



EWAS Project

Design of a Bioinformatics Workflow for DNA Methylation Analysis

Philippine Genome Center Internship Project.

Introduction

MethylSeq is a proprietary application developed by Illumina which analyzes whole genome and targeted bisulfite DNA sequences to determine methylation patterns from sequencing data. The application uses Bowtie2 (v2.2.2) and Bismark (v0.12.2) for alignment, deduplication, and methyl calling of reads. In this internship project, the MethylSeq workflow was reconstructed using the same tools for alignment and methylation extraction and other tools for quality control and report generation.

Documentation contents

The first part of the documentation covers some background material to get started, for those trying to replicate the project. The second involves the more technical parts of the project, for those who plan to use it in their own workflows.

Part I. Getting Started

- Epigenetics and EWAS
- Programming Concepts
- Tools
- Sample Data
- SLURM Directives

Part II. Scripts

- Retrieval of Reads
- Quality Control
- Methylation Analysis
- Report Generation

Part I. Getting Started

This section contains information that may be useful in replicating or adding onto the methylation analysis project.


Epigenetics and EWAS

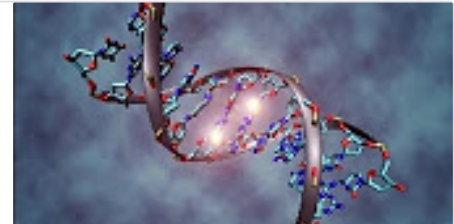
Epigenetics is the study of how traits are inherited across generations without changes to the DNA sequence while epigenome-wide association studies measure the association between epigenetic variation and a phenotype of interest. Below are lectures and papers on these topics.

Videos

Epigenetics Playlist

Share your videos with friends, family, and the world

 <https://www.youtube.com/playlist?list=PL89u8Q5z-cEPjqzqWmdwsgoHgPbQq50QBa>



Recommended videos:

1. [Genetics, epigenetics and disease](#): definitions of epigenetics, dinucleotide CG as a genomic signalling module
2. [Introduction to Epigenome Wide Association Studies \(EWAS\)](#): steps in performing epigenome-wide association studies

Papers

1. Flanagan J. M. (2015). Epigenome-wide association studies (EWAS): past, present, and future. *Methods in Molecular Biology*, 1238, 51–63. https://doi.org/10.1007/978-1-4939-1804-1_3
[Link to paper]
2. Michels, K. B., Binder, A. M., Dedeurwaerder, S., Epstein, C. B., Greally, J. M., Gut, I., Houseman, E. A., Izzi, B., Kelsey, K. T., Meissner, A., Milosavljevic, A., Siegmund, K. D., Bock, C., & Irizarry, R. A. (2013). Recommendations for the design and analysis of epigenome-wide association studies. *Nature Methods*, 10(10), 949–955. <https://doi.org/10.1038/nmeth.2632>
[Link to paper]
3. Birney, E., Smith, G. D., & Greally, J. M. (2016). Epigenome-wide association studies and the interpretation of disease-omics. *PLoS Genetics*, 12(6), e1006105. <https://doi.org/10.1371/journal.pgen.1006105>
[Link to paper]

4. Castro de Moura, M., Davalos, V., Planas-Serra, L., Alvarez-Errico, D., Arribas, C., Ruiz, M., Aguilera-Albesa, S., Troya, J., Valencia-Ramos, J., Vélez-Santamaria, V., Rodríguez-Palmero, A., Villar-Garcia, J., Horcajada, J. P., Albu, S., Casasnovas, C., Rull, A., Reverte, L., Dietl, B., Dalmau, D., Arranz, M. J., ... Esteller, M. (2021). **Epigenome-wide association study of COVID-19 severity with respiratory failure**. EBioMedicine, 66, 103339.
<https://doi.org/10.1016/j.ebiom.2021.103339>
[\[Link to paper\]](#)

Programming Concepts

This project requires you to have some knowledge in **command line**, **Python**, and **Shell scripting**. The following are some recommendations on where to start on each of these topics.

