NGS_assignment_wk12

XiaoyanWen

4/18/2022

Next Generation Sequence Analysis Homework Week 12

ChiP-seq is a common NGS application that aims to characterize either the locations of protein (e.g., transcription factors binding to DNA) or of histone modifications ("histone marks") which impact chromatin structure.

In Week 11, you conducted analysis of a transcription factor (TF) ChIP-seq dataset from a biopsy of prostate cancer tumors where you processed fastqs, aligned ChIP reads and control (="input") reads to the human reference genome, sorted and filtered the BAMs and called peaks with MACS2.

Here you will generate quality control metrics from ChIP-seq libraries which are essential steps to analysis of ChIP-seq data.

About the data

The data for Task 1 are transcription factor ChIP-seq data for Androgen Receptor from Week 11, while the data for Task 2 are H3K36me3 ChIP-seq data from the ENCODE project.

The SRA run accession numbers for the reads in Task 1 are:

- SRR7207011 is the Androgen Receptor ChIP for patient 1 (P1_AR_DSG)
- SRR7207017 is the Androgen Receptor ChIP for patient 2 (P2_AR_DSG)
- SRR7207089 is the "input" (P_Input_DSG)

Task 1: MACS2 peaks and FRiP scores

MACS2 outputs a **narrowPeak file** which is a format used by the ENCODE project to store peak intervals, P values, and related information. The format is defined at the UCSC website here:

http://genome.ucsc.edu/FAQ/FAQformat.html#format12

Review the column definitions of the narrowPeak format, then answer the following. Please provide the relevant R code used to answer the questions where applicable

1*point*

.

Q1.1a: How many peaks were called by MACS2 for each of the two Androgren Receptor ChIP-Seq samples in your MACS2 outputs from last week?

```
test1_peaks <- read.delim("test1_peaks.narrowPeak", header=FALSE)
colnames(test1_peaks) <- c('chrom','chromStart','chromEnd','name','score','strand','signa
lValue','pValue','qValue','peak')
head(test1_peaks)
```

```
##   chrom chromStart chromEnd         name score strand signalValue  pValue
## 1     1    2255323  2255706 test1_peak_1   126      .     8.11804 17.4005
## 2     1    2256038  2256255 test1_peak_2    61      .     5.72274 10.2513
## 3     1    2410902  2411135 test1_peak_3    90      .     6.75168 13.3976
## 4     1    3529493  3529758 test1_peak_4   194      .    11.17000 24.5662
## 5     1    5494583  5494855 test1_peak_5    80      .     6.94508 12.2990
## 6     1    5727288  5727543 test1_peak_6   166      .     7.20623 21.5840
##     qValue peak
## 1 12.69970  175
## 2  6.15537  156
## 3  9.01558   56
## 4 19.41290  125
## 5  8.01057  154
## 6 16.60940   98
```

```
nrow(test1_peaks)
```

```
## [1] 2001
```

```
test2_peaks <- read.delim("test2_peaks.narrowPeak", header=FALSE)
colnames(test2_peaks) <- c('chrom','chromStart','chromEnd','name','score','strand','signa
lValue','pValue','qValue','peak')
head(test2_peaks)
```

```
##   chrom chromStart chromEnd         name score strand signalValue   pValue
## 1     1    1433290  1433483 test2_peak_1    62      .     4.89448  9.25272
## 2     1    1658981  1659178 test2_peak_2   106      .     7.81004 13.84700
## 3     1    1758776  1758963 test2_peak_3    48      .     4.82947  7.76074
## 4     1    1782810  1783061 test2_peak_4    54      .     5.22227  8.41215
## 5     1    1892595  1892800 test2_peak_5    74      .     5.15279 10.47360
## 6     1    1894641  1894850 test2_peak_6    70      .     5.76460 10.06710
##     qValue peak
## 1  6.28193  142
## 2 10.62870  101
## 3  4.88303  128
## 4  5.49946  185
## 5  7.42504  120
## 6  7.05026   52
```

```
nrow(test2_peaks)
```

```
## [1] 20783
```

Answer: 2001 peaks were called for patient 1 (SRR7207011 vs. input); 20783 peaks were called for patient 2 (SRR7207017 vs. input).

