# ChIP sequencing quality control metrics definition

Myrto Kostadima[1], Sitanshu Gakkhar[2], Ewa Bergmann[3], Thomas Manke[3], Daniel Zerbino[1], Martin Hirst[2,3] and Paul Flicek[1]

1. European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Genome Campus, Hinxton, Cambridge CB10 1SD, United Kingdom
2. Department of Microbiology and Immunology, Michael Smith Laboratories, Centre for High-Throughput Biology, University of British Columbia, 2125 East Mall, Vancouver BC V6T1Z4, Canada
3. Canada's Michael Smith Genome Sciences Centre, BC Cancer Agency, 675 W. 10th Avenue, Vancouver, BC V5Z1L3, Canada
4. Max Planck Institute of Immunobiology and Epigenetics, 79108 Freiburg, Germany

This document aims to define the ChIP sequencing Quality Control Core Metrics and how to compute them. All commands given are in bash. The google document summarizing the metrics used across the different IHEC consortia can be found here:

https://docs.google.com/spreadsheets/d/1emtzMM9qBcOgPT6Jne0n5N-SpRdLUaCBsWCS5koMhRw/edit?usp=sharing

## 1. Pre-processing

Prior to calculating the metrics defined below we assume the following pre-processing steps for single-end reads following alignment using BWA:

```
## The following commands assume that there is a pair of BAM files, one for the ChIP and one for the Input, $ChIP_original_BAM_file and $Input_original_BAM_file for two samples labelled, $ChIP_sampleName and $Input_sampleName, respectively.

## The following steps are shown for the $ChIP_sampleName but have to be applied to the $Input_sampleName too:

## Sort the BAM file by coordinate

java -Xmx2048m -jar picard.jar SortSam INPUT=$ChIP_original_BAM_file OUTPUT=${ChIP_sampleName}_original.sorted.bam SORT_ORDER=coordinate VALIDATION_STRINGENCY=SILENT

## Mark, but not remove, duplicate reads

java -Xmx2048m -jar picard.jar MarkDuplicates INPUT=${ChIP_sampleName}_original.sorted.bam OUTPUT=${ChIP_sampleName}_markDup.bam METRICS_FILE=${ChIP_sampleName}_original.sorted_metrics.out REMOVE_DUPLICATES=false ASSUME_SORTED=true VALIDATION_STRINGENCY=SILENT

## Remove unmapped reads and those with mapping quality less than 5:

samtools view -b -F 4 -q 5 ${ChIP_sampleName}_markDup.bam > ${ChIP_sampleName}_quality_filtered.bam

## Remove duplicate reads:
```

```
samtools view -b -F 1024 ${ChIP_sampleName}_quality_filtered.bam > ${ChIP_sampleName}_dedup.bam



## Index the final deduplicated BAM file

samtools index ${ChIP_sampleName}_dedup.bam
```

## 2. Mappability

We want to extract the following mapping statistics:

```
## The original number of reads, the number of those aligned, the number of duplicate reads, the duplicate percentage and the final number of reads after deduplication and removal of reads with MAPQ<5:

samtools flagstat ${ChIP_sampleName}_markDup.bam > ${ChIP_sampleName}_markDup_flagstat.txt

total_reads=`grep "in total" ${ChIP_sampleName}_markDup_flagstat.txt | sed -e 's/ + [[:digit:]]* in total .*//'`

mapped_reads=`grep "mapped (" ${ChIP_sampleName}_markDup_flagstat.txt | sed -e 's/ + [[:digit:]]* mapped (.*)//'`

dupped_reads=`grep "duplicates" ${ChIP_sampleName}_markDup_flagstat.txt | sed -e 's/ + [[:digit:]]* duplicates$//'`

dup_rate=$(echo "${dupped_reads}/${mapped_reads}" | bc -l)

## Finally, the number of singletons for paired-end data sets can be calculated using:

left_singletons=`grep "singletons" ${ChIP_sampleName}_markDup_flagstat.txt | sed -e 's/ + [[:digit:]]* singletons .*//'`

right_singletons=`grep "singletons" ${ChIP_sampleName}_markDup_flagstat.txt | sed -e 's/[[:digit:]]* + //;s/ singletons .*//'`

singletons=$((left_singletons+right_singletons))



## The final number of reads:

samtools flagstat ${ChIP_sampleName}_dedup.bam > ${ChIP_sampleName}_dedup_flagstat.txt

final_reads=`grep "mapped (" ${ChIP_sampleName}_dedup_flagstat.txt | sed -e 's/ + [[:digit:]]* mapped (.*)//'`
```

## 3. Calculating Jensen-Shannon distance (JSD) and CHANCE divergence

To calculate those we run:

```
## Attention: Regarding the bin size (specified in the command below by the ‘-bs’ option) the agreement across the IHEC ASWG is 200 bp for sharp marks and 1,000 bp for broad marks.

## No need to remove the blacklisted regions for the JSD calculation.

  if [[ type == "H3K27ac" || type == "H3K4me3" || type == "H2AFZ" || type == "H3ac" || type == "H3K4me2" || type == "H3K9ac" ]]
  then
    bin_size=200
  else
    bin_size=1000
  fi

plotFingerprint -b ${ChIP_sampleName}_dedup.bam ${Input_sampleName}_dedup.bam -bs ${bin_size} -l $ChIP_sampleName $Input_SampleName --JSDsample ${Input_sampleName}_dedup.bam --outQualityMetrics ${ChIP_sampleName}_fingerprint.txt -plot ${ChIP_sampleName}_fingerprint.png -p 8

js_dist=`grep ${ChIP_sampleName} ${ChIP_sampleName}_fingerprint.txt | cut -f8`

chance_div=`grep ${ChIP_sampleName} ${ChIP_sampleName}_fingerprint.txt | cut -f12`
```

### 4. Calculating FRiP scores

```
## The following command assumes that there is a BED file, $bed_file, containing the peaks for ${ChIP_sampleName}.

reads_under_peaks=`bedtools intersect -wa -bed -abam ${ChIP_sampleName}_dedup.bam -b ${bed_file} | wc -l`

frip=$(echo "${reads_under_peaks}/${final_reads}" | bc -l)
```

### 5. Tools and versions:

To calculate the metrics we use:

- **samtools** v 1.3.1
- **picard** v 2.9.0

- **plotFingerprint** v 2.4.2
- **bedtools** v 2.26