Whole Genome Bisulfite Core Quality Control metrics definition

Thomas Sierocinski[1], Annaick Carles[1], Martin Hirst[1,2]

1.Department of Microbiology and Immunology, Michael Smith Laboratories, Centre for High-Throughput Biology, University of British Columbia, 2125 East Mall, Vancouver BC V6T1Z4, Canada.
2. Canada's Michael Smith Genome Sciences Centre, BC Cancer Agency, 675 W. 10th Avenue, Vancouver, BC V5Z 1L3, Canada.

This document aims to define the Whole Genome Bisulfite Sequencing Quality Control Core Metrics, their corresponding flagging thresholds and how to compute them.

1 Core metrics definition and flagging thresholds

Core metrics were selected among a set of 25 metrics in order to minimize redundancy between them, thus maximizing their diversity and the aspect of the WGBS process they characterize. Thresholds were defined using a two-sigma approach:

$$Threshold = \bar{X} \pm 2\sigma$$

Where $\bar{X}$ represents the population mean and σ its standard deviation using the Center for Epigenome Mapping Technologies Whole Genome Bisulfite Sequencing libraries (n = 143).

1.1 Mapping efficiency

Extracted from the bamstat file (see section 2.1.3), it equals to:

$$\frac{Number\ of\ reads\ aligned}{total\ number\ of\ reads} * 100$$

The value for that metric should exceed 85%.

1.2 Estimated average coverage

Extracted from the bamstat file (section 2.1.4), it characterizes the average coverage per base. Values for this metric should be above 12 for a single lane.

1.3 Proportion of uniquely aligned reads without duplicates and a quality score > 10

Extracted from the bamstat file (see section 2.1.5), it counts the uniquely aligned reads with duplicates removed that have a Phred (like) quality score over 10. See here for more details about the scoring system. The proportion is computed as follows:

$$\frac{Number\ of\ uniquely\ aligned\ reads\ without\ duplicates\ and\ Q > 10}{Total\ number\ of\ reads} * 100$$

Values for this metric should exceed 85%.

1.4 Median CpG coverage

Generated using bedtools and novo5mc, it outputs the median coverage for CpGs on chromosome 1 (see section 2.2.2). Values for this metrics should exceed 2.

1.5 Bisulfite Conversion rate (human, lambda)

The bisulfite conversion procedure may or may not fully convert all non-methylated cytosines into uracils, depending on how well the conversion reaction was performed in the laboratory. In order to assess the conversion rate, lambda phage is typically spiked into a library. Lambda does not have methylated cytosine residues, so if the conversion reaction is complete, all the C's should be converted to T's in the sequenced reads that align to lambda. If the conversion is not 100%, then one can apply the conversion efficiency (say, 92%) to any results seen in the target species reads (i.e. human). See section 2.3 for more details about its computation.

Values for this metrics should exceed 97% in both human and lambda.

1.6 GC bias

This metrics characterizes the GC bias in the coverage values by computing the Pearson correlation between coverage values and GC content for 1000 bins of 100 base pair (see section 2.4 for more details about its computation).
The absolute value of the correlation score should not exceed 0.4.

2 Metrics computation

2.1 Computing the bamstats

2.1.1 Software and files dependencies

Samtools:
Bamstat.py:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/bamStats.py

It requires a bam file as an input and the size of the genome against which the data were aligned. It outputs some core quality control metrics amongst other ones.

2.1.2 Usage

First you will need to add the path to your samtools installation to the bamStat.py script, at line 19, replace the specified path (/path/to/your/samtools/install/samtools") by your own.

The script can then be used as follows (for hg19 genome build):

```
bamStats.py --genomesize=2864785220 –b your_bam_file.bam
> your_bamstats_file.bamstats
```

2.1.3 Mapping efficiency

The total number of reads and total number of reads aligned can be found in the resulting bamstat file. The mapping efficiency can be computed according to the formula in section 1.1.

For instance, using shell bash:

```
Total_nb_reads=`grep "Total_Number_Of_Reads" your_bamstat_file |
awk '{print $2}'`
nb_reads_aligned=`grep " Number_Reads_Aligned" your_bamstat_file
| awk '{print $2}'`
mapping_efficiency=$(echo "scale=2;  $ nb_of_reads_aligned *100 /
$Total_nb_reads" | bc)
```

2.1.4 Estimated average coverage

The estimated average coverage can be directly extracted from the bamstat files.
For instance, in shell bash:

```
estim_X_coverage=`grep “stimate_for_genome_X_coverage”
your_bamstat_file`
```

