Processing RRBS samples with DMAP: a User Guide

Peter A. Stockwell

Dept. of Pathology, University of Otago.

Introduction: This guide describes the steps needed to use locally written and other software tools for analysing differential methylation on multiple individuals after RRBS runs. Much of this is also adaptable to whole genome bisulphite sequence data (WGBS).

Note that the DMAP2 package (`https://github.com/peterstockwell/DMAP2`) is a series of scripts designed to simplify running DMAP and installing the necessary genomic and annotation files. DMAP2 renders some of these instructions obsolete: notes are given here where this is likely.

1. Software required and installation:

You may already be using a system which has the necessary programs available. Otherwise install the prerequisites with the following:

DMAP works with BAM or SAM alignment files generated by bisulphite aligners, we have found `bismark` and `bsmapz` to be the most useful.

bismark: obtain the `bismark` distribution by downloading from:

https://github.com/FelixKreuger/Bismark

selecting '`Code`' and '`Download ZIP`'. Unpack the file with

```
unzip Bismark-master
```

which will produce a directory `Bismark-master` containing the executables which you should copy to a directory that makes them easy to execute. This may depend on your privilege and access to system directories. If you have appropriate privilege, then `/usr/local/bin/` is the usual location, otherwise you should use your own `bin` directory. In the first case (privilege) use commands like:

```
cd Bismark-master
sudo cp bismark /usr/local/bin/
```

otherwise (no privilege):

```
cd Bismark-master
cp bismark ~/bin/
```

The complete list of executables to be copied is:

```
NOMe_filtering
bam2nuc
bismark
bismark2bedGraph
bismark2report
bismark2summary
bismark_genome_preparation
bismark_methylation_extractor
coverage2cytosine
deduplicate_bismark
filter_non_conversion
methylation_consistency
```

An alternative source for `bismark` is to visit

```
https://www.bioinformatics.babraham.ac.uk/projects/download.html#bismark
```

which should produce a download like `bismark_v0.22.3.tar.gz`.
Unpack this with

```
gzip –dc bismark_v0.22.3.tar.gz | tar –xvf –
```

to produce a directory `bismark_v0.22.3` containing documentation as a .pdf and perl executables for `bismark`, `bismark_genome_preparation` and `bismark_methyation_extractor`. The executables should be put into an appropriate directory on your defined path – the usual choice is `/usr/local/bin/` or your own `~/bin/` directory, as above. Note that the `bismark` release from this source is not the latest, the `github.com` source is to be preferred.

bowtie2: `bismark` now uses bowtie2 as the aligner. Obtain this from:

```
http://bowtie-bio.sourceforge.net/bowtie2/
```

then

```
unzip bowtie2-2.5.1-source.zip
cd bowtie2-2.5.1
make
```

which requires a functional C++ compiler, or for a pre-compiled version:

```
unzip bowtie2-2.5.1-linux-x86_64.zip
cd bowtie2-2.5.1-linux-x86_64.zip
```

noting that '`linux`' can be replaced with '`macos`' etc. for other versions.

As before, the executables should be copied to an appropriate directory, either `/usr/local/bin/`, if accessible, or your own `~/bin/` with:

```
sudo cp bowtie2 bowtie2-align-[ls] /usr/local/bin/
sudo cp bowtie2-build bowtie2-build-[ls] /usr/local/bin/
sudo cp bowtie2-inspect bowtie2-inspect-[ls] /usr/local/bin/
```

or

```
cp bowtie2 bowtie2-align-[ls] ~/bin/
cp bowtie2-build bowtie2-build-[ls] ~/bin/
cp bowtie2-inspect bowtie2-inspect-[ls] ~/bin/
```

bsmapz: available from `https://github.com/zyndagj/BSMAPz`. Select the '`Latest`' option under '`Releases`', then '`Source code (tar.gz)`' under '`Assets`' to download '`BSMAPz-1.1.3.tar.gz`'. Unpack this with:

```
gzip -dc BSMAPz-1.1.3.tar.gz | tar -xvf -
```

producing a directory `BSMAPz-1.1.3` in which you can build the package with:

```
cd BSMAPz-1.1.3.tar.gz
make
```

