DNA Methylation Data Analysis

Leila Taher

WS 2022/2023

Contents

1 Introduction

The aim of this tutorial is to introduce you to a simple workflow to analyze bisulfite-sequencing data.

1.1 Tools

- FastQC
- Bismark
- Samtools
- deepTools
- IGV

Bismark is a program to map bisulfite treated sequencing reads to a genome of interest and perform methylation calls in a single step. Bismark needs a working version of Perl and it is run from the command line. Furthermore, it requires Bowtie or Bowtie 2 to be installed on your computer. Install the tool(s) if necessary.

1.2 Data

Sample	FASTQ file
H1 (chr21, forward reads)	ENCFF857QML_R1.fastq.gz
H1 (chr21, reverse reads)	ENCFF857QML_R2.fastq.gz

These sequencing data were generated by the laboratory of Joe Ecker for the Roadmap Epigenomics Project. You can find the original data here. The laboratory performed bisulfite treatment followed by sequencing (whole-genome bisulfite sequencing, WGBS) on the H1 human embryonic stem cell line. The sequencing libraries were prepared in a directional manner. The data set with which you will be working contains only sequencing reads from chromosome 21.

2 Quality control with FastQC

We will use FastQC to generate a report about the quality of our sequencing reads. FastQC is a java application designed to spot potential problems in high throughput sequencing data sets. Create a folder to store the output and run FastQC for each .fastq file. FastQC will generate two files for each input file, one compressed, and one not. Examine the .html files.

- In particular, have a close look at the “Per base sequence content”. Note the GC distribution and percentage of “T” and “C”. Is this what you expected?

3 Mapping with Bismark

3.1 Preparing the genome

First you need to download the sequence of the human reference genome assembly.

- Download the soft-masked sequence for chromosome 21 of the GRCh38 assembly of the human genome from the Ensembl website (Ensembl release 104, `http://ftp.ensembl.org/pub/release-104/fasta/homo_sapiens/dna/`). Create a folder called “GRCh38” and store the (uncompressed) file in the folder.

Bismark supports reference genome sequence files in FASTA format; allowed file extensions are either either .fa or .fasta.

To prepare the genome for mapping, type:

```
bismark_genome_preparation -verbose GRCh38
```

- Note that you may need to include the complete paths to the tool and input files.

This command will create a subfolder called `Bisulfite_Genome` in the folder where the .fasta file is. The folder contains two subfolders: `CT_conversion` and `GA_conversion`. Each of these subfolders has a .fa file and its corresponding Bowtie2 index.

- What is the content of the .fa files?
- Compute the percentage of “As”, “Cs”, “Gs”, and “Ts” for each of the .fa files. For this purpose, you can use the `wc` command together with `grep` command with the `-o` flag. `-o` stands for `-only-matching` and will print each match (non-empty) in an input line on a separate output line. For example:

```
for i in A C G T ; do
        grep -o $i genome_mfa.GA_conversion.fa | wc -l;
done
```

- In addition to the sequence of the converted genomes, you will find the Bowtie/Bowtie 2 indexes. Bowtie indexes the reference genome using a scheme based on the Burrows-Wheeler transform (BWT) and the FM index. This index permits fast substring queries while keeping a relatively small memory footprint.

