Epigenetics – Modifications and Analysis Software

Table of Contents

[Modifications 1](#_Toc32409749)

[WGBS 1](#_Toc32409750)

[Targeted Bisulfite Sequencing 2](#_Toc32409751)

[EPIC-TABSAT: 2](#_Toc32409752)

[Methods of Quality Control 3](#_Toc32409753)

[FastQC - 3](#_Toc32409754)

[ Basic Statistics: 3](#_Toc32409755)

[ Per Base Sequence Quality: 3](#_Toc32409756)

[ Per Sequence Quality: 3](#_Toc32409757)

[ Per base sequence content: 3](#_Toc32409758)

[ Per Sequence GC Content: 4](#_Toc32409759)

[ Per Base N Content: 4](#_Toc32409760)

[ Sequence length distribution: 4](#_Toc32409761)

[ Sequence Duplication Levels: 5](#_Toc32409762)

[ Overrepresented Sequences: 6](#_Toc32409763)

[ Adapter Content: 6](#_Toc32409764)

[ Kmer Content: 6](#_Toc32409765)

[ Per Tile Sequence Quality: 6](#_Toc32409766)

[Methods of Genome Alignment 6](#_Toc32409767)

[BowTie: 6](#_Toc32409768)

[Strand Bias 7](#_Toc32409769)

[Alignment Modes: Two modes are mutually exclusive 7](#_Toc32409770)

[Output and Reporting Modes 8](#_Toc32409771)

[Performing Alignments 8](#_Toc32409772)

[Fastq-screen 8](#_Toc32409773)

[Types of Sequencing 9](#_Toc32409774)

[HMEDIP-Seq 9](#_Toc32409775)

[Chip-Seq 9](#_Toc32409776)

[RNA-Seq 9](#_Toc32409777)

# Modifications

* Most common form of modification is methylation of cytosine bases in C-G base pair **(CpG)**

WGBS Whole genome bisulfite sequencing

* Whole Genome Bisulfite Sequencing (**WGBS**) is the gold standard of epigenetic modification analysis, however:
  + Expensive
  + Time-consuming
  + Difficult at single-base resolution

# Targeted Bisulfite Sequencing

* **TBS** (targeted bisulfite sequencing) is useful for single-base resolution analysis of specific loci
* Multiple computational tools for analysis of TBS:
  + BISMA – one ref. sequence, <400 specific region
  + QUMA – one ref. sequence
  + BisAMP – designed for RNA cytosine-5 methylation data with Illumina compatibility

EPIC-TABSAT: <https://academic.oup.com/nar/article/47/W1/W166/5491737>

Allows multiple regions of interest to be analysed via ‘Target’ file

* 1. Target file must be in excel format
  2. Up to 10GB file size for FastQ sequence
  3. Can perform optional adapter trimming by uploading file containing adapter sequences

Workflow

1. Quality assessment
2. Aligns the provided amplification primer sequences within the respective target region to the reference genome
3. Assignment of the exact primer locations and is used to determine the exact target regions
4. Uses Bismark (15) to map the reads to the selected reference genome using all four DNA strands that result from bisulfite treatment
5. Results are used to extract the methylation state and reads spanning the individual target
6. Reads mapping outside of targets are also summarized
7. The software deduces specific methylation-patterns, which are aggregated and compared between samples.
8. Results are combined and HTML output prepared

# Methods of Quality Control

FastQC - <https://www.bioinformatics.babraham.ac.uk/projects/fastqc/>

Basic Statistics: File name/type, total sequences, poor quality sequences, sequence length (shortest and longest, or one value if all the same), %GpC content

A close up of a logo

Description automatically generatedPer Base Sequence Quality: good for long reads where quality degrades with length,

* + Quality/adapter trimming can then be performed to increase quality
  + Multiple sequences of different lengths can result in poor coverage for a given base range, also reducing quality

Per Sequence Quality: Observed mean quality of below 27 = 0.2% error rate, below 20 = 1% error rate

* + Usually indicates loss of quality within a certain run

A close up of a device

Description automatically generated

Per base sequence content: should not be hugely imbalanced, should run relatively in parallel, especially with longer reads

* + **Biased fragmentation:** Random hexamer ligation/tagmentation libraries always have a selection bias in around the first 12bp of each run - biased selection of random primers. **Nearly all RNA-Seq libraries will fail this module** because of this bias, but this is not a problem which can be fixed by processing, and it doesn't seem to adversely affect the ablity to measure expression.
  + **Biased composition:** **Present in TBS** due to conversion of cytosines to uracils/thymines

Per Sequence GC Content: GC content across each sequence relative to modelled normal distribution

A close up of a device

Description automatically generated

* + **Issues here indicate a library problem**
  + Sharp peaks indicate specific contaminant (adapters etc.)
  + Broader peaks may indicate contamination with a different species

Per Base N Content: If sequencing unable to call base with sufficient confidence, base replaced with an N base.

