December 15, 2011

(1) Bismark - Quick Reference v0.6.beta2

Bismark needs a working version of Perl and is run from the command line. Furthermore, Bowtie (http://bowtie-bio.sourceforge.net/index.shtml) or Bowtie 2 (http://bowtie-bio.sourceforge.net/bowtie2) needs to be installed on your computer. For more information on how to run Bismark with Bowtie 2 please go to the end of this manual.

First you need to download a reference genome and place it in a genome folder. Genomes can be obtained e.g. from the Ensembl (http://www.ensembl.org/info/data/ftp/index.html/) or NCBI websites (ftp://ftp.ncbi.nih.gov/genomes/) (for the example below you would need to download the *Homo sapiens* genome. Bismark supports reference genome sequence files in FastA format, allowed file extensions are either either .fa or .fasta. Both single-entry or multiple-entry FastA files are supported.

The following examples will use the file `'test_dataset.fastq'` which is available for download from the Bismark homepage (it contains 10,000 reads in FastQ format, Phred33 qualities, 50 bp long reads, from a human directional BS-Seq library).

(I) Running `bismark_genome_preparation`

USAGE: `bismark_genome_preparation [options] <path_to_genome_folder>`

A typical genome indexing could look like this:

```
/bismark/bismark_genome_preparation --path_to_bowtie /usr/local/bowtie/ --verbose /data/genomes/homo_sapiens/GRCh37/
```

(II) Running `bismark`

USAGE: `bismark [options] <genome_folder> {-1 <mates1> -2 <mates2> | <singles>}`

Typical alignment example (tolerating one non-bisulfite mismatch per read):

```
/bismark/bismark -q --phred33-quals -n 1 -l 50 --directional /data/genomes/homo_sapiens/GRCh37/ test_dataset.fastq
```

This will produce two output files:

(a) `test_dataset.fastq_bismark.txt` (contains all alignments plus methylation call strings)

(b) `test_dataset.fastq_bismark_mapping_report.txt` (contains alignment and methylation summary)

(III) Running the Bismark `methylation_extractor`

USAGE: `methylation_extractor [options] <filenames>`

A typical command to extract context-dependent (CpG/CHG/CHH) methylation could look like this:

```
/bismark/methylation_extractor -s --comprehensive
test_dataset.fastq_bismark.txt
```

This will produce three output files:

(a) `CpG_context_test_dataset.fastq_bismark.txt`
(b) `CHG_context_test_dataset.fastq_bismark.txt`
(c) `CHH_context_test_dataset.fastq_bismark.txt`

(2) Bismark - General Information

What is Bismark?

Bismark is a set of tools for the time-efficient analysis of Bisulfite-Seq (BS-Seq) data. Bismark performs alignments of bisulfite-treated reads to a reference genome and cytosine methylation calls at the same time. Bismark is written in Perl and is run from the command line. Bisulfite-treated reads are mapped using the short read aligner Bowtie (Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biol 10:R25) and therefore it is a requirement that Bowtie is installed on your machine (see Dependencies).

All files associated with Bismark as well as a test BS-Seq data set can be downloaded from:
http://www.bioinformatics.bbsrc.ac.uk/projects/bismark/

We would like to hear your comments/suggestions about Bismark! Please email them to: felix.krueger@bbsrc.ac.uk.

Bugs can be reported to our bug tracking system at:
http://www.bioinformatics.bbsrc.ac.uk/bugzilla/

Installation notes

Bismark is written in Perl and is executed from the command line. To install Bismark simply copy the bismark_v0.X.Y.tar.gz file into a Bismark installation folder and extract all files by typing:

```
tar xzf bismark_v0.X.Y.tar.gz
```

Dependencies

Bismark requires a working of Perl and Bowtie to be installed on your machine (http://bowtie-bio.sourceforge.net/index.shtml). Bismark will assume that the Bowtie executable is in your path unless the path to Bowtie is specified manually with:

```
--path_to_bowtie </../../bowtie>.
```

In order to work properly the current working directory must contain the sequence files to be analysed.

Hardware requirements

Bismark holds the reference genome in memory and in addition to that runs four parallel instances of Bowtie. The memory usage is dependent on the size of the reference genome. For a large eukaryotic genome (human or mouse) we experienced a typical memory usage of around 8-10GB. We thus recommend running Bismark on a machine with 4 CPU cores and 12 GB of RAM.