Command Line

To use the command line interface, the terminal launcher is used. MacOS and Linux users have a terminal installed in their systems, in Windows you may use PowerShell or the Windows Subsystem for Linux (WSL). To learn more about a command, enter `man [command]` and a user manual of the given command will be returned. Below are common commands used.

Command	Description
<code>ls</code>	lists directories and files
<code>cd [dir]</code>	changes present working directory
<code>pwd</code>	shows full path of present working directory
<code>touch [file]</code>	creates a new file
<code>cp [file] [new_file]</code>	copies contents of <code>[file]</code> to <code>[new_file]</code>
<code>mv [file] [dir]</code>	moves contents of <code>[file]</code> to directory <code>[dir]</code> , if <code>[dir]</code> does not exist, renames <code>[file]</code>
<code>rm [file]</code>	removes a file
<code>mkdir [new_dir]</code>	creates new directory
<code>rmdir [dir]</code>	removes directory
<code>[command] > [file]</code>	saves output from a given command to <code>[file]</code>
<code>[command] >> [file]</code>	appends output from a given command to <code>[file]</code>
<code>less [file]</code>	view file contents
<code>wget [URL]</code>	download file from a given URL
<code>scp</code>	copies files securely between remote servers

Links to more references:

1. [Basic UNIX commands](#)

2. [Terminal commands with sample screenshots](#)
3. [UNIX command-line cheat sheet](#)

Python

There are plenty of resources on how to get started with Python such as [Codecademy](#), [Coursera](#), and a lot of YouTube videos. It is suggested, however, to read through [Automate the Boring Stuff with Python](#) as this project heavily focuses on automating workflows. For additional knowledge on how to apply Python (and problem solving in general) in bioinformatics, you may check [Rosalind](#).

Shell Scripting

Below is a short tutorial on shell scripting and automation.

Shell Scripting Crash Course - Beginner Level

This is an intro to shell scripting with Bash. We will learn what shell scripting is and create a cheat sheet with things like variables, conditionals, loops...

 <https://www.youtube.com/watch?v=v-F3Yld6oMw>



#!/bin/bash Shell Scripting
Crash Course
Beginner Level

[\[Supplementary material to the video\]](#)

Tools

MethylSeq

The project is based on the MethylSeq workflow by Illumina. The tool uses the following modules:

- Bismark (v0.12.2): for methylation calling
- Bowtie2 (v2.2.2): for aligning sequencing reads
- SAMtools (v0.1.19-isis-1.0.3): for post-processing read alignments

By using a bisulfite-converted reference genome and FASTQ reads as inputs, it produces the following outputs:

- BAM files: processed reads after alignment
- Cytosine report: cytosine methylation states in the genome
- Bismark processing report: generated by the Bismark tool ([sample](#))
- bedGraph file: cytosine methylation states for only the cytosines with sequencing coverage

The manual is found in this [link](#).

https://support.illumina.com/content/dam/illumina-support/documents/documentation/software_documentation/basespace/methylseq-app-guide-15069929-a.pdf

Fastp

Fastp is a tool used for quality control and for processing FastQ files. The tool filters out low quality reads, trims adapters, reduces overrepresented sequences, and returns an HTML processing report. Documentation is found in this [link](#).

A sample report is found [here](#).

fastp report at 2018-06-27 10:32:27

```
fastp -i /Users/shifu/data/fq/S010_20170320003-4_ffpedna_pan-cancer-v1_S10_R1_001.fastq -l
/Users/shifu/data/fq/S010_20170320003-4_ffpedna_pan-cancer-v1_S10_R2_001.fastq -o testdata/big.R1.fq -O
testdata/big.R2.fq -p fastp 0.17.0, at 2018-06-27 10:32:27
https://opengene.org/fastp/fastp.html
```


Bismark

Bismark maps bisulfite-converted reads to a reference genome and analyzes their methylation status. This is the core tool used in this project.

