Contents

Chapter 1

What is CGmapTools

DNA methylation is crucial for a wide variety of biological processes. With the development of high throughput methylome profiling methods, huge volumes of data are generated and in egent need of computational tools for data analysis. Though several tools have been proposed to fit this need, there is not a mainstream standard for bisulfite sequencing data storage and manipulation. What's more, the performence of available tools needs to be improved.

We proposed **CGmapTools**, a bisulfite sequencing analysis toolset with enhanced features on SNV calling and allele specific methylations and visualizations, in hope to set up a standard for bisulfite sequencing data related manipulation, including better data storage, extraction, visualization and improved performence in SNP calling. We also provide dozens of utilities and a seamless pipeline for bisulfite sequencing data analysis.

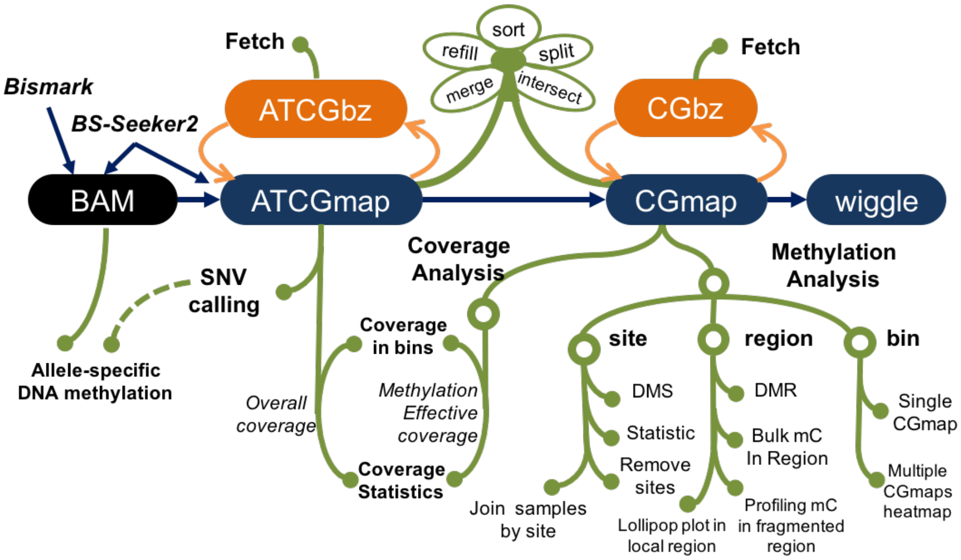

Figure 1.1: Schematic diagram of CGmapTools

Chapter 2

New File Format

To facilitate high throughput data manipulation and reduce storage usage, several file format have been proposed and generaly accepted as the standard. Due to these great efforts (e.g. SAM/BAM and VCF), data analysis and tool development become more easier and highly efficient. However, when it comes to bisulfite sequencing data, currently, available tools possess their own tool specific data format. In consequence, integrating results from several tools leads to extra efforts in unifying data format and developing custermized tools, which is time comsuming and error prone.

As one of the features of CGmapTools, we defined ATCGmap and CGmap file format to simplify downstream DNA methylation analysis and in hope to standardize the storage format of bisulfite sequencing data.

2.1 ATCGmap Format

After alignment of sequencing reads to the reference genome, all the detail information about read coverage and methylation level of a cytosine site are stored in BAM/SAM format files though requiring further interpretation. A well defined file format called **pileup** summarized the information of mapped reads covered on each nucleotide along the reference genome. But the pileup file does not designed for bisulfte sequencing data, which lacks DNA methylation estimation of cytosines.

Here, we defined ATCGmap file format to integrate both mapping and coverage of non-cytosine and cytosine sites with estimated DNA methylation in a single file.

Col	Field	Type	Regexp/Range	Brief description
1	CHR	String	[!-?A-~]{1,118}	Query template NAME
2	NUC	Char	[ATCGN-]	The nucleotide on reference genome
3	POS	Int	[0,232-1]	1-based leftmost mapping position
4	CONT	String	{"–", \"CG", "CHG", "CHH"}	Context
5	DINUC	String	{"–", "CA", "CT", "CC", "CG"}	Dinucleotide context
6	WA	Int	[0,214-1]	Counts of reads on Watson strand support Adenine
7	WT	Int	[0,214-1]	Counts of reads on Watson strand support Thymine
8	WC	Int	[0,214-1]	Counts of reads on Watson strand support Cytosine
9	WG	Int	[0,214-1]	Counts of reads on Watson strand support Guanine
10	WN	Int	[0,26-1]	Counts of reads on Watson strand support None
11	CA	Int	[0,214-1]	Counts of reads on Crick strand support Adenine
12	CT	Int	[0,214-1]	Counts of reads on Crick strand support Thymine
13	CC	Int	[0,214-1]	Counts of reads on Crick strand support Cytosine
14	CG	Int	[0,214-1]	Counts of reads on Crick strand support Guanine
15	CN	Int	[0,26-1]	Counts of reads on Crick strand support None
16	METH	Float	[0,1] or "na"	Methylation level or "Not Available"

2.2 CGmap Format

In cases we only want to retain DNA methylation on cytonsines to save storage usage, we defined another file format called **CGmap** which provides sequence context and estimated DNA methylation level of any covered cytosines on the reference genome.

Col	Field	Type	Regexp/Range	Brief description
1	CHR	String	[!-?A-~]{1,118}	Query template NAME
2	NUC	Char	[ATCGN-]	The nucleotide on reference genome
3	POS	Int	[0,2^32^-1]	1-based leftmost mapping position
4	CONT	String	{"–", "CG", "CHG", "CHH"}	Context
5	DINUC	String	{"–", "CA", "CT", "CC", "CG"}	Dinucleotide context
6	METH	Float	[0,1] or "na"	Methylation level or "Not Available"
7	MC	Int	[0,2^12^-1]	Counts of reads support methylated Cytosine
8	NC	Int	[0,2^12^-1]	Counts of reads support all Cytosine

Chapter 3

File Manipulation

CGmapTools provides multiple utilities to manipulate files in ATCGmap and CGmap format or compressed ATCGbz/CGbz format.

Usage: `cgmaptools <convert|fetch|refill|intersect|merge2|mergelist|sort|split|select|> [options]`

3.1 convert

- **Description**: File format coversion.
- **Usage**: `cgmaptools convert <command> [options]`
- **sub-command table**:

Commands	From	To
bam2cgmap	BAM	CGmap & ATCGmap
atcgmap2atcgbz	ATCGmap	ATCGbz
atcgbz2atcgmap	ATCGbz	ATCGmap
atcgmap2cgmap	ATCGmap	CGmap
cgmap2cgbz	CGamp	CGbz
cgbz2cgmap	CGbz	CGmap
cgmap2wig	CGmap	WIG

- **Example commands** :

  BAM to CGmap `cgmaptools convert bam2cgmap -b WG.bam -g genome.fa --rmOverlap -o WG`

  BAM to CGmap `cgmaptools convert bam2cgmap -b RR.bam -g genome.fa --rmOverlap -o RR`

  ATCGmap to ATCGbz `cgmaptools convert atcgmap2atcgbz -c WG.ATCGmap.gz -b WG.ATCGbz`

  ATCGvz to ATCGmap `cgmaptools convert atcgbz2atcgmap -c WG2.ATCGmap.gz -b WG.ATCGbz`

  CGmap to CGbz `cgmaptools convert cgmap2cgbz -c RR.CGmap.gz -b RR.CGbz`

  CGbz to CGmap `cgmaptools convert cgbz2cgmap -c RR2.CGmap.gz -b RR.CGbz`

  CGmap => WIG `cgmaptools convert cgmap2wig [-i <CGmap>] [-w <wig>] [-c <INT> -b <float>]`

  Note: please refer to the help message for usage details using `-h` option.