Alignment speed depends largely on the read length and bowtie parameters used. Allowing many mismatches and using a short seed length (which is the default option, see below) tends to be fairly slow, whereas looking for near perfect matches can align around 5-25 million sequences per hour.

BS-Seq test data set

A test BS-Seq data set is available for download from the Bismark homepage. It contains 10,000 single-end shotgun BS reads from human ES cells in FastQ format (from SRR020138, Lister et al, 2009; trimmed to 50 bp; base call qualities are Sanger encoded Phred values (Phred33)).

Which kind of BS-Seq files and/or experiments are supported?

Bismark supports the alignment of bisulfite-treated reads for the following conditions:

- sequence format either fastQ or fastA
- single-end or paired-end reads
- input files can be uncompressed or gzip compressed (ending in `.gz`)
- variable read length support
- directional or non-directional BS-Seq libraries

In addition, Bismark retains much of the flexibility of Bowtie (adjustable seed length, number of mismatches, insert size ...). For a full list of options please run `./bismark --help` or see Appendix at the end of this User Guide.

It should be mentioned that Bismark supports only reads in base-space, such as from the Illumina platform. There are currently no plans to extend its functionality to colour-space reads.

How does Bismark work?

Sequence reads are first transformed into fully bisulfite-converted forward (C->T) and reverse read (G->A conversion of the forward strand) versions, before they are aligned to similarly converted versions of the genome (also C->T and G->A converted). Sequence reads that produce a unique best alignment from the four alignment processes against the bisulfite genomes (which are running in parallel) are then compared to the normal genomic sequence and the methylation state of all cytosine positions in the read is inferred.

Bismark alignment and methylation call report

Upon completion, Bismark produces a run report which 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 of methylation of cytosines in either CpG, CHG or CHH context (where H can be either A, T or C). This percentage is calculated individually for each context following the equation:

% methylation (context) = 100 * methylated Cs (context) / (methylated Cs (context) + unmethylated Cs (context)).

It should be stressed that the percent methylation value (context) is just a very rough calculation performed directly at the mapping step. Actual methylation levels after post-processing or filtering have been applied may vary.

(3) Running Bismark

Running Bismark is split up into three individual steps:

(I) First, the genome of interest needs to be bisulfite converted and indexed to allow Bowtie alignments. This step needs to be carried out only once for each genome.

(II) Bismark read alignment step. Simply specify a file to be analysed, a reference genome and alignment parameters. Bismark will produce a combined alignment/methylation call output as well as a run statistics report.

(III) Bismark methylation extractor. This step is optional and will extract the methylation information from the Bismark alignment output. Running this additional step allows splitting the methylation information up into the different contexts, getting strand-specific methylation information and offers some filtering options.

Each of these steps will be described in more detail (with examples) in the following sections.

(I) Bismark Genome Preparation

This script needs to be run only once to prepare the genome of interest for bisulfite alignments. You need to specify a directory containing the genome you want to align your reads against (please be aware that the `bismark_genome_preparation` script currently expects fastA files in this folder (with either `.fa` or `.fasta` extension, single or multiple sequence entries per file). Bismark will create two individual folders within this directory, one for a C->T converted genome and the other one for the G->A converted genome. After creating C->T and G->A versions of the genome they will be indexed in parallel using the indexer `bowtie-build`. Once both C->T and G->A genome indices have been created you do not need to use the genome preparation script again (unless you want to align against a different genome....).

(I) Running `bismark_genome_preparation`

USAGE: `bismark_genome_preparation [options] <path_to_genome_folder>`

A typical command could look like this:

```
./bismark_genome_preparation --path_to_bowtie /usr/local/bowtie/ --verbose
/data/genomes/homo_sapiens/GRCh37/
```

(II) Bismark alignment step

This step represents the actual bisulfite alignment and methylation calling part. Bismark requires the user to specify only two things:

(a) The directory containing the genome of interest. This folder must contain the unmodified genome (as `.fa` or `.fasta` files) as well as the two bisulfite genome subdirectories which were generated in the Bismark Genome Preparations step (see above).

(b) The sequence file(s) to be analysed (in either FastQ or FastA format).

All other parameters are optional.

In the current version, it is required that the current working directory also contains the sequence files to be analysed. For each sequence file or each set of paired-end sequence files, Bismark produces one comprehensive alignment and methylation call output file as well as a report file detailing alignment and methylation call statistics for your information and record keeping.