Q1.1b: What is the mean peak width for each sample? Show the R command (or other approach) you used to arrive at your answer.

```
test1_peaks$peakWidth <- test1_peaks$chromEnd - test1_peaks$chromStart
summary(test1_peaks$peakWidth)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   184.0   206.0   238.0   255.8   284.0   827.0
```

```
test2_peaks$peakWidth <- test2_peaks$chromEnd - test2_peaks$chromStart
summary(test2_peaks$peakWidth)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
##   187.0   231.0   297.0   337.3   400.0  1912.0 
```

Answer: the mean peak Width for patient 1 sample is 255.8; for patient 2 sample is 337.3

Q1.1c: What is meant by the "signalValue" in column 7?

Answer: 'signalValue' means signal to noise ratio.the value is considered 'good' when > 1.05.

Q1.2. The Fraction of Reads in Peaks (FRiP) score is a simple measure of library quality that is useful in identifying low-quality libraries from a set of ChIP-seq libraries (i.e., it is useful in a comparative context). The FRiP score is calculated as implied by the name and is simply the number of reads overlapping called peaks divided by the total number of reads.

High quality libraries typically have FRiP scores > 1% (See Week 11 pre-recorded videos and the Landt et al. Week 11 reading)

The bedtools software suite is a useful set of methods for working with intervals (e.g., peak regions) and is a commandline set of tools you should be aware of when considering problems in NGS analysis. Here you will use the **bedtools intersect function** to calculate the FRiP score separately for each of the two Androgen Receptor CHiP BAMs from last week.

https://bedtools.readthedocs.io/en/latest/

Review the bedtools intersect tool:

https://bedtools.readthedocs.io/en/latest/content/tools/intersect.html

Begin by logging in to Greene and check out a compute node with 8 GB of memory.

```
srun --time=4:00:00 --mem=8GB --pty /bin/bash
```

Now load the most recent bedtools module and run bedtools intersect to count the reads in each BAM that intersect intervals in the narrowPeak files (numerator in the FRiP score). Please use the corresponding MACS2 *narrowPeak* file for each BAM.

```
module load bedtools/intel/2.29.2

bedtools intersect -a test1_peaks.narrowPeak -b SRR7207011.sorted.lowqfiltered.bam -C -bed >both_test1.bed

bedtools intersect -a test2_peaks.narrowPeak -b SRR7207017.sorted.lowqfiltered.bam -C -bed >both_test2.bed

module purge
```

```
both_test1 <- read.delim("both_test1.bed", header=FALSE)

colnames(both_test1) <- c('chrom','chromStart','chromEnd','name','score','strand','signalValue','pValue','qValue','peak', 'intersect_count')

head(both_test1)
```

```
##   chrom chromStart chromEnd         name score strand signalValue  pValue
## 1     1    2255323  2255706 test1_peak_1   126      .     8.11804 17.4005
## 2     1    2256038  2256255 test1_peak_2    61      .     5.72274 10.2513
## 3     1    2410902  2411135 test1_peak_3    90      .     6.75168 13.3976
## 4     1    3529493  3529758 test1_peak_4   194      .    11.17000 24.5662
## 5     1    5494583  5494855 test1_peak_5    80      .     6.94508 12.2990
## 6     1    5727288  5727543 test1_peak_6   166      .     7.20623 21.5840
##     qValue peak intersect_count
## 1 12.69970  175              42
## 2  6.15537  156              22
## 3  9.01558   56              28
## 4 19.41290  125              36
## 5  8.01057  154              26
## 6 16.60940   98              47

sum(both_test1$intersect_count)

## [1] 69002

both_test2 <- read.delim("both_test2.bed", header=FALSE)

colnames(both_test2) <- c('chrom','chromStart','chromEnd','name','score','strand','signal
Value','pValue','qValue','peak', 'intersect_count')

head(both_test2)

##   chrom chromStart chromEnd         name score strand signalValue   pValue
## 1     1    1433290  1433483 test2_peak_1    62      .     4.89448  9.25272
## 2     1    1658981  1659178 test2_peak_2   106      .     7.81004 13.84700
## 3     1    1758776  1758963 test2_peak_3    48      .     4.82947  7.76074
## 4     1    1782810  1783061 test2_peak_4    54      .     5.22227  8.41215
## 5     1    1892595  1892800 test2_peak_5    74      .     5.15279 10.47360
## 6     1    1894641  1894850 test2_peak_6    70      .     5.76460 10.06710
##     qValue peak intersect_count
## 1  6.28193  142              22
## 2 10.62870  101              16
## 3  4.88303  128              14
## 4  5.49946  185              20
## 5  7.42504  120              24
## 6  7.05026   52              21

sum(both_test2$intersect_count)

## [1] 877004
```