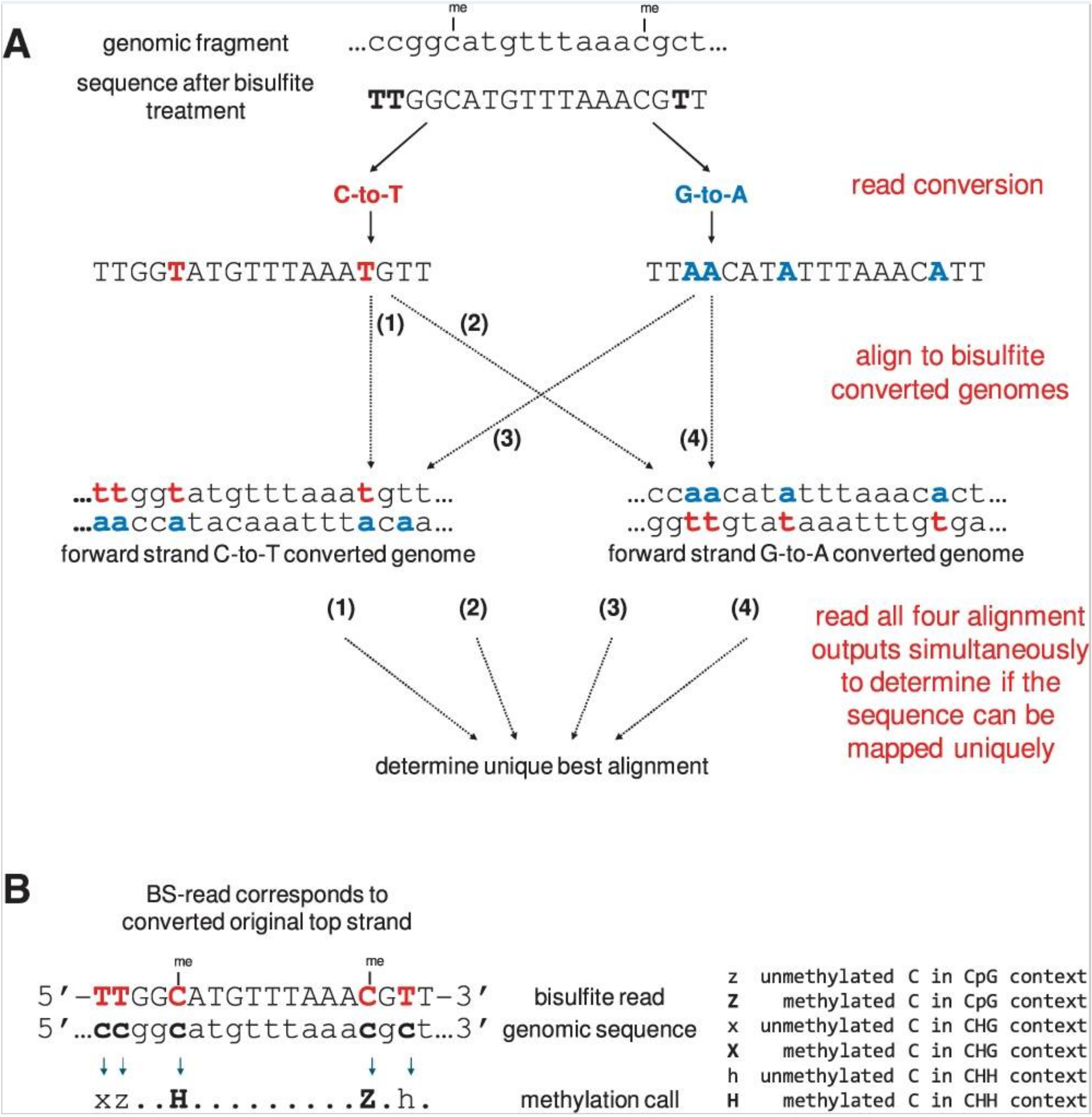
A
genomic fragment
...ccggcatgtttaaacgct...
me
me
sequence after bisulfite treatment
TTGGCATGTTTAAACGTT
C-to-T
G-to-A
read conversion
TTGGTATGTTTAAATGTT
TTAACATATTTAAACATT
(1)
(2)
(3)
(4)
align to bisulfite converted genomes
...ttggtatgtttaaatgtt...
...aaccatacaaatttacaa...
forward strand C-to-T converted genome
...ccaacatatttaaacact...
...ggttgtataaatttgtga...
forward strand G-to-A converted genome
(1)
(2)
(3)
(4)
read all four alignment outputs simultaneously to determine if the sequence can be mapped uniquely
determine unique best alignment
B
BS-read corresponds to converted original top strand
me
me
5'-TTGGCATGTTTAAACGTT-3'
bisulfite read
5'...ccggcatgtttaaacgct...3'
genomic sequence
xz..H.........Z.h.
methylation call
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

