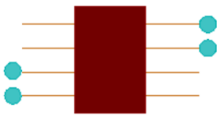

HPG-Methyl – User Guide – v3.1

HPG-Methyl is an ultrafast and highly sensitive Next-Generation Sequencing (NGS) read mapper and methylation context extractor. Compared with other current mapping and methylation extraction tools, HPG-Methyl offers better sensitivity and shorter execution times even for long reads.

Since the files generated by HPG-Methyl are fully compatible with the files generated by other popular tools, it can be used as a drop-in replacement to accelerate existing methylation analysis pipelines.

Contact any of the following developers for any enquiry:

- Juanma Orduña (juan.orduna@uv.es).
- Mariano Pérez (mariano.perez@uv.es).
- Ricardo Olanda (ricardo.olanda@uv.es).
- César González (cesar.gonzalez-segura@uv.es).

Changelog

The following list contains the releases of HPG-Methyl available on the repository. This document is relevant for the 3.1 and 3.0 versions, older versions are unsupported but available to download.

v3.1 (current)

- Now methylation context information is stored in the output BAM file, using the same format as Bismark (http://www.bioinformatics.babraham.ac.uk/projects/bismark/).
- Minor bugfixes and optimizations.

v3

- Optimized using bidirectional Burrows-Wheeler Transform and a modified parallel pipeline.

v2

- Improved version with CpG context information switch available as a command line option.

v1

- Original version with CpG context information disabled.

1) Quick Start Guide

1. Downloading and extracting the software

HPG-Methyl runs on a terminal. Get into the destination directory and download the latest compiled version of HPG-Methyl from the public repository:

```
$ sftp anonymous@clariano.uv.es:/src/version_3_1-HPG-
Methyl2/hpg_methyl_3_1_x86_64.tar.gz .
```

When prompted for a password for any download, use `anonymous`.

Extract the contents of the tarball:

```
$ tar xvzf hpg_methyl_3_1_x86_64.tar.gz
```

2. Process the reference genome

You need a reference genome in FASTA format to compare your sequenced reads with. If you do not have an available genome, an example *homo sapiens* genome can be downloaded from the repository. Create a directory to store your genome and download:

```
$ mkdir genome/
```

```
$ sftp anonymous@clariano.uv.es:/datasets/Homo_sapiens.GRCh37.68.dna.fa genome/
```

The example genome dataset size is **3,15 GB**. Make sure that your connection is stable and fast enough to download the file.

HPG-Methyl needs a special index to access the reference genome, which has to be generated before doing any read mapping. This process has to be done **only once per reference genome**.

Create a directory to store the index genome, and let HPG-Methyl generate it. This process will take a while depending on the CPU processing power of your machine. Extensive information on the HPG-Methyl command line options is available at the end of this document:

```
$ mkdir genome/index/
```

```
$ ./hpg-aligner build-index -g ./genome/Homo_sapiens.GRCh37.68.dna.fa -i
genome/index/ -r 10 --bs-index
```

The genome index files take a considerable amount of space. The directory holding the index must have at least **250 GB** of available free space. Storing the index in fast media is recommended (like a solid state drive or a fast network drive).

3. Mapping and methylation analysis of the sequenced reads

When the genome index creation has been completed, sequenced read files can be mapped to the reference genome and the methylation context analysis can be performed. The read files must be in the FASTQ format and must contain only single-ended reads. Paired-ended reads are unsupported. If you don't have any available sequenced read data, download an example read from the repository:

```
$ sftp anonymous@clariano.uv.es:/datasets/real/SRR309230_1_075nt_15M.fastq
genome/
```

The example sequenced read file size is **3,31 GB**. Make sure that your connection is stable and fast enough to download the file.

When the download has been completed, run HPG-Methyl to align the sequence with the reference genome and generate the methylation context analysis. This process will create three output files:

- **alignments.bam** Binary SAM file containing the aligned reads and methylation context information for each read. The file can be converted to a text SAM file using *samtools*. The methylation context information is stored in SAM tags, following the *Bismark* tag format.
- **CpG_context.txt / CHG_context.txt / CHH_context.txt / MUT_context.txt** CSV files containing the context-dependent methylation analysis information.