Now load the most recent samtools view module and count the total number of reads in each BAM. This is the denominator in the FRiP score.

```
module load samtools/intel/1.14
samtools view -c SRR7207011.sorted.lowqfiltered.bam
## 22517899
samtools view -c SRR7207017.sorted.lowqfiltered.bam
## 20702991
```

For your Q1.2 answer, report the FRiP score for each of the two androgen receptor ChIP-seq libraries and the command lines you used to generate your answer. Do the two androgen receptor libraries pass the 1% threshold typical of high quality ChIP-seq libraries?

2points

.

```
sprintf('FRiP score for patient 1 androgen receptor ChIP-seq library: %f percent', sum(bo
th_test1$intersect_count)/22517899*100)

## [1] "FRiP score for patient 1 androgen receptor ChIP-seq library: 0.306432 percent"

sprintf('FRiP score for patient 2 androgen receptor ChIP-seq library: %f percent', sum(bo
th_test2$intersect_count)/20702991*100)

## [1] "FRiP score for patient 2 androgen receptor ChIP-seq library: 4.236122 percent"
```

Answer: only patient 2 androgen receptor ChIP-seq library (SRR7207017) pass the pass the 1% threshold typical of high quality ChIP-seq library.

Task 2: QC analysis of H3K36me3 ChIP-seq with the BioConductor ChIC package

In this Task, you will work with histone ChIP-seq data. Histone ChIP-seq aims to identify regions of the genome with a given type of histone modification.

In the Week 11 pre-recorded video and web session, we discussed a number of metrics that are used to evaluate the quality of ChIP-seq libraries. These included the FRiP score and cross-correlation profiling metrics (e.g., NSC). Here you will generate quality metrics for a H3K36me3 histone modification ChIP-seq experiment from the ENCODE project.

The BioConductor ChIC package provides functions for calculating these and related metrics for evaluating ChIP-seq library quality. You can read more about it at the BioConductor page for the package, the package vignette (tutorial), and the package documentation:

https://bioconductor.org/packages/release/bioc/html/ChIC.html

https://bioconductor.riken.jp/packages/3.10/bioc/vignettes/ChIC/inst/doc/ChIC-Vignette.pdf

https://bioconductor.riken.jp/packages/3.10/bioc/manuals/ChIC/man/ChIC.pdf

This section of the assignment is derived primarily from the ChIC Vignette.

To illustrate the use of the ChIC package, we will use toy control and sample H3K36me3 ChIP-seq BAMs from ENCODE.

You can conduct the following analysis on Greene or locally on your personal computer. The BAM files are small (~400 Mb each), so you may download these files directly to your personal computer.

The ChIP BAM is on Greene at: `/scratch/work/courses/BI7653/hw12.2022/ENCFF000BFX.bam`

The control ("input") BAM is on Greene at: `/scratch/work/courses/BI7653/hw12.2022/ENCFF000BDQ.bam`

Now, create a directory for this task (e.g., in your `/scratch` if you are working on hpc)

Copy the two BAMs to your task directory.

Next, install the ChIC BioConductor package from the R console using the installation instructions at the Bioconductor page for the package (link above).

Once installed, load the ChIC library in R.

```
BiocManager::install("ChIC")

## 'getOption("repos")' replaces Bioconductor standard repositories, see
## '?repositories' for details
##
## replacement repositories:
##     CRAN: https://cloud.r-project.org

## Bioconductor version 3.14 (BiocManager 1.30.16), R 4.1.3 (2022-03-10)

## Installation paths not writeable, unable to update packages
##   path: /usr/lib/R/library
##   packages:
##     spatial, survival

library(ChIC)
```

```
## Loading required package: spp

## Loading required package: Rcpp

setwd("/mnt/c/Users/wen_x/Downloads/NGS/wk12")
```

Now define two variables with the prefix for the two BAM filenames (required by ChIC) and read the BAM files.

```
chipName <- "ENCFF000BFX"
inputName <- "ENCFF000BDQ"

chipBam <- readBamFile(chipName)
inputBam <- readBamFile(inputName)
```

```
## 9650808 reads from ENCFF000BFX
## 9983747 reads from ENCFF000BDQ
```

Define variables which we will use to run the analysis with multiple threads below.

```
mc <- 4
cluster <- parallel::makeCluster( mc )
```

Now we will execute functions required for generating a cross-correlation profile plot. Cross-correlation profiling is a key method used by the ENCODE project to assess ChIP-seq library quality. We discussed the cross-correlation profile method in some detail in the pre-recorded videos from Week 11. You may also read about this method in Landt et al. 2012 primary reading from Week 10.