3.2 fetch

- **Description**: Fastly acess methylation data in specified region.
- **Usage**: `cgmaptools fetch <command> [options]`
- **Commands**:

  atcgbz: fetch lines from ATCGbz file.

- **Usage**: `cgmaptools fetch atcgbz -b <ATCGbz> -C <CHR> -L <LeftPos> -R <RightPos>`

```
-b, --ATCGbz <arg>     input ATCGbz file
-C, --CHR <arg>        specify the chromosome name
-L, --leftPos <arg>    the left position
-R, --rightPos <arg>   the right position
```

  cgbz: fetch lines from CGbz file.

  Usage: `cgmaptools fetch cgbz -b <CGbz> -C <CHR> -L <LeftPos> -R <RightPos>`

```
-b, --CGbz <arg>       intput CGbz file
-C, --CHR <arg>        specify the chromosome name
-L, --leftPos <arg>    the left position
-R, --rightPos <arg>   the right position
```

- **Example commands** :

```
cgmaptools fetch atcgbz -b WG.ATCGbz -C chr2 -L 90 -R 100
```

```
cgmaptools fetch cgbz -b RR.CGbz -C chr3 -L 2200 -R 2400
```

3.3 refill

- **Description**: Fill the CG/CHG/CHH and CA/CC/CT/CG context to CGmap or ATCGmap files. Other fields will not be affected.
- **Usage**: `cgmaptools refill [-i <CGmap>] -g <genome.fa> [-o output]`

```
-i STRING     Input CGmap file (CGmap or CGmap.gz)
-g STRING     genome file, FASTA format (gzipped if end with '.gz')
-o STRING     Output file name (gzipped if end with '.gz')
-0, --0-base  0-based genome if specified [Default: 1-based]
```

- **File formats**:

  The input CGmap file, which is lacking C context on the 3rd and 4th columns:

```
Chr1    C       3541    -       -       0.0     0       1
```

  After `refill` processing, the CGmap file would be as below, added C context information:

```
Chr1    C       3541    CG      CG      0.0     0       1
```

- **Example command**:

```
zcat RR2.CGmap.gz | gawk -F"\t" -vOFS="\t" '{$4="-"; $5="-"; print;}' | cgmaptools
refill -g genome.fa -o RR3.CGmap.gz
```

3.4 intersect

- **Description**: Get the intersection of two CGmap files.
- **Usage**: `cgmaptools intersect [-1 <CGmap_1>] -2 <CGmap_2> [-o <output>]`

  ```
  -1 CGmap File  File name, end with .CGmap or .CGmap.gz.
  -2 CGmap File  standard input if not specified
  -o OUTFILE     To standard output if not specified. Compressed output if end
                 with .gz
  ```

- **File formats**:

  Suppose you have two CGmap file from two samples, the first one is:

  ```
  Chr1  C  3541  CG  CG  0.8  4  5
  ```

  and the second CGmap file is:

  ```
  Chr1  C  3541  CG  CG  0.4  4  10
  ```

  After intersction, the output contains sites covered in both CGmap files. And the last three columns of the output are extracted from the second CGmap file:

  ```
  Chr1  C  3541  CG  CG  0.8  4  5  0.4  4  10
  ```

- **Example command** :

  ```
  cgmaptools intersect -1 WG.CGmap.gz -2 RR.CGmap.gz -C CG -o intersect_CG.gz
  ```

3.5 merge2

- **Description**: Merge two CGmap or ATCGmap files together.
- **Usage**: `cgmaptools merge2 <command> [options]`
- **Commands**:

  atcgmap: merge two ATCGmap files into one.

- **Usage**: `cgmaptools merge2 atcgmap -1 <ATCGmap> -2 <ATCGmap>`

  ```
  -1  Input, 1st ATCGmap file
  -2  Input, 2nd ATCGmap file
  Output to STDOUT in ATCGmap format
  Tips: Two input files should have the same order of chromosomes
  ```

  cgmap: merge two CGmap files into one.

- **Usage**: `cgmaptools merge2 cgmap -1 <CGmap_1> -2 <CGmap_2> [-o <output>]`

  ```
  -1 FILE     File name end with .CGmap or .CGmap.gz
  -2 FILE     If not specified, STDIN will be used.
  -o OUTFILE  CGmap, output file. Use STDOUT if omitted (gzipped if end with
              '.gz').
  ```

- **Example command** :

  ```
  cgmaptools merge2 atcgmap -1 WG.ATCGmap.gz -2 RR.ATCGmap.gz | gzip > merge.ATCGmap.gz
  ```

  ```
  cgmaptools merge2 cgmap -1 WG.CGmap.gz -2 RR.CGmap.gz | gzip > merge.CGmap.gz
  ```

3.6 mergelist

- **Description**: merge a list of files.
- **Usage**: `cgmaptools mergelist <command> [options]`
- **Commands**:

  tomatrix: Fill methylation levels according to the Index file for CGmap files in list.

- **Usage**: `cgmaptools mergelist tomatrix  [-i <index>] -f <IN1,IN2,..> -t <tag1,tag2,..> [-o output]`

```
-i FILE     TXT file, index file, use STDIN if omitted
-f STRING   List of (input) CGmap files (CGmap or CGmap.gz)
-t STRING   List of tags, same order with '-f'
-c INT      minimum coverage [default: 1]
-C INT      maximum coverage [default: 200]
-o STRING   Output file name (gzipped if end with '.gz')
```

- **Example commands** :

```
zcat RR*.CGmap.gz WG.CGmap.gz | gawk '$8>=5' | cut -f1,3 | sort -u | cgmaptools sort
-c 1 -p 2 > index
```

```
cgmaptools mergelist tomatrix -i index -f RR.CGmap.gz,RR2.CGmap.gz,WG.CGmap.gz -t
RR,RR2,WG -c 5 -C 100 -o matrix.CG.gz
```

  tosingle: merge list of input files into one.

  Usage: `cgmaptools mergelist tosingle -i f1,f2,..,fn [-o <output>]`

```
-i FILE     List of input files; gzipped file ends with '.gz'
-f FILE     cgmap or atcgmap [Default: cgmap]
-o OUTFILE  To standard output if not specified; gzipped file if end with
            '.gz'
```

3.7 sort

- **Description**: Sort the input files by chromosome and position.
- **Usage**: `cgmaptools sort [-i <input>] [-c 1] [-p 3] [-o output]`

```
-i FILE            File name end with .CGmap or .CGmap.gz.
                   If not specified, STDIN will be used.
-c INT, --chr=INT  The column of chromosome [default: 1]
-p INT, --pos=INT  The column of position [default: 2]
-o OUTFILE         To standard output if not specified
```

3.8 split

- **Description**: Split the files by each chromosomes.
- **Usage**: `cgmaptools split -i <input> -p <prefix[.chr.]> -s <[.chr.]suffix>`

```
-i FILE     Input file, CGmap or ATCGmap foramt, use STDIN when not specified.
            (gzipped if end with 'gz').
-p STRING   The prefix for output file
```

```
-s STRING   The suffix for output file (gzipped if end with 'gz').
```

- **Example command** :

```
cgmaptools split -i WG.CGmap.gz -p WG -s CGmap.gz
```

3.9 select

- **Description**: Split the files by each chromosomes.
- **Usage**: `cgmaptools select <command> [options]`
- **Commands**:

region: Lines in input CGmap/ATCGmap be selected/excluded by BED file. Strand is NOT considered. Output to STDOUT in same format with input.