(II) Running `bismark`

Before running Bismark we recommend spending some time on quality control of the raw sequence files using FastQC (www.bioinformatics.bbsrc.ac.uk/projects/fastqc/). FastQC might be able to spot irregularities associated with your BS-Seq file, such as high base calling error rates or contaminating sequences such as PCR primers or Illumina adapters. Many sources of error will impact detrimentally the alignment efficiencies and/or alignment positions, and thus possibly also affect the methylation calls and conclusions drawn from the experiment.

If no additional options are specified Bismark will use a set of default values, some of which are:

- if no specific path to Bowtie is specified it is assumed that the `bowtie` executable is in the path.
- Bowtie is run `--best mode` (it is not recommended to turn this off)
- standard alignments allow up to 2 mismatches in the seed region (which is defined as the first 28 bp by default). These parameters can be modified using the options `-n` and `-l`, respectively.

Even though the user is not required to specify additional alignment options it is often advisable to do so. To see a full list of options please type `./bismark --help` on the command line or see the Appendix at the end of this User Guide.

Directional BS-Seq libraries (`--directional`)

Bisulfite treatment of DNA and subsequent PCR amplification can give rise to four – bisulfite converted – strands for a given locus. Depending on the adapters used, BS-Seq libraries can be constructed in two different ways:

(a) If a library is directional, only reads which are (bisulfite converted) versions of the original top strand (OT) or the original bottom strand (OB) will be sequenced. Even though the strands complementary to OT (CTOT) and OB (CTOB) are generated in the BS-PCR step they will not be sequenced as they carry the wrong kind of adapter at their 5'-end. Specifying '`--directional`' will instruct Bismark to ignore any uniquely best alignments coming from the complementary strands as they shouldn't exist in theory and won't be present in the BS-Seq library in question.

(b) Alternatively, BS-Seq libraries can be constructed so that all four different strands generated in the BS-PCR can and will end up in the sequencing library with roughly the same likelihood. In this case all four strands (OT, CTOT, OB, CTOB) can produce valid alignments and the library is called non-directional.

To summarise this again: alignments to the original top strand or to the strand complementary to the original top strand (OT and CTOT) will both yield methylation information for cytosines on the top strand. Alignments to the original bottom strand or to the strand complementary to the original bottom strand (OB and CTOB) will both yield methylation information for cytosines on the bottom strand, i.e. they will appear to yield methylation information for G positions on the top strand of the reference genome.

For more information about how to extract methylation information of the four different alignment strands please see below in the section on the Bismark methylation extractor.

USAGE: `./bismark [options] <genome_folder> {-1 <mates1> -2 <mates2> | <singles>}`

A typical single-end analysis of a 40 bp sequencing run could look like this:

```
./bismark -q --phred64-quals -n 1 -l 40 --directional /data/genomes/homo_sapiens/GRCh37/ s_1_sequence.txt
```

What does the Bismark output look like?

Bismark produces a comprehensive alignment and methylation call output file for each input file or set of paired-end input files. As of version 0.5.0 the sequence basecall qualities of the input FastQ files are written out into the Bismark output file as well to allow filtering on quality thresholds. Please note that the quality values are encoded in Sanger format (Phred 33 scale), even if the input was in Phred64 or the old Solexa format.

The single-end output contains the following information (1 line per sequence, tab separated):

```
 (1) seq-ID
 (2) alignment strand
 (3) chromosome
 (4) start
 (5) end
 (6) original bisulfite read sequence
 (7) equivalent genomic sequence (+2 extra bp)
 (8) methylation call string
 (9) read conversion
(10) genome conversion
(11) read quality score (Phred33 scale)
```

Single-end alignment example:

```
 (1) HWUSI-EAS611_0001:3:1:1186:18334#0/1
 (2) +
 (3) 4
 (4) 122893213
 (5) 122893242
 (6) TGGGTTTTTAAGATTTTGTGTAGTTGGGGTTTGGAGATGG
 (7) CGGGCCCTCAAGACCCTGCGCAGCTGGGGCCTGGAGACGGAG
 (8) z...hhh.h....hhx..z.x..x.....hx......z..
 (9) CT
(10) CT
(11) IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
```

The paired-end output looks like this (1 line per sequence pair, tab separated):

```
 (1) seq-ID
 (2) alignment strand
 (3) chromosome
 (4) start
 (5) end
 (6) original bisulfite read sequence 1
 (7) equivalent genomic sequence 1 (+2 extra bp)
 (8) methylation call string 1
 (9) original bisulfite read sequence 2
```

```
(10) equivalent genomic sequence 2 (+2 extra bp)
(11) methylation call string 2
(12) read 1 conversion
(13) genome conversion
(14) read 1 quality score (Phred33 scale)
(15) read 2 quality score (Phred33 scale)
```

