
    c. No sample level demultiplexing
2) ScaleMethyl Sample demultiplexxing
    a. Using ScaleBio *bcParser* tool
    b. Based on tagmentation barcodes
    c. Tagmentation wells grouped into samples based on sample barcode table
3) Read QC
    a. fastQC and barcode passing rate
    b. Read trimmimg and adapter removal
4) Whole-Genome Alignment
    a. Bisulfite aware alignment using bsbolt
5) Duplicate removal
    a. Removing PCR duplicate fragments based on cell-barcode and mapping position
6) Methylation data extraction
    a. Determine (un-)methylated read-counts per cell and genome position
7) Per-cell Methylation signal output
    a. Generate per-cell methylation output in various formats for downstream analysis
8) Methylation matrix file generation
    a. Generate genome-bin by cell methylation matrix for clustering, etc.
9) QC (Library and Sample) Report generation

3.1. FASTQ generation

- This is performed only if a Sequencer RunFolder (BCL files) is provided as input, rather than FASTQ files that were pre-generated using Illumina *bcl-convert*.
- A samplesheet.csv in Illumina samplesheet v1 format is automatically generated and Illumina bcl-convert is run to generate fastq files
    - All necessary FASTQ files are generated, including the reads 1 and 2, i7 and i5 index reads (R1, R2, I1 and I2, respectively)
    - Fastq files are **not yet** demultiplexed into the individual samples loaded into the ScaleBio Methylation assay at this point
    - Reads not matching the expected ScaleBio Met i5 Index Barcode or Met i7 Index Barcode sequences are filtered (into the *Undetermined* FASTQ files).
- If --splitFastq is enabled, a separate set of FASTQ file is produced for each i5 and i7 index combination, resulting in multiple files, that can be processed in parallel for decreased overall runtime
- 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, i5 and i7 Index Barcodes) are extracted from the reads and error corrected using ScaleBio *bcParser* tool.
- 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.

3.3. Sample Demultiplexing

- Samples loaded into different wells of the tagmentation plate during the ScaleBio Single Cell Methylation workflow are separated at this point for further analysis.
- Implemented in *bcParser* together with Barcode Parsing (see above).
- The assignment of tagmentation plate wells to samples is done in the sample barcode table (samples.csv; see Chapter 2 for more information).

3.4. Read Trimming

- The previous step removed barcode and constant sequences from the reads. Next cutadapt is used to remove any occurance of the sequencing adapters from the 3' end of read 1 and 2
    - CTATCTCTTATA; AGATCGGAAGAGC
- Additionally 10bp are trimmed from the end of read2 corresponding to possible adapter locations.
- Read pairs with reads shorter than 20 bp after trimming are discarded.

3.5. Genome Alignment

- BiSuflite Bolt (BSBolt) is used to align the trimmed and filtered reads to the genome.
- BSBolt performs local, paired-end sequence alignment against the original and in-silico converted reference genome using a modified BWA version internally.
- Unmapped reads are discarded
- Optionally outputs a sorted BAM file with alignments for custom downstream analysis or visualization [*--bamOut*]

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, such that reads from different cells are never 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 cell calling in the ScaleMethyl workflow is based on the number of aligned, deduplicated reads per cell-barcode.
- The cell barcode corresponds to the combination of the Tagmentation Barcode and the Met i5 Index Barcode and Met i7 Index Barcode.
- In the first pass, all cell-barcodes with less than a fixed threshold [--minUniqCount, default 1000] are excluded from further analysis.
- Later, the final 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 signal extraction

- Every genomic position where an aligned and deduplicated read overlaps a cytosine ('C') base corresponds to a methylation call (converted or unconverted, i.e. unmethylated or methylated).
    - This information is stored 'XB' BAM tags, see BSBolt/docs/alignment_output.md at master · NuttyLogic/BSBolt (github.com)
- Methylation calls are grouped by the sequence context of the cytosine, into either 'CG' or 'CH', the latter representing all other contexts (CA, CT, CC)
    - 'CH' methylation is low in most cell types, but relevant e.g. in brain samples
    - CG sites are relatively rare in the genome compared to other dinucleotides and hence the CG methylation files are much smaller than CH, making CG-only analysis faster.
- All methylation calls for each potential cell are extracted from the BAM file and stored in a collection of parquet files sorted by genomic position
    - For each position to total number of converted and unconvered read-counts is stored
- This same information can optionally be output in a variety of related, but different formats for different downstream tools
    - H5 for amethyst [--amethystOut true]
    - .allc for allcools [--allcOut true]
    - .cov (bismark-like format) for Methscan and others. [--covOut true]

