NGSmethPipe Manual (v1.1)

NGSmethPipe is a program for the generation of single base-pair-resolution methylation maps from bisulfite conversion high throughput sequencing experiments. The program has 4 steps: (i) indexing (sequence conversion into 3 letter alphabet building the Bowtie index), (ii) preprocessing of the reads (adapter removal, quality trimming), (iii) Bowtie alignment (single-end or pair-end reads) and (vi) post-processing including SNV and bisulfite failure detection, sequence error handling and extraction of methylation data for different sequence contexts.

Main features of NGSmethPipe:

1. The program implements three quality filters: (i) putative bisulfite failures can be detected, (ii) the false positive rate can be controlled by means of the Phred Scores, (iii) SNV (single nucleotide variants) can be detected and removed.
2. Usage of a "seed extension" method applied to the Bowtie alignments allowing to map a higher number of reads without compromising the mapping quality.
3. Extensive output options including all possible cytosine sequence contexts (CG, CHG and CHH; where H is A, T or C) and the possibility to join the information from both strands (useful for the detection of hemi-methylation).
4. Complete statistics of the whole process, including aligned reads, discarded reads, discarded positions, chromosome data coverage, etc.
5. The memory and CPU needs can be adapted to the user's computer resources.
6. Fastq input files are accepted in zip, gzip, bzip2 or uncompressed.

1. Installation of NGSmethPipe (Unix platforms)

- Install the Perl interpreter (http://www.perl.org/) and resolve two dependencies. From the command-line (with super-user mode):
  - ✓ *Bundle::Thread* (perl –MCPAN –e "install Bundle::Thread").
  - ✓ *IO::Uncompress::AnyUncompress* (perl -MCPAN -e "install IO::Uncompress::AnyUncompress).
- Download and install the Bowtie aligner http://bowtie-bio.sourceforge.net/index.shtml (for example in /home/user/Bowtie)
- Download NGSmethPipe http://bioinfo2.ugr.es/NGSmethPipe/downloads/NGSmethPipe.tgz. Uncompress the file (tar xzvf NGSmethPipe.tgz) to a folder of your choice (e.g. /home/user/NGSmethPipe).
- Download the examples http://bioinfo2.ugr.es/NGSmethPipe/downloads/Examples.tgz (for example to

(home/user/Examples/). The compressed example file includes all necessary sequences and reads to test the available options included in the program.

- Download the genome sequences into a folder of your choice (/home/user/seq/assembly). The genome sequences can be downloaded for example from the UCSC genome browser http://hgdownload.cse.ucsc.edu/downloads.html. Note that this step is not necessary to test the program.

2. Quick start

Within the example directory, three folders can be found: "h1_exampleChr22" for human assembly (MethylC-Seq reads), "wtshoots_example" for *Arabidopsis thaliana* assembly (BS-Seq reads, sequenced with tags) and "wa09fibro_exampleChr21" for pair-end reads example. Next, we will give some examples to explain how to use the program:

i. Quick Launch (default Options)

'perl NGSmethPipe.pl seqDir=*/home/user/Examples/h1_exampleChr22/seqDir/* inDir=*/home/user/Examples/h1_exampleChr22/inDir/* bowtieDir=*/home/user/bowtie/* '

This command will run the Meta-script with default options, which will launch sequentially all the scripts that compose the program, from the Bowtie indexer to the methylation level extraction.

ii. ConfigFile Launch (default Options)

'perl NGSmethPipe.pl /home/user/Examples/h1_exampleChr22/NGSmethPipeConfigFile_h1.dat'

Both example folders contain a configuration file with *.dat extension that allows to set the parameters. Note that the file is just accepted by the Meta-script. See **Parameters** section for more information.

iii. Alignment Options (l=25 n=2)

'perl NGSmethPipe.pl seqDir=*/home/user/Examples/h1_exampleChr22/seqDir/* inDir=*/home/user/Examples/h1_exampleChr22/inDir/* bowtieDir=*/home/user/bowtie/* l=25 n=2 '

With these parameters Bowtie will search for seed alignments with 25 bp length allowing 2 mismatches within the seed (n) (see Alignment selection).

iv. Parallelization Options (p=6 maxChunk=10000)

perl NGSmethPipe.pl seqDir=*/home/user/Examples/h1_exampleChr22/seqDir/* inDir=*/home/user/ Examples/h1_exampleChr22/inDir/* bowtieDir=*/home/user/bowtie/* p=6 maxChunk=10000