- **Usage**: `cgmaptools select region [-i <CGmap/ATCGmap>] -r <BED> [-R]`

```
-i  Input, CGmap/ATCGmap file; use STDIN if not specified
    Please use "gunzip -c <input>.gz " and pipe as input for gzipped file.
    Ex: chr12   G   19898796    ...
-r  Input, Region file, BED file to store regions
    At least 3 columns are required
    Ex: chr12 19898766 19898966 XX XXX XXX
-R  [optional] Reverse selection. Sites in region file will be excluded when specified
```

- **Example command** :

```
for CHR in 1 2 3 4 5; do (for P in 1 2 3 4 5; do echo | gawk -vC=$CHR -vP=$P -vOFS="\t"
'{print "chr"C, P*1000, P*1000+200, "+";}' ; done) ; done > region.bed
```

```
zcat WG.CGmap.gz | cgmaptools select region -r region.bed | head
```

site: Select lines from input CGmap/ATCGmap in index or reverse.

- **Usage** : `cgmaptools select site -i <index> [-f <CGmap/ATCGmap>] [-r] [-o output]`

```
-i FILE     Name of Index file required (gzipped if end with '.gz').
-r          reverse selected, remove site in index if specified
-f STRING   Input CGmap/ATCGmap files. Use STDIN if not specified
-o STRING   CGmap, Output file name (gzipped if end with '.gz').
```

- **Example commands** :

```
gawk 'NR%100==50' index > site
```

```
cgmaptools select site -f RR.CGmap.gz -i site -o RR_select.CGmap.gz
```

Chapter 4

SNV calling

Bisulfite sequencing data contains information of both methylation and genome sequences. In addition to DNA methylation analysis, we can also call variants using bisulfite data. Due to bisulfite coversion and PCR amplification during library preparation, the unmethylated cytosines on the DNA fragments would be converted to thymines. Thus, it's difficult to distinguish thymine produced by bisulfite coversion with the real thymine allele.

In recent years, few tools are adapted to bisulfite data for SNP calling. The main idea is removing vague reads that may contain unmethylated cytosines for a given positoin. Consequently, the rest reads can be regarded as reads generated from a normal genome DNA without bisulfite treatment and can be used to call variants using regular methods without consideration of bisulfite conversion.

However, removing the vague reads leads to information lost in most cases making variant calling less confident, especially when the sequencing depth is low. To solve this problem, we proposed two independent methods called BinomWC (based on binomial) and BayesWC (based on bayesian), taking vague reads into consideration.

- **Usage**: `cgmaptools snv [-i <ATCGmap>] [-o <output> -v <VCF>]`

```
-i FILE               ATCGmap format, STDIN if not specified
-v FILE, --vcf=FILE   VCF format file for output
-a, --all_nt          Show all sites with enough coverage (-l). Only show
                      SNP sites if not specified.
-o OUTFILE            STDOUT if not specified
-m MODE, --mode=MODE  Mode for calling SNP [Default: binom]
                      binom: binomial,  separate strands
                      bayes: bayesian mode
--bayes-e=BAYES_ER    (BayesWC mode) Error rate for calling a nucleotide
                      [Default: 0.05]
--bayes-p=BAYES_PV    (BayesWC mode) P value as cut-off [Default: 0.001]
--bayes-dynamicP      (BayesWC mode) Use dynamic p-value for different
                      coverages install of specific p-value. (Recomended)
                      "--bayes-p" will be ignored if "--bayes-dynamicP" is
                      specified.
--binom-e=BINOM_ER    (BinomWC mode) Error rate for calling a nucleotide
                      [Default: 0.05]
--binom-p=BINOM_PV    (BinomWC mode) P value as cut-off [Default: 0.01]
--binom-cov=BINOM_COV
                      (BinomWC mode) The coverage checkpoint [Default: 10]
```

- **Example commands** :

```
cgmaptools snv -i WG.ATCGmap.gz -m bayes -v bayes.vcf -o bayes.snv --bayes-dynamicP
cgmaptools snv -i WG.ATCGmap.gz -m binom -o binom.snv
```

Chapter 5

Methylation Analysis

5.1 dms

- **Description**: Get the differentially methylated sites between two samples.
- **Usage**: `cgmaptools dms [-i <CGmapInter>] [-m 5 -M 100] [-o output]`

  ```
  -i FILE               File name for CGmapInter, STDIN if omitted
  -m INT, --min=INT     min coverage [default : 0]
  -M INT, --max=INT     max coverage [default : 100]
  -o OUTFILE            To standard output if omitted. Compressed output if
                        end with .gz
  -t STRING, --test-method=STRING
                        chisq, fisher [default : chisq]
  ```

- **Example command** :

  ```
  cgmaptools dms -i intersect_CG.gz -m 4 -M 100 -o DMS.gz -t fisher
  ```

- **File format**:

  #1 Using the output of `intersect` as input:

  ```
  Chr1  C  3541  CG  CG  0.8  4  5  0.4  4  10
  ```

  The output of `dms` is:

  ```
  chr1    C   4654    CG  CG  0.92    1.00    8.40e-01
  chr1    C   4658    CHH CC  0.50    0.00    3.68e-04
  chr1    G   8376    CG  CG  0.62    0.64    9.35e-01
  ```

5.2 dmr

- **Description**: Get the differentially methylated region by Fisher's exact test.
- **Usage**: `cgmaptools dmr [-i <CGmapInter>] [-m 5 -M 100] [-o output]`

  ```
  -i FILE                 File name for CGmapInter, STDIN if omitted
  -c INT, --minCov=INT    min coverage [default : 0]
  -C INT, --maxCov=INT    max coverage [default : 100]
  -s INT, --minStep=INT   min step in bp [default : 100]
  -S INT, --maxStep=INT   max step in bp [default : 500]
  ```

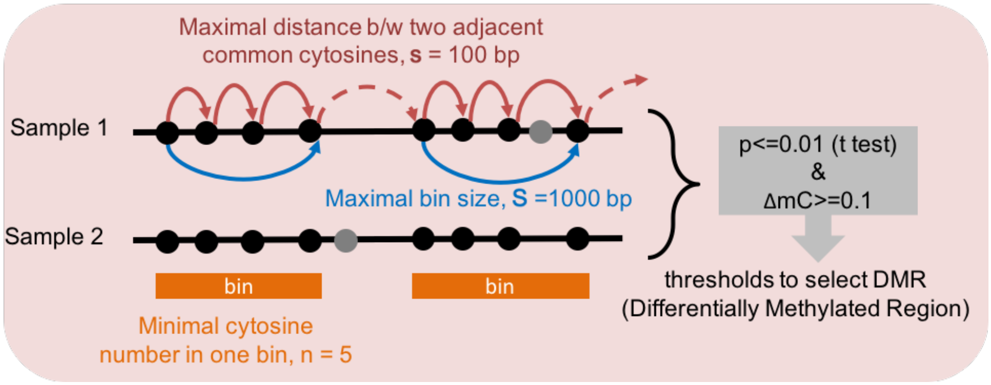
Maximal distance b/w two adjacent common cytosines, s = 100 bp
Sample 1
Sample 2
Maximal bin size, S =1000 bp
p<=0.01 (t test) & ΔmC>=0.1
thresholds to select DMR (Differentially Methylated Region)
bin
bin
Minimal cytosine number in one bin, n = 5