2.1.5 Proportion of uniquely aligned reads without duplicates and a quality score > 10

This metric can be derived for the bamstat file by applying formula detailed in section X. For instance, using shell bash:

```
Total_nb_reads=`grep "Total_Number_Of_Reads" your_bamstat_file |
awk '{print $2}'`
number_unique_aligned_reads_wodups_Q10=`grep “
Number_of_Uniquely_Aligned_Reads_without_Dups_and_”
your_bamstat_file`
proportion_uniquely_aligned_wodups_Q10=$(echo "scale=2;
$number_unique_aligned_reads_wodups_Q10 *100 / $Total_nb_reads" |
bc)
```

2.2 Median CpG coverage

2.2.1 Software and files dependencies

Computing this metric require the following softwares and scripts:
R: https://cran.r-project.org/
Samtools: `git repository://github.com/samtools/samtools.git`
Java Runtime environment (build 1.7.0_03-b04):
http://www.oracle.com/technetwork/java/javase/downloads/java-archive-downloads-javase7-521261.html#jre-7u3-oth-JPR
Novo5mc:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/Novo5mC.jar
novo5mc.CpGcoverage.sh:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/novo5mc.CpGcoverage.sh
novo5mc_meanMedianCpGcov_4all.R:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/novo5mc_meanMedianCpGcov_4all.R

Required files:
bam file: must be sorted by position.
Region file:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/chr1.regionfile
Reference genome file:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/GRCh37-lite.fa.gz
(need to be unzipped before usage)

2.2.2 Computation

First, we run Novo5mc:

```
/path/to/java/runtime/environment/bin/java -jar -Xmx10G /path/to
/file/Novo5mC.jar -bam /path/to/sorted/bam -out
/path/to/the/desired/output/folder/  -genome
/path/to/reference/genome/used/for/alignment/genome.fa -q5 -F 1540
-minCoverage 3 -name output_filename -samtools /path/to/samtools -
regions 1 > /desired/path/to/your/log/file.log
```

This generates a file bearing the "`.5mC.CpG`" extension in the output folder. We then run the novo5mc.CpGcoverage.sh script on it:

```
/path/to/novo5mc.CpGcoverage.sh your_file.5mC.CpG
```

This generates a file bearing the "`.avCov`" extension in the current folder. We then run the novo5mc_meanMedianCpGcov_4all.R:

```
Rscript novo5mc_meanMedianCpGcov_4all.R your_avCov_file.avCov >
CpG_coverage.output
```

This generates a file with the "CpG_coverage" prefix from which we can extract the median CpG coverage, for instance using shell bash:

```
cpg_coverage_median=`grep -m4 "\[1\] " your_CpG_coverage_file |
tail -n1 | sed 's/\[1\] //g'`
```

2.3 Bisulfite Conversion rate

2.3.1 Software and files dependencies

Samtools: git repository://github.com/samtools/samtools.git
Novomethyl (novocraft-3.02.10): http://www.novocraft.com/support/download/
Human Reference genome file:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/GRCh37-lite.fa.gz
Lambda Reference genome file:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/lambda_NC_001416.1.fasta
Bam file: sorted by position and duplicates marked

2.3.2 Computation

Novomethyl requires two preliminary steps: split of the bam file into CT and GA strands:

```
samtools view -h your_bam | grep -e "^@\|ZB:Z:CT" | samtools view -ubS - |
samtools sort – output_filename.CT
samtools view -h your_bam | grep -e "^@\|ZB:Z:GA" | samtools view -ubS - |
samtools sort – output_filename.GA
```

Then running samtools mpileup

NOTE : CT bam file should be first

```
samtools mpileup -BC 0 -q 30 -f reference_genome_file
output_filename.CT.bam output_filename.GA.bam |
gzip -c > output_filename.mpileup.txt.gz
```

Running Novomethyl (for human and lambda)

```
gunzip -c output_filename.mpileup.txt.gz | Path_To_Novomethyl -o Consensus -% 2>
output_filename.methylation.log
```

The last lines of the methylation log file shows the estimated bisulfite conversion rate.

```
tail -2 output_filename.methylation.log
```

2.4 GC Bias

2.4.1 Software and file dependencies

R: https://cran.r-project.org/
Bedtools: https://github.com/arq5x/bedtools2/releases/download/v2.25.0/bedtools-2.25.0.tar.gz
GCbias_all.R:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/GCbias_all.R
Bins position file:
https://svn.bcgsc.ca/bitbucket/projects/EDCC/repos/opencemt/browse/WGBS/QC/hg19a.gc.1M

2.4.2 Computation

We first compute the coverage for the given bam

```
bedtools coverage -abam your_bam -b path_to_bins_position_file -
counts > path/to/desired/output.gc_cov.1M.txt
```

Using the ".gc_cov.1M.txt" output file and the GCBias_all.R script, we can compute the CG bias:

```
Rscript GCbias_all.R $input_file > $out_file
```

The results can be extracted from the output:

```
gc_content_avcov_spearman_rho=`awk 'f{print;f=0} /rho/{f=1}'
$out_file`
```