The parallelization of the process and the memory usage can be controlled, using 'p' (number of threads) and 'maxChunk' (number of reads processed by each thread).

v. Reads with Tags

'perl NGSmethPipe.pl seqDir=*/home/user/Examples/wtshoots_example/seqDir/*
inDir=*/home/user/reads/Examples/wtshoots_example/inDir/*
bowtieDir=*/home/user/bowtie/* fw=TCTGT rc=TCCAT'

In this example, we are going to align reads sequenced with tags. Example reads can be found in the folder wtshoots_example (reads extracted from Feng et al.). The user should specify the sequences of the forward and reverse tags, 'fw' and 'rc' respectively.

vi. Pair-End reads

'perl NGSmethPipe.pl seqDir=*/home/user/Examples/wa09fibro_exampleChr21/seqDir/*
inDir=*/home/user/reads/Examples/wa09fibro_exampleChr21/inDir/*
bowtieDir=*/home/user/bowtie/ m1=_1 m2=_2 I=250 X=300*'

In case of pair-end reads (extracted from Laurent dataset), the user needs to specify the mates file endings ('m1' and 'm2') and optionally the minimum and maximum insert size for valid pair-end alignments can be selected.

vii. Output ratios Options

'perl NGSmethPipe.pl
seqDir=*/home/user/Examples/h1_exampleChr22/seqDir/*inDir=*/home/user/Examples/h1_exampleChr22/inDir/* bowtieDir=*/home/user/bowtie/* minQ=40
methNonCpGs=3 pattern=CG uniStrand=N'

This last example shows how to obtain the final methylation levels. The options include the quality filters as well as the sequence context to be studied. In the example, the 'minQ' parameter is set to 40. This means that all positions with less than a Phred Score of 40 (probability of an erroneous base call less than 0.0001) are ignored. Setting the methNonCpGs to 3, the Pipe will discard all reads with more than 3 methylated non-CpG contexts (default is more than 90% methylated non-CpG contexts: methNonCpGs=0.9). The parameter 'pattern' allows to choose the sequence context which should be processed (on the example just CG context, ALL by default) and 'uniStrand' specifies how the methylation ratios will be printed out (uniStrand=N, will return the methylation ratios for each strand separately).

3. Running NGSmethPipe

NGSmethPipe is composed of three perl scripts. This modular architecture makes it possible to launch just parts of the whole pipeline (for example to test different parameters). Nevertheless, a whole workflow can be easily launched by means of a meta-script. All scripts display a help text when launched without arguments.

- Running the Meta-Script:

  'perl NGSmethPipe.pl seqDir=<> inDir=<> bowtieDir=<> [Options]'
  See NGSmethPipe **Parameters**.

- Running each step independently:

  a. Building 3-letter alphabet index:
  'perl NGSmethPipeIndex.pl seqDir=<> bowtieDir=<> [Options]'
  See NGSmethPipe **Parameters**..

  b. Aligning reads:
  'perl NGSmethPipeAlign.pl seqDir=<> bowtieDir=<> inDir=<> [Options]'
  See NGSmethPipe **Parameters**.

  c. Retrieving methylation levels:
  'perl NGSmethPipeRatios.pl seqDir=<> alignDir=<> [Options]'
  See NGSmethPipe. **Parameters**.

4. Parameters

Each script has its own parameter set; however, some are used in more than one-step. The parameters are supplied with this format: <*name of the option*>=<*value*>.

- **NGSmethPipeIndex:**
  - Required:
    - **seqDir** → path of the reference genome sequences, in fasta format (/home/user/Examples/h1_exampleChr22/seqDir/ or /home/user/Examples/wtshoots_example/seqDir/ in our examples). NGSmethPipe does not allow multifasta format.
    - **bowtieDir** → path of the Bowtie executables (/home/user/bowtie/ in our example).
  - Optional:
    - **p** → number of threads to parallelize the process (by default *p=4*).

- **NGSmethPipeAlign:**
  - Required:
    - **seqDir** → path of the reference genome sequences, in fasta format (/home/user/Examples/h1_exampleChr22/seqDir/ or /home/user/Examples/wtshoots_example/seqDir/ in our examples). NGSmethPipe does not allow multifasta format.
    - **bowtieDir** → path of the Bowtie executable files (/home/user/bowtie in our example).
    - **inDir** → path with the reads (/home/user/Examples/h1_exampleChr22/inDir/ or

/home/user/Examples/wtshoots_example/inDir/). FastQ files must have *.fastq extension to be accepted as input by NGSmethPipe.

