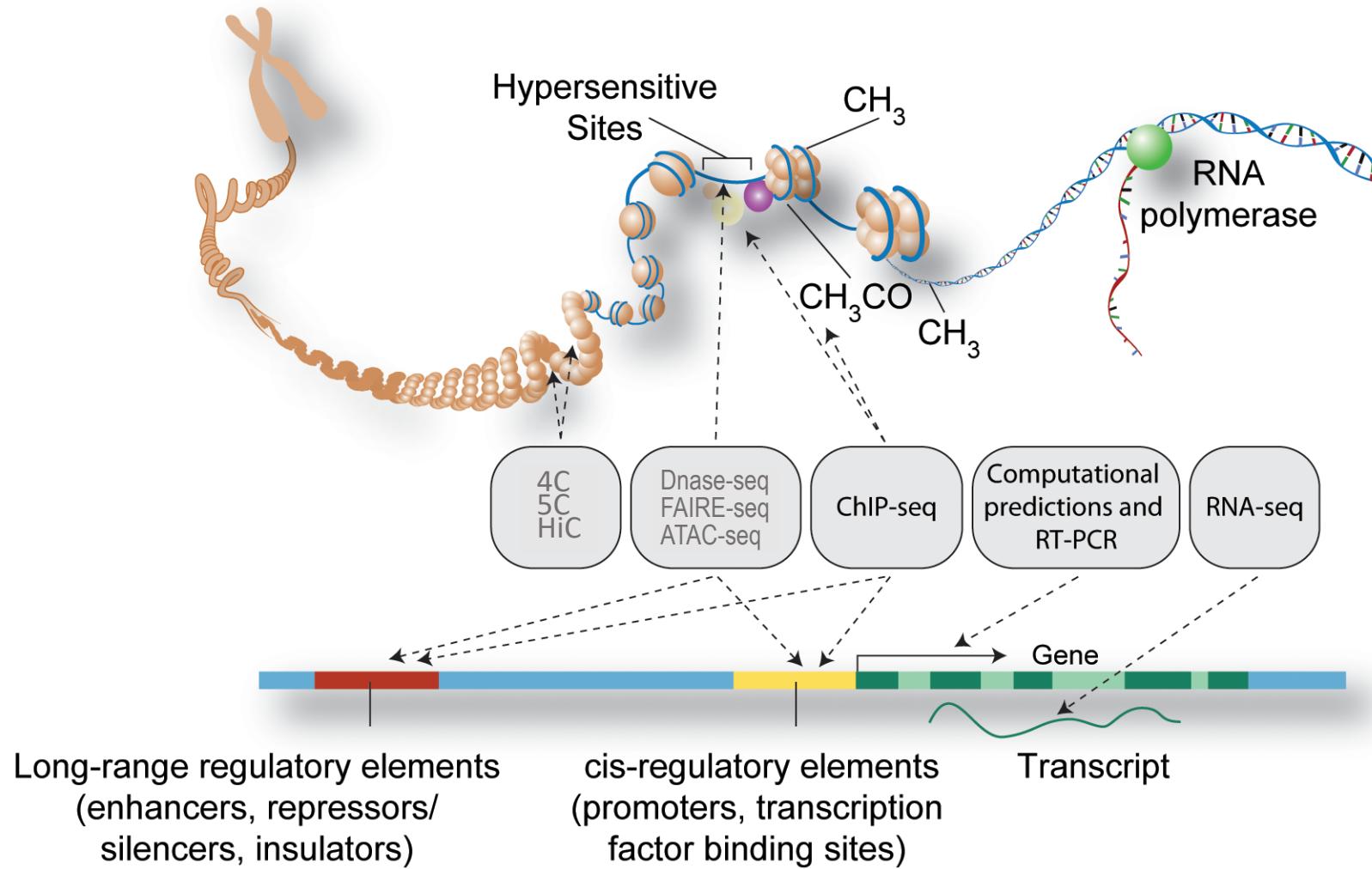
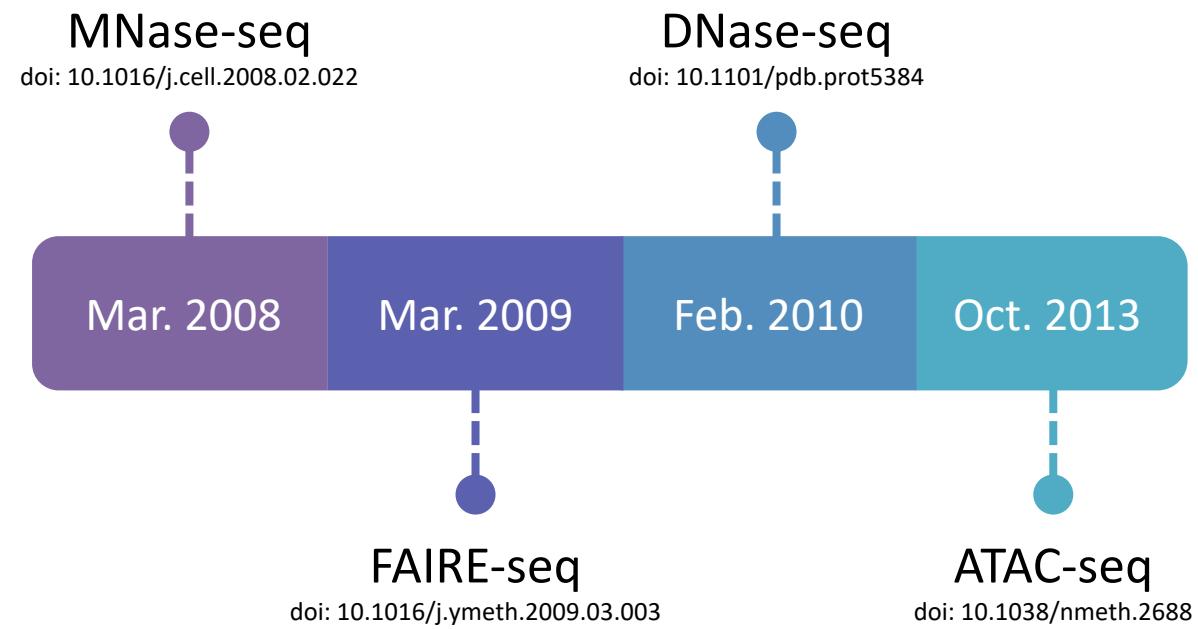


「ATACseqQC」

Jianhong Ou
Haibo Liu
Julie Zhu



PUBLISH TIME



THE ATAC-SEQ DATASETS SUBMITTED TO GEO

FAIRE-seq 116

DNase-Seq 4076

MNase-Seq 6647

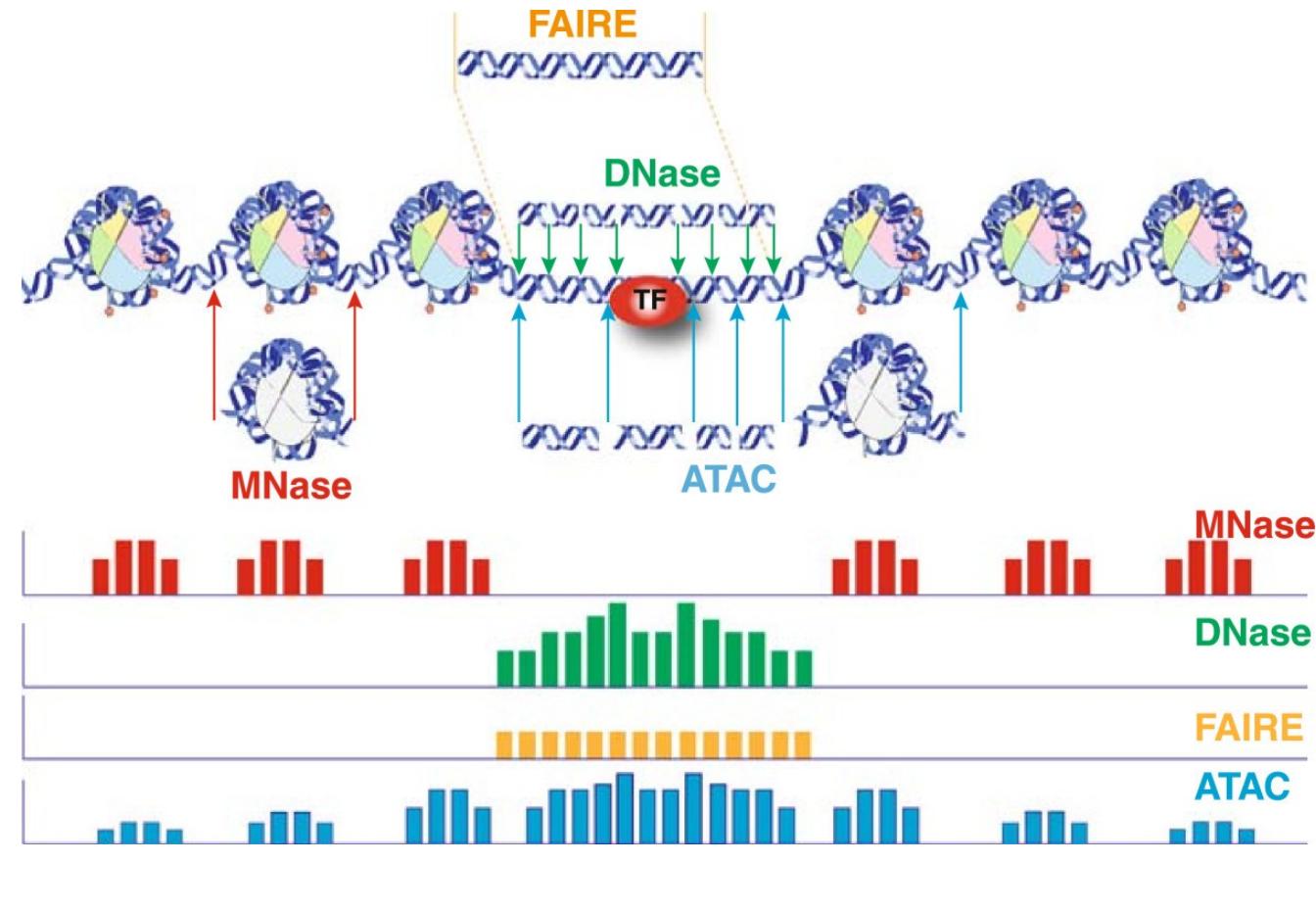
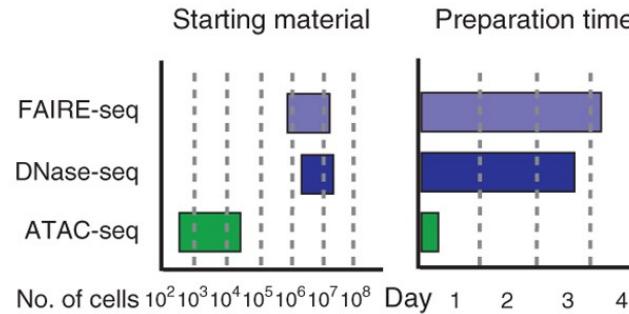
ATAC-seq 21236

```
library(SRAdb)
sqlfile <- getSRAdbFile()
sra_con <- dbConnect(SQLite(),sqlfile)
(sra_tables <- dbListTables(sra_con))
dbListFields(sra_con,"sra")
library_strategy <-
  dbGetQuery(sra_con,
  'SELECT library_strategy,
  COUNT(*) AS count
  FROM sra
  GROUP BY library_strategy')
subs <- library_strategy[
  library_strategy$library_strategy %in%
  c("ATAC-seq", "FAIRE-seq",
  "MNase-Seq", "DNase-Hypersensitivity"), ]
barplot(subs$count)
```