Paired-end alignment example:

```
 (1) HWUSI-EAS611_100205:2:1:13:1732#0
 (2) +
 (3) 14
 (4) 62880539
 (5) 62880652
 (6) CGGGATTTCGCGGAGTACGGGTGATCGTGTGGAATATAGA
 (7) CGGGACTCCGCGGAGCACGGGTGACCGTGTGGAATACAGAGT
 (8) Z....h.xZ.Z....h.Z......xZ..........x...
 (9)   CAACTATCTAAAACTAAAAATAACGCCGCCCAAAAACTCT
(10) TCCGGCTGTCTGGAGCTGAAGATGGCGCCGCCCAGAAGCTCT
(11)   .zx..x...xh.h..x..h..hh.Z..Z....x..h....
(12) CT
(13) CT
(14) IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
(15) IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
```

To see the location of the mapped reads the Bismark output file can be imported into a genome viewer, such as SeqMonk, using the chromosome, start and end positions (this can be useful to identify regions in the genome which display an artefactually high number of aligned reads). The alignment output can also be used to apply post-processing steps such as de-duplication (allowing only 1 read for each position in the genome to remove PCR artefacts) or filtering on the number of bisulfite conversion related non-bisulfite mismatches * (please note that such post-processing scripts are not part of the Bismark package).

* Bisulfite conversion related non-bisulfite mismatches are mismatch positions which have a C in the BS-read but a T in the genome; such mismatches may occur due to the way bisulfite read alignments are performed. Reads containing this kind of mismatches are not automatically removed from the alignment output in order not to introduce a bias for unmethylated reads (please send me an email if you have further questions about this). It should be noted that, even though no methylation calls are performed for these positions, reads containing bisulfite conversion related non-bisulfite mismatches might lead to false alignments if particularly lax alignment parameters were specified.

Methylation call

The methylation call string contains a dot '.' for every position in the BS-read not involving a cytosine, or contains one of the following letters for the three different cytosine methylation contexts (UPPER CASE = METHYLATED, lower case = unmethylated):

```
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
```

(III) Bismark methylation extractor

Bismark comes with a supplementary `methylation_extractor` script which operates on Bismark result files and extracts the methylation call for every single C analysed. The position of every single C will be written out to a new output file, depending on its context (CpG, CHG or CHH), whereby methylated Cs will be labelled as forward reads (+), non-methylated Cs as reverse reads (-). The resulting files can be imported into a genome viewer such as SeqMonk (using the generic text import filter) and the analysis of methylation data can commence. Alternatively, the output of the methylation extractor can be transformed into a bedGraph file using the script genome_methylation_bismark2bedGraph which was kindly provided by Dr Oliver Tam (also available for download at www.bioinformatics.bbsrc.ac.uk/projects/bismark/).

The methylation extractor output looks like this (tab separated):

```
(1) seq-ID
(2) methylation state
(3) chromosome
(4) start position (= end position)
(5) methylation call
```

Methylated cytosines will receive a '+' orientation, unmethylated cytosines will receive a '-' orientation.

Examples for cytosines 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
```

Examples for cytosines 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
```

Examples for cytosines in CHH context:

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

The Bismark `methylation_extractor` comes with a few options, such as ignoring the first <int> number of positions in the methylation call string, e.g. to remove a restriction enzyme site (if RRBS is performed with non-directional BS-Seq libraries it might be required to remove reconstituted MspI sites at the beginning of each read as they will introduce a bias into the first methylation call (please send me an email if you require further information on this)). Another useful option for paired-end reads is called `'--no_overlap'`: specifying this option will extract the methylation calls of overlapping parts in the middle of paired-end reads only once (using the calls from the first read which is presumably the one with a lowest error rate).

For a full list of options type `./methylation_extractor --help` at the command line or refer to the Appendix section at the end of this User Guide.

Methylation extractor output

By default, the Bismark `methylation_extractor` discriminates between cytosines in CpG, CHG or CHH context. If desired, CHG and CHH contexts can be merged into a single non-CpG context by specifying the option `'--merge_non_CpG'` (as a word of warning, this might produce files with up to several hundred million lines…).