As before, the executable should be copied to an appropriate directory, either `/usr/local/bin/`, if accessible, or your own `~/bin/` with:

```
sudo cp bsmapz /usr/local/bin/
```

or

```
cp bsmapz ~/bin/
```

Note: see (b) Genome preparation below for `bsmapz` genome requirements.

samtools: needed by `bismark` in order to generate bam alignment files: available from various sources. A convenient download is:

```
https://www.htslib.org/download/
```

selecting the '`samtools-1.x`' link where '`x`' is the version. The downloaded file is unpacked with:

```
bzip2 -dc samtools-1.x.tar.bz2 | tar -xvf -
```

if you have sudo privilege followed by:

```
cd samtools-1.x/
./configure
make
sudo make install
```

alternatively, without sudo privilege:

```
cd samtools-1.x/
./configure --prefix=$HOME
make
```

```
make install
```

which will install the samtools executables into your own `bin` directory.

fastqc: obtained from

```
http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
```

and unpacked with:

```
unzip fastqc_v0.10.1.zip
cd FastQC
chmod a+x fastqc
```

if you have sudo privilege

```
sudo ln –s fastqc /usr/local/bin/fastqc
```

otherwise

```
ln -s fastqc ~/bin/
```

zlib: needed by the `cleanadaptors` and `diffmeth` programs, preferably 1.2.5 or later, in order to perform compression/decompression on the fly. This is available from `http://www.zlib.net` at the link under ' **zlib** source code, version 1.2.13, tar.gz' format/ and can be unpacked and built with commands like:

```
gzip -dc zlib-1.2.13.tar.gz | tar -xvf -
cd zlib-1.2.13
./configure
make
sudo make install
```

which will install files in /usr/local. The option exists for installing elsewhere by using

```
./configure --prefix=<somewhereIcanwrite>
```

if you don't have sudo access to `/usr/local`: `<somewhereIcanwrite>` being some directory to which you have access. Some changes will be needed in the DMAP make scripts in order to accommodate this.

DMAP: now distributed from github and obtained with the command:

```
git clone https://github.com/peterstockwell/DMAP
```

producing a directory `DMAP` or by going to the github respository at

```
https://github.com/peterstockwell/DMAP
```

and using the '`Code`' and '`Download ZIP`' links to get the file `DMAP-master.zip` which can be unpacked with:

```
unzip DMAP-master.zip
```

creating a directory `DMAP-master`.

In either case `cd` into the directory (`DMAP/src` or `DMAP-master/src`) and execute

```
make
```

Depending on your C compiler, there may be a number of warnings, these can usually be ignored. The most useful executables that will be compiled are `cleanadaptors`, `diffmeth` and `identgeneloc` which should be put into `/usr/local/bin` if accessible to you, otherwise put them into your own top level `bin` directory or somewhere else that is accessible.

If compiling fails through the lack of zlib, then you should obtain and install this (see above). If that is not possible then you can build the program without zlib, but it will not be able to read BAM files and `cleanadaptors` won't be able to read or write `.gz` compressed fastq files. The command:

```
make nozlib
```

will compile these programs without the compression options.

fastx_trimmer: part of the the `fastx-toolkit` and `libgtextutils`. Note that functions now available in the DMAP `cleanadaptors` program make `fastx_trimmer` superfluous, but if you do need it, then download the latest source versions from `http://hannonlab.cshl.edu/fastx_toolkit/download.html` and build with:

```
gzip -dc tars/libgtextutils-0.7.tar.gz | tar -xvf -
cd libgtextutils-0.7
```

then, with sudo privilege

```
./configure
make CXXFLAGS="-std=c++03 -O1"
sudo make install
```

or

```
./configure --prefix=$HOME
make CXXFLAGS="-std=c++03 -O1"
make install
```

Then build the fastx_toolkit with:

```
cd ../
bzip2 –dc fastx_toolkit-0.0.14.tar.bz2
cd fastx_toolkit-0.0.14
```

with sudo privilege

```
./configure
make CXXFLAGS="-std=c++03 -O1"
sudo make install
```

or without

```
./configure --prefix=$HOME
make CXXFLAGS="-std=c++03 -O1"
make install
```

2. Files needed:

(a) **Genome:** fasta files for each chromosome, named `<Header>1.fa`, `<Header>2.fa` to `<Header>X.fa` & `<Header>Y.fa`. Normally bisulphite sequencing doesn't need the mitochondrial genome, though that is so small very few reads would map to it.