METHODS FOR PROFILING CHROMATIN ACCESSIBILITY

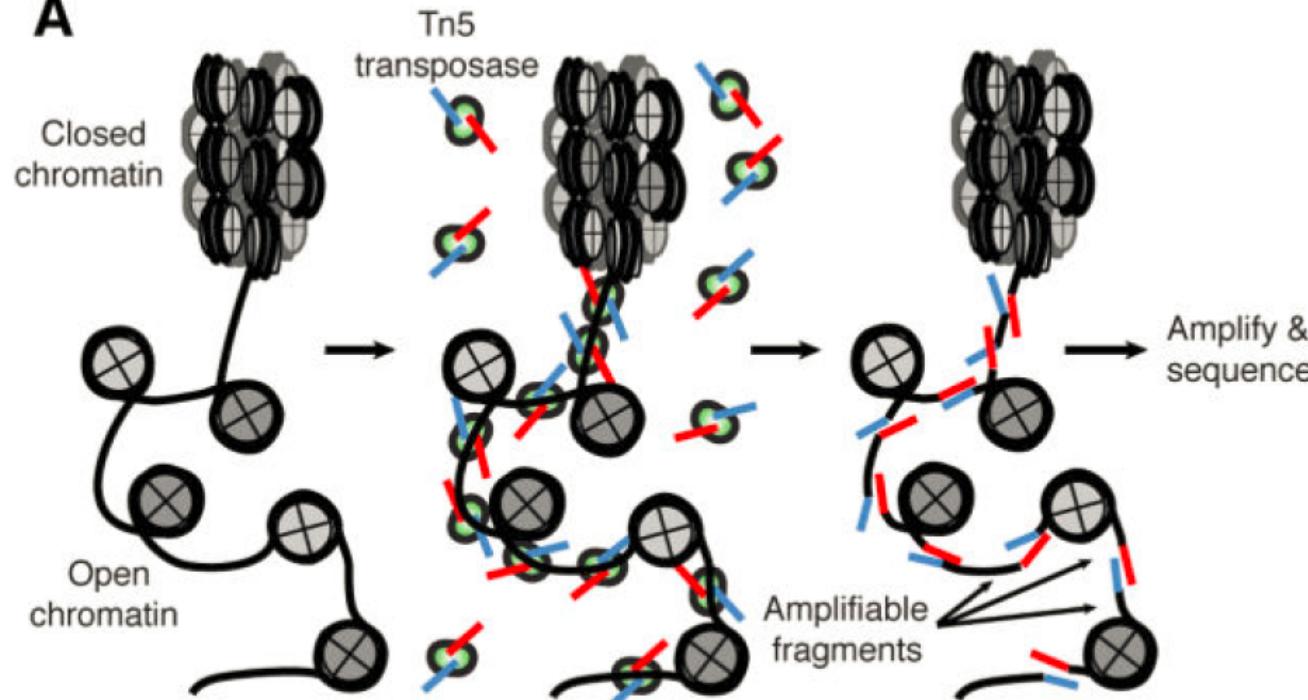
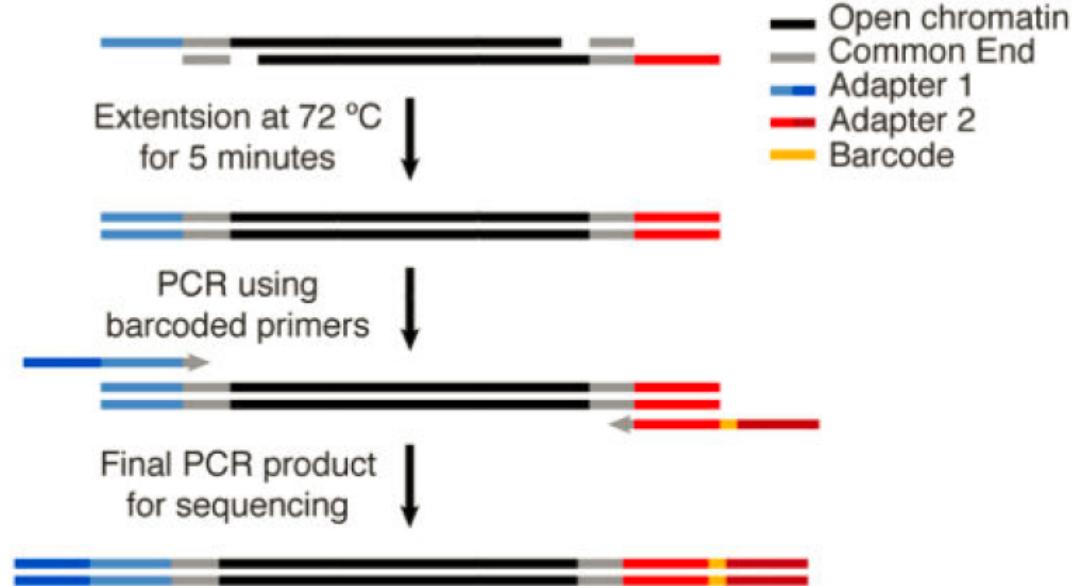
Why ATAC-seq

- ✿ rapid and sensitive



Ref: Buenrostro et.al., 2013. doi: 10.1038/nmeth.2688

Ref: Tsompana and Buck. 2014. doi: 10.1186/1756-8935-7-33

A**B**

Ref: Buenrostro et.al., 2015. doi: 10.1002/0471142727.mb2129s109

ATAC-SEQ IS A SIMPLE TWO-STEP PROTOCOL

1. Insertion of Tn5 transposase with adaptors
2. PCR amplification

QUESTION TO BE



ANSWERED IN QC

Q: Does the sequencing succeed?

S: fastQC

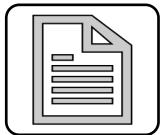
K: sequencing score (or mapping score)
library complexity (duplicates rate)

Q: Does the library provide information of open chromatin, TF and nucleosome occupancy?

S: ATACseqQC

K: periodicity fragment size distribution
TSSs enrichment for nucleosome-free fragments
detectability of genome-wide TF occupancy

WHY ATACSEQQC



Well documented



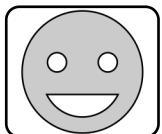
Well maintained



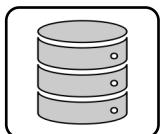
cross platform



Bioconductor support site



Easy to use



Provide files for downstream analysis

Access & Citations

10k

Article Accesses

16

[Web of Science](#)

17

[CrossRef](#)

Online attention



41 tweeters

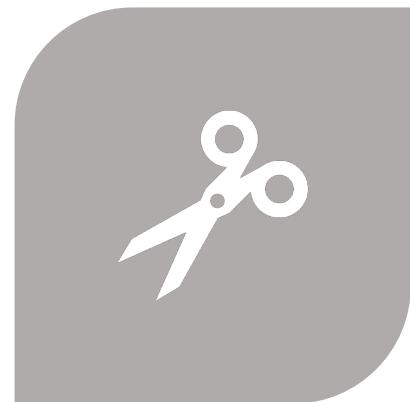
131 Mendeley

1 Google+ users

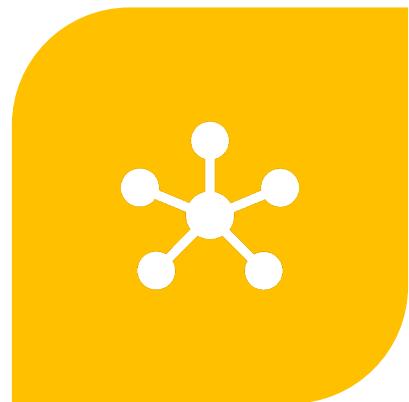
This article is in the 88th percentile (ranked 31,409th) of the 275,422 tracked articles of a similar age in all journals and the 91st percentile (ranked 1st) of the 12 tracked articles of a similar age in *BMC Genomics*

PREPARE SAMPLE DATA SET

Sample data is downloaded from <https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM1155959>



TRIM



ALIGN



FILTER

PREPARE FOR INPUTS

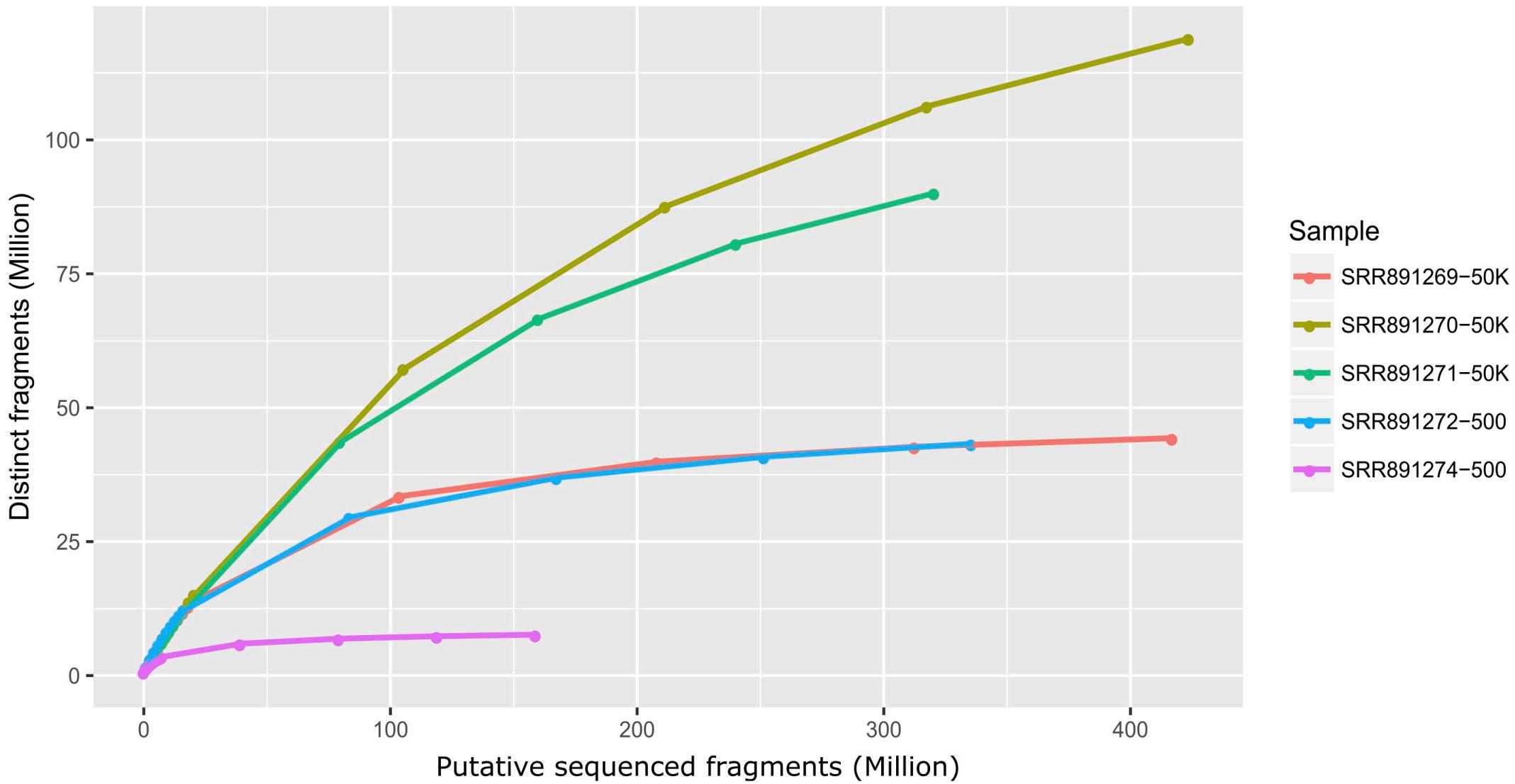
```
## set the working directory,  
## replace "~/Downloads/ATACseqQCworkshop" by your path  
wd <- "~/Downloads/ATACseqQCworkshop"  
dir.create(wd)  
setwd(wd)  
library(BiocManager)  
install("jianhong/workshop2020")  
## input the sample bamFile from the installed package  
## the data is pre-processed chr1 data of GSM1155959  
extfilePath <- system.file("extdata", "ATACseqQC",  
                           package="workshop2020", mustWork = TRUE)  
dir(extfilePath)## see what we have in extdata/ATACseqQC directory
```

```
## [1] "GL3.chr1.bam"          "GL3.chr1.bam.bai"  
## [3] "GL3.chr1.rmdup.bam"   "GL3.chr1.rmdup.bam.bai"
```

<https://github.com/jianhong/workshop2020>
<https://bioconductor.org/packages/ATACseqQC>

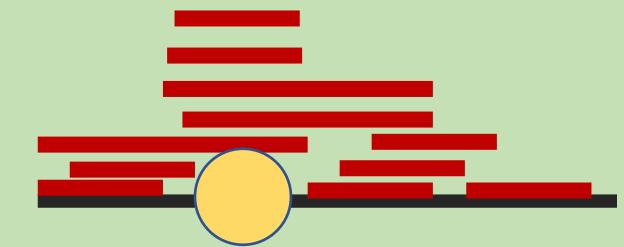
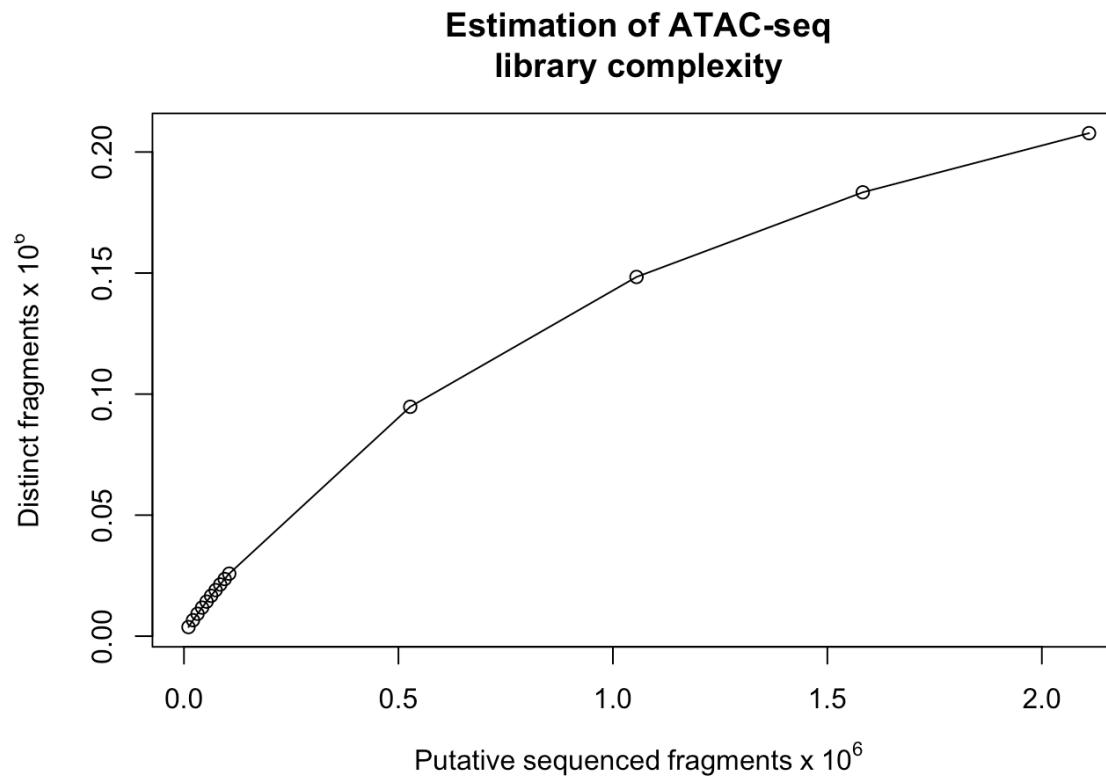