HPG-Methyl has a multi-threaded pipeline and can fully exploit all the processing cores in your computer platform. Use the following command to run the process with 4 threads:

```
$ ./hpg-aligner bs -i genome/index/ -f genome/SRR309230_1_075nt_15M.fastq -o .
--cpu-threads 4 –write-mcontext
```

Using more threads only improves performance if your system has enough CPU cores or threads to do so. The recommendation is to run as many threads as virtual processors are available in the machine.

2) Building

1. Downloading the source and file structure

First, download the source tarball from the SFTP repository. This manual corresponds to the latest version available, HPG-Methyl v3.1, located on the `/src/version_3_1-HPG-Methyl2` directory. The complete connection details to download the tarball are:

Address	`clariano.uv.es`
Username	`anonymous`
Password	`anonymous`
Protocol	SFTP
Complete path	`sftp://anonymous:anonymous@clariano.uv.es/src/version_3_1-HPG-Methyl2/hpg-methyl2_v3_1.tar.gz`

Extract the source tarball into the desired directory as usual:

```
$ tar xvzf hpg-methyl2_v3_1.tar.gz
```

The extracted tarball has the following directory structure:

`/`	Root directory containing the SCons build script, README and COPYING notices.
`/src`	Source code of HPG-Methyl
`/lib`	Source code of OpenCB bioformats and common libraries. The version of this libraries included with HPG-Methyl is modified and might not be compatible with the version available at GitHub.
`/bin`	Directory where the compiled binary will be generated.

2. Compilation and installation

Building HPG-Methyl requires having a working installation of GCC 4.4 or greater, SCons and the following software packages, which can be installed using `apt-get` on Ubuntu or Debian based distributions or `yum` on Red Hat based distributions:

Ubuntu / Debian	Fedora / CentOS / Red Hat
zlib1g-dev	zlib-devel
libcurl4-gnutls-dev	libcurl-devel
libxml2-dev	libxml2-devel
libncurses5-dev	ncurses-devel
libgsl0-dev	gsl-devel
check	check-devel

When all packages are installed, get into the directory that was created and build the software using the command:

```
$ scons
```

When the process finishes, the compiled binary will be available at the `/bin` folder, and it can be run from there. If a system-wide installation of the tool is desired, copy the binary to the applications directory:

```
$ cd bin/
```

```
$ cp ./hpg-methyl /user/bin/hpg-methyl
```

3. Extending and debugging

This section is only relevant if you plan to extend or debug HPG-Methyl. Otherwise, you can skip it.

In order to make changes to HPG-Methyl, modify the code on the `/src` or `/lib` directories and build the toolchain. To compile without optimizations and with debug symbols, use the command:

```
$ scons debug=1
```

The source is bundled with Eclipse and Visual Studio Code projects solutions to ease the development process. Under Eclipse, import the tarball directory into the workspace to work with the source code. Under Visual Studio Code, use the Open Folder command and open the tarball directory.

The Eclipse solution requires the SConsolidator plugin to integrate with SCons. The Visual Studio Code doesn't have any additional dependencies. Check the included VS Code tasks for building, debugging and running information.

3) Running

The syntax for running HPG-Methyl is:

```
$ hpg-methyl <mode> <options>
```

The available modes are **`bs`**, used to map bisulphite-treated sequences and **`build-index`**, used to create the Burrows-Wheeler index of the reference genome.

The bisulphite mapping mode aligns the sequenced reads using the Burrows-Wheeler Transform, the Smith-Waterman algorithm and several post-process and pre-process filters. The details of the pipeline are beyond the scope of this manual, and they are available on an article. Contact the developers for more information.

The pipeline has been designed to be easily tunable to any particular application, and most algorithm parameters can be adjusted through command line options. For most applications, the default values should suffice since all algorithms have been throughly tested and optimized to obtain the best performance in a variety of situations.

Following is the description of the different algorithms on HPG-Methyl and their tunable parameters. The annex available at the end of this document shows the complete list of optional parameters and their description.

1. Smith-Waterman Algorithm parameters

The Smith-Waterman algorithm is used to perform the mapping on reads that were not fully mapped on the Burrows-Wheeler stage, allowing gaps and mismatches with the reference genome.