Figure 1: Bismark's approach to bisulfite mapping and methylation calling. (A) Reads from a BS-Seq experiment are converted into a C-to-T and a G-to-A version and are then aligned to equivalently converted versions of the reference genome. A unique best alignment is then determined from the four parallel alignment processes [in this example, the best alignment has no mismatches and comes from thread (1)]. (B) The methylation state of positions involving cytosines is determined by comparing the read sequence with the corresponding genomic sequence. Depending on the strand a read mapped against this can involve looking for C-to-T (as shown here) or G-to-A substitutions. Source

3.2 Running Bismark

We will now process and map the reads using Bismark. (Running this command will take a while!)

```
bismark -n 1 ./GRCh38/ -1 H1_chr21_R1.fastq.gz -2 H1_chr21_R2.fastq.gz
```

- `./GRCh38/` specifies the paths to the reference assembly to be used.
- `-n 1` defines the maximum number of mismatches permitted in the seed.
- `-1 H1_chr21_R1.fastq` specifies the location of the file with the forward reads.
- `-2 H1_chr21_R2.fastq` specifies the location of the file with the reverse reads.

For more details, please refer to the Bismark user guide and documentation.
Bismark will produce two output files:

- A .bam file with the mappings and methylation calls.
- A report file called `*_bismark_PE_report.txt`.

The report file contains information about the following:

- Summary of alignment parameters used.
- Number of sequences analysed.
- Number of sequences with a unique best alignment (mapping efficiency).
- Statistics summarising the bisulfite strand the unique best alignments came from.
- Number of cytosines analysed.
- Number of methylated and unmethylated cytosines.
- Percentage methylation of cytosines in CpG, CHG or CHH context (where H can be “A”, “T” or “C”).

This percentage is calculated individually for each context, according to:

$$\text{percent of methylation} = 100\,\frac{\text{methylated Cs}}{\text{methylated Cs} + \text{unmethylated Cs}}.$$

The percent of methylation is only a very rough calculation. Actual methylation levels after post-processing or filtering may vary.

- Inspect the report file. How do you interpret?

4 Extracting methylation calls

An important step is the removal of PCR artifacts. You could use the Picard tools for this step. Like Picard tools, `deduplicate_bismark` identifies reads mapping to the the very same position and in the same orientation, and removes all of them but one:

```
deduplicate_bismark --bam bismark_bam_file
```

Please, note that read deduplication is recommended for WGBS samples, but not for reduced representation bisulfite sequencing (RRBS) libraries. Why?

- Sort and index the deduplicated bam files using Samtools. Check former exercises to recall the appropriate commands.

The methylation calls are already stored in the .bam files:

```
samtools view bismark_deduplicated_bam_file | less
```

Nevertheless, Bismark includes a tool to extract the methylation call for each “C” and write the information in a more readable format. This tool is called `bismark_methylation_extractor`:

```
bismark_methylation_extractor --report -s \
--counts \
--bedGraph \
--gzip \
-o Methylation \
bismark_deduplicated_bam_file
```

The command above will create several output files in a folder called “Methylation”. The position of each “C” is written to a different output file, depending on its context: CpG, CHG or CHH.

The output is tab-separated and includes:

1. the seq-ID;
2. the methylation state; Methylated “Cs” are labelled with a “+”, non-methylated “Cs”, with a “-”.
3. the chromosome;
4. the start (and end) position; and
5. the methylation call (stsring). The methylation call string contains a dot (“.”) for every position in the read not involving a cytosine and one of the following letters cytosines in different methylation contexts:

Methylation call	Meaning
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
u	unmethylated C in Unknown context (CN or CHN)
U	methylated C in Unknown context (CN or CHN)

For example, in CpG context:

```
HWUSI-EAS611_0006:3:1:1058:15806#0/1 - 6 91793279 z
HWUSI-EAS611_0006:3:1:1058:17564#0/1 + 8 122855484 Z
```

In CHG context:

```
HWUSI-EAS611_0006:3:1:1054:1405#0/1 - 7 89920171 x
HWUSI-EAS611_0006:3:1:1054:1405#0/1 + 7 89920172 X
```

In CHH context:

```
HWUSI-EAS611_0006:3:1:1054:1405#0/1 - 7 89920184 h
```

The “`--counts`” option can be used to report the number of each CpG (optionally, each cytosine) in the genome, irrespective of whether it is covered by any reads or not.