* + Often more N reads near end of sequence due to poor base quality
  + High N content suggests pipeline could not interpret data to call bases

Sequence length distribution: single peak = one length across sequences, multiple peaks = multiple lengths

A close up of a device

Description automatically generated

Sequence Duplication Levels: High level of duplication can indicate enrichment bias

* + Looks at first 100K sequences
  + Blue line – shows duplication levels of all sequences
  + Red line – proportion of deduplicated sequences in original data
  + Highly diverse library should display peaks at far left of the graph, over-sequencing may flatten lines at lower end and create inconsistent enriched peaks
  + Contaminants will produce peaks at the right
  + Top line shows number of sequences remaining if all duplicates are removed
  + Unable to differentiate between PCR artefacts or true random biological duplicates
  + Warning here simply indicates that this region of the library has been exhausted, maximum diversity has been sequenced
  + In RNA-Seq libraries sequences will be present at v. different levels in starting population
  + To observe lowly expressed transcripts, it is common to greatly over-sequence high expressed transcripts – creating peaks in high regions
  + **Duplication will come from physically connected regions, examination of the distribution of duplicates in a specific genomic region allows distinction between over-sequencing and general technical duplication, but these distinctions are not possible from raw fastq files.**
  + A similar situation can arise in highly enriched CHIP-Seq libraries although the duplication there is less pronounced
  + Finally, if you have a library constructed around restriction sites, or an unfragmented small RNA library, then the constrained start sites will generate huge duplication levels which should not be treated as a problem, nor removed by deduplication. Should consider using random barcoding to allow distinction of technical/biological duplicates



Overrepresented Sequences:indicates highly biologically significant sequences, or a contaminated/non-diverse library

* + Lists sequences which make up more than 0.1% of the total
  + Looks up sequences in list of known common contaminants, hits must be at least 20bp in length – hits indicate true source, may not be exact
  + Triggered when analysing small RNA libraries – sequences not subjected to random fragmentation – 1 sequence naturally present in ^ proportions

Adapter Content:good to assess whether you need to adapter trim or not

* + Denotes specific Kmers and evaluates if they are overrepresented
  + If adapters are longer than inserts then this sequence will automatically fail this fastqc module

Kmer Content: tests for the overrepresentation of short Kmers in the sequences

* + To allow this module to run in a reasonable time only 2% of the whole library is analysed and the results are extrapolated to the rest of the library. Sequences longer than 500bp are truncated to 500bp for this analysis
  + Individually, overrepresented sequences, even if not present at threshold to trigger overrepresented sequences module will cause Kmers from those sequences to be highly enriched in this module. Appearing as sharp spikes of enrichment at a single point in the sequence, rather than broad enrichment
  + Libraries which derive from random priming will nearly always show Kmer bias at the start of the library due to an incomplete sampling of the possible random primers.

### Per Tile Sequence Quality:

* + Will only appear in results if Illumina library was used
  + Can look at quality scores from each tile (across all bases) to see if particular tile in flow-cell caused issues
  + Colour-scale = cold-to-hot, therefore, cold colours indicate average and above quality for that base, hot colours indicate worse quality relative to other tiles for that base
  + Ideally, graphic should be all blue
  + Errors may be flagged due to bubbles in the flow-cell, or permanent issues with smudges/debris on the flow-cell line

Fastq-screen: <https://www.bioinformatics.babraham.ac.uk/projects/fastq_screen/>

Way of checking if reads align with expectations regarding species alignment. Form of QC as ensures no species contamination

# Methods of Genome Alignment

BowTie: <http://bowtie-bio.sourceforge.net/manual.shtml>

* Capable of aligning 35bp reads to human genome at 25 million reads/hour, output can be in SAM output for use in SAMTools
* Forms basis of other software tools: TopHat, Cufflinks, Crossbow, Myrna
* Works best when aligning short reads to large genomes
* Requires: many reads with at least one good quality alignment, many high-quality reads, small number of alignments per read
* -v/-e/-n/-l defines which options are legal
* -k/-a/-m/-M/--best/--strata defines which/how many legal alignments should be reported
* BowTie is ineffective for any ambiguous characters in your reference sequence
* Reports sample alignments in a randomised order

### Strand Bias

* **However – BowTie can exhibit strand bias** – differences in genotype inferred from positive and negative strand
  + When this occurs, software selects one strand or other with 50% probability, then reports alignments from sites on the selected strand
  + Selection often over-assigned to strand with fewer alignment sites
  + Mitigated when longer or paired end reads are used, or use ‘Bowtie --best’ which forces strand selection to be proportional to no. of best sites on strand

### Alignment Modes: Two modes are mutually exclusive

* -n Alignment Mode: this is the default mode for BowTie, uses two criteria
  1. Alignments may have no more than -n mismatches (e.g. ‘-n 3’, -n can only be between 0-3) in the first L bases (set by ‘-l’ option, 5 or greater)
     + Using -n 2/3 reduces sensitivity, time is minimised spent on inaccurate alignments yet this may result in genuine -n 2/3 mismatch alignments not being reported
     + Use Phred quality score; Q (standard Sanger scoring) – where p is the probability that the base call is incorrect
     + Q = -10log(p)
     + Q is inverse log proportional to p
  2. Phred quality scores for mismatched positions must not exceed -e option
* -v Alignment Mode:

1. Alignments may have no more than -v mismatches (where -v is between 0-3)
   * If there are multiple legal alignments, preference given to alignments with fewer mismatches,

### Output and Reporting Modes

* Standard output below:
  + +/- = forward/reverse strand respectively, below shows four hits on reverse strand
  + All hits are inexact, containing mismatches denoted by the ‘NT>NT’ syntax, four contain two mismatches, one contains 1
  + The ‘-a’ option means all hits are reported, -a --best results in all hits being reported in ‘best-to-worst’ order, not shown below
  + -a --best --strata = reports only alignments in best alignment

A screenshot of a social media post

Description automatically generated

* -k option: -k x results in x number of hits being reported, not guaranteed to be the best hit, using -k 1 --best would guarantee top result reported
* -m option: -m x reports alignments when reads have more than (or equal to) x alignments

Link in subheading has all other possible options.

### Performing Alignments

* Comes in format ‘bowtie (basename of reference genome) (name of fastq file for reads), e.g. ‘bowtie e\_coli reads/e\_coli\_1000.fq’
* The alignments can be printed to an output file, e.g. ‘bowtie -t e\_coli reads/e\_coli\_1000.fq e\_coli.map’
  + -t argument prints the timing signatures for each line

# Methods of Adapter Trimming

Cutadapt - <https://cutadapt.readthedocs.io/en/stable/guide.html>

### Basics

* Basic command-line, where output.fastq and input.fastq file names can be altered

cutadapt -a AACCGGTT -o output.fastq input.fastq

* Cutadapt also supports .gz compressed filetypes

tail -n 4 input.fastq | cutadapt -a AACCGGTT - > output.fastq

* The ‘-‘ above instead of standard input replaces ‘input.fastq’

### Adapter Types

| **Adapter type** | **Command-line option** |
| --- | --- |
| [Regular 3’ adapter](https://cutadapt.readthedocs.io/en/stable/guide.html#three-prime-adapters) | -a ADAPTER |
| [Regular 5’ adapter](https://cutadapt.readthedocs.io/en/stable/guide.html#five-prime-adapters) | -g ADAPTER |
| [Non-internal 3’ adapter](https://cutadapt.readthedocs.io/en/stable/guide.html#non-internal) | -a ADAPTERX |
| [Non-internal 5’ adapter](https://cutadapt.readthedocs.io/en/stable/guide.html#non-internal) | -g XADAPTER |
| [Anchored 3’ adapter](https://cutadapt.readthedocs.io/en/stable/guide.html#anchored-3adapters) | -a ADAPTER$ |
| [Anchored 5’ adapter](https://cutadapt.readthedocs.io/en/stable/guide.html#anchored-5adapters) | -g ^ADAPTER |
| [5’ or 3’ (both possible)](https://cutadapt.readthedocs.io/en/stable/guide.html#anywhere-adapters) | -b ADAPTER |
| [Linked adapter](https://cutadapt.readthedocs.io/en/stable/guide.html#linked-adapters) | -a ^ADAPTER1...ADAPTER2 or…. -g ADAPTER1...ADAPTER2 |

### Alignment Algorithm

* Most algorithms use a scoring mechanism for alignments, where insertions/deletions/mismatches = negative score, and alignments = positive score
* Cutadapt uses unit costs, where mismatches etc. = one error, errors within one alignment are then minimised by alignment selection

1. Consider all possible overlaps between the two sequences and compute an alignment for each, minimizing the total number of errors in each one.
2. Keep only those alignments that do not exceed the specified maximum error rate.
3. Then, keep only those alignments that have a maximal number of matches (that is, there is no alignment with more matches).
4. If there are multiple alignments with the same number of matches, then keep only those that have the smallest error rate.
5. If there are still multiple candidates left, choose the alignment that starts at the leftmost position within the read.

### Quality Trimming Algorithm

* The basic idea is to remove all bases starting from the end of the read whose quality is smaller than the given threshold. This is refined a bit by allowing some good-quality bases among the bad-quality ones. In the following example, we assume that the 3’ end is to be quality-trimmed.

1. Assume you use a threshold of 10 and have these quality values:
   1. 42, 40, 26, 27, 8, 7, 11, 4, 2, 3
2. Subtracting the threshold gives:
   1. 32, 30, 16, 17, -2, -3, 1, -6, -8, -7
3. Then sum up the numbers, starting from the end (partial sums). Stop early if the sum is greater than zero:
   1. (70), (38), 8, -8, -25, -23, -20, -21, -15, -7

The numbers in parentheses are not computed (because 8 is greater than zero), but shown here for completeness. The position of the minimum (-25) is used as the trimming position. Therefore, the read is trimmed to the first four bases, which have quality values 42, 40, 26, 27.

# Types of Sequencing

## HMEDIP-Seq

## Chip-Seq

## RNA-Seq