Steps performed in Bismark:

1. `bismark_genome_preparation` : a genome of interest is first downloaded before running this step
2. `bismark` : processing and alignment of reads
3. `deduplicate_bismark` : remove duplicate reads
4. `bismark_methylation_extractor` : execute methylation calls and analysis
5. `bismark2report` : produce a processing report
6. `bismark2summary` : produce a summary report for all reads processed


A guided tutorial is found in the Training Section of the Babraham Bioinformatics Group (<https://www.bioinformatics.babraham.ac.uk/training.html>).

 [https://www.bioinformatics.babraham.ac.uk/training/Methylation_Course/Basic%20BS-Seq%20processing%20Exercises.p](https://www.bioinformatics.babraham.ac.uk/training/Methylation_Course/Basic%20BS-Seq%20processing%20Exercises.pdf)
df

Documentation for Bismark is found [here](#).

Bismark/Docs at master · FelixKrueger/Bismark

This User Guide outlines the Bismark suite of tools and gives more details for each individual step. For troubleshooting some of the more commonly experienced problems in sequencing in general and bisulfite-sequencing in

 <https://github.com/FelixKrueger/Bismark/tree/master/Docs>

FelixKrueger/
Bismark

A tool to map bisulfite converted sequence reads and determine cytosine methylation states

 10 Contributors  11 Issues  248 Stars  77 Forks



Overleaf

Overleaf is an online LaTeX editor used to format technical reports. There are other open-source LaTeX editors like [Texmaker](#) and [TeXstudio](#) that can be used for the project since the LaTeX editor is used for testing the automation scripts only.

To get familiar with the commands, these are some links to LaTeX tutorials:

1. <https://latex-tutorial.com>
2. <https://wch.github.io/latexsheet/>
3. LaTeX using Overleaf Introduction ([Video](#))

Sample Data

Other than the provided datasets from the Fastp and Bismark documentation, data can be found from the National Center for Biotechnology Information (NCBI).

As an example, data used for the project can be found in this [paper](#) by Sureshchandra et al. (2021).

Phenotypic and Epigenetic Adaptations of Cord Blood CD4+ T Cells to Maternal Obesity

Pregavid obesity has been shown to disrupt the development of the offspring's immune system and increase susceptibility to infection. While the mechanisms underlying the impact of maternal obesity on fetal myeloid cells are emerging, the consequences for T cells remain poorly defined.

 <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8071865/>



To retrieve sequences from NCBI, watch this video below.

NCBI Minute: How to Quickly Retrieve Sequences from NCBI

Presented February 14, 2018. This NCBI Minute will show you how to quickly grab a protein or nucleotide sequence in FASTA or another format from NCBI using th...

 <https://www.youtube.com/watch?v=Ahrx9JsaIU>

NCBI Minute

How to Quickly
Retrieve Sequences
from NCBI



SLURM Directives

In the project, the scripts that use the high-performance computing (HPC) server have header lines that look like this:

```
1 ##--Resource Allocation--##
2 #SBATCH --job-name=job_name
3 #SBATCH --account=my_account
4 #SBATCH --qos=shortjobs
5 #SBATCH --cpus-per-task=8
6 #SBATCH --mem=20G
7 #SBATCH --out=%x.out
8 #SBATCH --err=%x.err
```

The lines are SLURM directives, with more information found below. These are important in properly accessing the HPC since the server handles a complex queue of tasks from different people in different locations. These directives help the server assess which tasks need to be prioritized in the queue based on memory needed, task length, and the group executing the task.

Slurm Workload Manager


Section: Slurm Commands (1) Updated: Slurm Commands Index sbatch - Submit a batch script to Slurm. Option(s) define multiple jobs in a co-scheduled heterogeneous job. For more details about heterogeneous jobs see the document https://slurm.schedmd.com/heterogeneous_jobs.html sbatch submits a batch script to Slurm.