Strand-specific methylation output files (default):

As its default option, the `methylation_extractor` will produce a strand-specific output which will use the following abbreviations in the output file name to indicate the strand the alignment came from:

OT – original top strand
CTOT – complementary to original top strand

OB – original bottom strand
CTOB – complementary to original bottom strand

Methylation calls from OT and CTOT will be informative for cytosine methylation positions on the original top strand, calls from OB and CTOB will be informative for cytosine methylation positions on the original bottom strand. Please note that specifying the `'--directional'` option in the Bismark alignment step will not report any alignments to the CTOT or CTOB strands.

As cytosines can exist in any of three different sequence contexts (CpG, CHG or CHH) the `methylation_extractor` default output will consist of 12 individual output files per input file (`CpG_OT_...`, `CpG_CTOT_....`, `CpG_OB_...` etc.).

Note for SeqMonk users: if you were running a non-directional BS-Seq library and are interested in strand-specific methylation, the CT and CTOT output files can be merged into a 'top strand' data group and OB and CTOB can be merged into a 'bottom strand' data group in SeqMonk before starting the analysis.

Context-dependent methylation output files (`--comprehensive` option):

If strand-specific methylation is not of interest, all available methylation information can be pooled into a single context-dependent file (information from any of the four strands will be pooled). This will default to three output files (CpG-context, CHG-context and CHH-context), or result in 2 output files (CpG-context and Non-CpG-context) if `'--merge_non_CpG'` was selected (note that this can result in enormous file sizes for the non-CpG output).

Both strand-specific and context-dependent options can be combined with the `'--merge_non_CpG'` option.

(III) Running the Bismark `methylation_extractor`

USAGE: `./methylation_extractor [options] <filenames>`

A typical command for a single-end file could look like this:

```
./methylation_extractor -s --comprehensive s_1_sequence.txt_bismark.txt
```

If you get stuck at any point or have any questions/comments please contact me via email:
felix.krueger@bbsrc.ac.uk

(4) APPENDIX - Full list of options

(I) Bismark Genome Preparation

A full list of options can also be viewed by typing: `./bismark_genome_preparation --help`

USAGE: `./bismark_genome_preparation [options] <arguments>`

OPTIONS:

`--help/--man`	Displays this help file.
`--verbose`	Print verbose output for more details or debugging.
`--yes/--yes_to_all`	Answer yes to safety related questions (such as "Are you sure you want to overwrite any existing folder called Bisulfite_Genomes?").
`--path_to_bowtie </../../>`	The full path to the bowtie installation on your system. If the path is not provided as an option you will be prompted for it later.
`--single_fasta`	Instruct the Bowtie Indexer to write the converted genomes into single-entry FastA files instead of making one multi-FastA file (MFA) per chromosome. This might be useful if individual bisulfite converted chromosomes are needed (e.g. for debugging), however it can cause a problem with indexing if the number of chromosomes is vast (this is likely to be in the range of several thousand files; operating systems can only handle lists up to a certain length. Some newly assembled genomes may contain 20000-50000 contig of scaffold files which do exceed this list length limit).

ARGUMENTS:

`<path_to_genome_folder>`	The path to the folder containing the genome to be bisulfite converted (this may be an absolute or relative path). Bismark Genome Preparation expects one or more fastA files in the folder (valid file extensions: `.fa` or `.fasta`). If the path is not provided as an argument you will beprompted for it later.

(II) Bismark

A brief description of Bismark and a full list of options can also be viewed by typing: `./Bismark --help`

USAGE: `bismark [options] <genome_folder> {-1 <mates1> -2 <mates2> | <singles>}`

ARGUMENTS:

`<genome_folder>`	The full path to the folder containing the unmodified reference genome as well as the subfolders created by the Bismark_Genome_Preparation script (`/Bisulfite_Genome/CT_conversion/` and `Bisulfite_Genome/GA_conversion/`). Bismark expects one or more fastA files in this folder (file extension: `.fa` or `.fasta`). The path to the genome folder can be relative or absolute.
`-1 <mates1>`	Comma-separated list of files containing the #1 mates (filename usually includes "_1"), e.g. `flyA_1.fq,flyB_1.fq`). Sequences specified with this option must correspond file-for-file and read-for-read with those specified in `<mates2>`. Reads may be a mix of different lengths. Bismark will produce one mapping result and one report file per paired-end input file pair.
`-2 <mates2>`	Comma-separated list of files containing the #2 mates (filename usually includes "_2"), e.g. `flyA_1.fq,flyB_1.fq`). Sequences specified with this option must correspond file-for-file and read-for-read with those specified in `<mates1>`. Reads may be a mix of different lengths.
`<singles>`	A comma or space separated list of files containing the reads to be aligned (e.g. `lane1.fq, lane2.fq,lane3.fq`). Reads may be a mix of different lengths. Bismark will produce one mapping result and one report file per input file.