The first steps to conducting cross-correlation profiling with ChIC is to determine binding peak separation distance and approximate window size that should be used for binding detection. Please execute the following interactively.

```
chip_binding.characteristics <- get.binding.characteristics(chipBam,
                                                            srange=c(0,500),
                                                            bin = 5,
                                                            accept.all.tags = TRUE,
                                                            cluster = cluster)

input_binding.characteristics <- get.binding.characteristics(inputBam,
                                                             srange=c(0,500),
                                                             bin = 5,
                                                             accept.all.tags = TRUE,
                                                             cluster = cluster)
parallel::stopCluster( cluster )
```

Note that the output variables are lists with information describing the cross-correlation profile. For example, the chip_binding.characteristics and input_binding.characteristics variables contain cross-correlation data required to construct the cross-correlation profile.

```
class(chip_binding.characteristics)
```

```
## [1] "list"
## $cross.correlation

## $peak
## $peak$x
## [1] 195

## $peak$y
## [1] 0.2263048


## $whs
## [1] 395
```

```
chip_binding.characteristics
```

Description:df [101 × 2]

x <dbl>	y <dbl>
0	0.2038080
5	0.2050042
10	0.2081550
15	0.2102577
20	0.2128025
25	0.2158619
30	0.2193696
35	0.2218041
40	0.2183645
45	*0.2153960*

Now using the output binding characteristics from above, we can calculate the cross correlation QC-metrics (e.g., NSC and RSC) that are used to measure signal-to-noise for the ChIP sample and generate the cross-correlation plot. Execution of the command below should create a pdf in your working directory called CrossCorrelation.pdf.

```
crossvalues_Chip <- getCrossCorrelationScores( chipBam,
                                               chip_binding.characteristics,
                                               read_length = 36,
                                               annotationID="hg19",
                                               savePlotPath = ".",
                                               mc = mc)
```

```
hg19 valid annotation...
load chrom_info
(-) [=======>---------------------------------------]  17%
Phantom peak and cross-correlation...
(|) [===============>-------------------------------]  33%
smoothing...
Check strandshift...
200
strandshift not adjusted...
(/) [===================>---------------------------]  42%
Phantom peak with smoothing...
Crosscorrelation plot saved under .
plot cross correlation curve with smoothing
pdf saved under./ChIPCrossCorrelation.pdf
(-) [======================>------------------------]  50%
```

```
NRF calculation
(\) [============================>--------------------]  58%
calculate alternative QC scores...
(|) [================================>----------------]  67%System has not been booted with
systemd as init system (PID 1). Can't operate.
Failed to create bus connection: Host is down
Show Traceback

no loop for break/next, jumping to top level
```

Confirm that CrossCorrelation.pdf was generated successfully.

Review the contents of the crossvalues_Chip variable noting that this variable is a list that includes quality control metrics NSC and RSC.

```
class(crossvalues_Chip)
crossvalues_Chip
```