- Optional:
  - **qscore** → quality format used in the fastq files. Select between *phred33-quals*, *phred64-quals*, *solexa-quals* or *solexa1.3-quals*. The default is *NA:* the pipeline will try to detect automatically the quality format used on the files. Determining the quality format is important for running Bowtie with quality options, otherwise Bowtie will work with false quality values.
  - **extension** → the pipeline will try to disambiguate reads with multiple mappings (by default Y).
  - **outDir** → output folder (by default all output files will be written into the inDir directory).
  - **PHREDtrim** → Minimum Phred score: if a base call has a lower value, the read is trimmed at this position (default: 2).
  - **adapter** → the 3-prime adapter sequence. If the sequence is not detected an iterative search is performed, reducing the sequence in 1 pb from the 3-prime end and looking for an exact match until the sequence is found or finish its length (by default null).
  - **p** → number of threads to parallelize the process (by default *p=4*).
  - **maxChunk** → is the maximum number of reads processed by each thread (by default maxChunk=20.000). The maximum number of reads should be manipulated carefully, as it could increase drastically the required memory.
  - **chunkmbs** → maximum megabytes of memory used by Bowtie for each read (default chunkmbs=512).
  - **tags** → tag detection will be turned on specifying their sequences, reads processed with tags will be mapped to 4 Bowtie instances (non-directional reads):

    fw → forward tag sequence (by default fw=TCTGT).

    rc → reverse tag sequence (by default rc=TACCT).
  - **l** → l is the Bowtie "seed length" parameter (by default l=26).
  - **n** → maximum number of mismatches allowed within the seed (by default n=1).
  - **e** → maximum allowed total sum of quality values at all mismatch positions throughout the entire alignment (by default e=70).

        - **nomaqround** → Bowtie, as Maq, internally rounds base qualities to the nearest 10 and rounds qualities greater than 30 to 30. Rounding could be suppressed with nomaqround parameter (by default nomaqround=N).
        - **tryhard** → Try as hard as possible to find valid alignments when they exist, including paired-end alignments (by default tryhard=N).
        - **mates file endings** → File ends that differentiate the #1 mates from #2 mates. If the user specifies both mates file ends, the pair-end mode will be turned on.
            m1 → File ending of #1 mates (by default m1=null, example m1=_1).
            m2 → File ending of #2 mates (by default m2=null, example m2=_2).
        - **I** → minimum insert size for valid paired-end alignments (by default I=1).
        - **X**→ maximum insert size for valid paired-end alignments (by default X=500).

- NGSmethPipeRatios:
    - Required:
        - **seqDir** → path of the reference genome sequences, in fasta format (/home/user/Examples/h1_exampleChr22/seqDir/ or /home/user/wtshoots_example/seqDir/ in our examples). NGSmethPipe does not allow multifasta format.
        - **inDir** → path where the results from the align step has been kept.
    - Optional:
        - **p** → number of threads to parallelize the process (by default *p=4*).
        - **outDir** → output folder (by default all output files will be written into the inDir directory).
        - **qscore** → quality format used in the fastq files. Select between *phred33-quals*, *phred64-quals*, *solexa-quals* or *solexa1.3-quals*. The default is *NA:* the pipeline will try to detect automatically the quality format used on the files. Determining the quality format is important for running Bowtie with quality options, otherwise Bowtie will work with false quality values.
        - **minQ** → minimum allowed PHRED quality value for a particular base call. All positions with lower values are discarded (default minQ=20).
        - **methNonCpGs** → methylation outside the CpG context to detect putative bisufite failures. The value can be the

fraction of methylated non-CpG contexts within a read (values between 0 and 1) or the absolute number of methylated non-CpG contexts (integers higher than 1). Default value is 0.9, i.e. a read is discarded if more than 90% of its non-CpG contexts are methylated. A value of 0 will turn of the bisulfite check. Default value: 0.9.

- **minSNVperc** → fraction of reads that cover a given position indicating the existence of a SNV at this position. Default value 0.75 (more than three quartes of the reads at the position must indicate a SNV to discard the position).
- **pattern** → methylation contexts. The options are: CG, CHG, CHH or ALL (where H is any nucleotide except G) (by default pattern=CG).
- **uniStrand** → binary option (*Y* or *N*). This option selects the output format. The methylation levels can be presented as a weighted mean of the values from the Watson and Crick strands (sum of reads that indicate methylation on both strands divided by the total number of reads mapped) if the user select *Y*, or as the methylation levels for each strand separately with N (by default uniStrand=N). See **Output formats**.
- **bedOut** → results will be also given in bed format (by default N).
- **wigOut** → results will be also given in wig format (by default N).