OPTIONS:

Input:

`-q/--fastq`	The query input files (specified as `<mate1>`, `<mate2>` or `<singles>` are FastQ files (usually having extension `.fq` or `.fastq`). This is the default. See also `--solexa-quals` and `--integer-quals`.
`-f/--fasta`	The query input files (specified as `<mate1>`, `<mate2>` or `<singles>` are FastA files (usually having extension `.fa`, `.mfa`, `.fna` or similar). All quality values are assumed to be 40 on the Phred scale.
`-s/--skip <int>`	Skip (i.e. do not align) the first `<int>` reads or read pairs from the input.
`-u/--qupto <int>`	Only aligns the first `<int>` reads or read pairs from the input. Default: no limit.

--phred33-quals FastQ qualities are ASCII chars equal to the Phred quality plus 33. Default: on.

--phred64-quals FastQ qualities are ASCII chars equal to the Phred quality plus 64. Default: off.

--solexa-quals Convert FastQ qualities from solexa-scaled (which can be negative) to phred-scaled (which can't). The formula for conversion is: phred-qual = 10 * log(1 + 10 ** (solexa-qual/10.0)) / log(10). Used with -q. This is usually the right option for use with (unconverted) reads emitted by the GA Pipeline versions prior to 1.3. Default: off.

--solexa1.3-quals Same as --phred64-quals. This is usually the right option for use with (unconverted) reads emitted by GA Pipeline version 1.3 or later. Default: off.

--path_to_bowtie The full path </../../> to the Bowtie (1 or 2) installation on your system. If not specified it will be assumed that Bowtie is in the path.

Alignment:

-n/--seedmms <int> The maximum number of mismatches permitted in the "seed", i.e. the first base pairs of the read (where L is set with -l/--seedlen). This may be 0, 1, 2 or 3 and the default is 2. This option is only available for Bowtie 1 (for Bowtie 2 see -N).

-l/--seedlen The "seed length"; i.e., the number of bases of the high quality end of the read to which the -n ceiling applies. The default is 28. Bowtie (and thus Bismark) is faster for larger values of -l. This option is only available for Bowtie 1 (for Bowtie 2 see -L).

-e/--maqerr <int> Maximum permitted total of quality values at all mismatched read positions throughout the entire alignment, not just in the "seed". The default is 70. Like Maq, bowtie rounds quality values to the nearest 10 and saturates at 30.

--chunkmbs <int> The number of megabytes of memory a given thread is given to store path descriptors in --best mode. Best-first search must keep track of many paths at once to ensure it is always extending the path with the lowest cumulative cost. Bowtie tries to minimize the memory impact of the descriptors, but they can still grow very large in some cases. If you receive an error message saying that chunk memory has been exhausted in --best mode, try adjusting this parameter up to dedicate more memory to the descriptors. Default: 512.

-I/--minins <int> The minimum insert size for valid paired-end alignments. E.g. if -I 60 is specified and a paired-end alignment consists of two 20-bp alignments in the appropriate orientation with a 20-bp gap between them, that alignment is considered valid (as long as -X is also satisfied). A 19-bp gap would not be valid in that case. Default: 0.

-X/--maxins <int> The maximum insert size for valid paired-end alignments. E.g. if -X 100 is specified and a paired-end alignment consists of two 20-bp alignments in

the proper orientation with a 60-bp gap between them, that alignment is considered valid (as long as `-I` is also satisfied). A 61-bp gap would not be valid in that case. Default: 250.

Bowtie 1 Reporting:

`-k <2>` Due to the way Bismark works Bowtie will report up to 2 valid alignments. This option will be used by default and cannot be changed.

`--best` Make Bowtie guarantee that reported singleton alignments are "best" in terms of stratum (i.e. number of mismatches, or mismatches in the seed in the case if `-n` mode) and in terms of the quality; e.g. a 1-mismatch alignment where the mismatch position has Phred quality 40 is preferred over a 2-mismatch alignment where the mismatched positions both have Phred quality 10. When `--best i`s not specified, Bowtie may report alignments that are sub-optimal in terms of stratum and/or quality (though an effort is made to report the best alignment). `--best` mode also removes all strand bias. Note that `--best` does not affect which alignments are considered "valid" by Bowtie, only which valid alignments are reported by Bowtie. Bowtie is about 1-2.5 times slower when `--best` is specified. Default: on.