3.9. Generation of Matrix

- The per-cell methylation files above arethe primary output of the workflow, representing the full, basepair resolution single-cell methylation data. However due to the single-basepair resolution, the files are very large and sparse for ease of initial analyses, such as visualizations and clustering, the workflow also generates matrix files with a methylation score per cell and genomic region (bin).
    - This lower resolution data can be used directly as input to e.g. PCA and then UMAP and standard single-cell clustering methods.
- These output files are dense matrices (.mtx file format) where columns represent single cells, and rows are non-overlapping genomic bins from `genomeTiles` in the genome.json.
    - Separate matrices are created for CG and CH context methylation.

- The matrix files (mtx format) can be read into Amethyst or general purpose single-cell tools such as Seurat or ScanPy.
- Martrix generation can be turned off using --`matrixGenerationCG false` --`matrixGenerationCH false` [default true for both options].
    - CH methylation matrices are only relevant for a subset of sample-types and disabling them will save time and storage spacer.
- 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 measures the relative number of reads (depths of coverage) around Transcript Start Sites comapred to the rest of the genome. It is as a QC metric to assess nucleosome disruption during the library preparation, with expected values around or below 1.
- TSS enrichment is reported in the workflow as the ratio between reads in the 200 bp centered on the TSS and 200bp control windows 1kb upstream (-) from TSSs
    - 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 methylation metrics.
- Includes distribution of Targmentation 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

Key output files (for information on all output files, see github documentation here):

Directory	File	Description
report		QC reports and statistics
report/sample_reports/ <sample>	<sample>.report.html	An interactive standalone HTML report including key metrics/figures for each sample
	csv/<sample>.*.csv	Additional sample metrics in csv format
report/library_report/ <library>	library.<libName> .report.html	Barcode summary and demultiplexing statistics for the whole library (potentially multiple samples)
	csv/<libName>.combine dPassingCellStats.csv	Key metrics for passing cells for the whole library
samples		Single-cell methylation outputs
	<sample>.allCells.csv	Metrics per cell-barcode, including barcodes / well positions
samples/genome_bin_matrix	<sample>.{CG,CH}.{score }.mtx.gz	CG and CH binned genome-wide matrix files in Matrix Market format
samples/methylation_coverage	<format>/<sample>/ {CG,CH}/<barcode>.*	Per-cell methylation calls in bismark .cov, .allc or amethyst .h5 format

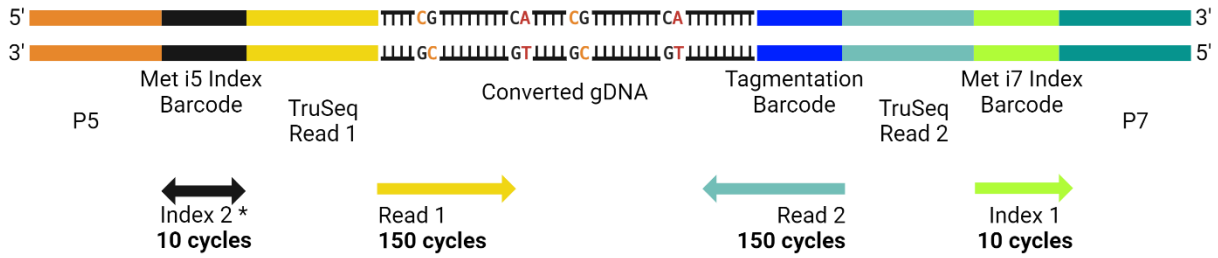
Please see distributed downstream tutorials on how to use these output files in downstream analysis with various software tools. Additionally, sample reports have been linked below:

- 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 1.

Figure 1: 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.
Rev B	Oct 2024	1020783	Methylation v1.1 release.