- Meta-Script NGSmethPipe:

The parameters of the Meta-Script are inherited from the individual scripts that compose the pipeline. In the Meta-Script the options for each step could be provided to the program with a configuration file. The format of the configuration file is simple: one parameter per line using <parameter name>=<value>. A line starting with "#" will not be considered. For example:

```
#output directory
outDir=/home/user/h1_output/
#number of threads
p=4
#maximun chunks size
maxChunk=20000
#qscore range
qscore=NA
```

The Meta-Script using the configuration file can be launched as "perl NGSmethPipe.pl NGSmethPipeConfigFile.dat".

5. Output formats

- NGSmethPipeIndex:

  Bowtie indexes will be stored in the fasta sequences directory, specified by 'seqDir'. The script will create two genomic indexes, one for each strand: cytosines are converted to thymines in the Watson strand, and guanines to adenines in the Crick strand. The indexes files have *.ebwt extension, which is the output extension used by Bowtie.

- NGSmethPipeAlign:

  The output of this step will be stored in the reads directory by default, or in that specified by 'outDir'. The aligned reads are reported in files with **.align* extension (bowtie alignment format). The pair-end mode output will be the same as single-end mode, except a mate identifier at the end of the read id (/1 for the #1 mates and /2 for the #2 mates). The fields in each line are:

  - ID → original identifier of the read
  - Strand → the strand where the read maps (+ or -)
  - Chromosome → chromosome where the read maps (chr1,chr2,chrX, etc…)
  - Start position → start position of the read in the chromosome (0-based). The coordinate refers to the Watson-strand
  - Read → The sequence of the read with its original alphabet. In case of Crick-strand reads, the sequence returned is the reverse complement of the original sequence
  - Quality line → Encoded Phred Quality Scores
  - Number of alignments for the ID → NGSmethPipe only retrieves unique alignments (the value will always be 0).
  - Mismatches line → Mismatches positions specifying base in the reference and in the read

The log file keeps the process running time, the used parameters, the number of processed reads, number of bowtie unique mapped reads, number of ambiguous reads retrieved by bowtie and aligned reads.

- NGSmethPipeRatios:

  The methylation output files are named after the analyzed pattern (methylation context). The output file format depends on whether the uniStrand option is set or not.

  - uniStrand=N
    - Chromosome → the chromosome

        - Start position → start position of the methylation context (1-based), in the positive strand
        - End position → end position (1-based)
        - Strand → the strand in which the methylation is detected
        - Number of reads → number of reads covering the cytosine position
        - Methylation ratio → methylation level of the cytosine, calculated as the number of cytosines not converted to thymines divided by the number of reads mapped to the position

    - uniStrand=Y (by default)
        - Chromosome → the chromosome
        - Start position → start position of the methylation context (1-based), in the positive strand
        - End position → end position (1-based)
        - Number of reads → number of reads covering the context, on both strands
        - Methylation ratio → methylation level of the cytosine, calculated as the number of cytosines not converted to thymines divided by the number of reads mapped to the position, in both strands
        - Methylation difference → the absolute difference between the methylation ratios on each strand
        - Number of reads on the Watson strand
        - Methylation level on the Watson strand
        - Number of reads on the Crick strand
        - Methylation level on the Crick strand

The SNVs files depend on the context selected, the SNVs detected will correspond with the methylation positions of the patterns (C and G in the CG context or C in a CAA pattern for example).

    - SNV output
        - Context → context where the SNV has been detected
        - Chromosome
        - Strand
        - Start position → start position of the methylation context (1-based), in the positive strand
        - End position → end position (1-based)
        - Number of reads
        - Adenine frequency
        - Thymine frequency
        - Cytosine frequency
        - Guanine frequency

- Unknown base frequency
- SNV frequency → frequency of bases that differ from the reference

The log file stores the number of reads discarded by the bisulfite check, the number of positions discarded by means of the Q value threshold and the coverage (% of positions covered) as a function of chromosome, strand and context.

6. NGSmethPipe implementation and specific functions for each step

This section will explain briefly the implementation of each script, trying to offer the user a clear idea of how the processes work.