Figure 5.1: Dynamic Fragmentation Strategy

```
-n INT, --minNSite=INT  min N sites [default : 2]
-o OUTFILE              To standard output if omitted. Compressed output if
                        end with .gz
```

- **File format**:

  #1 Using the output of `intersect` as input:

  ```
  Chr1  C  3541  CG  CG  0.8  4  5  0.4  4  10
  ```

  The output of `dms` is:

  ```
  chr1    1004572 1004574 inf     0.00e+00    0.1100  0.0000
  chr1    1009552 1009566 -0.2774 8.08e-01    0.0200  0.0300
  chr1    1063405 1063498 0.1435  8.93e-01    0.6333  0.5733
  ```

- **Example command**:

  ```
  cgmaptools dmr -i intersect_CG.gz -o DMR.gz
  ```

5.3 asm

- **Description**: Allele specific methylation analysis.
- **Usage**: `cgmaptools asm [options] -r <ref.fa> -b <input.bam> -l <snp.vcf>`

```
-r     Samtools indexed reference genome seqeunce, fasta format. eg. hg19.fa
       - use samtools to index reference first: samtools faidx hg19.fa
-b     Samtools indexed Bam format file.
       - use samtools to index bam file first: samtools index <input.bam>
-l     SNPs in vcf file format.
-s     Path to samtools eg. /home/user/bin/samtools
       - by defualt, we try to search samtools in your system PATH,
-o     Output results to file. [default: STDOUT]
-t     C context. [default: CG]
       - available context: C, CG, CH, CW, CC, CA, CT, CHG, CHH
-m     Specify calling mode. [default: asr]
       - alternative: ass
       - asr: allele specific methylated region
       - ass: allele specific methylated site
-d     Minimum number of read for each allele linked site to call ass. [default: 3]
```

```
      - ass specific.
-n    Minimum number of C site each allele linked to call asr. [default: 2]
      - asr specific.
-D    Minimum read depth for C site to call methylation level when calling asr.
      [default: 1]
      - asr specific.
-L    Low methylation level threshold. [default: 0.2]
      - allele linked region [or site] with low methylation level should be
        no greater than this threshold.
-H    High methylation level threshold. [default: 0.8]
      - allele linked region[or site] with high methylation level should be
        no less than this threshold.
-q    Adjusted p value using Benjamini & Hochberg (1995)
      ("BH" or its alias "fdr"). [default: 0.05]
```

- **Example command**:

```
gawk '{if(/^#/){print}else{print "chr"$0;}}' bayes.vcf > bayes2.vcf
```

```
cgmaptools asm -r genome.fa -b WG.bam -l bayes2.vcf > WG.asm
```

5.4 mbed

- **Description**: Calculate average methylation levels in given regions.
- **Usage**: `cgmaptools mbed [-i <CGmap>]  -b <regin.bed> [-c 5 -C 500 -s]`

```
-i  String, CGmap file; use STDIN if not specified
    Ex: chr1    G   3000851 CHH CC  0.1 1   10
-b  String, BED file
    Ex: chr1    3000000 3005000 -
-c  Int, minimum Coverage [Default: 5]
-C  Int, maximum Coverage [Default: 500]
-s  Strands would be distinguished when specified
```

- **Example command**:

```
zcat WG.CGmap.gz | cgmaptools mbed -b region.bed
```

- **File format**:

The output format:

```
chr   sense_Count   sense_mC    anti_Count  anti_mC all_Count   all_mC
chr1    203   0.08127     178     0.1148      381     0.09692
chr2    185   0.07045     257     0.05586     442     0.06197
chr3    313   0.1042      250     0.1358      563     0.1182
chr4    300   0.1218      271     0.13      571   0.1257
chr5    282   0.1272      222     0.1589      504     0.1412
```

5.5 mbin

- **Description**: Generate the methylation level in Bins.
- **Usage**: `cgmaptools mbin [-i <CGmap>] [-c 10 --CXY 5 -B 5000000]`

```
-i FILE              File name end with .CGmap or .CGmap.gz.
                     If not specified, STDIN will be used.
-B BIN_SIZE          Define the size of bins [Default: 5000000]
-c COVERAGE          The minimum coverage for site selection [Default: 10]
-C CONTEXT, --context=CONTEXT
                     specific context: CG, CH, CHG, CHH, CA, CC, CT, CW
                     use all sites if not specified
--cXY=COVERAGEXY     Coverage for chrX/Y should be half that of autosome
                     for male [Default: same with -c]
-f FIGTYPE, --figure-type=FIGTYPE
                     png, pdf, eps. Will not generate figure if not
                     specified
-p STRING            Prefix for output figures
-t STRING, --title=STRING
                     title in the output figures
```

- **Example command**:

```
cgmaptools mbin -i WG.CGmap.gz -B 500 -c 4 -f png -t WG -p WG > mbin.WG.data
```

- **File format**:

  The output format:

```
chr1    1       5000    0.0000
chr1    5001    10000   0.0396
chr2    1       5000    0.0755
chr2    5001    10000   0.0027
chr3    1       5000    na
```

5.6 mmbin

- **Description**: Generate the methylation level in Bins for multiple samples.
- **Usage**: `cgmaptools mmbin [-l <1.CGmap[,2.CGmap,..]>] [-c 10 --CXY 5 -B 5000000]`

```
-l FILE              File name list, end with .CGmap or .CGmap.gz. If not
                   specified, STDIN will be used.
-t FILE              List of samples
-B BIN_SIZE          Define the size of bins [Default: 5000000]
-C CONTEXT, --context=CONTEXT
                     specific context: CG, CH, CHG, CHH, CA, CC, CT, CW
                     use all sites if not specified
-c COVERAGE          The minimum coverage for site selection [Default: 10]
--cXY=COVERAGEXY     Coverage for chrX/Y should be half that of autosome
                     for male [Default: same with -c]
```

- **Example command**:

```
cgmaptools mmbin -l WG.CGmap.gz,RR.CGmap.gz,RR2.CGmap.gz,merge.CGmap.gz -c 4 -B 2000
| gawk '{printf("%s:%s-%s", $1, $2, $3); for(i=4;i<=NF;i++){printf("\t%s", $i);}
printf("\n");}' > mmbin
```

- **File format**:

  The output format:

```
chr1    1       5000    0.0000
```

```
chr1    5001    10000   0.0396
chr2    1       5000    0.0755
chr2    5001    10000   0.0027
chr3    1       5000    na
```

5.7 mfg

- **Description**: Calculated methylation profile across fragmented regions.
- **Usage**: `cgmaptools mfg [-i <CGmap>]  -r <region> [-c 5 -C 500]`

```
-i  String, CGmap file; use STDIN if not specified
    chr1    G   851 CHH CC  0.1 1   10
-r  String, Region file, at least 4 columns
    Format: chr strand  pos_1   pos_2   pos_3   ...
    Regions would be considered as [pos_1, pos_2), [pos_2, pos_3)
    Strand information will be used for distinguish sense/antisense strand
    Ex:
    chr1    +   600 700 800 900 950
    chr1    -   1600    1500    1400    1300    1250
-c  Int, minimum Coverage [Default: 5]
-C  Int, maximum Coverage [Default: 500]
    Sites exceed the coverage range will be discarded
```