LIBRARY COMPLEXITY

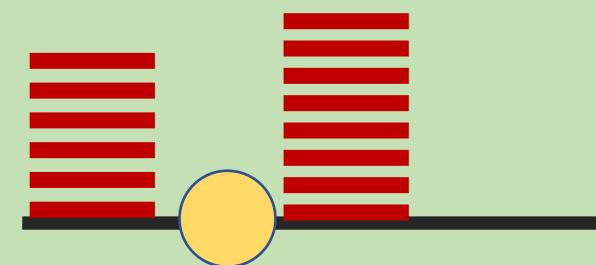


ESTIMATE THE LIBRARY COMPLEXITY

```
## load the library
library(ATACseqQC)
## mapping status, such as mapping rate,
## duplicate rate, genome-wide distribution,
## mapping quality, and contamination.
## NOTE: requires sorted BAM files with duplicate reads marked as input.
## library complexity
estimateLibComplexity(readsDupFreq(file.path(extfilePath, "GL3.chr1.bam")))
```



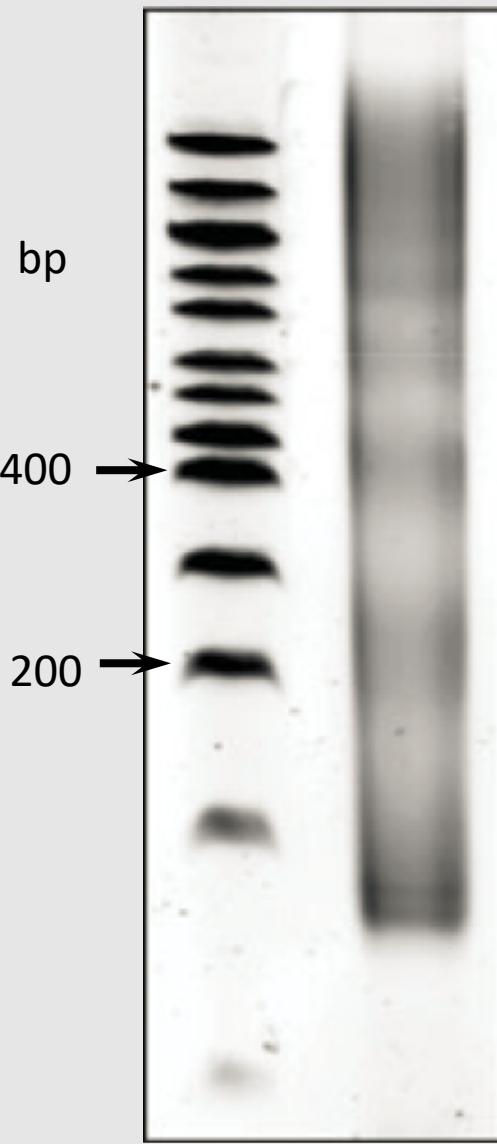
Typical ATAC-seq peak



Low-complexity ATAC-seq peak

Library fragment size

Agarose Gel

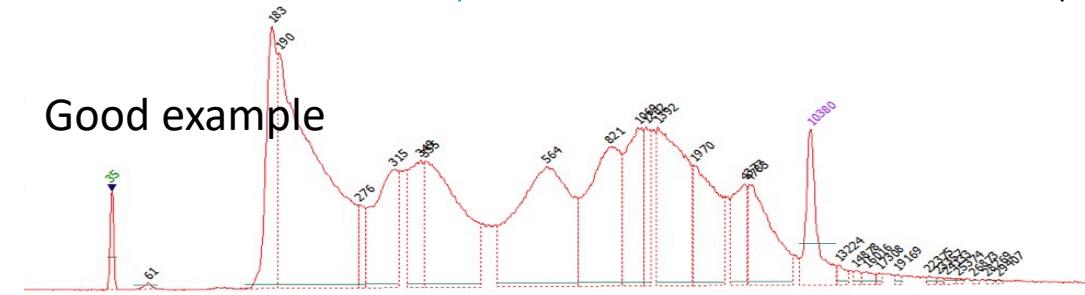


Bioanalyzer



ATAC-SEQ LIBRARY FRAGMENTATION PROFILE

Good example



Good

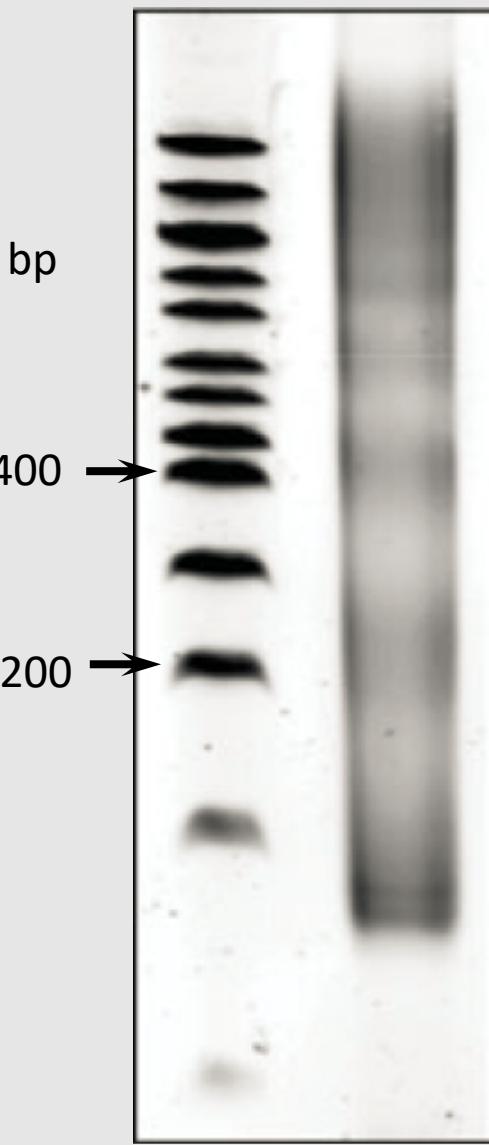
Need to be improved

bad

Adapter/primer
Dimer (<200 nt)
or nucleosome-free fragment
(dimer+80)

Library fragment size

Agarose Gel



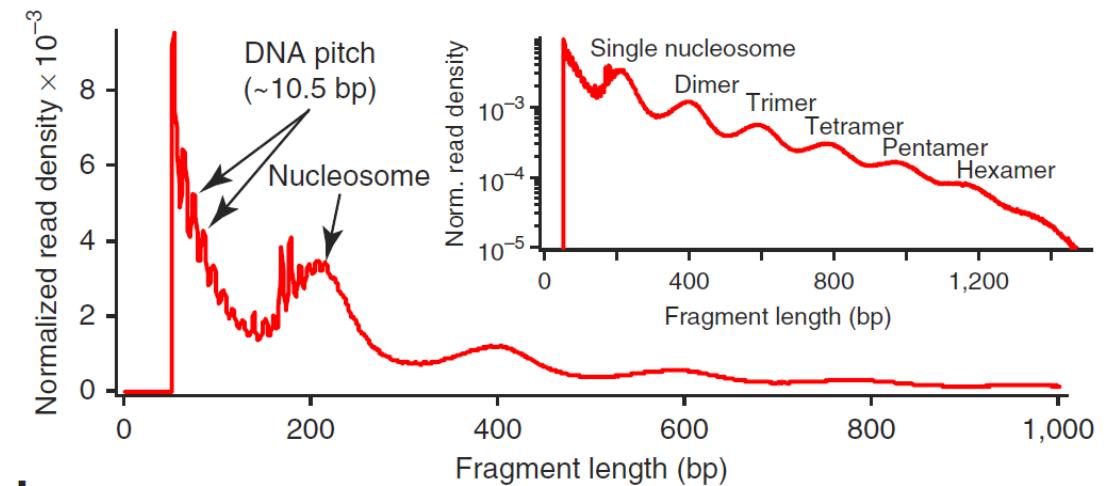
Bioanalyzer



Ref: Buenrostro et.al., 2015. doi: 10.1002/0471142727.mb2129s109

ATAC-SEQ LIBRARY INSERTION PROFILE

Library insert size (=size in gel – adapter/primer size)



Ref: Buenrostro et.al., 2013. doi: 10.1038/nmeth.2688

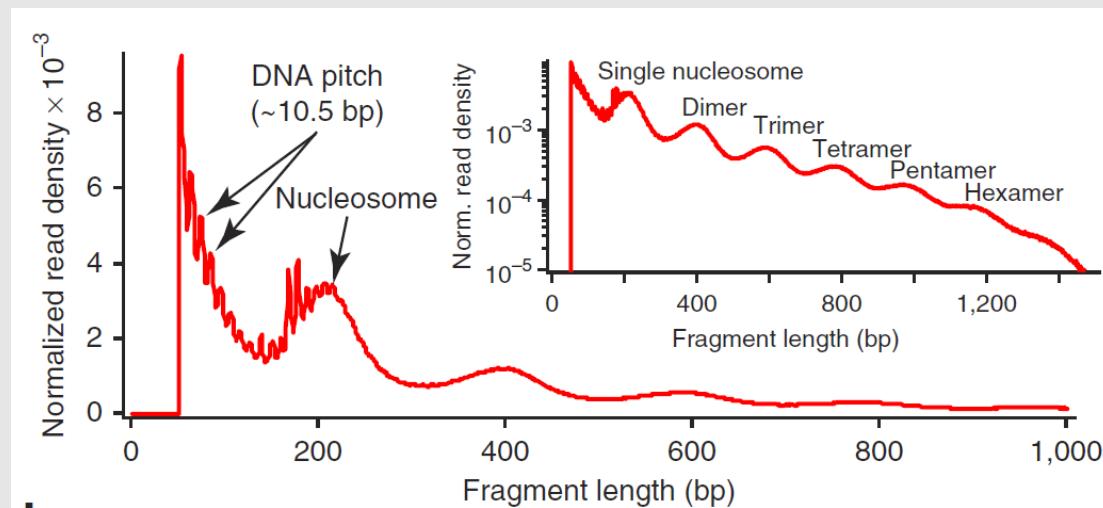
Must be Paired End reads



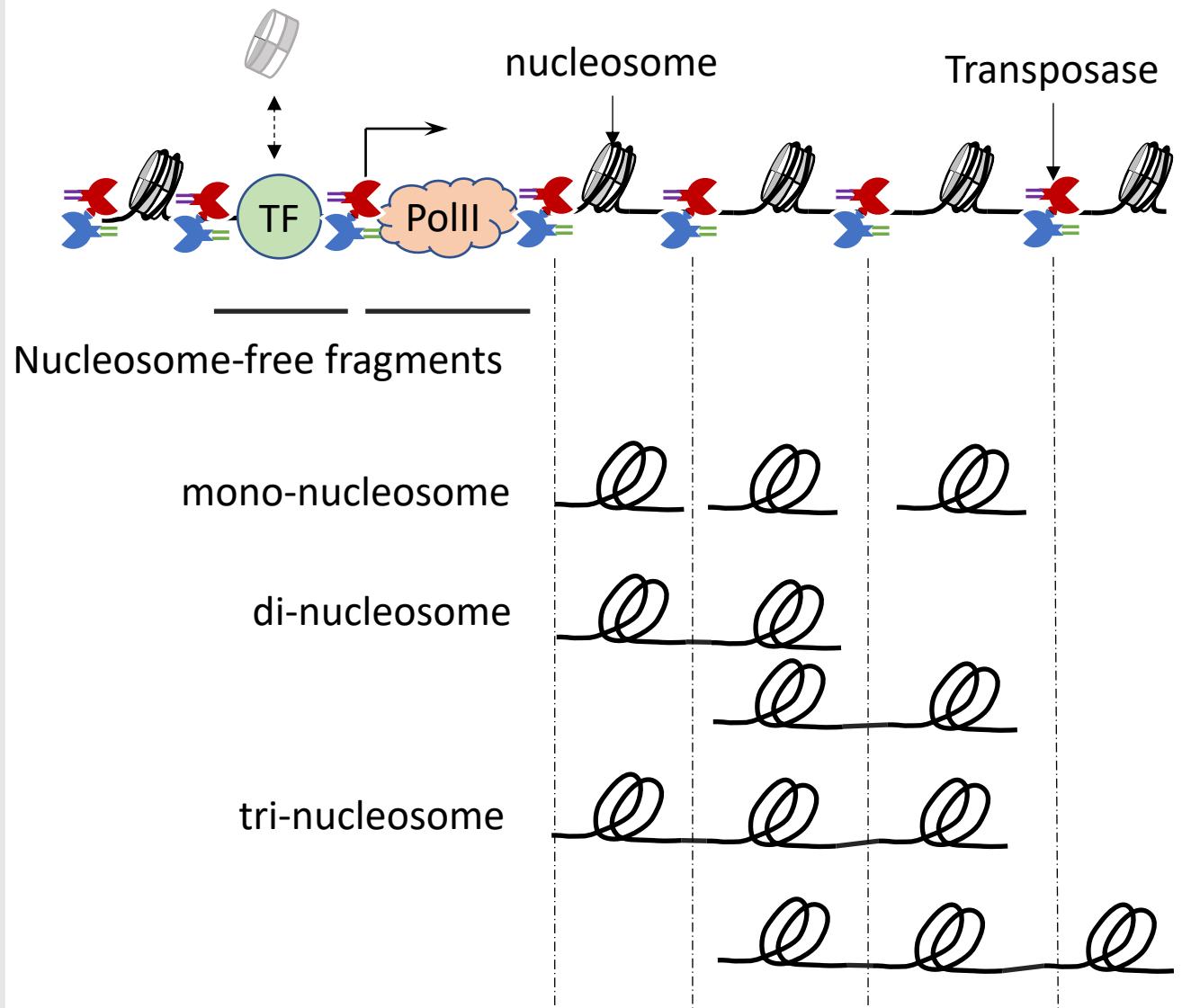
fragSizeDist(bamfile, bamfile.labels)