Note: DMAP2 has an improved method of preparing the genome files: I'd recommend checking and following that documentation. Otherwise various means of downloading these files are available, but one method is to visit:

```
https://www.ensembl.org/info/data/ftp/index.html
```

and for the species select '`FASTA`' in the '`DNA (FASTA)`' column. On the following page, select the files `<Species><Version>.dna.chromosome.1.fa.gz`, etc. to download the required chromosomes as compressed fasta files.

Another option is using command line ftp with:

```
ftp ftp.ensembl.org
```

log on as `anonymous` using email as password, then:

```
cd pub/release-109/ftp/homo_sapiens
```

obtain compressed fasta files with:

```
mget Homo_sapiens.GRCh37.71.dna*.fa.gz
```

and wait, informational messages should appear, then finally disconnect with:

```
bye
```

Then unpack the files with a command like:

```
gunzip *.fa.gz
```

which will inflate them by about 3 times, producing `.fa` files.

(b) **Genome preparation:** if you are using the bismark aligner, build the libraries in the directory where you have the genome sequence files with:

```
bismark_genome_preparation ./
```

which will take a while to run: best done overnight. It will create a directory with the name Bisulfite_Genome containing bowtie2 index files.

The bsmapz aligner requires the genome sequences as a single fasta file which doesn't need preprocessing.

Note: DMAP2 incorporates preparing the bismark genome into its dmap_index_build.sh script, along with producing the genome and feature information files for diffmeth and identgeneloc. It also configures bsmapz genome files as required for that aligner. You may find it preferable to use this method.

(c) **Feature table information:** the choice of format depends on your preferences and requirements. EMBL and Genbank are suitable for relating sequence regions to genes but with DMAP return gene names as ENSEMBL gene IDs (e.g. ENSG00000108001.13_2). SeqMonk files have the advantage of containing additional information on Transcription Start Sites and CpG islands and will return gene IDs as the more conventional HCNG names (e.g. EBF3). A further advantage is they give you the ability to choose specific biotypes (e.g. protein_coding) for features. GTF/GFF3 annotations contain considerable information about transcript variants, and also return conventional gene names. Further, DMAP allows you to select specific feature types and GTF attributes.

(i) EMBL/Genbank: DMAP requires 1 file for each chromosome. The files (sequence and annotation) are most easily obtained from ENSEMBL either from the web resource at https://www.ensembl.org/info/data/ftp/index.html as 'Annotated sequence (EMBL)' or 'Annotated sequence (GenBank)' or you could consider using command line utilities like ftp or curl to transfer the files at the command line, with a command (for Homo sapiens) like:

```
curl -O \
ftp.ensembl.org/pub/current_embl/homo_sapiens/Homo_sapiens.GRCh38.105.chromosome.[1-22].dat.gz
```

X and Y chromosomes will need [1-22] replaced by [X-Y]. To get Genbank files replace '/current_embl/' with '/current_genbank/'.

(ii) SeqMonk: 1 file/chromosome is needed for SeqMonk files which are uploaded using the SeqMonk mapped sequence data application, available from:

https://www.bioinformatics.babraham.ac.uk/projects/seqmonk/

Start the application and choose File > New Project then 'Import Genome From Server' and select and download the genome of choice. This should create a directory seqmonk_genomes, probably in your top level directory, containing the organism and genome version, which contain a series of files `1.dat`, `2.dat`, etc. These files contain the annotations for each chromosome.

(iii) GTF/GFF3: either obtained from `https://www.gencodegenes.org/` for mouse or human, by choosing the organism and genome version, then downloading the comprehensive gene annotation file as GTF or GFF3, which will download a file with a name like: `gencode.v39.annotation.gtf.gz` that you can move to an appropriate location.

The same annotation is available from Ensembl, either with a browser at

```
https://ftp.ensembl.org/pub/current_gtf/homo_sapiens/
```

There are two options: '`.chr.gtf.gz`' and '`.gtf.gz`' - the former restricts annotation to actual chromosomes while the latter includes unassigned contigs.