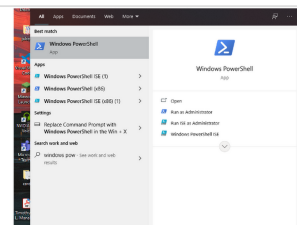
 <https://slurm.schedmd.com/sbatch.html>

Here are notes compiled by 2021 interns on how to access the HPC server:

Server access notes

Outline Generating RSA keys Accessing the server Trying things out Running tools in HPC Before generating our keys, we first need to open the terminal. For Windows users Open Windows Powershell (type it in Windows key) For Mac

 <https://docs.google.com/document/d/1deGTzSyKrEh-NSXOg-gQQ1HVqdP-GJZW1DynJvQQII6A/edit>



Part II. Scripts

The scripts are divided into four directories: `00-get_reads`, `01-fastp`, `02-bismark`, and `03-reports`.



Retrieval of Reads

Found in the `00-get_reads` directory. This is an **optional step** in case the reads are not available locally.

Inside this folder, we have three scripts: `00-wget.sh` for downloading raw sequencing data from the Sequence Read Archive (SRA), `01-extract-reads.sh` for converting the raw data to processed reads, and `get-reads.sh` as a wrapper script for the two previous scripts.

00-wget.sh

- This script downloads reads from SRA.
- Input: Plain text file (TXT) containing the **directory** where the reads will be saved and **links** where the reads will be downloaded from.
- Output: Unprocessed reads retrieved from SRA.

How to use:

1. Run the following command:

```
./00-wget.sh <.txt_file_of_reads>
```

Example: `./00-wget.sh 00-get_reads_-_sample.txt`

2. The .txt file should contain:
 - a. Line 1: Directory where the reads will be placed. Directory must exist for the script to work properly.
 - b. Lines 2-onwards: Download links of the reads.

`00-get_reads_-_sample.txt`

```
1 ./data/pregavid_obesity
2 https://sra-downloadb.be-md.ncbi.nlm.nih.gov/sos3/sra-pub-run-20/SRR13377238/SRR13377238.1
3 https://sra-downloadb.be-md.ncbi.nlm.nih.gov/sos3/sra-pub-run-21/SRR13377239/SRR13377239.1
4 https://sra-downloadb.be-md.ncbi.nlm.nih.gov/sos3/sra-pub-run-20/SRR13377240/SRR13377240.1
5 https://sra-downloadb.be-md.ncbi.nlm.nih.gov/sos3/sra-pub-run-19/SRR13377241/SRR13377241.1
6 https://sra-downloadb.be-md.ncbi.nlm.nih.gov/sos3/sra-pub-run-21/SRR13377242/SRR13377242.1
```

01-extract_reads.sh

- This script uses the [SRA-Toolkit](#) to convert unprocessed SRA data into their FASTQ format.
- Input: One line of text containing the directory where the unprocessed SRA reads are located.
- Output: Directory named `paired_reads`. Inside the directory, for each unprocessed SRA file (`<read_identifier>`), `<read_identifier>_1.fastq` and `<read_identifier>_2.fastq` can be found.

How to use:

1. Run the following command:

```
./01-extract_reads.sh <reads_directory>
```

Example: `./01-extract_reads.sh /location/of/reads`

2. `<reads_directory>` should exist and must contain unprocessed SRA reads.
3. After execution, the output directory `paired_reads` is found in `<reads_directory>`.
4. If the script is not executed in the PGC HPC, kindly pull an image of SRA-Toolkit in Docker Hub (recommended: [inutano/sra-toolkit](#)). In `01-extract_reads.sh`, replace `hpc` on the line below to `docker run` (or how images are run locally in your computer):

```
hpc inutano/sra-toolkit fasterq-dump "$rds"
```

get_reads.sh

- This is a wrapper script for `00-wget.sh` and `01-extract-reads.sh`. The script exists in case both the downloading and conversion of reads have yet to be done.
- Input: Plain text file (TXT) containing the **directory** where the reads will be saved and **links** where the reads will be downloaded from.
- Output: Directory containing unprocessed reads retrieved from SRA and their respective conversions to FASTQ format.
- In `get_reads.sh`, the input is similar to the input of `00-wget.sh`, while the output is the combination of outputs from `00-wget.sh` and `01-extract-reads.sh`.