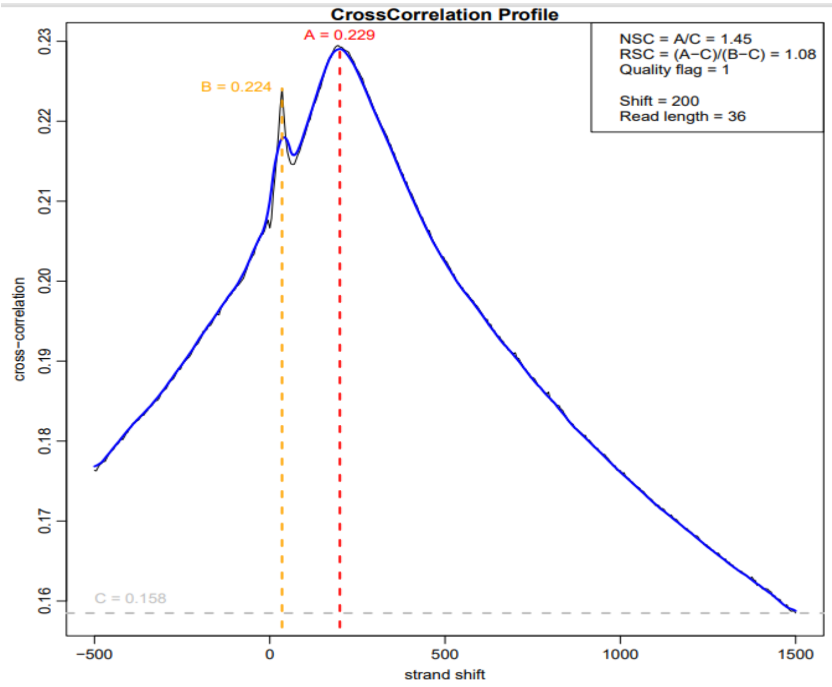
```
Error: object 'crossvalues_Chip' not found
```

Now use the cross-correlation profile figure (CrossCorrelation.pdf) and crossvalues_Chip list contents to answer the following questions.

Q2.1a.Include the cross-corrleation profile plot in your answers file

$$1point$$

.



Answer:

Q2.1b Select all that are true statements about the cross-correlation profile

$$1point$$

:

1. aligned read asymmetries between DNA strands are the basis for the cross-correlation profile approach to ChIP-seq QC
2. cross-correlation profiling requires peaks to be called first using a tool like MACS2
3. correlations between strand-specific depths are recalculated after performing a “strand shift”
4. a single correlation is calculated between strand-specific depth
5. high quality ChIP-seq libraries will typically have a single cross-correlation peak
6. NSC values below a pre-defined threshold are of acceptable quality for downstream analysis
7. NSC values above a pre-defined threshold are of acceptable quality for downstream analysis

   Answer: 1,3,4,7

Q2.2. Now answer the following questions

$$2 points$$

.

Q2.2a What is the NSC value for the ChIP sample?

Answer: NSC = A/C = 1.45

Q2.2b What is the RSC values for the ChIP sample?

Answer: RSC = (A-C)/(B-C) = 1.08

Q2.2c Do either of the metrics incorporate the “shaddow peak” height in how they are calculated? Which one(s)?

Answer: RSC calculation incorporated the “shadow peak” height B.

Q2.2d Landt et al. 2012 (see “Cross-correlation Analysis”) provide minimum NSC and RSC values for libraries with acceptable signal-to-noise ratios. What are these minimum values and does this library pass ENCODE standards for quality control? Select one:

1. minimum NSC = 0.95, minimum RSC = 0.7, the library fails QC
2. minimum NSC = 0.95, minimum RSC = 0.7, the library passes QC
3. minimum NSC = 1.0, minimum RSC = 0.75, the library fails QC
4. minimum NSC = 1.0, minimum RSC = 0.75, the library passes QC
5. minimum NSC = 1.05, minimum RSC = 0.8, the library fails QC
6. minimum NSC = 1.05, minimum RSC = 0.8, the library passes QC

   Answer: 6

Now execute the <getCrossCorrelationScores function> for the control ("input") sample and generate a cross-correlation profile plot. **Be sure to change the BAM input and the binding characteristics argument in the getCrossCorrelationScores function.**

```
crossvalues_Input <- getCrossCorrelationScores(inputBam,
                                               input_binding.characteristics,
                                               read_length = 36,
                                               annotationID="hg19",
                                               savePlotPath = ".",
                                               mc = mc)
```

```
hg19 valid annotation...
load chrom_info
(-) [=======>---------------------------------------]  17%
Phantom peak and cross-correlation...
(|) [==============>--------------------------------]  33%
smoothing...
Check strandshift...
140
Strandshift is adjusted
(/) [==================>----------------------------]  42%
Phantom peak with smoothing...
Crosscorrelation plot saved under .
plot cross correlation curve with smoothing
pdf saved under./ChIPCrossCorrelation.pdf
(-) [======================>------------------------]  50%
NRF calculation
(\) [===========================>-------------------]  58%
calculate alternative QC scores...
(|) [================================>--------------]  67%System has not been booted with
systemd as init system (PID 1). Can't operate.
Failed to create bus connection: Host is down
```