Function in ATACseqQC

ATAC-SEQ LIBRARY FRAGMENTATION PROFILE

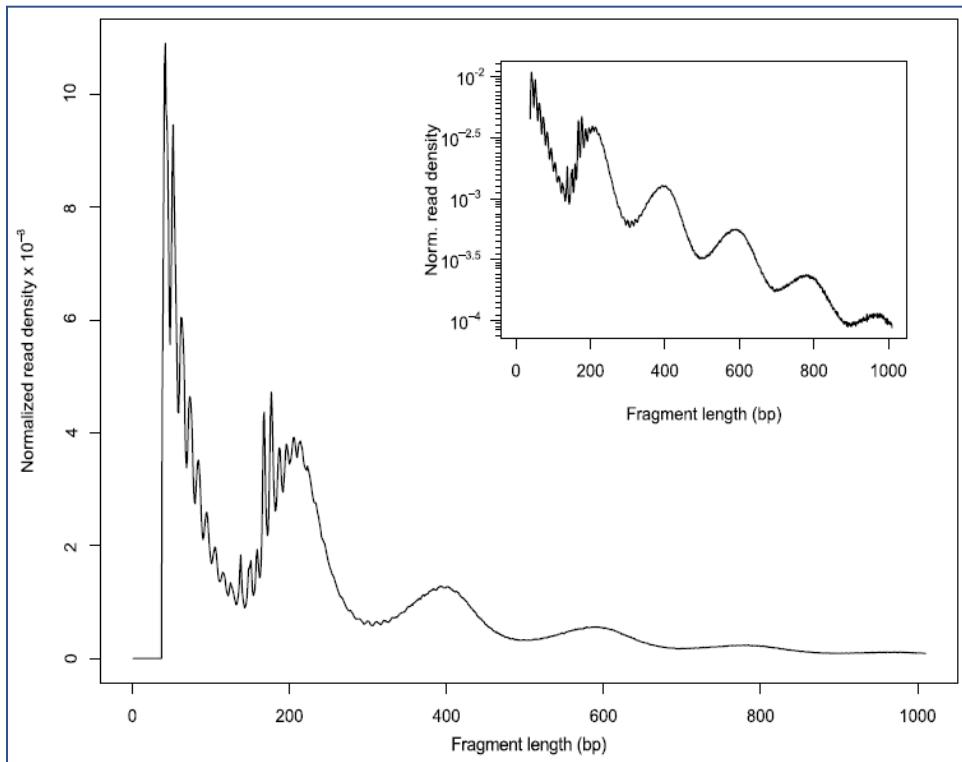


Ref: Buenrostro et.al., 2015. doi: 10.1002/0471142727.mb2129s109

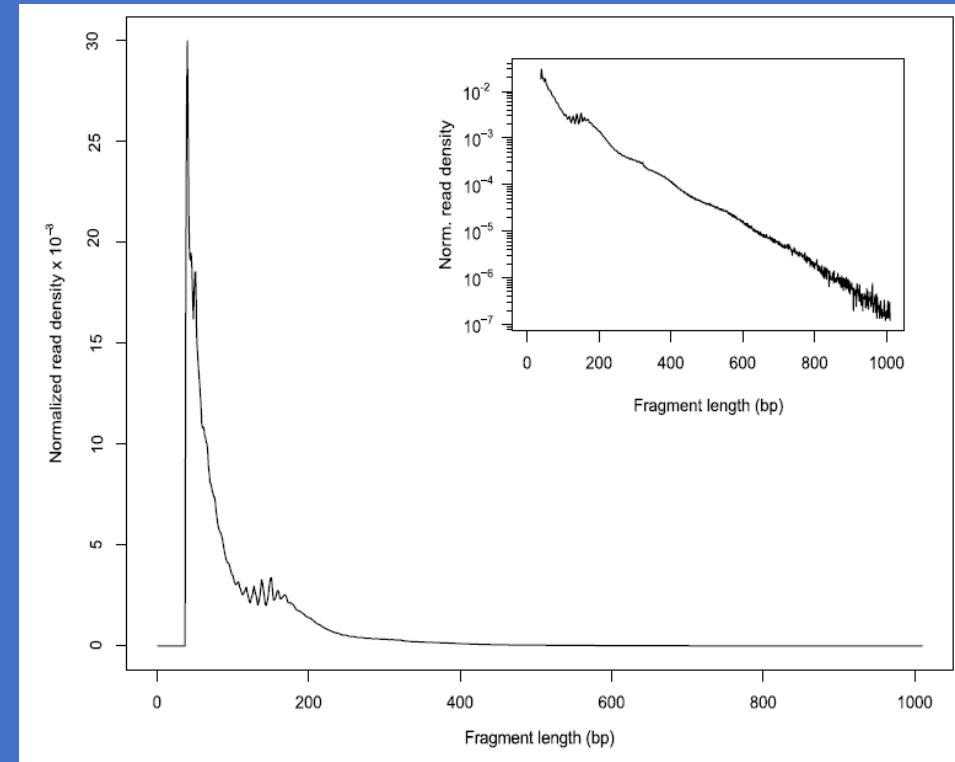


ASSESSING INSERT SIZE DISTRIBUTION

A (High quality)

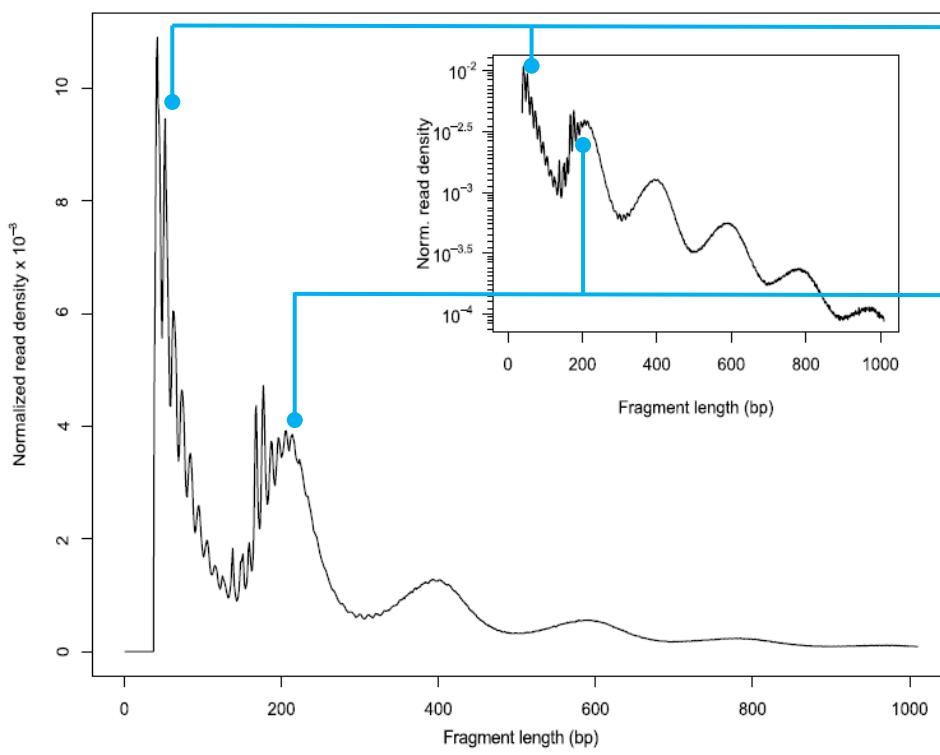


B (Poor quality)



MINIMAL CRITERIA FOR INSERT SIZE DISTRIBUTION

You must see in the distribution of insert size:



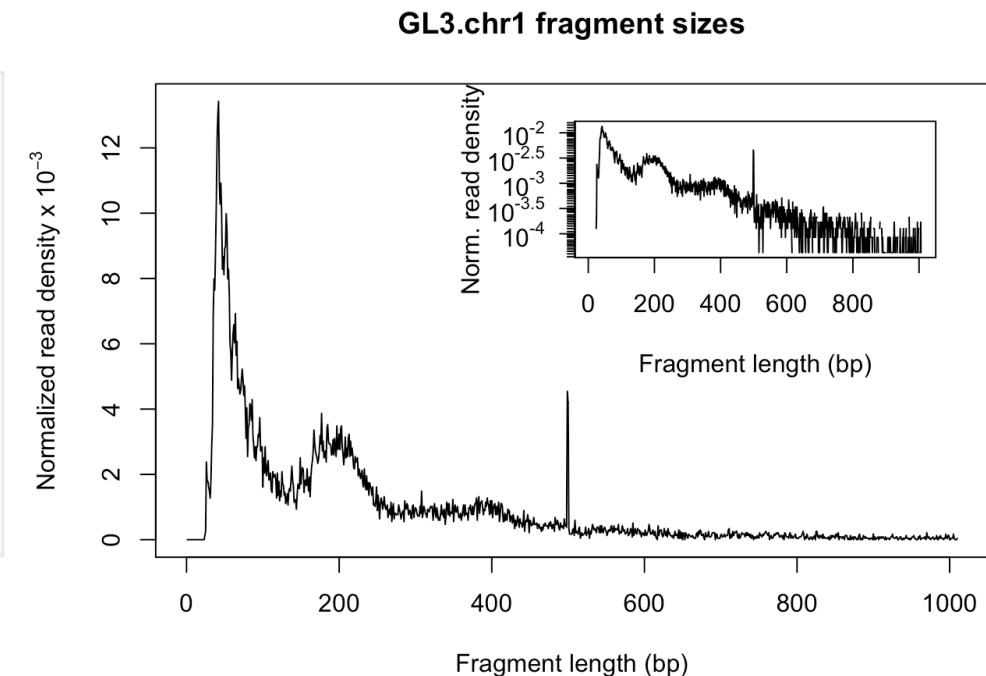
A nucleosome-free region (NFR) peak

A mono-nucleosome region peak (147~294 bp).