`--no_best` Disables the `--best` option which is on by default. This can speed up the alignment process, e.g. for testing purposes, but for credible results it is not recommended to disable `--best`.

Output:

`--directional` The user may specify if the sequencing library was constructed in a strand-specific manner. In this case the strands complementary to the original strands are merely theoretical and should not exist in reality. Thus, specifying `--directional` will only report alignments to the original top or bottom strands. This is the recommended option for strand-specific libraries (both single or paired-end).

`--sam-no-hd` Suppress SAM header lines (starting with @). This might be useful when very large input files are split up into several smaller files to run concurrently and the output files are to be merged.

`--quiet` Print nothing besides alignments.

`--vanilla` Performs bisulfite mapping with Bowtie 1 and prints the 'old' output (as in Bismark 0.5.X) instead of SAM format output.

`--un` Write all reads that could not be aligned to the file `_unmapped_reads.txt` in the output directory. Written reads will appear as they did in the input, without any translation of quality values that may have taken place within `Bowtie` or `Bismark`. Paired-end reads will be written to two parallel files with _1 and _2 inserted in their filenames, i.e. `unmapped_reads_1.txt` and `unmapped_reads_2.txt`. Reads with more than one valid alignments with the

same number of lowest mismatches (ambiguous mapping) are also written to `unmapped_reads.txt` unless `--ambiguous` is also specified.

`--ambiguous` Write all reads which produce more than one valid alignment with the same number of lowest mismatches or other reads that fail to align uniquely to `_ambiguous_reads.txt`. Written reads will appear as they did in the input, without any of the translation of quality values that may have taken place within `Bowtie` or `Bismark`. Paired-end reads will be written to two parallel files with _1 and _2 inserted in their filenames, i.e. `_ambiguous_reads_1.txt` and `_ambiguous_reads_2.txt.fq`. These reads are not written to the file specified with `--un`.

`-o/--output_dir <dir>` Write all output files into this directory. By default the output files will be written into the same folder as the input file. If the specified folder does not exist, Bismark will attempt to create it first. The path to the output folder can be either relative or absolute.

Other:

`-h/--help` Displays this help file.

`-v/--version` Displays version information.

BOWTIE 2 SPECIFIC OPTIONS

`--bowtie2` Uses Bowtie 2 instead of Bowtie 1. Bismark limits Bowtie 2 to only perform end-to-end alignments, i.e. searches for alignments involving all read characters (also called untrimmed or unclipped alignments). Bismark assumes that raw sequence data is adapter and/or quality trimmed where appropriate. Default: off.

Bowtie 2 alignment options:

`-N <int>` Sets the number of mismatches to allowed in a seed alignment during multispeed alignment. Can be set to 0 or 1. Setting this higher makes alignment slower (often much slower) but increases sensitivity. Default: 0. This option is only available for Bowtie 2 (forBowtie 1 see -n).

`-L <int>` Sets the length of the seed substrings to align during multiseed alignment. Smaller values make alignment slower but more senstive. Default: the --sensitive preset of Bowtie 2 is used by default, which sets -L to 20. This option is only available for Bowtie 2 (for Bowtie 1 see -l).

`--ignore-quals` When calculating a mismatch penalty, always consider the quality value at the mismatched position to be the highest possible, regardless of the actual value. I.e. input is treated as though all quality values are high. This is also the default behavior when the input doesn't specify quality values (e.g. in -f mode). This option is invariable and on by default.

Bowtie 2 paired-end options:

`--no-mixed` This option disables Bowtie 2's behavior to try to find alignments for the individual mates if it cannot find a concordant or discordant alignment for a pair. This option is invariable and and on by default.

`--no-discordant` Normally, Bowtie 2 looks for discordant alignments if it cannot find any concordant alignments. A discordant alignment is an alignment where both mates align uniquely, but that does not satisfy the paired-end constraints (--fr/--rf/--ff, -I, -X). This option disables that behaviour and it is on by default.

Bowtie 2 Effort options:

`-D <int>` Up to <int> consecutive seed extension attempts can "fail" before Bowtie 2 moves on, using the alignments found so far. A seed extension "fails" if it does not yield a new best or a new second-best alignment. Default: 15.