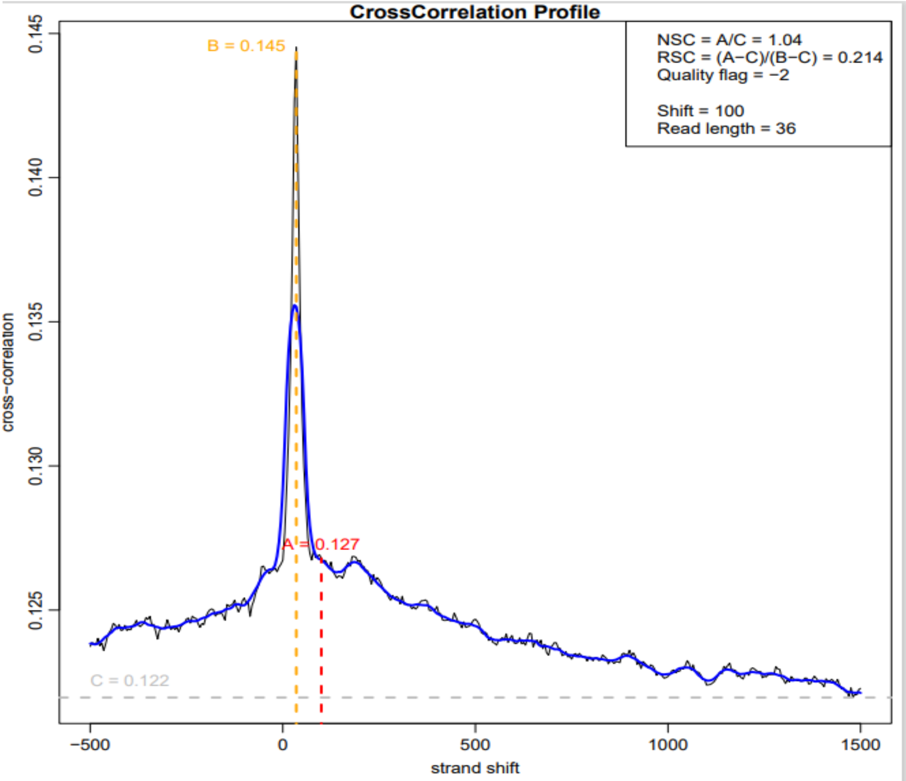
```
Show Traceback

no loop for break/next, jumping to top level
```

Q2.3 Now answer the following for the input sample. **Be sure to include the cross-correlation profile figure for the input sample in your answer**

$$1point$$

.



Answer:

Q2.3a What steps are typically included/excluded in the preparation of the control/input sample? (see Week 11 pre-recorded video). Select all true statements.

1. chemical cross-linking of DNA and protein
2. fragmentation of DNA
3. chromatin immunoprecipitation with an appropriate antibody
4. unlinking of DNA and protein
5. library preparation and sequencing of DNA

Answer: 1,2,3,4,5

Q2.3b What are the NSC and RSC values for the input sample? Would this library pass the quality control standards for the ENCODE project if it were a ChIP sample?

Answer: NSC = A/C = 1.04; RSC = (A−C)/(B−C) = 0.214

No, this library wouldn't pass the quality control standards for the ENCODE project if it were a ChIP sample.

Q2.4. ChIC calculates "Global Metrics" for ChIP-seq quality control. These metrics center around the fingerprint plot and nine metrics that can be derived from such plots (which are not discussed in the pre-recorded videos or elsewhere in the course).

Answer:

To generate a fingerprint plot, we need to calculate smoothed tag (read) density distributions. We first remove anomalies in the BAMs as follows (see vignette)

```
    selectedTags <- removeLocalTagAnomalies(chipBam,
                                            inputBam,
                                            chip_binding.characteristics
                                            )
    inputBamSelected <- selectedTags$input.dataSelected
    chipBamSelected <- selectedTags$chip.dataSelected
```

```
Filter tags
SKIP select.informative.tags filter
remove.local.tag.anomalies filter
Warning in for (a in args) { :
  closing unused connection 10 (<-localhost:11661)
Warning in for (a in args) { :
  closing unused connection 9 (<-localhost:11661)
Warning in for (a in args) { :
  closing unused connection 8 (<-localhost:11661)
Warning in for (a in args) { :
  closing unused connection 7 (<-localhost:11661)
```