The “`--bedGraph`” option can be used to report additionally to .bam format also in .bedGraph format which might be needed for some downstream analysis tools.

4.1 M-bias plot

Methylation bias (M-bias) plots are useful for quality control. The methylation level per position averaged across all reads should be nearly constant, but often shows bias at the 5' and/or 3' end of reads. The first (red) and second (blue) read in each pair is plotted separately. Likewise, results of reads mapping to the original top (OT, left) strand and original bottom (OB, right) are kept separate. When a non-directional library is used, graphs for the complementary to original top (CTOT) and complementary to original bottom (CTOB) are also produced. Vertical lines indicate the suggested bounds of the region for inclusion during methylation extraction. Here you can find more information on this topic.

The `bismark_methylation_extractor` tool produces an M-bias plot. The plot also contains the absolute number of methylation calls (both methylated and unmethylated) per position. For pairedend, reads two individual M-bias plots will be drawn. The data used to generate the M-bias plot is written into a text file and is in the following format:

```
<read position> <count methylated> <count unmethylated> <% methylation> <total coverage>
```

This allows generating nice graphs by alternative means, e.g. using R.

- Inspect the M-bias plot for potential problems. Some problems like methylation bias are explained here.

5 Visualizing methylation data

5.1 Generating heatmaps and profiles with deepTools

Now we will use deepTools to visualize the methylation level around all transcription start sites (TSS). At gene promoters, DNA methylation is usually a repressive mark. deepTools is a suite of python tools for efficient analysis of high-throughput sequencing data. Download and install the tool if necessary.

Therefore, we first need to get the information for the transcription start sites (TSS) in the genome:

- Go to the Ensembl website and click on "BioMart" in the top menu.
  1. Choose database: "Ensembl Genes". The number that follows the name of the database indicates the version.
  2. Choose data set "Human Genes". Check that the assembly matches the one you used for mapping the sequencing reads.
  3. Go to the "Attributes" tab on the left pane and choose the following:
     (a) Chromosome/scaffold name
     (b) Transcription start site (TSS)
     (c) Gene stable ID
     (d) Gene name
     (e) Gene type

     Please note that orders matters when selecting the attributes. If necessary, start by unclicking all boxes.
  4. Click on "Results". Export unique results to a file, in TSV format. You should obtain something like this:

```
MT      577     ENSG00000210049 MT-TF   Mt_tRNA
MT      648     ENSG00000211459 MT-RNR1 Mt_rRNA
MT      1602    ENSG00000210077 MT-TV   Mt_tRNA
MT      1671    ENSG00000210082 MT-RNR2 Mt_rRNA
MT      3230    ENSG00000209082 MT-TL1  Mt_tRNA
MT      3307    ENSG00000198888 MT-ND1  protein_coding
```

5. Use your favorite tool to 1) filter for TSS corresponding to protein-coding genes (we want the protein coding sites as methylation status could have a big impact on gene expression), and 2) add a column in between the TSS and gene stable ID containing the position of the TSS plus 1 (why?). Remove the header and save the resulting file as a .bed file.

The `bamCoverage` tool takes a BAM-formatted file as input and generates a *coverage track*, in bigWig or bedGraph format, as output. The coverage is calculated as the number of reads per *bin*, where the bins are short consecutive counting windows of a defined size. BigWig files are much smaller than BAM files, especially as your bin size increases. The coverage can be normalized, which is important for comparing different samples to each other (because they will most likely vary in terms of sequencing depth).