- **Example command**:

```
for CHR in 1 2 3 4 5; do (for P in 1 2 3 4 5; do echo | gawk -vC=$CHR -vP=$P -vOFS="\t"
'{print "chr"C, P*1000, P*1000+1000, "+";}' ; done) ; done | cgmaptools bed2fragreg
-n 30 -F 50,50,50,50,50,50,50,50,50,50 -T 50,50,50,50,50,50,50,50,50,50 > fragreg.bed
```

```
gunzip -c WG.CGmap.gz | cgmaptools mfg -r fragreg.bed -c 2 -x CG > WG.mfg
```

```
cgmaptools fragreg -i WG.mfg -f pdf -o WG_mfg.pdf
```

- **File format**:

The output format:

```
Region_ID       R_1     R_2     R_3     R_4
sense_ave_mC    0.50    0.40    0.30    0.20
sense_sum_mC    5.0     4.0     3.0     2.0
sense_sum_NO    10      10      10      10
anti_ave_mC     0.40    0.20    0.10    NaN
anti_sum_mC     8.0     4.0     2.0     0.0
anti_sum_NO     20      20      20      0
total_ave_mC    0.43    0.27    0.17    0.2
total_sum_mC    13.0    8.0     5.0     2.0
total_sum_NO    30      30      30      10
```

5.8 mstat

- **Description**: Methyaltion statistic.
- **Usage**: `cgmaptools mstat [-i <CGmap>]`

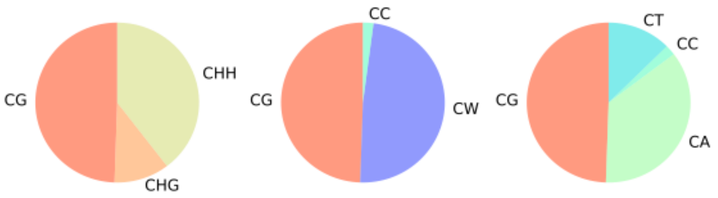

Figure 5.2: mC contribution example

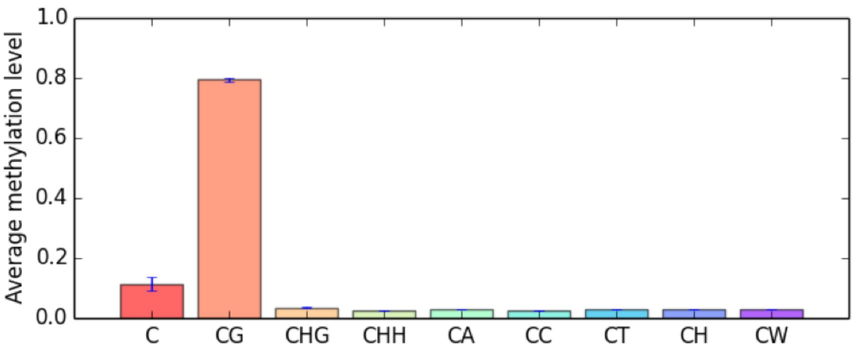

Figure 5.3: Bulk mC example

```
-i FILE               File name end with .CGmap or .CGmap.gz. If not
                      specified, STDIN will be used.
-c COVERAGE           The minimum coverage for site selection [Default: 10]
-f FILE, --figure-type=FILE
                      png, pdf, eps. Will not generate figure if not
                      specified
-p STRING             Prefix for output figures
-t STRING, --title=STRING
                      title in the output figures
```

- **Example command**:

```
cgmaptools mstat -i WG.CGmap.gz -c 4 -f png -p WG -t WG > WG.mstat.data
```

- **File format**:

The output format:

```
MethStat        context C       CG      CHG     CHH     CA      CC      CT      CH      CW
mean_mC         global  0.0798  0.3719  0.0465  0.0403  0.0891  0.0071  0.0241  0.0419  0.0559
sd_mCbyChr      global  0.0078  0.0341  0.0163  0.0110  0.0252  0.0049  0.0076  0.0096  0.0148
count_C         global  10000   1147    2332    6521    3090    2539    3224    8853    6314
contrib_mC      global  1.0000  0.5348  0.1360  0.3292  0.3452  0.0228  0.0973  0.4652  0.4424
quant_mC        [0]     8266    471     2012    5783    2422    2421    2952    7795    5374
quant_mC   (0.00 ,0.20] 705     182     155     368     272     97      154     523     426
mean_mC_byChr   chr1    0.0840  0.4181  0.0340  0.0412  0.0794  0.0065  0.0251  0.0393  0.0513
mean_mC_byChr   chr10   0.0917  0.4106  0.0758  0.0421  0.0968  0.0097  0.0349  0.0502  0.0655
```

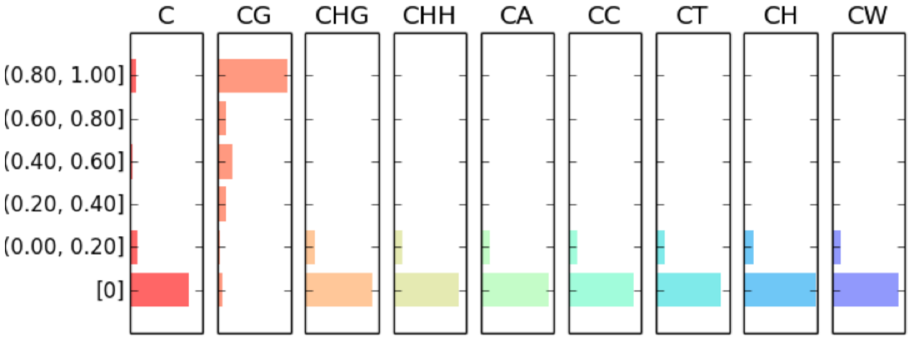

Figure 5.4: mC fragmented distribution example

5.9 mtr

- **Description**: Calculate the methylation levels in regions in two ways.
- **Usage**: `cgmaptools mtr [-i <CGmap>] -r <region> [-o <output>]`

```
-i FILE     File name end with .CGmap or .CGmap.gz. If not specified, STDIN
            will be used.
-r FILE     Filename for region file, support *.gz
-o OUTFILE  To standard output if not specified.
```

- **Example command**:

```
cgmaptools mtr -i WG.CGmap.gz -r region.bed -o WG.mtr.gz
```

- **File formats**:

The input file format:

```
#chr    start_pos  end_pos
chr1    8275       8429
```

The output format:

```
#chr   start_pos   end_pos   mean(mC)   #_C   #read(C)/#read(T+C)   #read(T+C)
chr1    8275        8429      0.34      72          0.40                164
```

Chapter 6

Coverage Analysis

6.1 oac

- **Description**: Overall coverage (for ATCGmap).
- **Usage**: `cgmaptools oac <command> [options]`
- **Command**:

  bin: Overall coverage in bins.