How to use:

1. Run the following command:

```
./get_reads.sh <.txt_file_of_reads>
```

Example: `./get_reads.sh 00-get_reads_-_sample.txt`

2. The `.txt` file should contain:
 - a. Line 1: Directory where the reads will be placed. Directory must exist for the script to work

properly.

b. Lines 2-onwards: Download links of the reads.

Quality Control

The reads are filtered and processed using Fastp, a FASTQ pre-processing tool. Its script, `fastp_script.sh`, is found in the `01-fastp` directory.

fastp_script.sh

- Input: For each SRA file (`<read_identifier>`), `<read_identifier>_1.fastq` and `<read_identifier>_2.fastq` . See `01-extract_reads.sh` output as an example.
- Output: Reports in HTML and JSON format, binary output files (`.cleaned`). These files are found in the `html` , `json` , and `output_files` folders under the present working directory.

How to use:

1. Change the following variables in the script, if needed:
 - a. `READS_LOCATION` : location of reads processed by the SRA-Toolkit.
 - b. `FASTP_LOCATION` : command of executable Fastp tool. In the PGC HPC, it is `hpc pgcbioinfo` .
 - c. `EXT` : filename extension of input reads (e.g. `.fq` , `.fastq` , `.fastq.gz`).
 - d. `R1_ID` and `R2_ID` : identifiers for read1 and read2 respectively (e.g. `_1` , `_2`).
2. Change the filter parameters using the line below in the Fastp script, according to what is necessary in the project. See the Fastp documentation for more examples.

```
$FASTP_LOCATION/fastp --qualified_quality_phred 30 --length_required 50 \  
--in1 $in1 --in2 $in2 --out1 $out1 --out2 $out2 --html $result \  
--json $result_json
```

3. A step-by-step explanation is found in the bottom half of the script, commented out.
-

Methylation Analysis

Methylation patterns are processed using Bismark, its scripts found in the `02-bismark` directory. The version used in this project is v0.23.1 and its image is pulled from biocontainers (link).

00-main.sh

- This is a wrapper script for all the steps using Bismark. Lines containing commands can be commented out in the code especially if those steps have been executed previously (e.g. if

reference genome has been prepared for bisulfite alignments, there is no need to execute

```
./01-genome_preparation.sh
```

- Input: Directory containing reference genome and directory containing reads.
- Output: HTML and plaintext reports generated by Bismark, BAM output files.
- Parameters: `-g / --genome` to indicate path for genome directory, `r / --reads` for reads directory, `-p / --paired` to indicate that the reads are paired-end

How to use:

```
./00-main.sh -g <genome_folder> -r <reads_folder> -p # for paired-end reads  
./00-main.sh -g <genome_folder> -r <reads_folder> # for single-end reads  
sbatch 00-main.sh -g <genome_folder> -r <reads_folder> -p # run in slurm
```

01-genome_preparation.sh

- The script calls the `bismark_genome_preparation` step from Bismark.
- Input: Directory containing a reference genome in `.fa` or `.fasta`
- Output:

How to use:

```
./01-genome_preparation.sh <path_to_genome_folder>
```

From the Bismark documentation, it's necessary to also place the path to where Bowtie2 is downloaded. Since the Bismark container pulled from [biocontainers](#) already has Bowtie2 packaged in it, there is no need to direct a path to Bowtie2.

02-bismark.sh

- The script calls the `bismark` step.
- Input: Directory containing reference genome and path to reads.
- Output: BAM file containing all alignments and a plaintext report on alignment and methylation calls on single-end reads (`_SE_report.txt`).