- Execute the following command for the deduplicated read mappings:

```
bamCoverage \
--binSize 10 \
--extendReads \
--ignoreDuplicates \
--normalizeUsing RPKM \
--bam input_bam_file \
--outFileFormat bigwig \
--outFileName sample.bigwig \
--numberOfProcessors 2
```

- `--binSize/-bs INT`: size of the bins, in base pairs (Default: 50).
- `--extendReads/-e`: extend the reads to the size of the fragments.
- `--ignoreDuplicates`: consider only once (pairs of) reads that have the same orientation and start position.
- `--normalizeUsing RPKM`: normalize the number of reads per bin using "Reads Per Kilobase per Million mapped reads" (RPKM); other possible choices are `CPM` (Counts Per Million mapped reads); `BPM` (Bins Per Million mapped reads); `RPGC` (reads per genomic content, i.e., mapped reads are considered after blacklist filtering, if applied).
- `--outFileName`. File needs to end with .bigwig otherwise troubles will occur.
- `--outFileFormat/-of bigwig`. bigWig files are compressed, binary files. If you would like to see the coverage values, choose "`bedGraph`" as output format.

Given one or more bigWig-formatted files and a set of genomic coordinates, the `computeMatrix` tool constructs a matrix that can be used for visualization:

```
computeMatrix reference-point -S sample1.bigwig sample2.bigwig ... \
-R consensus_peaks \
--beforeRegionStartLength=1500 --afterRegionStartLength=1500 \
--referencePoint=center \
-o matrix_sample.gz
```

`computeMatrix` will then:

(i) Subtract the value indicated in `--beforeRegionStartLength/-b` from the center of the regions in the BED-formatted file.

(ii) Add the value indicated in `--afterRegionStartLength/-a` to the center of the regions in the BED-formatted file.

(iii) Split the resulting region into 50 bp bins (or the value specified by `--binSize`).

(iv) Calculate the mean of the values in the bigWig files for each of the bins.

(v) Write out a matrix where each row corresponds to a region in the BED file and each column to a bigWig file.

The reference point for the plots can be either the region start (TSS), the region end (TES) or the region center Note that regardless of what you specify, plotHeatmap/plotProfile will default to using “TSS” as the label.

You can use both the `plotHeatmap` and `plotProfile` commands to visualize the matrix:

```
plotHeatmap --matrixFile matrix_file_name.gz --outFileName output_file.png
```

and

```
plotProfile --matrixFile matrix_file_name.gz --outFileName output_file.png
```

- Generate a heatmap and a profile visualizing the methylation signal in a 500-bp genomic window centered at the TSSs retrieved from the Ensembl website. plotHeatmap has several parameters that you can adjust to beautify the image (see here and here).
- How do you interpret these figures?

5.2 Visualizing mappings in IGV

The Integrative Genomics Viewer (IGV) is a high-performance, easy-to-use, interactive tool for the visual exploration of genomic data. It supports flexible integration of all the common types of genomic data and metadata, investigator-generated or publicly available, loaded from local or cloud sources.

- Launch IGV and load the deduplicated mappings using `File -> Load from File`.

When bisulfite sequence tracks are initially loaded, default coloring of mismatches against the reference will show red “T’s” and green “A’s”. Right click on the track to open the pop-up menu for the alignments. Under “color alignments by” you should be able to select the “bisulfite mode” and, under the bisulfite mode, “CpG”, among other modes. (Attention: This step will not work in the web version, the desktop version is necessary!) In the “bisulfite mode”, a new coloring schema are applied and together allow you to visually distinguish read strand and bisulfite conversion status. Reads are colored by DNA strand. For paired reads, this is the same as coloring by first-of-pair strand:

- Gray for forward reads or forward read first pairs (F1 or F1R2)
- Sage for reverse reads or reverse read first pairs (R1 or R1F2)

The mode is highlighted in reads with a red or blue nucleotide corresponding to the position of the cytosine in the reference genome.

- For forward reads, a red “C” denotes a nonconverted cytosine, implying methyl or other protection, while a blue “T” denotes a bisulfite-converted cytosine.
- For reverse reads, a red “G” denotes a nonconverted cytosine, implying methyl or other protection, while a blue “A” denotes a bisulfite converted cytosine.

In zoomed-out views, colored nucleotides are represented by colored lines.

Here you can find detailed information.

- Explore the data looking for interesting regions. Load the annotation of the chromosome if necessary.

6 Tool versions used

fastqc 0.11.9, bismark 0.22.3, samtools 1.10, deeptools 3.5.0, IGV 2.4.9, bowtie2 2.3.5.1, perl v5.30.0