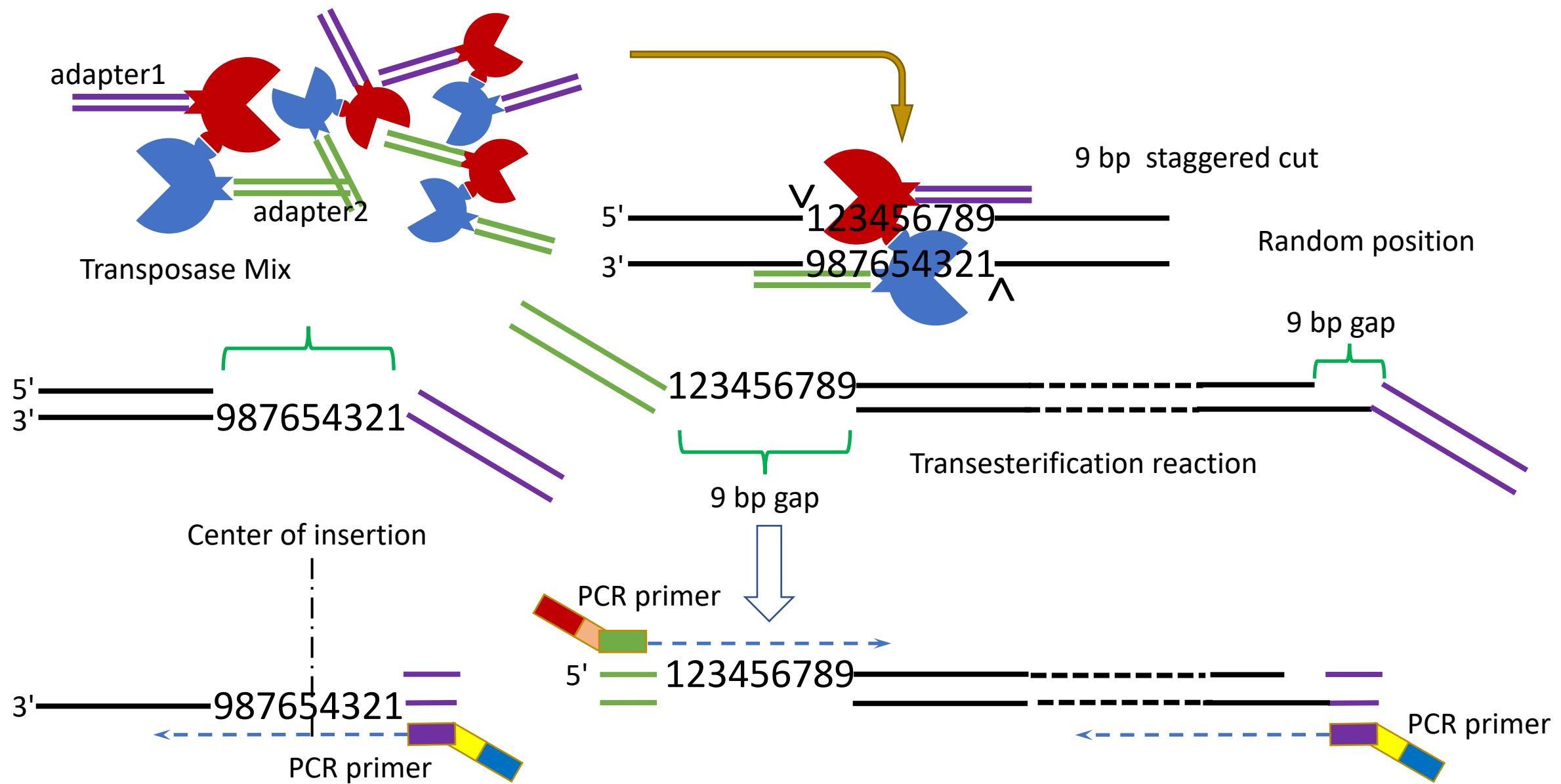
ASSESSING INSERT SIZE DISTRIBUTION

First, there should be a large proportion of reads with less than 100 bp, which represents the nucleosome-free region. Second, the fragment size distribution should have a clear periodicity, which is evident in the inset figure, indicative of nucleosome occupancy (present in integer multiples).

```
## set bam file name, replace file.path(extfilePath,
"GL3.chr1.rmdup.bam")
## by your bam file name
bamfile <- file.path(extfilePath, "GL3.chr1.rmdup.bam" )
(bamfile.labels <- gsub(".rmdup.bam", "", basename(bamfile)))
## [1] "GL3.chr1"
## generate fragment size distribution
fragSize <- fragSizeDist(bamfile, bamfile.labels)
```



SHIFTING ALIGNED READS



SHIFTING ALIGNED READS

```
## bamfile tags to be read in
possibleTag <- combn(LETTERS, 2)
possibleTag <- c(paste0(possibleTag[1, ], possibleTag[2, ]),
                paste0(possibleTag[2, ], possibleTag[1, ]))
library(Rsamtools)
## use top 100 reads to test possible tags
bamTop100 <- scanBam(BamFile(bamfile, yieldSize = 100),
                      param = ScanBamParam(tag=possibleTag))[[1]]$tag
tags <- names(bamTop100)[lengths(bamTop100)==100]
tags
## [1] "AS" "MD" "PG" "XG" "NM" "XM" "XN" "XO" "XS" "YS" "YT"
## files will be output into outPath
outPath <- "splited"
dir.create(outPath)
## shift the coordinates of 5'ends of alignments in the bam file
library(BSgenome.Hsapiens.UCSC.hg38)
seqlev <- "chr1" ## subsample data for quick run
which <- as(seqinfo(Hsapiens)[seqlev], "GRanges")
gal <- readBamFile(bamfile, tag=tags, which=which,
                   asMates=TRUE, bigFile=TRUE)
shiftedBamfile <- file.path(outPath, "shifted.bam")
gal1 <- shiftGAlignmentsList(gal, outbam=shiftedBamfile)
```

SAM file example

```
@HD VN:1.6 SO:coordinate
@SQ SN:ref LN:45
r001 99 ch1 7 30 8M2I4M = 37 39 TTAGATAAAGGCTG *
r002 0 ch1 9 30 3S6M. * 0 0 AAAAGATAAGGATA *
r003 0 ch1 9 30 5S6M * 0 0 GCCTAAGCTAA * SA:Z:ref,29,-,6H5M,17,0;
r004 0 ch1 16 30 6M14M * 0 0 ATAGCTTCAGC *
r003 2064 ch1 29 17 6H5M * 0 0 TAGGC * SA:Z:ref,9,+,5S6M,30,1;
r001 147 ch1 37 30 9M. = 7 -39 CAGCGGCAT * NM:i:1
```

The alignment section: mandatory fields

Col	Field	Type	Brief description
1	QNAME	String	Query template
2	FLAG	Int	bitwise FLAG
3	RNAME	String	Reference sequence NAME
4	POS	Int	1-based leftmost mapping POSition
5	MAPQ	Int	MAPping Quality
6	CIGAR	String	CIGAR string
7	RNEXT	String	Reference name of the mate/next read
8	PNEXT	Int	Position of the mate/next read
9	TLEN	Int	observed Template LENGTH
10	SEQ	String	segment SEQuence
11	QUAL	String	ASCII of Phred-scaled base QUALity+33

The alignment section: optional fields

All optional fields follow the TAG:TYPE:VALUE format where TAG is a two-character string that matches /[A-Za-z][A-Za-z0-9]/. Each TAG can only appear once in one alignment line.

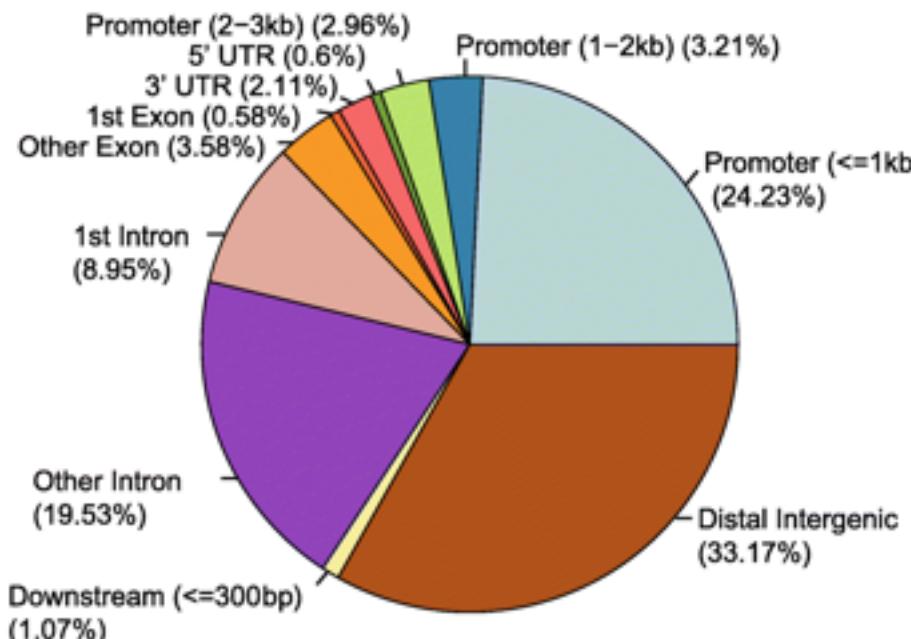
ERRORS WHEN SHIFTING ALIGNED READS

```
> gal1 <- shiftGAlignmentsList(gal, outbam=shiftedBamfile)
[E::sam_parse1] unrecognized type N
[W::sam_read1] Parse error at line 26
```

```
> gal1 <- shiftGAlignmentsList(gal, outbam=shiftedBamfile.outPath)
[bam_translate] PG tag "MarkDuplicates" on read "M01263:122:000000000-CWJNL:1:2106:23406:17450"
encountered with no corresponding entry in header, tag lost. Unknown tags are only reported once per input file
for each tag ID.
[bam_translate] PG tag "MarkDuplicates" on read "M01263:122:000000000-CWJNL:1:2104:12761:8947"
encountered with no corresponding entry in header, tag lost. Unknown tags are only reported once per input file
for each tag ID.
```

```
tags <- tags[tags != "PG"]
```

ENRICHMENT IN PROMOTERS

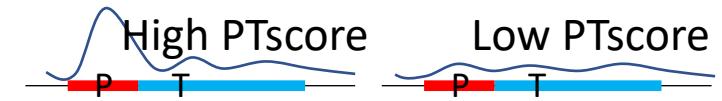


Typical peak annotation pie chart

Ref: Yan et.al., 2020. doi: 10.1186/s13059-020-1929-3

PT score

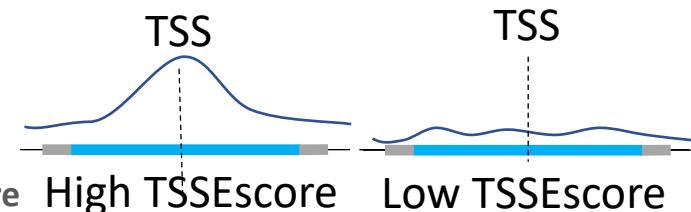
Promoter/Transcript body score



PT score is calculated as the coverage of promoter divided by the coverage of its transcript body. PT score will show if the signal is enriched in promoters.

TSSE score

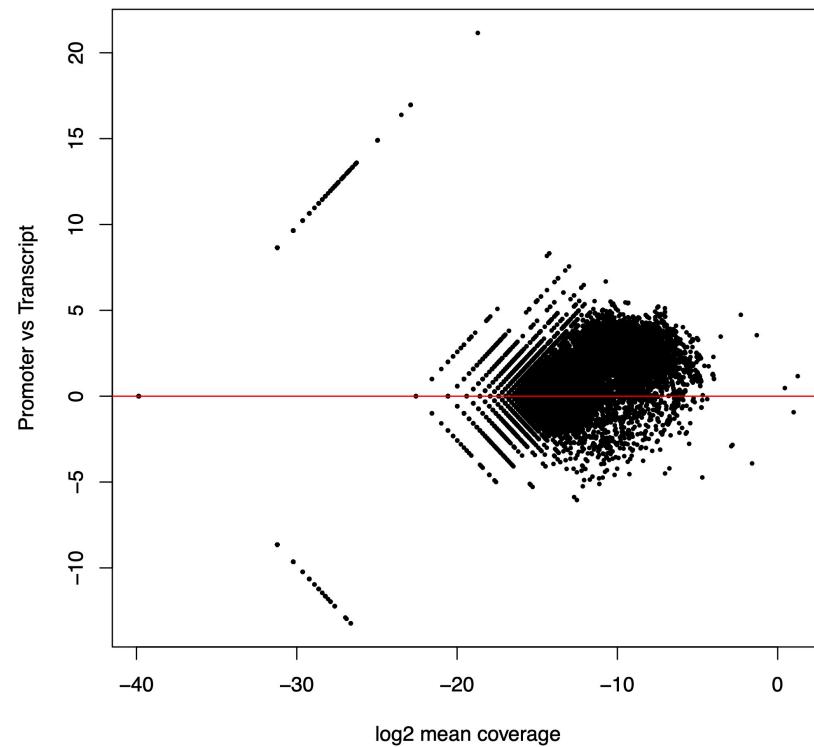
Transcription Start Site (TSS) Enrichment Score



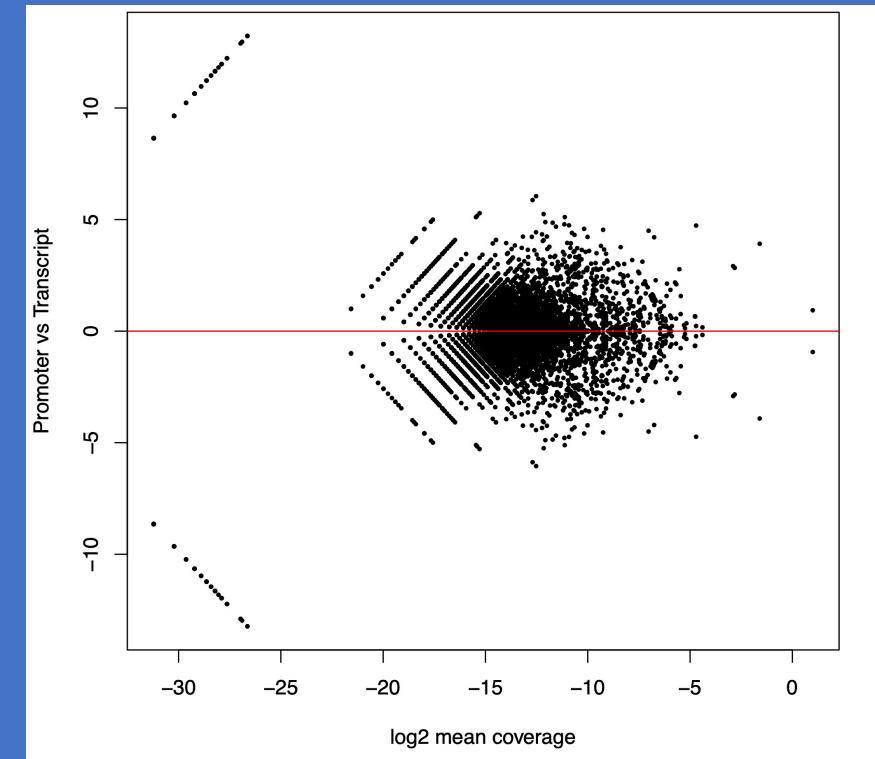
TSS enrichment score is a ratio between aggregate distribution of reads centered on TSSs and that flanking the corresponding TSSs. TSS score = the depth of TSS (1000 bp each side) / the depth of end flanks (100bp each end).