Alternatively the files can be downloaded at the command line with something like:

```
curl -f -O \
'https://ftp.ensembl.org/pub/current_gtf/homo_sapiens/Homo_sapiens.GRCh38.[100-120].chr.gtf.gz'
```

where '`100-120`' will get the actual current version (108 at the time of writing this) and may generate a list of errors for the versions it can't find.

3. NGS Files delivered from Illumina systems:

since an entire flowcell must be run with the same protocol, sequencing is frequently done in paired-end mode, where each read is primed from the 5’ adaptor and sequenced in the forward direction (READ1), then reprimed from the 3’ adaptor and sequenced back from the 3’ end (READ2). The process also involves sequencing the index on the 3’ adaptor in order to demultiplex multiple samples run in the same lane. For RRBS work, it is usual to work only with the 5’ read, since the short fragments will cause the forward and reverse reads to overlap frequently, biasing CpG mapping. It is possible to use the `FLASH` application (`https://sourceforge.net/projects/flashpage/files/`) to make longer reads for those showing overlaps leaving non-overlapping 5' and 3' reads, all of which can be used in mapping. In our experience, the extra effort did not result in significantly improved mapping, so when second reads have been provided, we ignore them.

(a) **Decompression:** the bulk of data files means that they are distributed as compressed data, indicated by having the suffix `.gz` appended to the name. Most processing (`fastqc`, adaptor trimming and mapping) can work directly with compressed files. Viewing the contents of compressed files is possible with commands like:

```
gzip –dc mydata_R1.fastq.gz | more
```

(b) **Quality trimming:** typical Illumina runs are now taken to 100 cycles or more but we note that the read quality with bisulphite treatment often deteriorates signficantly before then so that a check of the quality is needed. It is usual for **`fastqc`** to be run as part of the sequencing operation, and this information may be provided along with the data. If not you can run **`fastqc`** graphically or from the command line with:

```
fastqc --outdir qc mydata_R1.fastq.gz
```

which will write a range of quality assessments to a pre-existing directory `qc`. Within the resulting data will be a file `fastqc_report.html` which you can open with a web browser. The page contains a summary of the run, including the number of reads and a graphical display of the run quality. You can assess an appropriate length for hard trimming by looking at this graphic, generally where the median quality has fallen below Q20.

Let's assume that we want to trim to 90bp for this example, so that the command:

```
gzip -dc mydata_R1.fastq.gz | fastx_trimmer –l 90 -z \
  -o mydata_tr90_R1.fastq.gz
```

will generate a gzip compressed file of trimmed data or see `cleanadaptors` below which is capable of performing hard trimming and adaptor trimming simultaneously.

It is possible to use some trimming utilities (e.g. `fastq_quality_trimmer` [from `fastx_`toolkit, `-t` option] or `cleanadaptors` [`-q` option]) to scan reads from the 5' end and truncate them at the point where the quality falls below a threshold. In our experience, there are often cases where a single low Q value early in the read causes most or all of the read to be rejected, losing significant high quality read data.

(c) **Adaptor trimming:**

100bp or longer reads on a 40-220 RRBS library will frequently read into the 3' adaptor. Such reads need to have the adaptor trimmed from them or they will not map to the genome. A file of adaptor sequences is needed and we shall assume that this file is available locally for now: two formats can be used: (1) a simple text file with one adaptor sequence/line (`-i` option) or (2) a fasta file (`-I` option). `cleanadaptors` v1.22 and later have many additional features including the ability to read and write gzip-compressed fastq files, to hard trim reads and to quality trim. Adaptor trimming is usually complete now with a single pass through the `cleanadaptors` program with commands like:

```
cleanadaptors –I contam.fa –t 3 -x 20 -z –F mydata_tr90_R1.fastq.gz \
 -o mydata_tr90ad3pp_R1.fastq.gz
```

where:

`–t 3` trims 3 bases back from the adaptor to delete the C incorporated during library preparation

`–x 20` rejects any reads which are trimmed to less than 20 bp.

`cleanadaptors` can simplify the required commands for decompressing and hard trimming to something like:

```
cleanadaptors -I contam.fa -t 3 -x 20 -l 90 -z -F mydata_R1.fastq.gz \
-o mydata_tr90ad3pp_R1.fastq.gz
```

which will hard trim all reads to 90bp (`-l 90`), check for adaptors in fasta formatted `contam.fa`, rejecting reads which would be less than 20bp after these operations. Adaptor matching trims are further shortened by 3 bp. The input and output files are gzip compressed (`-z` option).