`-R <int>` <int> is the maximum number of times Bowtie 2 will "re-seed" reads with repetitive seeds. When "re-seeding," Bowtie 2 simply chooses a new set of reads (same length, same number of mismatches allowed) at different offsets and searches for more alignments. A read is considered to have repetitive seeds if the total number of seed hits divided by the number of seeds that aligned at least once is greater than 300. Default: 2.

Bowtie 2 parallelization options:

`-p NTHREADS` Launch NTHREADS parallel search threads (default: 1). Threads will run on separate processors/cores and synchronize when parsing reads and outputting alignments. Searching for alignments is highly parallel, and speedup is close to linear. Increasing -p increases Bowtie 2's memory footprint. E.g. when aligning to a human genome index, increasing -p from 1 to 8 increases the memory footprint by a few hundred megabytes. This option is only available if bowtie is linked with the pthreads library (i.e. if BOWTIE_PTHREADS=0 is not specified at build time). In addition, this option will automatically use the option '--reorder', which guarantees that output SAM records are printed in an order corresponding to the order of the reads in the original input file, even when -p is set greater than 1 (Bismark requires the Bowtie 2 output to be this way). Specifying --reorder and setting -p greater than 1 causes Bowtie 2 to run somewhat slower and use somewhat more memory then if --reorder were not specified. Has no effect if -p is set to 1, since output order will naturally correspond to input order in that case.

Bowtie 2 Scoring options:

`--score_min <func>` Sets a function governing the minimum alignment score needed for an alignment to be considered "valid" (i.e. good enough to report). This is a function of read length. For instance, specifying L,0,-0.6 sets the minimum-score function f to f(x) = 0 + -0.6 * x, where x is the read length. See also: setting function options at http://bowtie-bio.sourceforge.net/bowtie2. The default is L,0,-0.6.

Bowtie 2 Reporting options:

`--most_valid_alignments <int>` This is the Bowtie 2 parameter -M. Bowtie 2 searches for at most <int>+1 distinct, valid alignments for each read. The search terminates when it can't find more distinct valid alignments, or when it finds <int>+1 distinct alignments, whichever happens first. Only the best alignment is reported. Information from the other alignments is used to estimate mapping quality and to set SAM optional fields, such as AS:i and XS:i. Increasing -M makes Bowtie 2 slower, but increases the likelihood that it will pick the correct alignment for a read that aligns many places. for reads that have more than <int>+1 distinct, valid alignments, Bowtie 2 does not guarantee that the alignment reported is the best possible in terms of alignment score. -M is always used and its default value is set to 10.

(III) Bismark Methylation Extractor

A brief description of the Bismark methylation extractor and a full list of options can also be viewed by typing:
`./methylation_extractor --help`

USAGE: `./methylation_extractor [options] <filenames>`

ARGUMENTS:

`<filenames>`	A space-separated list of result files in Bismark format from which methylation information is extracted for every cytosine in the read.

OPTIONS:

`-s/--single-end`	Input file(s) are Bismark result file(s) generated from single-end read data. Specifying either `--single-end` or `--paired-end` is mandatory.
`-p/--paired-end`	Input file(s) are Bismark result file(s) generated from paired-end read data. Specifying either `--paired-end` or `--single-end` is mandatory.
`--no_overlap`	For paired-end reads it is theoretically possible that read_1 and read_2 overlap. This option avoids scoring overlapping methylation calls twice. Whilst this removes a bias towards more methylation calls towards the center of sequenced fragments it can de facto remove a good proportion of the data.
`--fasta`	Choosing this option will print out the genomic sequences that correspond to the bisulfite mapped reads in FastA format. This might be useful for certain applications where the bisulfite read cannot be used (such as repeat analyses).
`--ignore <int>`	Ignore the first `<int>` bp when processing the methylation call string. As all reads are sorted in a forward direction this can remove e.g. a restriction enzyme site at the start of each read (this can become relevant for non-directional libraries in RRBS type experiments).
`--comprehensive`	Specifying this option will merge all four possible strand-specific methylation info into context-dependent output files. The default contexts are: (i) CpG context (ii) CHG context (iii) CHH context (Depending on the C content of the Bismark result file, the output file size might reach 10-30GB!).
`--merge_non_CpG`	This will produce two output files (in `--comprehensive` mode) or eight strand-specific output files (default) for Cs in (i) CpG context (ii) any non-CpG context (Depending on the C content of the Bismark result file, the output file size might reach 10-30GB!).

`--report`	Prints out a short methylation summary and the parameters used to run this script.
`--version`	Displays the version information.
`-h/--help`	Displays this help file and exits.