The weight matrix for the Smith-Waterman algorithm can be tuned through the `sw-match`, `sw-mismatch`, `sw-gap-open`, `sw-gap-extend` and `sw-min-score` optional parameters. The default SWA weight table has been optimized by hand to have the best performance on most cases.

Minimum score	0,8

Default SWA weight table.

2. Filtering and seeding parameters

Between the different mapping stages and at the end of the pipeline, there are several filtering and seeding stages which are used to accelerate the mapping process.

The seeding and CAL stages default parameters are shown below:

Minimum CAL size	30
Maximum distance between seeds	100
CAL flank size	5
Number of seeds per read	10
CAL umbral length factor	4
Read minimum discard length	100
Read maximum inner gap	-1

Seeding and CAL stages default parameters

When the mapping (and optionally methylation analysis) have been completed, there is a last filtering stage where the alignments being written to the BAM file can be filtered. The default parameters are shown below:

Report all hits	No
Report only best hits	Yes
Report only *n* first best hits	No
Report only *n* first hits	No

Reporting stage default parameters

3. Methylation context analysis parameters

If the `write-mcontext` command line option is enabled, the last pipeline stage will obtain the methylation status of the current read and will store it on the tags of the output BAM file. After all

reads have been mapped, the context-dependent methylation information is stored into text files.

When generating the Burrows-Wheeler index, use the `--bs-index` option to create an index compatible with the methylation analysis stage.

Annex: Complete list of command line options

`-f / --fq / --fastq <path>` Path to the sequenced read file in FASTQ format.

`-i / --bwt-index <path>`	Path to the BWT index directory. It must have been previously generated using the `build-index` option.
`-l / --log-level <level>`	Console debug message level. <level> must be one of the following numeric values, from more verbose to less verbose. • 0: Show all debug messages (used to troubleshoot the application). • 1: Show information messages, warnings and errors. This is the default log level. • 2: Report only warnings or worse. • 3: Report only errors or worse. • 4: Report only fatal errors. • 5: Don't log any messages.
`--report-all`	Report all alignments.
`-o / --outdir`	Path to the output directory.
`--cpu-threads`	Number of CPU threads to use.
`-r / --index-ratio`	Index compression ratio for the Burrows-Wheeler Transform. Used when creating the genome index.
`--min-cal-size`	Minimum CAL size.
`--max-distance-seeds`	Maximum distance between seeds.
`--read-batch-size`	Read batch size in bytes.
`--write-batch-size`	Write batch size in bytes.
`--cal-flank-size`	Flank length for CALs.
`--sw-match`	Match value for Smith-Waterman algorithm.
`--sw-mismatch`	Mismatch value for Smith-Waterman algorithm.
`--sw-gap-open`	Gap open penalty for Smith-Waterman algorithm.
`--sw-gap-extend`	Gap extend penalty for Smith-Waterman algorithm.
`--sw-min-score`	Minimum score for valid mappings.
`-t / --time`	Profile timing of all the application stages and log the elapsed time when the execution finishes through the console.
`-s / --stats`	Generate and save execution statistics.
`-h / --help`	Print the application help.
`--prefix`	File prefix name.
`-g / --ref-genome`	Path to the reference genome.
`--report-n-best <n>`	Report the <n> best alignments.
`--report-n-hits <n>`	Report <n> hits.
`--num-seeds <n>`	Number of seeds per read.

`--min-num-seeds <n>`	Minimum number of seeds to create a CAL (if -1, the maxixum will be taken).
`--filter-read-mappings <n>`	Reads that map in more than <n> locations are discarded.
`--filter-seed-mappings <n>`	Seeds that map in more than <n> locations are discarded.
`--report-best`	Report all alignments with best score.
`--bs-index`	Indicate the use of bisulphite generation of the index. Used when creating the genome index.
`--write-mcontext`	Save the methylation context report files (CHG, CHH, CpG, MUT) and embed the alignment methylation information in the BAM file.
`--umbral-cal-length-factor <f>`	CAL is accepted as a candidate if it has been mapped more than CAL size / <f>
`--min-read-discard <n>`	Discard small anchors for reads larger or equal than <n>.
`--max-inner-gap <n>`	Recursive Burrows-Wheeler stops when inner gap is lower than <n>.