A further issue sometimes arises with RRBS data where adaptors may mismatch in the first two base positions which will prevent `cleanadaptors` from finding them, although `fastqc` scans will still observe them. This behaviour can be avoided by using the `-T` option which 5' trims the adaptor sequences before the run, and then increases the `-t` trim back by the same amount. `-T 2` seems to work appropriately for this.

Note: the DMAP2 package automatically performs adaptor trimming as part of the mapping process: you may prefer to use the DMAP2 scripts to simplify the process.

4. Mapping: we have usually used `bismark` to map the reads with the following:

```
bismark –N 1 <Path_to_Genome_directory> mydata_tr90ad3pp_R1.fastq
```

which will generate a BAM file `mydata_tr90ad3pp_bismark_bt2.bam` and a report file `mydata_tr90ad3pp_R1_bismark_bt2_report.txt`.

Bismark may generate messages like:

```
Chromosomal sequence could not be extracted for
  HWI-ST871:252:C21CFACXX:5:2102:16275:48963_1:N:0:ATTCCT
```

which indicate that a read had mapped at the very end of a chromosome, overlapping beyond the end. These messages can be ignored.

Mapping is performed for each of the different samples in the study – often we keep each of the sample files in its own directory, relying on Unix symbolic links (see `man ln`) to make the fastq data available locally without copying it, if appropriate. Examining the report files will show if the bisulphite chemistry had been successful: specifically the number and percentage of unique alignments and the C methylated in a CpG context should indicate this, whereas the non-CpG methylation (CHG, CHH, CN or CHN contexts) should be low, typically <5%).

Alternatively, mapping can be done with `bsmapz` which uses a variant of the SOAP aligner and which can be run with commands like:

```
bsmapz -p 4 -v 2 -a mydata_tr90ad3pp_R1.fastq \
 -d <genome_location>/Homo_sapiens.wholegenome.fa \
 -o mydata_tr90ad3pp_R1.fastq_bsmapz.bam
```

where:
 `-p 4` performs the mapping on 4 CPU cores
 `-v 2` controls the number of mismatches permitted in a read
 `-d` gives the genome, as a single fasta file and located in `<genome_location>`
 `-o` writes the mapping to `mydata_tr90ad3pp_R1.fastq_bsmapz.bam`

`bsmapz` doesn't provide non-CpG methylation figures.

Note: DMAP2 scripts simplify running these mapping operations.

5. Differential methylation:

It is not necessary for mapping files all to be in the same place for the next stage of analysis. Optionally if your file system allows it, symbolic links can be used to provide convenient access to mapping output files from elsewhere. Run diffmeth with commands like:

```
diffmeth -F 2 -t 10 -X 40,220 -I 9 -N -z \
-G /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt \
-R ../X12/bismark_at3/X12_ad3tr85.fastq_bismark.bam \
-R ../X14/bismark_at3/X14_ad3tr90.fastq_bismark.bam \
-R ../X16_twinA/bismark_at3/X16_ad3tr80.fastq_bismark.bam \
-R ../X18/bismark_at3/X18_ad3tr85.fastq_bismark.bam \
-R ../X19/bismark_at3/X19_ad3tr80.fastq_bismark.bam \
-R ../X20/bismark_at3/X20_ad3tr80.fastq_bismark.bam \
-R ../X21/bismark_at3/X21_ad3tr80.fastq_bismark.bam \
-R ../X9006/fastq/bismark_at3/s6_ad3tr65.fastq_bismark.bam \
-R ../X9007/bismark_at3/X9007_ad3tr100.fastq_bismark.bam \
-R ../X9010/bismark_at3/X9010_ad3tr80_40.fastq_bismark.bam \
-R ../X9015/fastq/bismark_at3/s1_ad3tr75.fastq_bismark.bam \
> 11ind_3pp_allpr_F2t10.txt
```

Noting that putting such commands into a shell script is often useful, since it allows various options to be tried with minimal risk of errors. Alternatively scripts in the DMAP2 package simplify use of the various options.