PT SCORE

A (High quality)

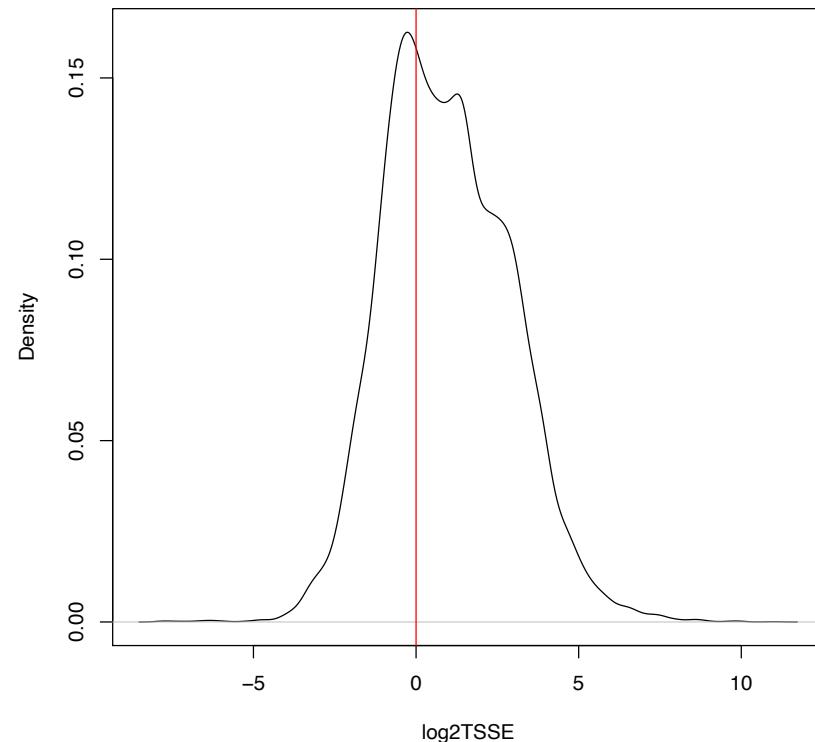


B (Poor quality)

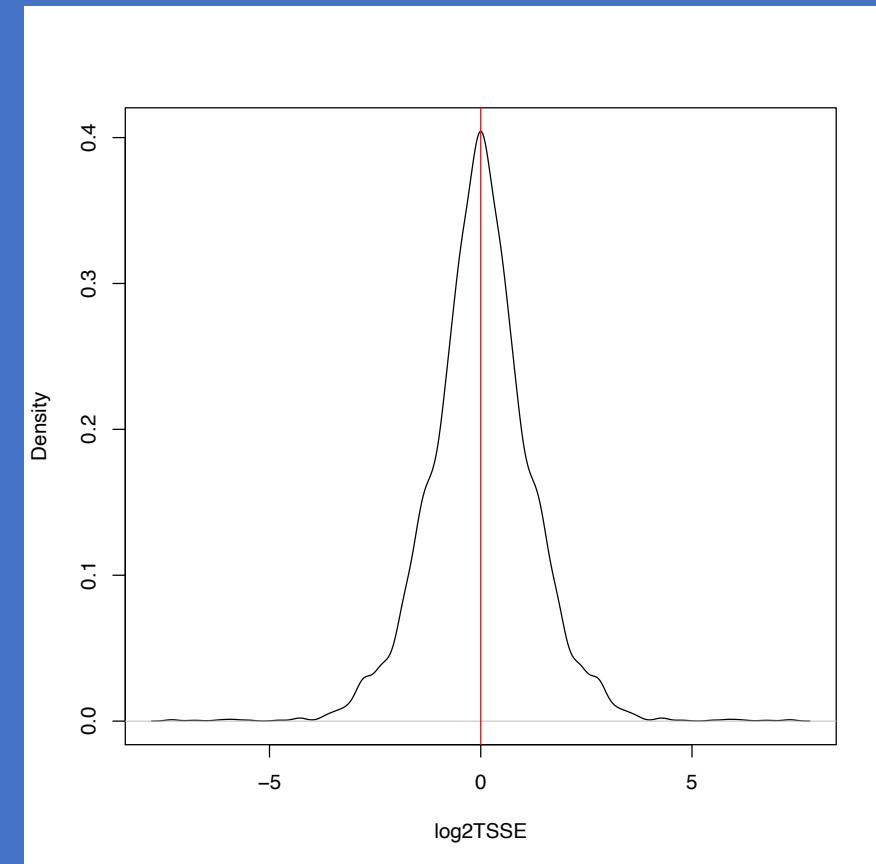


TSSE SCORE

A (High quality)

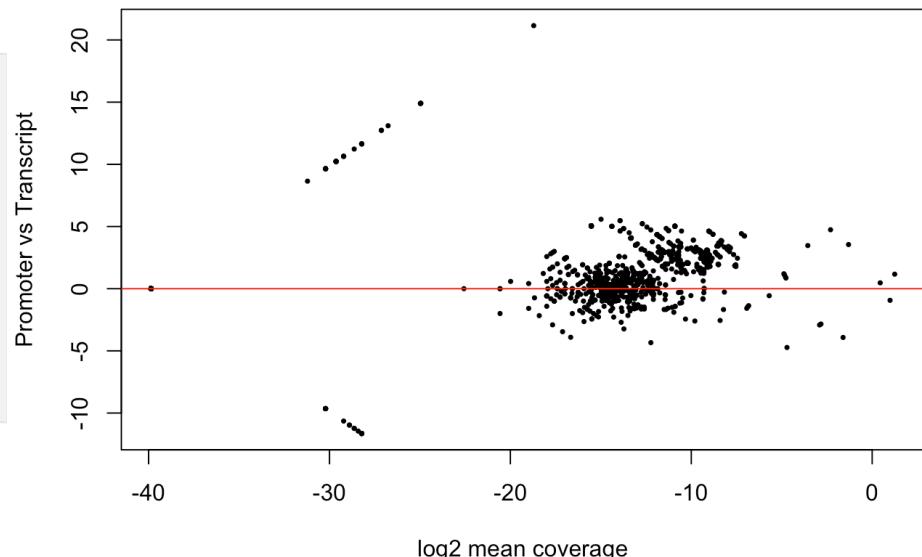


B (Poor quality)

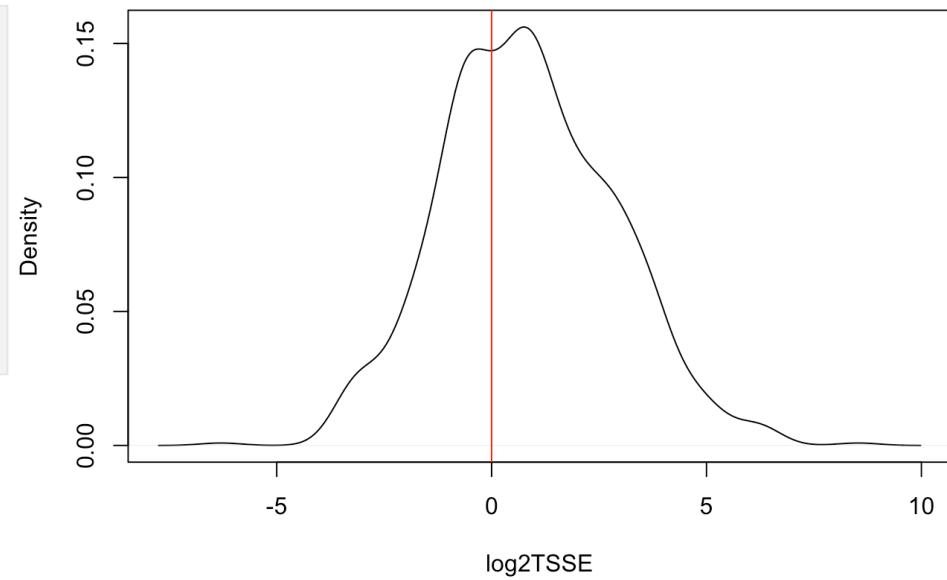


ENRICHMENT IN PROMOTERS

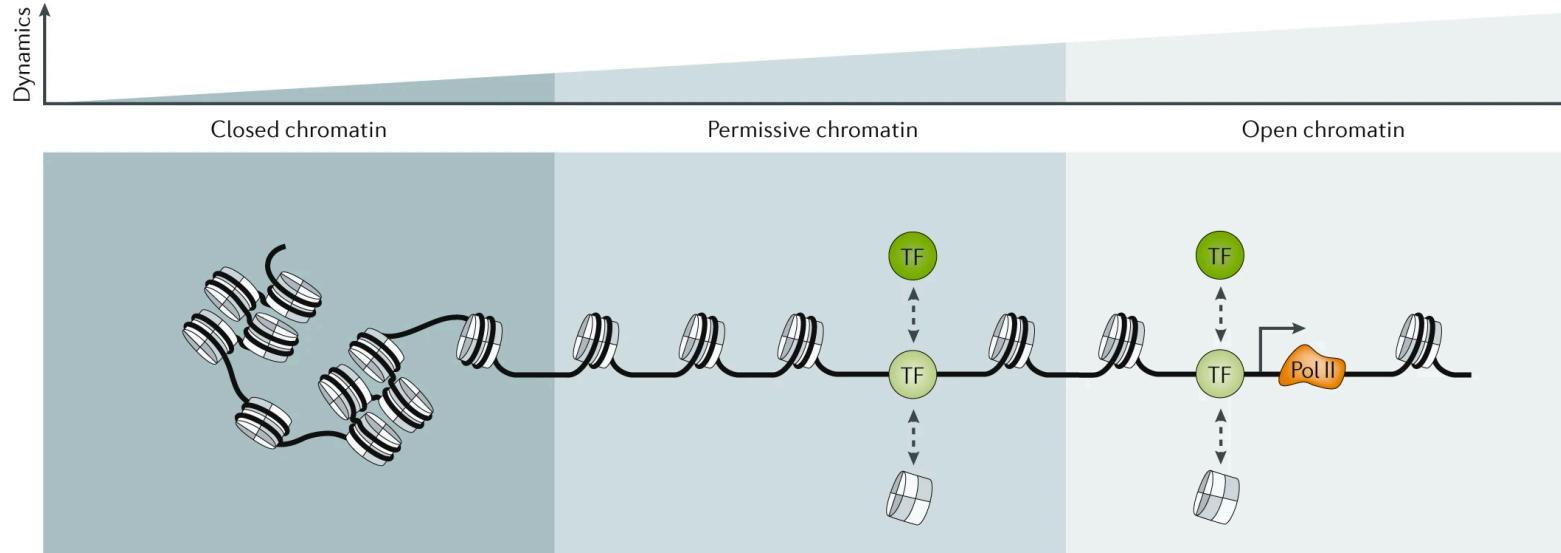
```
library(TxDb.Hsapiens.UCSC.hg38.knownGene)
txs <- transcripts(TxDb.Hsapiens.UCSC.hg38.knownGene)
pt <- PTscore(gal1, txs)
plot(pt$log2meanCoverage, pt$PT_score, pch=16, cex = .5,
     xlab="log2 mean coverage", ylab="Promoter vs Transcript")
abline(h=0, col="red")
```



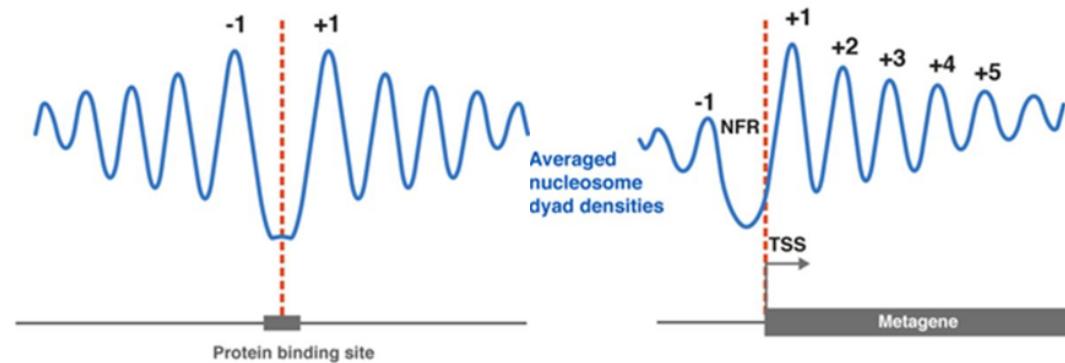
```
tsse <- TSSEscore(gal1, txs)
log2TSSE <- log2(tsse$TSS.enrichment.score)
log2TSSE <- log2TSSE[!is.na(log2TSSE)]
plot(density(log2TSSE), xlab = "log2TSSE", main = "")
abline(v=0, col="red")
```