We then extract the “tag shift” value from the cross-correlation output.

```
    finalTagShift <- crossvalues_Chip$tag.shift
    smoothedChip <- tagDensity(chipBamSelected,
                               annotationID = "hg19",
                               tag.shift = finalTagShift, mc = mc)
    smoothedInput <- tagDensity(inputBamSelected,
                                annotationID = "hg19",
                               tag.shift = finalTagShift, mc = mc)
```

```
Error: object 'crossvalues_Chip' not found
```

Now we will create a fingerprint plot and calculate global entrichment metrics

```
    Ch_Results <- qualityScores_GM(densityChip = smoothedChip,
                                     densityInput = smoothedInput,
                                     savePlotPath = ".")
```

The above will produce FingerPrintPlot.pdf in your present working directory.

Imagine the input library is also a ChIP sample (not a control). Which of the following is the best interpretations of the fingerprint plot

$$1point$$

.

1. the ChIP sample is higher quality because the library is more complex (i.e., has fewer duplicate reads)
2. the ChIP library is of higher quality because the read depth in peaks is higher
3. the ChIP sample is higher quality because a greater proportion of reads are in higher ranked bins (i.e., bins ranked by depth) than the input

4. none of the above

Answer: 4

A common way of evaluating quality of ChIP-seq data and gaining biological insight into transcription factors or histone marks is to create a profile, or “enrichment” plot. A common type of plot in this category is the Transcription Start Site (TSS) profile plot which is described in the ChIC vignette categorized as a “Local Enrichment Metric (LM)”.

A TSS profile plot for the sample and control BAMs you analyzed above is provided for you in Fig. 1.

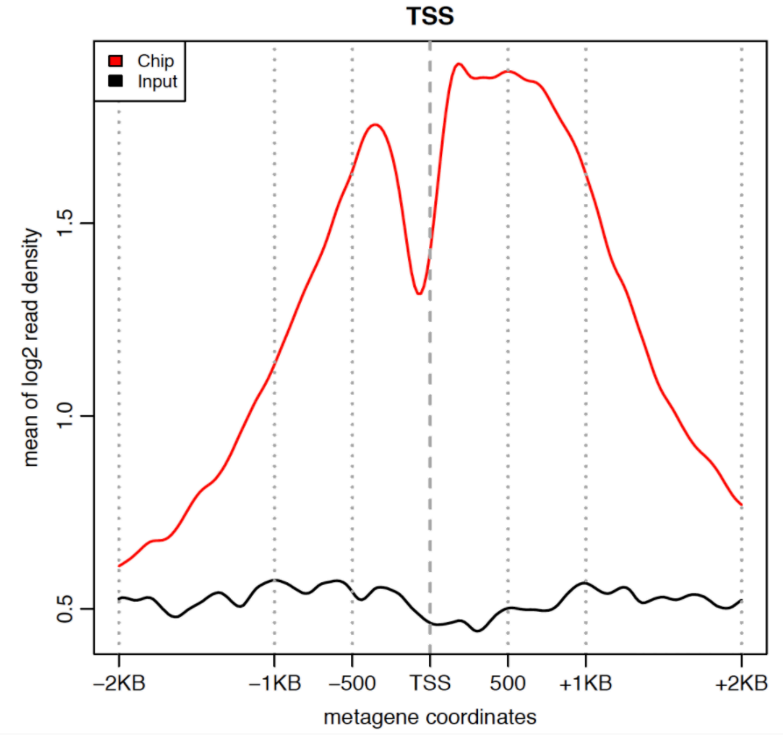


Fig 1. TSS profile plot for H3K36me3 ChIP-seq

Q2.5. What is shown in the TSS plot? Please provide a detailed interpretation of the plot. Based on the TSS profile plot, describe where are H3K36me3 modifications typically located relative to protein coding genes.

1*point*

Answer: TSS plot shows the library-size normalized read depth with transcription start sites (TSS) at the center (accessible chromatin region). Regions further upstream and downstream of TSS should be nucleosome bound. based on TSS profile plot, H3K36me3 is enriched on the gene body region and associated with active gene transcription.