The options used above can be found with:

```
diffmeth –h
```

or in the program documentation. To describe those used here:

`-F 2` indicates that 2 CpGs in each fragment must qualify for the criteria…
`-t 10` with 10 or more hits
`-X 40,220` performs a χ^2 statistic on 40-220bp fragments
`-I 9` requires that 9 of the 11 individuals in this run have valid fragments for that fragment to be considered
`-N` causes the leading CpG of 3' mapping reads to be assigned to the previous fragment
`-z` indicates that the following alignment files are in compressed BAM format
`-G` gives a file which indicates the location of the chromosomal fasta files. An example file is in Appendix I. This option supersedes the obsolete `-g` method of generating chromosome fasta file names by adding `1.fa`, `2.fa`, etc. to the given string up to limits suggested by `-k` and `-Y`.
`-R` is the series of sample BAM alignment files.

`diffmeth` writes output to the Unix/Linux `stdout` stream so `> 11ind_3pp_allpr_F2t10.txt` catches that into a named file. The output is a tab-delimited file of chromosome, fragment start and stop, CpG counts, χ^2 probability and statistic. It is suitable for importing into Excel or for further processing. The above run saved all fragments but it is possible to filter for low probabilities.

diffmeth can read either BAM and SAM files with the use of the -z and -Z options which are positional and affect the mapping files following them on the command line. A mixture of SAM and BAM files can be processed by multiple occurrences of -z and -Z. The default is SAM file input (-Z). The use of these options is similar to that in cleanadaptors.

diffmeth can process non-CpG methylation with the -J command option.

diffmeth defaults to the use of the usual MspI cleavage sites for RRBS processing. More cleavage options are possible with the -n option to use a file of restriction sites instead of the default MspI. The file should only contain a list of cleavage sites (case independent) one per line, with the cutting position indicated by a '^' character. The following would be for a MspI and TaqI combined digest:

```
C^CGG
T^CGA
```

Some specific examples:

(i) Identifying differentially methylated fragments (DMFs) pairwise: the following command applies Fisher's Exact statistic (-P 40,220) to a pair of BAM files. The options given require that at least 2 CpGs in each fragment have 10 or more hits (-F 2 -t 10) and that the leading CpG of 3' mapped reads is assigned to the previous fragment (-N). Only chromosome 21 is to be used (-c 21). Result written to ctrl_mds_FE_diffmeth.txt:

```
diffmeth -c 21 -P 40,220 -F 2 -t 10 -N -z \
-G /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt \
-R ctrl_1.fastq_bismark.sam -R mds_1.fastq_bismark.sam \
 > ctrl_mds_FE_diffmeth.txt
```

(ii) Identifying DMFs - ChiSQ test: on a cohort of 6 individuals. The following command runs diffmeth using the ChiSQ statistic (-X 40,220) requiring that at least 2 CpGs in each fragment have 10 or more hits (-F 2 -t 10), that at least 4 individuals contribute to the statistic (-I 4) and that the leading CpG of 3' mapped reads is assigned to the previous fragment (-N). Only chromosome 21 is to be used (-c 21). Results written to stdout (usually the terminal). The comparison including control and MDS individuals is intended to illustrate the use of this statistic with the test data set, not to imply that there is no difference between the groups:

```
diffmeth -c 21 -X 40,220 -F 2 -t 10 -N -I 4 -z \
 -G /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt \
 -R ctrl_1.fastq_bismark.bam \
 -R ctrl_2.fastq_bismark.bam -R ctrl_3.fastq_bismark.bam \
 -R mds_1.fastq_bismark.bam -R mds_2.fastq_bismark.bam \
 -R mds_3.fastq_bismark.bam
```

(iii) Comparing methylation between two groups: this applies the ANOVA F ratio test (-a 40,220), requiring that at least 4 individuals show counts for a fragment to be included (-I 4) and the leading CpG of 3' mapped reads is assigned to the preceding fragment (-N). BAM data for the treatment/disease group is identified with -S:

```
diffmeth -c 21 -a 40,220 -N -I 4 -z \
-G /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt \
-R ctrl_1.fastq_bismark.bam \
-R ctrl_2.fastq_bismark.bam \
-R ctrl_3.fastq_bismark.bam \
-S mds_1.fastq_bismark.bam \
-S mds_2.fastq_bismark.bam \
-S mds_3.fastq_bismark.bam
```

Probabilities for the F statistic are calculated using a continued fraction method modified from 'Numerical Recipes in C: the Art of Scientifc Computing' (ISBM 0-521-43108-5). Using the other ANOVA options (-A 40,220 or -B 40,220) return progressively more information about more methylated groups and sample counts.