- NGSmethPipeIndex

The script will generate the indexes for two different alphabets. Both indexes are necessary, because the BS-Seq protocol generates forward and reverse reads for each strand, resulting in reads with cytosines transformed to thymines (forward reads) and guanines transformed to adenines (reverse reads). The process is parallelized by a static queue (Thread::Queue module).

During the process, temporal files for each chromosome and alphabet will be generated with **.BSfasta* extension. Then two processes will run at once building both indexes, one for each alphabet.

Finally, the temporal files will be deleted and the index files with **.ebwt* extension will be stored in the folder with the genomic reference. The original fasta files must be retained in the directory (they will be necessary to compare the nucleotides found in the reads to those in the genomic sequence, which allows detecting the methylation state).

- NGSmethPipeAlign

The explanation of this script is divided into the possible steps, depending on the parameters selected in the command-line input. All the preprocessing steps (uncompress, Q score detection, quality trimmed and adapter trimmed) are parallelized by the Threads::Queue module.

a. Uncompress step
   Each selected thread will decompress files with the IO::Uncompress::AnyUncompress module (bzip2, gzip or zip). The 'inDir' folder will keep the compressed original

files and the temporally uncompressed fastq files, the temporal files will be deleted after processing the files.

b. Detecting quality score
   For each fastq file, the first 50 quality lines will be extracted and the quality format will be detected by means of the Q values range. It is however recommended to specify the format manually with the 'qscore' parameter.

c. Quality trimming
   The user can introduce a quality value with the 'PHREDtrim' parameter. The reads will be trimmed at the first base having a equal or lower quality. This option removes the low quality end of the reads.

d. Trim 3-prime adapter sequence
   The 'adapter' option allows to provide the sequence of the 3-prime adapter, activating an iterative search of this sequence at the 3' end of the read. If the entire sequence is found, it will be trimmed from the read. If it is not found, , the adapter will be trimmed sequentially by 1 pb from its 3-prime end and searched again. The program searches for an exact match of the adapter sequence, possibly failing to trim the adapter sequence in reads with low quality ends.

The main thread will continuously preparing and distributing to the worker threads a given number of reads defined by the 'maxChunk' option. The worker threads will (i) convert them into the appropriate alphabet, (ii) retain the changed positions, (iii) launch Bowtie and (iv) select the best alignments (see Alignment selection).

e. Pair-end mates
   If the user specifies the file endings for #1 and #2 mates, the pair-end mode will be turned on. Pair-end mode doesn't support Tags alignment and in the '*seed extension mode*' each mate of the pair must have the longest extension to disambiguate the pair alignment. There are two additional Bowtie parameters to control the valid pair mappings: I and X.

f. Tags
   When the tag sequences are provided, the program will search for the forward and reverse tags within the reads. If a tag is detected, the program will remove the sequence in

order to avoid errors during the alignment. Then every read will be align to all possible strands and orientations.

g. Alignment selection
NGSmethPipe offers a new way for disambiguating reads with multiple alignments based on the seed extension: (i) Bowtie finds the best alignments for each strand using by default the following command: `bowtie –"qscore" -e 70 -l 36 -n 1 –best –strata –k 2 –nomaqround`. (ii) The Pipe will select these alignments with the lowest e value. (iii) Then, the seed is extended in each possible alignment until the next mismatch occurs, selecting the longest alignment seed extension for each read.

Finally, the pipe converts the selected reads back to the original 4-letter alphabet. The alignment step is stored into files sorted by chromosome names with the extension **.align* (see **Output formats**).

- NGSmethPipeRatios

The methylation levels for each cytosine context are detected comparing the genomic fasta sequences with the alignments obtained in the previous step. This step includes three quality control steps:

a. Bisulfite check
One of the possible experimental errors in methylation profiling is that the bisulfite might fail to act in some reads. Trying to reduce this error source, the program checks the number of methylated non-CpG contexts in the reads at the beginning of the process, discarding those reads with a high number (or percentage) of methylated non-CpG contexts.

b. Quality score value check
The quality value is checked for each position discarding those with lower Phred scores than a value given by 'minQ'. The discarded positions allow to control the false positive rate caused by sequencing errors.

c. SNV detection
Another error source is the presence of SNVs (single nucleotide variants) that can be erroneously taken as an unmethylated cytosine in the case of an C/T SNV. These positions will be discarded if the fraction of reads indicating the presence of a SNV is higher than the threshold set by the 'minSNVperc' parameter. This threshold controls the

fraction of non-G in the reads mapped into the complementary strand.

The methylation levels output will be ordered by the methylation pattern in files with **.output* extension (see **Output formats**).