NUCLEOSOME POSITIONING AROUND TSSs AND TF-BOUND SITES



Ref: Klemm et.al., 2019. doi: 10.1038/s41576-018-0089-8

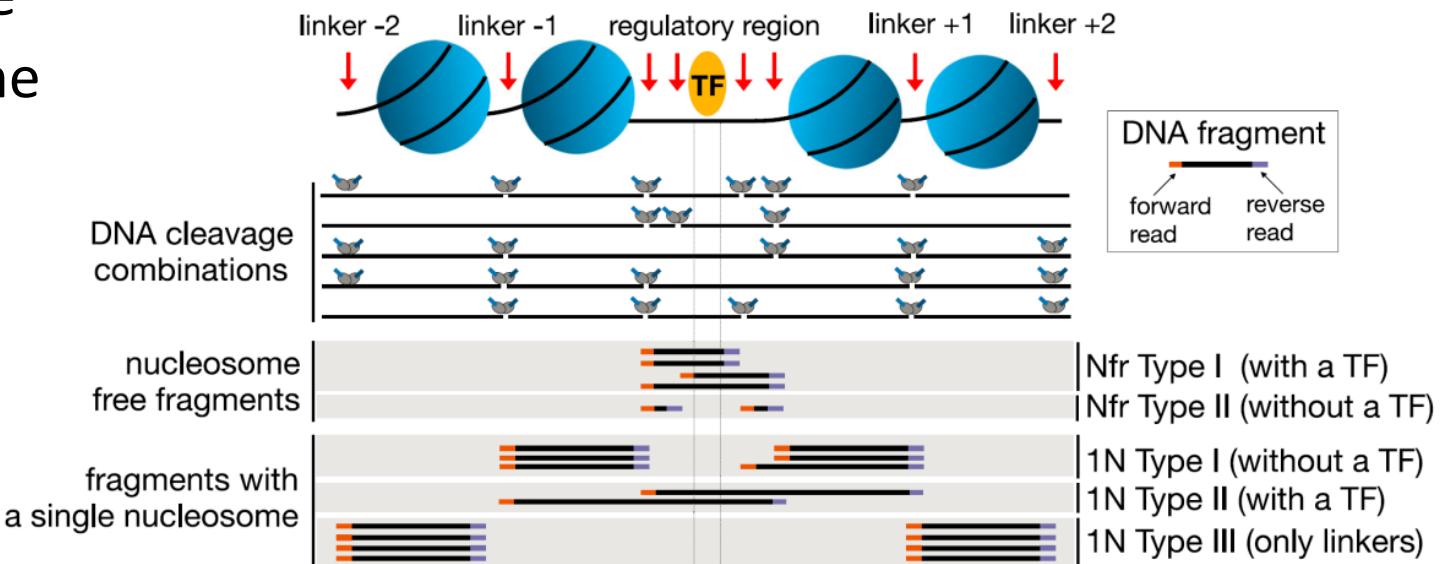


Ref: Baldi. 2018 doi: 10.1042/EBC20180058

SPLITTING BAM FILES

split aligned reads into different bins:

- Nucleosome-free
 - Mononucleosome
 - Dinucleosome
 - Trinucleosome



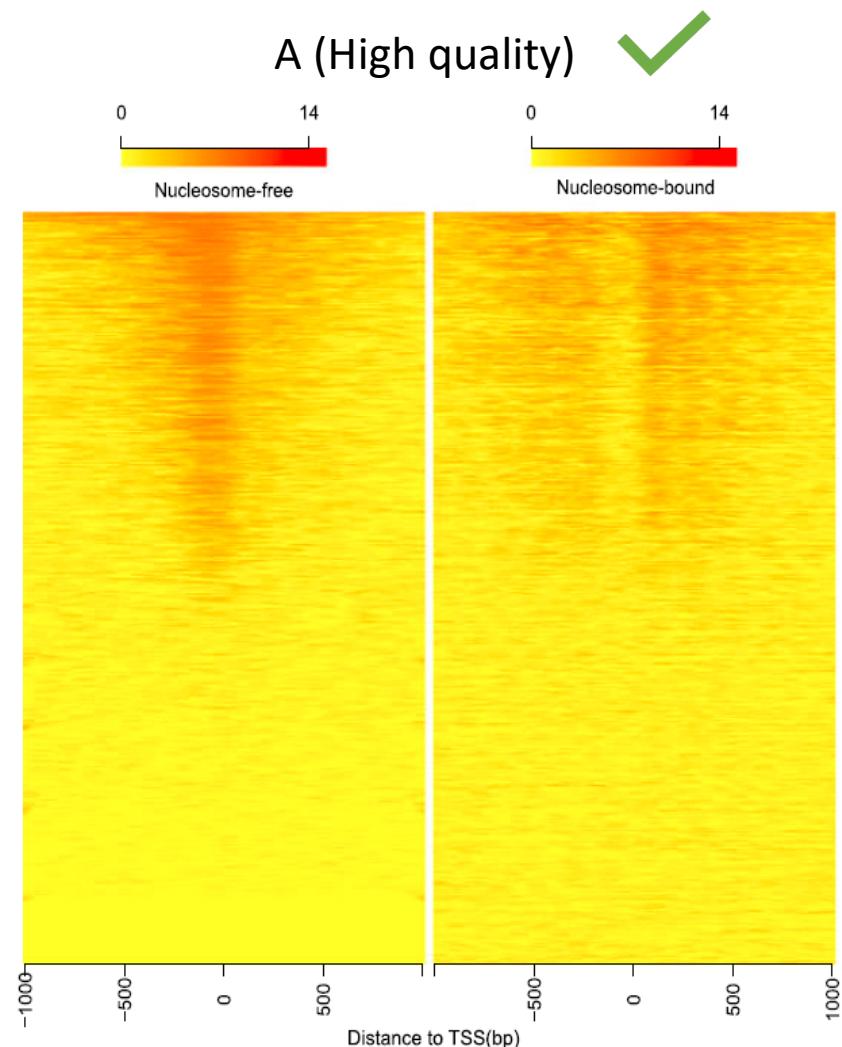
Li et al. 2019. doi: 10.1186/s13059-019-1642-2

SPLIT READS

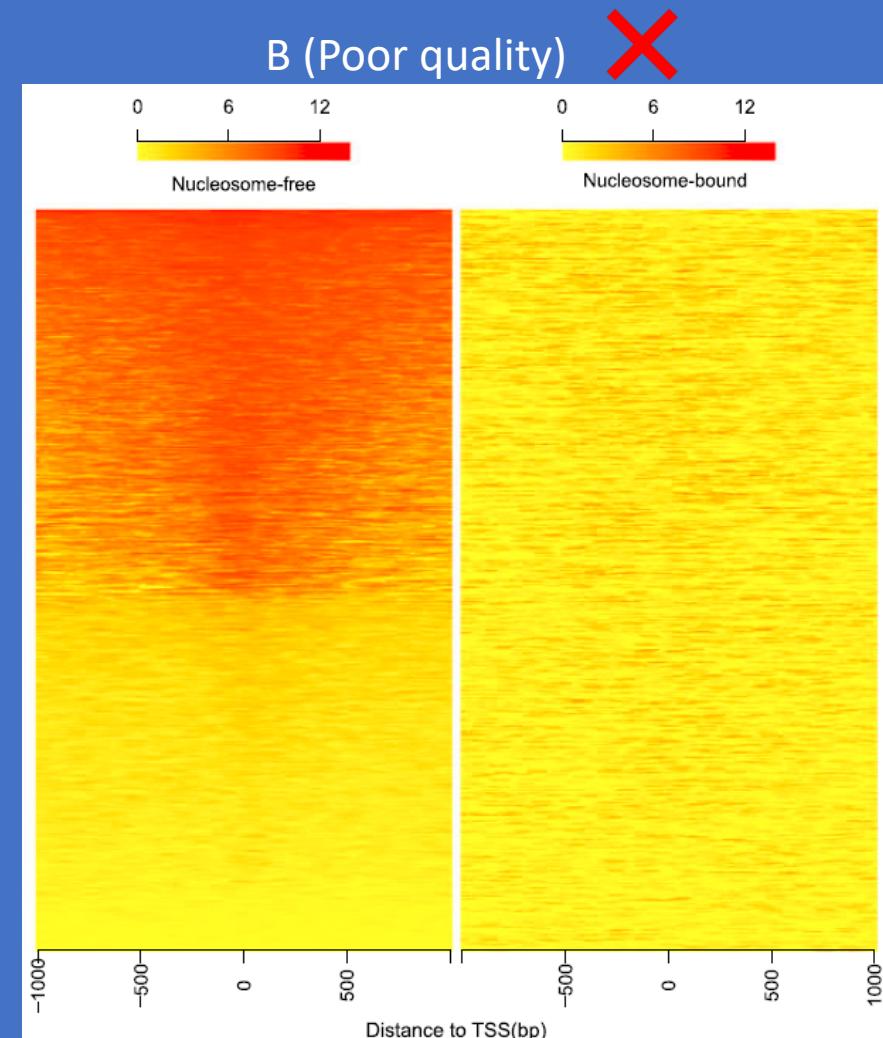
```
## run program for chromosome 1 only
txs <- txs[seqnames(txs) %in% "chr1"]
genome <- Hsapiens
## split the reads into NucleosomeFree, mononucleosome,
## dinucleosome and trinucleosome.
## and save the binned alignments into bam files.
objs <- splitGAlignmentsByCut(gal1, txs=txs, genome=genome, outPath = outPath)
## list the files generated by splitGAlignmentsByCut.
dir(outPath)
```

## [1] "dinucleosome.bam"	"dinucleosome.bam.bai"	"inter1.bam"
## [4] "inter1.bam.bai"	"inter2.bam"	"inter2.bam.bai"
## [7] "inter3.bam"	"inter3.bam.bai"	"mononucleosome.bam"
## [10] "mononucleosome.bam.bai"	"NucleosomeFree.bam"	"NucleosomeFree.bam.bai"
## [13] "others.bam"	"others.bam.bai"	"shifted.bam"
## [16] "shifted.bam.bai"	"trinucleosome.bam"	"trinucleosome.bam.bai"

HEATMAPS SHOWING FROM DIFFERENT BINS

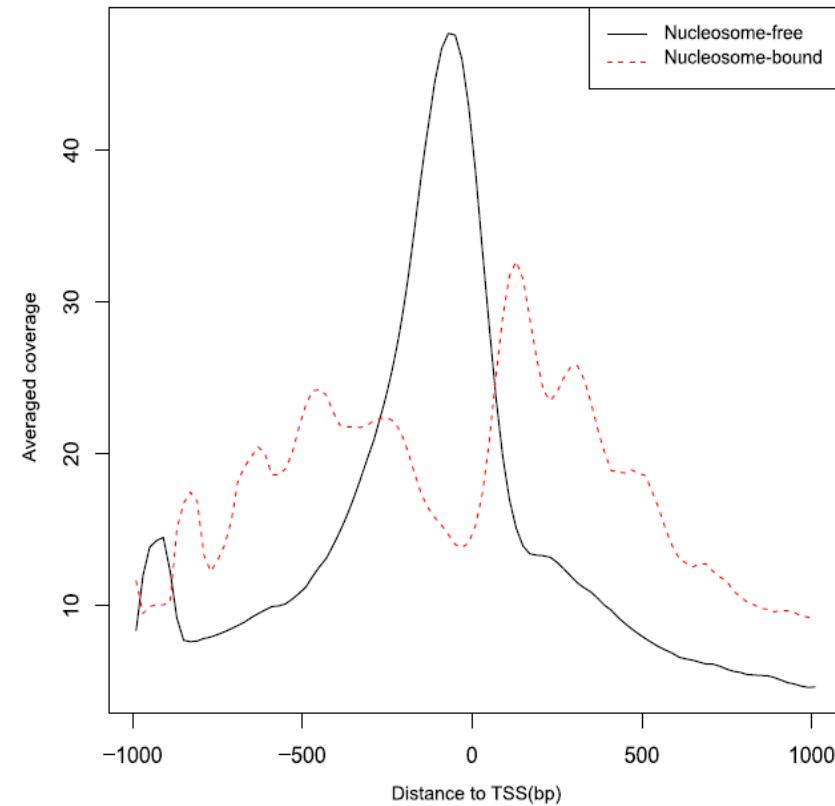


DISTRIBUTION OF READS AROUND TSSs



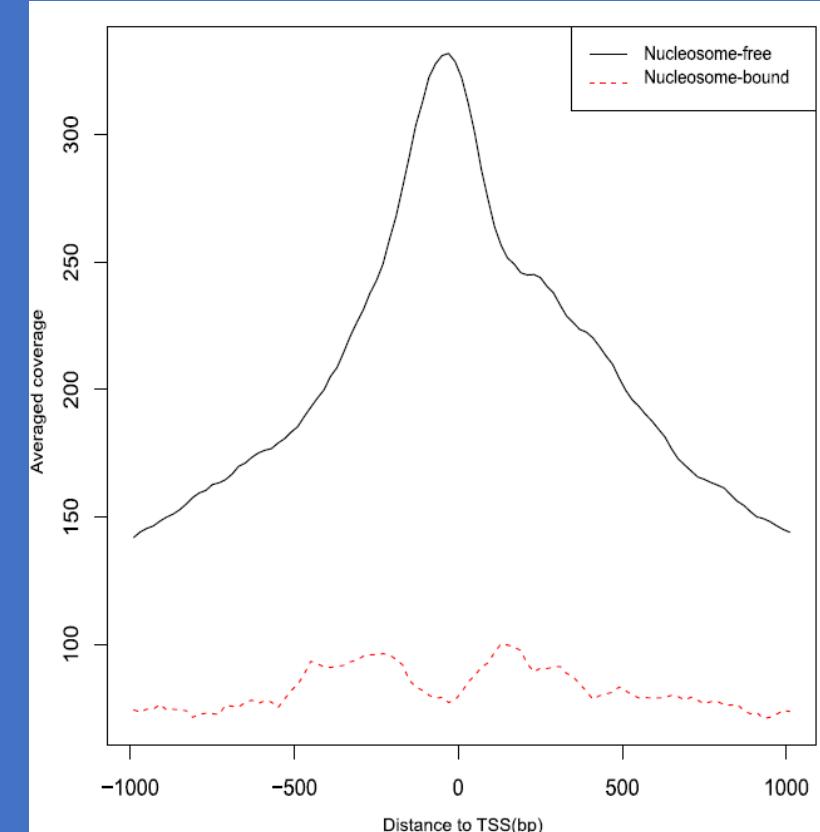
DENSITY PLOTS SHOWING READS FROM DIFFERENT

A (High quality)