(iv) As for iii, but indicate the more methylated group (-A 40,220) and restrict the output to Pr < 0.01 (-U 0.01). Each line is suffixed with a 'R' or 'S' character to indicate which group had higher methylation and a summary of the valid sample counts for R & S groups.

```
diffmeth -c 21 -A 40,220 -U 0.01 -N -I 4 -z \
-G /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt \
-R ctrl_1.fastq_bismark.bam \
-R ctrl_2.fastq_bismark.bam \
-R ctrl_3.fastq_bismark.bam \
-S mds_1.fastq_bismark.bam \
-S mds_2.fastq_bismark.bam \
-S mds_3.fastq_bismark.bam
```

(v) Compare two individuals with Fisher's Exact Test, using the -R and -S group formality to make diffmeth generate columns showing the methylation proportion for each and which is the more methylated:

```
diffmeth -P 40,220 -N -F 2 -t 10 -z \
-G /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt \
-R Ind_1.fastq_bismark.bam  -S Ind_2.fastq_bismark.bam
```

(vi) For WGBS analysis: for tiled fixed window analysis. The option -W <windowlength> is added to the command. E.g. as for iv, but with fixed width windows of 1000 bp rather than fragments (-W 1000):

```
diffmeth -c 21 -A 40,220 -W 1000 -U 0.01 -N -I 4 -z \
-G /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt \
-R ctrl_1.fastq_bismark.bam \
-R ctrl_2.fastq_bismark.bam \
-R ctrl_3.fastq_bismark.bam \
-S mds_1.fastq_bismark.bam \
-S mds_2.fastq_bismark.bam \
-S mds_3.fastq_bismark.bam
```

Note that the -A 40,220 RRBS option is still needed in order to define the required operation, the 40,220 values are ignored.

(vii) Generate a list of CpG counts for a combined Taq1 & Msp1 digest, showing CpGs with > 0 counts. Do this for H. sapiens chromosome 1 only. `rstsites.txt` contains the sites as indicated above (`-n`), and the mapping is for a SAM file (`-Z`). The info file `c1_chrinfo.txt` contains only chromosome 1, so the analysis will be confined to that - equivalent to putting `-c 1`.

```
diffmeth -G c1_chrinfo.txt -n rstsites.txt -L 40,220 \
  -Z -R Ind_2.fastq_bismark.sam
```

(viii) List all Cs with non-zero hits for a BAM file, only process chromosome 1:

```
diffmeth -G c1_chrinfo.txt -J -E 40,220 -z -R x12_c1_bismark.bam
```

`-J` directs `diffmeth` to process each C.

6. Proximal gene location:

`identgeneloc` takes the output file from `diffmeth` and compares the positions of fragments or any region of interest with feature table information. Typically this would be processed with:

```
identgeneloc -i -Q -U -R -B "protein_coding" \
-G "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/chrinfo.txt \
-r dmeth_10ind_lopr.txt
```

where the options in this run are as follows:
`-i` relates fragments to intron/exon boundaries and looks internally within genes
`-Q` expects feature table information from SeqMonk data files
`-U` scans for nearest upstream CpG Island
`-R` shows ranges for CpG Islands
`-B "protein_coding"` restricts the search to genes with /biotype="protein_coding" (noting that the SeqMonk Human data includes this, but not for zebrafish)
`-r` the `diffmeth` output file
-G gives the name of a file which specifies the location of feature table files for each chromosome which is to be processed. The file contains lines like:

```
1 "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/1.dat"
2 "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/2.dat"
```

A complete example is in Appendix II.

Like other programs, the output is to the Unix/Linux `stdout` stream which defaults to the terminal or which can be redirected into a file. The output is tab-delimited and can be imported into Excel or used for other processing.

Note: the DMAP2 package simplifies many aspects of running `identgeneloc`.