  Usage: `cgmaptools oac bin  [-i <ATCGmap>] [-B 5000000]`

```
-i FILE               File name end with .ATCGmap or .ATCGmap.gz. If not
                      specified, STDIN will be used.
-B BIN_SIZE           Define the size of bins [Default: 5000000]
-f FILE, --figure-type=FILE
                      png, pdf, eps. Will not generate figure if not
                      specified
-p STRING             Prefix for output figures
-t STRING, --title=STRING
                      title in the output figures
```

- **Example command**:

```
cgmaptools oac bin -i WG.ATCGmap.gz -B 1000 -f png -p WG -t WG > WG.oac_bin.data
```

  stat: Get the distribution of overall coverages.

- **Usage**: `cgmaptools oac stat [-i <ATCGmap>]`

```
-i FILE               File name end with .ATCGmap or .ATCGmap.gz. If not
                      specified, STDIN will be used.
-f FILE, --figure-type=FILE
                      png, pdf, eps. Will not generate figure if not
                      specified
-p STRING             Prefix for output figures
```

- **Example command**:

```
cgmaptools oac stat -i WG.ATCGmap.gz -p WG -f png > WG.oac_stat.data
```

- **output format**:

  The output format of `bin`:

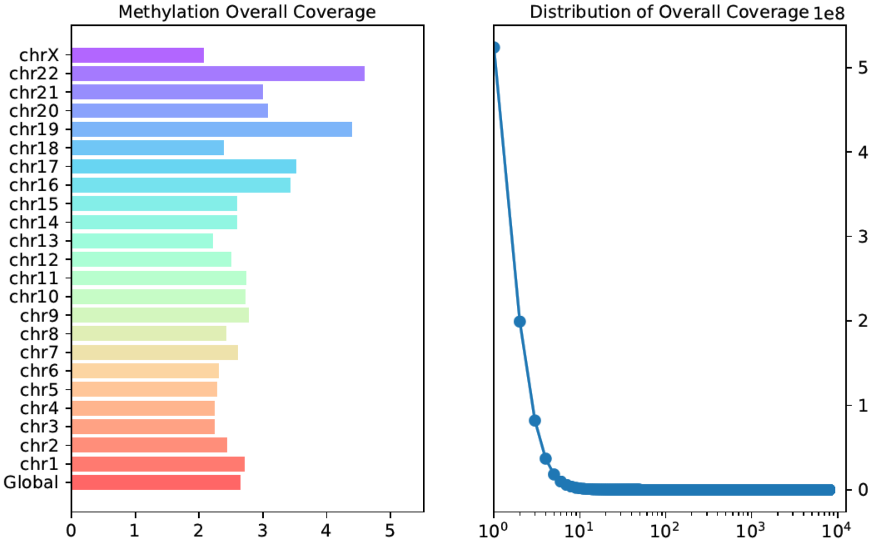

Figure 6.1: MEC example

```
chr1     1        5000     29.0000
chr1     5001     10000    30.0396
chr2     1        5000     35.0755
chr2     5001     10000    40.0027
chr3     1        5000     na
```

The output format of `stat`:

```
OverAllCov       global   47.0395
OverAllCov       chr1     45.3157
OverAllCov       chr10    47.7380
CovAndCount      1        1567
CovAndCount      2        655
CovAndCount      3        380
```

6.2 mec

- **Description**: Methylation effective coverage (for CGmap).
- **Usage**: `cgmaptools mec <command> [options]`
- **Commands**:

  bin: Generate the methylation-effective coverage in Bins.

  Usage: `cgmaptools mec bin [-i <CGmap>] [-B 5000000]`

```
-i FILE                 File name end with .CGmap or .CGmap.gz. If not
                        specified, STDIN will be used.
-B BIN_SIZE             Define the size of bins [Default: 5000000]
-f FILE, --figure-type=FILE
                        png, pdf, eps. Will not generate figure if not
```

```
                      specified
-p STRING             Prefix for output figures
-t STRING, --title=STRING
                      title in the output figures
```

- **Example command**:

  ```
  cgmaptools mec bin -i WG.CGmap.gz -B 1000 -f png -p WG -t WG > WG.mec_bin.data
  ```

- **output format**:

  The output format of `bin`:

  ```
  chr1    1       5000    29.0000
  chr1    5001    10000   30.0396
  chr2    1       5000    35.0755
  chr2    5001    10000   40.0027
  chr3    1       5000    na
  ```

  stat: Get the distribution of methylation-effective coverages.

- **Usage**: `cgmaptools mec stat [-i <CGmap>]`

  ```
  -i FILE               File name end with .CGmap or .CGmap.gz. If not
                        specified, STDIN will be used.
  -f FILE, --figure-type=FILE
                        png, pdf, eps. Will not generate figure if not
                        specified
  -p STRING             Prefix for output figures
  ```

- **Example command**:

  ```
  cgmaptools mec stat -i WG.CGmap.gz -p WG -f png > WG.mec_stat.data
  ```

- **Output format**:

  The output format of `stat`:

  ```
  OverAllCov      global  47.0395
  OverAllCov      chr1    45.3157
  OverAllCov      chr10   47.7380
  CovAndCount     1       1567
  CovAndCount     2       655
  CovAndCount     3       380
  ```

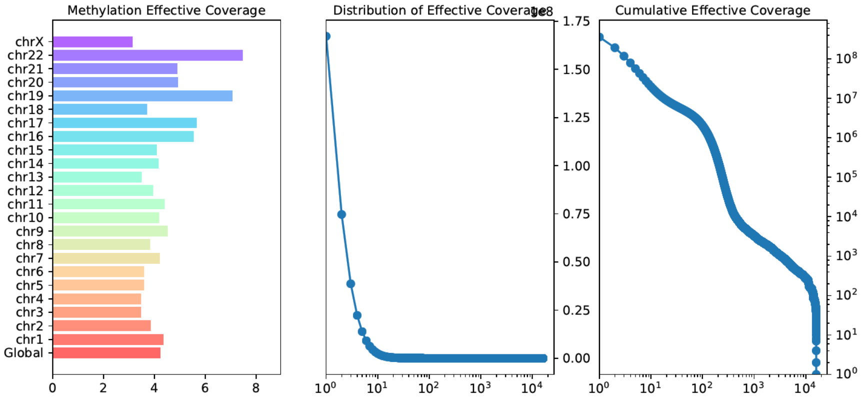

Figure 6.2: MEC example

Chapter 7

Graphics

7.1 lollipop

- **Description**: Plot local mC level for multiple samples.
- **Usage**: `cgmaptools lollipop [options] file`

  ```
  -i INFILE, --infile=INFILE
        input file
  -a ANNOTATION, --annotation=ANNOTATION
      [opt] sample name
  -o OUTFILE, --outfile=OUTFILE
      [opt] output file
  -f FORMAT, --format=FORMAT
      [opt] the format for output figure: pdf (default), png, eps
  -l LEFT, --left=LEFT
      [opt] Left-most position
  -r RIGHT, --right=RIGHT
      [opt] Right-most position
  -c CHR, --chr=CHR
      [opt] chromosome name
  -t TITLE, --title=TITLE
      [opt] text shown on title
  -w WIDTH, --width=WIDTH
      [opt] width (in inch). Default: 8.
  --height=HEIGHT
      [opt] height (in inch). Default: 8.
  -s SITE, --site=SITE
      [opt] file of site to be marked
  -b BED, --bed=BED
      [opt] BED file for region to be markered
  ```

- **Example command**:

  ```
  cgmaptools lollipop -i matrix.CG.gz -a anno.refFlat -f pdf
  ```

- **File format**:

  The input file format:

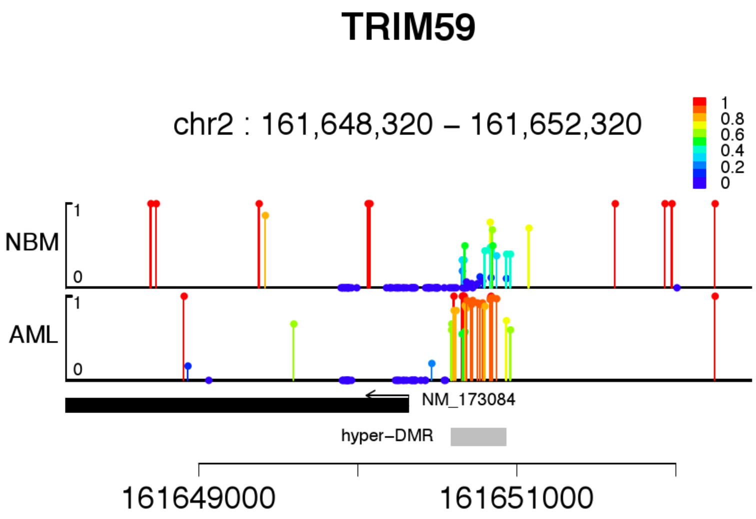

Figure 7.1: Lollipop example-1

> >= 3 columns, 1st line is the header, using R color name or "NaN". Can be output by CGmapFillIndex.py. Use STDIN if omitted.

```
chr    pos       E_vs_EMT    EMT_vs_M    E_vs_M
chr1    13116801    NaN        NaN          darkgreen
chr1    13116899    NaN        red          NaN
```

The bed file format:

> the first 4 columns are required.

```
chr1  213941196  213942363  REGION-1
chr1  213942363  213943530  REGION-2
```

7.2 heatmap

- **Description**: Plot methylation dynamics of target region for multiple samples heatmap.
- **Usage**: `cgmaptools heatmap [options]`

```
-i INFILE, --infile=INFILE
    input file
-o OUTFILE, --outfile=OUTFILE
    [opt] output file name. [default: mCBinHeatmap.SysDate.pdf]
-c, --cluster
    [opt] cluster samples by methylation in regions. [default: FALSE]
-l COLORLOW, --colorLow=COLORLOW
    [opt] color used for the lowest methylation value. [default: cyan3]
-m COLORMID, --colorMid=COLORMID
    [opt] color used for the middle methylation value. [default: null]
-b COLORHIGH, --colorHigh=COLORHIGH
    [opt] color used for the highest methylation value. [default: coral2]
```

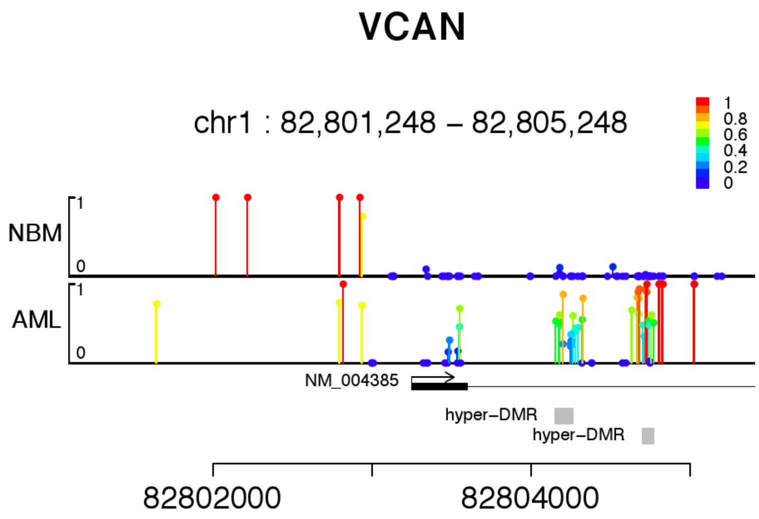

Figure 7.2: Lollipop example-2

```
-n COLORNUMBER, --colorNumber=COLORNUMBER
    [opt] desired number of color elements in the panel. [default: 10]
-W WIDTH, --width=WIDTH
    [opt] width of figure (inch). [default: 7]
-H HEIGHT, --height=HEIGHT
    [opt] height of figure (inch). [default: 7]
-f FORMAT, --format=FORMAT
    [opt] format of output figure. Alternative: png. [default: pdf]
-R RESOLUTION, --resolution=RESOLUTION
    [opt] Resolution in ppi. Only available for png format. [default: 300]
```

- **Example command**:

  ```
  cgmaptools heatmap -i mmbin -c -o cluster.pdf -f pdf
  ```

- **File format**:

  The input file format:

  The 1st line is the header. Each column contains methylation measurements of a sample.

  ```
  Region  Sample1  Sample2 ...
  Region1 0.1      0.1     ...
  Region2 0.1      0.1     ...
  ```

7.3 fragreg

- **Description**: Plot methylation dynamics of target and flanking region for multiple samples.
- **Usage**: `cgmaptools fragreg [options]`

  ```
  -i INFILE, --infile=INFILE
      input file
  ```

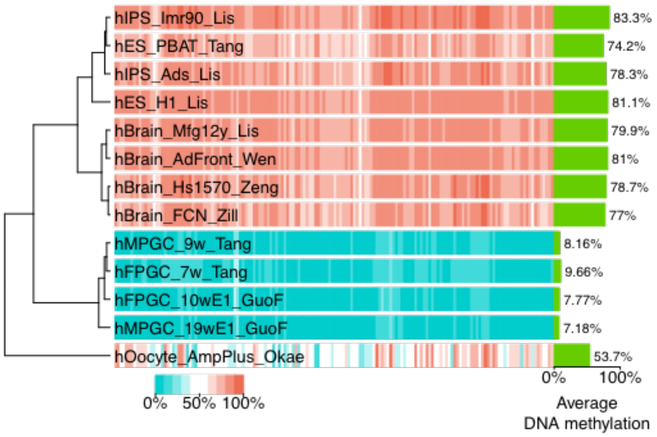

Figure 7.3: heatmap example-1

```
-r RATIO, --ratio=RATIO
    [opt] range ratio between target region and flanking region in plot. [default: 5]
-o OUTFILE, --outfile=OUTFILE
    [opt] output file name. [default: FragRegView.SysDate.pdf]
-W WIDTH, --width=WIDTH
    [opt] width of figure (inch). [default: 7]
-H HEIGHT, --height=HEIGHT
    [opt] height of figure (inch). [default: 7]
-f FORMAT, --format=FORMAT
    [opt] format of output figure. Alternative: png. [default: pdf]
-R RESOLUTION, --resolution=RESOLUTION
    [opt] Resolution in ppi. Only available for png format. [default: 300]
```

- **File format**:

  The input file format:

  The 1st line is the header. Each row contains methylation measurements of a sample.

```
Sample  Up1  Up2  ...  Region1  Region2 ...  Down1  Down2  ...
Sample1 0.1  0.1  ...  0.2      0.2     ...  0.3    0.3    ...
Sample2 0.1  0.1  ...  0.2      0.2     ...  0.3    0.3    ...
```

7.4 tanghulu

- **Description**: Show local mapped reads in Tanghulu shape.
- **Usage**: `cgmaptools tanghulu [options] -r <ref> -b <bam> -l chr1:133-144`

```
-r    Samtools indexed reference genome seqeunce, fasta format. eg. hg19.fa
      - use samtools to index reference: samtools faidx <hg19.fa>
-b    Samtools indexed Bam file to view.
      - use samtools to index bam file: samtools index <input.bam>
```