How to use:

```
./02-bismark.sh <genome_folder> <read1>
```

02-paired_bismark.sh

- The script calls the `bismark` step having paired reads as input.
- Input: Directory containing the reference genome, paths to reads 1 and 2.
- Output: BAM file containing all alignments and a plaintext report on alignment and methylation calls on paired-end reads (`_PE_report.txt`).

How to use:

```
./02-paired_bismark.sh <genome_directory> <read1> <read2>
```

03-deduplicate.sh

- Optional step recommended for whole-genome bisulfite samples. This script will remove duplicates in the produced BAM file by the `bismark` step.
- Input: BAM file produced by the `bismark` step, usually found in the directory where the `bismark` step was called.
- Output: Processed BAM file.

How to use:

```
./03-deduplicate.sh <.bam_file>
```

04-methylation_extractor.sh

- This script provides the methylation processing and splitting reports in plaintext.
- Input: Directory to genome folder and produced BAM file.
- Output: Plaintext methylation processing reports, bedGraph file, and Bismark coverage file (`bismark.cov`)

How to use:

```
./04-methylation_extractor.sh <genome_directory> <.bam_file>
```

05-bismark2report.sh

- This script creates an interactive HTML processing report for each of the reads processed in Bismark.
- It is required to run this script where the alignment, processing, and splitting reports are produced.

How to use:

```
./05-bismark2report.sh
```

06-bismark2summary.sh

- Similar to `05-bismark2report.sh`, this script also creates an interactive HTML report, but does this for all the reads processed in Bismark and summarizes the alignment and processing statistics in one file.
- It is required to run this script where the alignment, processing and splitting reports are produced.

How to use:

```
./06-bismark2summary.sh
```

Report generation

coverage_to_tex.py

- This script converts the plaintext coverage report produced by Bismark to a TeX file.
- Input: A plaintext `.bismark.cov` file (tab-delimited) and TeX file to be edited.
- Output: Formatted TeX file, which can be used in Overleaf and [Pandoc](#) to convert to PDF.

How to use:

```
./coverage_to_tex.py -h #to access help interface  
./coverage_to_tex.py -r <.bismark.cov_file> -o <.tex_file>  
./coverage_to_tex.py --report <.bismark.cov_file> --output <.tex_file> #long-form command
```

report_to_tex.py

- This script converts the HTML summary and processing reports of Bismark to a TeX file.
- Input: HTML file containing summary or processing report, and folder containing images of the reports.
- Output: Formatted TeX file, which can be used in Overleaf and Pandoc to convert to PDF.

How to use:

```
./report_to_tex.py -h # access help interface
./report_to_tex.py -r <HTML_report> -d <image_directory>
./report_to_tex.py --report <HTML_report> --directory <image_directory> #long-form command
```

report_conversion.py

- This is an alternate solution to `report_to_tex.py`, which saves the Plotly graphs as static images.
- Input: HTML summary or processing report and filename of PDF output.
- Output: PDF version of the Bismark summary or processing report.

How to use:

```
./report_conversion.py -h #access help interface
./report_conversion.py -r <HTML_report> -o <filename_of_output>
./report_conversion.py --report <HTML_report> --output <filename_of_output> #long-form command
```

Recommendations

Moving forward, here are possible tasks future interns could use for their projects:

1. Generate better reports from Bismark HTML processing and summary reports
 - a. Problem encountered: unable to automate extraction of plotly graphs from HTML.
 - i. This is possible if the processing and summary graphs were first generated in Python or JS (and not through `bismark2report` or `bismark2summary`) like so:
 - ii. <https://plotly.com/javascript/static-image-export/> and <https://plotly.com/python/static-image-export/>
 - b. How to potentially solve this: revise bismark2report and bismark2summary scripts and call functions from 1.a.ii.
 - i. Script for bismark2report:
<https://github.com/FelixKrueger/Bismark/blob/master/bismark2report>
 - ii. Script for bismark2summary:
<https://github.com/FelixKrueger/Bismark/blob/master/bismark2summary>
2. Use a faster alternative to Bismark: BiSulfite Bolt.
 - a. Link to paper: <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8106542/>