7. Sorting BAM files for IGV use:

If you wish to use IGV (Integrative Genome Viewer: https://software.broadinstitute.org/software/igv/) to view alignments, you need to sort and index the BAM files before they can be loaded. The following steps will do this:

```
samtools sort -o mydata_bismark_sort.bam mydata_bismark_bt2.bam
samtools index mydata_bismark_sort.bam
```

noting that the sort step may take some time. The output will be two files:

mydata_bismark_sort.bam and mydata_bismark_sort.bam.bai

8. %methylation for individual CpGs:

Can be performed with a combination of diffmeth and the awk script getcpgpcmeth.awk (distributed with DMAP). diffmeth is run using the –e or –E options to generate a detailed CpG list, then the script decomposes the output into a tabbed list of chromosome, position and %methylation. CpGs with no hits are shown as '-'.
An example:

```
diffmeth -G /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt \ –
e 40,220 -R ../X12/bismark_at3/X12_ad3tr85.fastq_bismark.sam | awk –f getcpgpcmeth.awk
```

will write the resulting list to <stdout>.

9. ANOVA tests for multiple sample groups:

diffmeth can perform two-way Analysis of Variance for more than two groups. The option letters (-R or -S) can be followed by an integer (1,2,3...) to indicate to which group each sample file belongs. The integers would normally be sequential, but need not necessarily be so. The first group should be indicated by -R1, the second by -R2 and so forth. If this method is being used, then -S1 and -R1 become equivalent - it would not be desirable to mix -Rn and -S since -S would always indicate the second group. If only two groups are required, then -R and -S do not need the suffixed number.

E.g.:

```
diffmeth -c 21 \
-G /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt \
-B 40,220 -z -R1 ctrl_1.bam -R1 ctrl_2.bam -R1 ctrl_3.bam \
-R2 treat1_1.bam -R2 treat1_2.bam -R2 trest1_3.bam \
-R3 treat2_1.bam -R3 treat2_2.bam -R3 treat2_3.bam
```

performs Analysis of Variance (ANOVA) on 3 control samples (-R1) and 3 samples for each of treatment1 and treatment 2 (-R2 and -R3), in this case only for chromosome 21 (-c 21). The -A option returns additional information about which of the sample groups has greater methylation and gives counts of the samples which contributed to the ANOVA F statistic.

10. Conclusion:

These notes are intended as a guide, they do not give all the possible options of the programs mentioned. Further, changes in sequencing systems and downstream processing will date some of this material.

Peter A. Stockwell
Dept. of Pathology, University of Otago, Dunedin, New Zealand.
4-Apr-2023.

Appendix I: Example chromosome sequence information file.

The preferred method of specifying chromosome sequence data and chromosome ID's for `diffmeth`. This file is `/Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/hs_GRCh37_chr_info.txt`

```
1 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.1.fa
2 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.2.fa
3 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.3.fa
4 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.4.fa
5 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.5.fa
6 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.6.fa
7 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.7.fa
8 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.8.fa
9 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.9.fa
10 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.10.fa
11 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.11.fa
12 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.12.fa
13 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.13.fa
14 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.14.fa
15 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.15.fa
16 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.16.fa
17 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.17.fa
18 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.18.fa
19 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.19.fa
20 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.20.fa
21 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.21.fa
22 /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.22.fa
MT /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.MT.fa
X /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.X.fa
Y /Volumes/Data2/HomoSapiens_genome/hs_ref_GRCh37/Homo_sapiens.GRCh37.65.dna.chromosome.Y.fa
```

Appendix II: Example file of genome annotation file location for `identgeneloc`. Note the use of double quotes to enclose the file names because they include spaces.

```
1     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/1.dat"
2     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/2.dat"
3     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/3.dat"
4     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/4.dat"
5     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/5.dat"
6     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/6.dat"
7     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/7.dat"
8     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/8.dat"
9     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/9.dat"
10    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/10.dat"
11    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/11.dat"
12    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/12.dat"
13    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/13.dat"
14    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/14.dat"
15    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/15.dat"
16    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/16.dat"
17    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/17.dat"
18    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/18.dat"
19    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/19.dat"
20    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/20.dat"
21    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/21.dat"
22    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/22.dat"
MT    "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/MT.dat"
X     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/X.dat"
Y     "/Volumes/Data2/SeqMonk_Genomes/Homo sapiens/GRCh37/Y.dat"
```