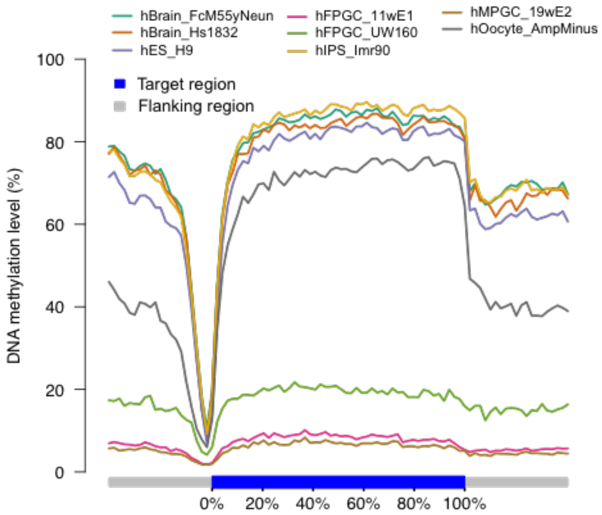

Figure 7.4: FragRegMC example

```
-l    Region in which to display DNA methylation.
      - or specify a single position (eg. heterozygous SNP site),
        we will show allele specific methylation.
-s    Path to samtools eg. /home/user/bin/samtools
      - by defualt, we try to search samtools in your system PATH.
-o    Output results to file [default: CirclePlot.Ctype.region.Date.pdf].
-t    C context. [default: CG]
      - available context: C, CG, CH, CW, CC, CA, CT, CHG, CHH
-d    Ouput device. [default: pdf]
      - alternative: png
-c    Seperate reads by chain. [default: OFF]
      - specify this option to turn ON.
-v    Show vague allele linked reads. [ default: OFF]
-g    Genotype of heterozygous SNP site.
      - This option provides two alleles of htSNP site. eg. AT
      - The genotype information can be used to reduce vague alleles.
      - This option is specific to display methylation in allele specific mode.
-D    Minimum number of reads (depth) covered in this region or allele linked.
      [default: 0|OFF]
-C    Minimum number of C (specified type) covered in this region or allele
      linked. [default: 0|OFF]
-W    Width of graphics reigon in inches. [default: 4]
-H    Height of graphics reigon in inches. [default: 4]
-R    Resolution in ppi. [default: 300]
      - only available for png device.
```

- **Example command**:

```
cgmaptools tanghulu -r genome.fa -b WG.bam -l chr1:2000-2400 -t CG
```

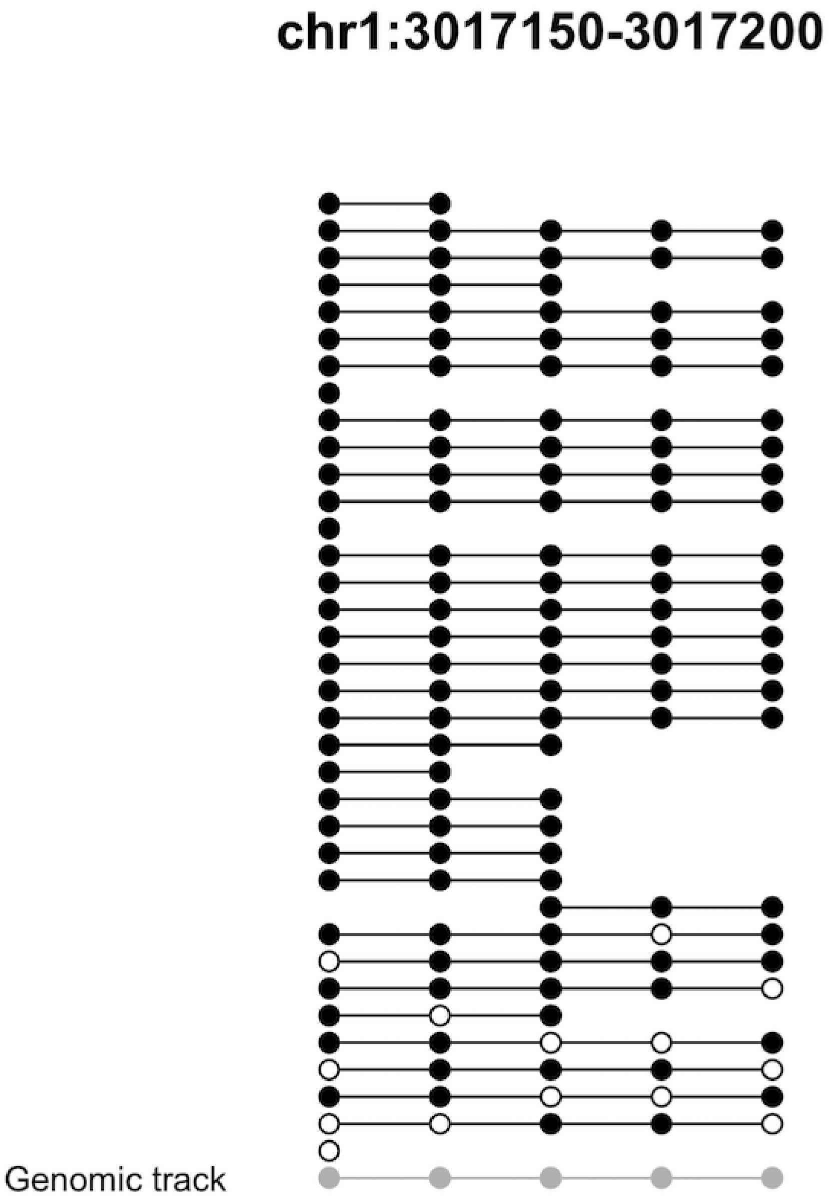

Figure 7.5: FragRegMC example

Chapter 8

Other Ultilities

8.1 findCCGG

- **Description**: Get MspI cutting sites for RRBS.
- **Usage**: `cgmaptools findCCGG -i <genome.fa> [-o <output>]`

  ```
  -i INFILE, --infile=INFILE
  -i FILE     Genome sequence file in Fasta format
  -o FILE     Name of the output file (standard output if not
              specified).Format: chr cCgg_pos ccGg_pos (0-base)
  ```

- **Example command**:

  ```
  cgmaptools findCCGG -i genome.fa -o genome.ccgg
  ```

- **File format**:

  The output file format:

  ```
  chr1    4025    5652
  chr1    8274    8431
  ```

8.2 bed2fragreg

- **Description**: Generate fragmented regions from BED file.
- **Usage**: `cgmaptools bed2fragreg [-i <BED>] [-n <N>] [-F <50,50,..> -T <50,..>] [-o output]`

  ```
  -i FILE      BED format, STDIN if omitted
  -F INT_list  List of region lengths in upstream of 5' end, Ex: 10,50. List
               is from 5'end->3'end
  -T INT_list  List of region lengths in downstream of 3' end, Ex: 40,20. List
               is from 5'end->3'end
  -n INT       Number of bins to be equally split [Default:1]
  ```

- **File format**:

  The input file format:

```
chr1   1000     2000    +
chr2   9000     8000    -
```

The output file format:

```
chr1   +    940  950  1000 1200 1400 1600 1800 1850
chr2   -    9060 9050 9000 8800 8600 8400 8200 8150
```