DISTRIBUTION OF BINS AROUND TSSs

B (Poor quality)



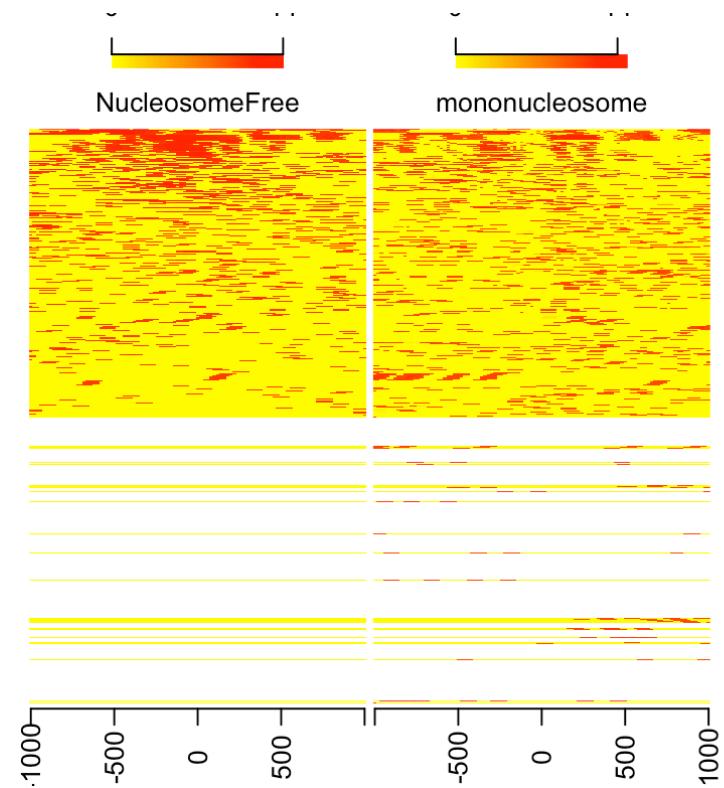
HEATMAP FOR NUCLEOSOME POSITIONS

```
library(ChIPpeakAnno)
bamfiles <- file.path(outPath,
  c("NucleosomeFree.bam", "mononucleosome.bam",
    "dinucleosome.bam", "trinucleosome.bam"))
TSS <- promoters(txs, upstream=0, downstream=1)
TSS <- unique(TSS)
## estimate the library size for normalization
(librarySize <- estLibSize(bamfiles))
```

	splited/NucleosomeFree.bam	splited/mononucleosome.bam
##	10140	3426
##	splited/dinucleosome.bam	splited/trinucleosome.bam
##	2748	0

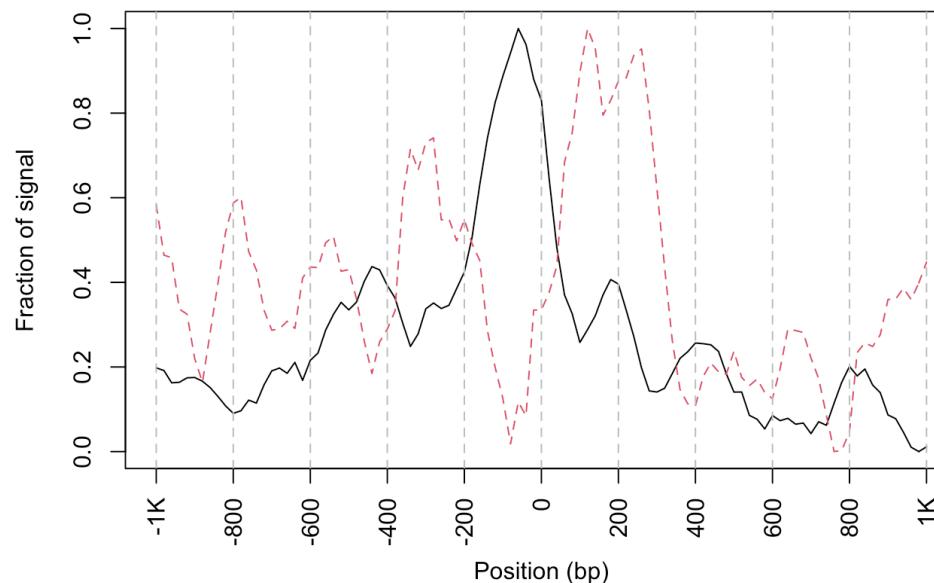
```
## calculate the signals around TSSs.
NTILE <- 101
dws <- ups <- 1010
sigs <- enrichedFragments(gal=objs[c("NucleosomeFree", "mononucleosome",
  "dinucleosome", "trinucleosome")],
  TSS=TSS, librarySize=librarySize, seqlev=seqlev,
  TSS.filter=0.5, n.tile = NTILE,
  upstream = ups, downstream = dws)
```

```
## log2 transformed signals
sigs.log2 <- lapply(sigs, function(.ele) log2(.ele+1))
featureAlignedHeatmap(sigs.log2, reCenterPeaks(TSS, width=ups+dws),
  zeroAt=.5, n.tile=NTILE) #plot heatmap
```



COVERAGE CURVE FOR NUCLEOSOME POSITIONS

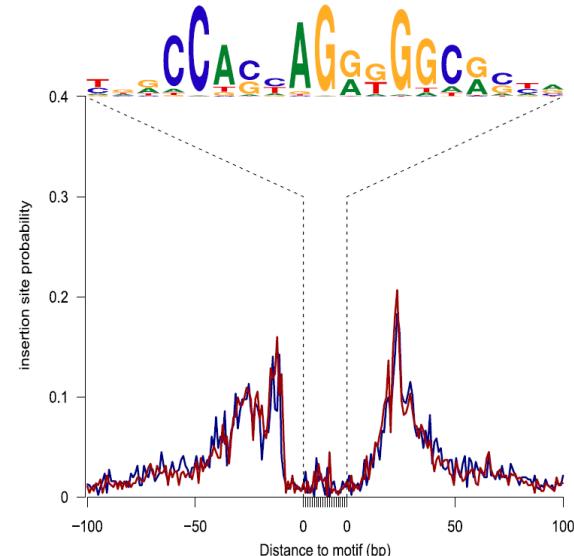
```
## get signals normalized for nucleosome-free and nucleosome-bound regions.  
out <- featureAlignedDistribution(sigs, reCenterPeaks(TSS, width=ups+dws),  
                                   zeroAt=.5, n.tile=NTILE, type="I", ylab="Averaged coverage")  
  
## rescale the nucleosome-free and nucleosome signals to 0~1  
range01 <- function(x){(x-min(x))/(max(x)-min(x))}  
out <- apply(out, 2, range01)  
matplot(out, type="I", xaxt="n", xlab="Position (bp)", ylab="Fraction of signal")  
axis(1, at=seq(0, 100, by=10)+1, labels=c("-1K", seq(-800, 800, by=200), "1K"), las=2)  
abline(v=seq(0, 100, by=10)+1, lty=2, col="gray")
```



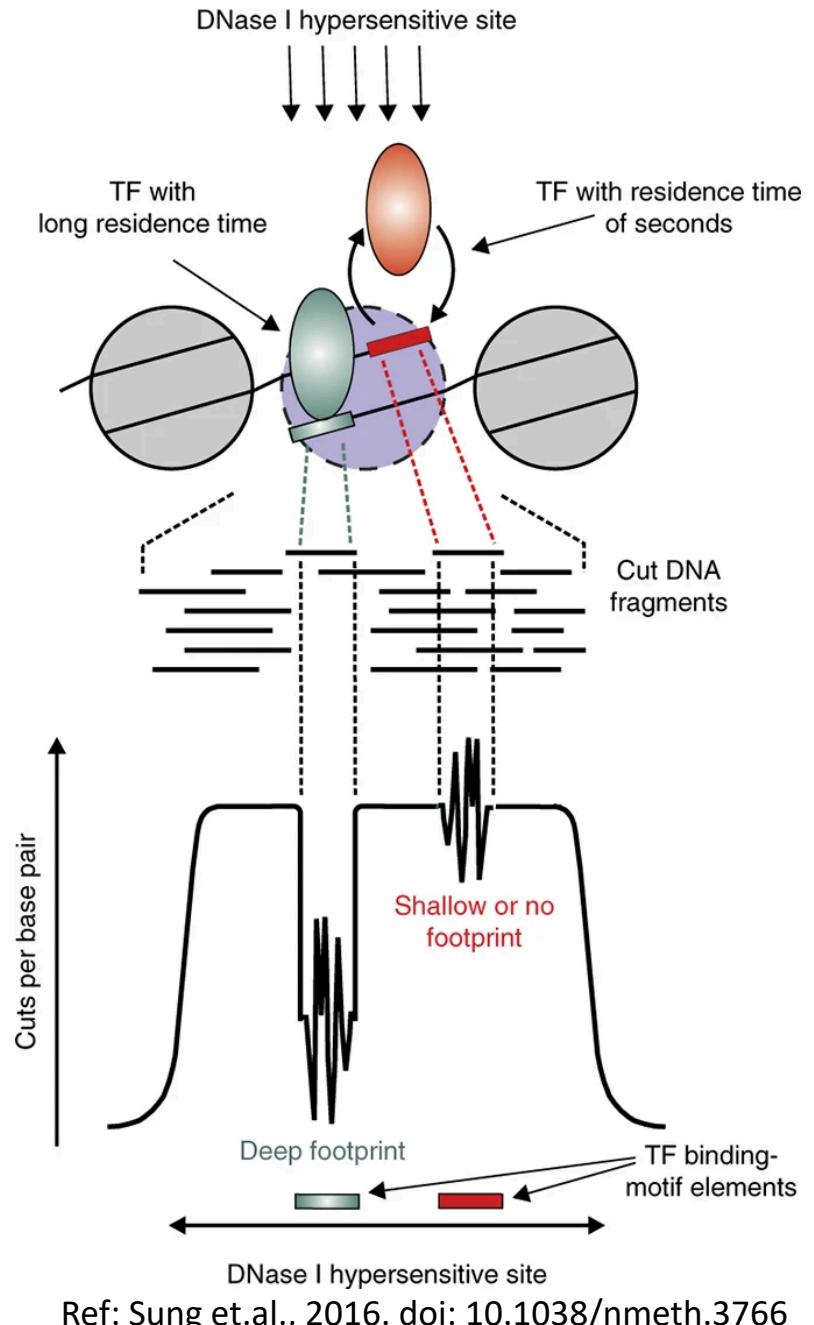
ASSESSING FOOTPRINTS OF DNA-BINDING FACTORS

```
## get PWM of TFs
CTCF <- query(MotifDb, c("CTCF"))
CTCF <- as.list(CTCF)

sigs <- factorFootprints(shiftedBamFile, pfm=CTCF[[1]],
  genome=genome, min.score="90%", seqlev=seqlev,
  upstream=100, downstream=100)
```



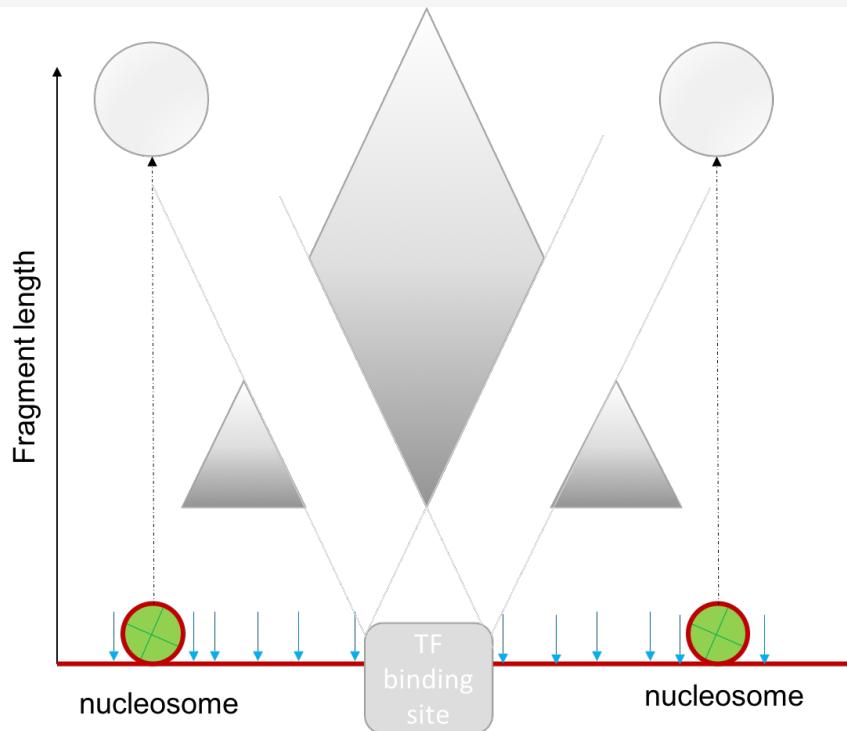
Ref: Ou et.al., 2018. doi: 10.1186/s12864-018-4559-3



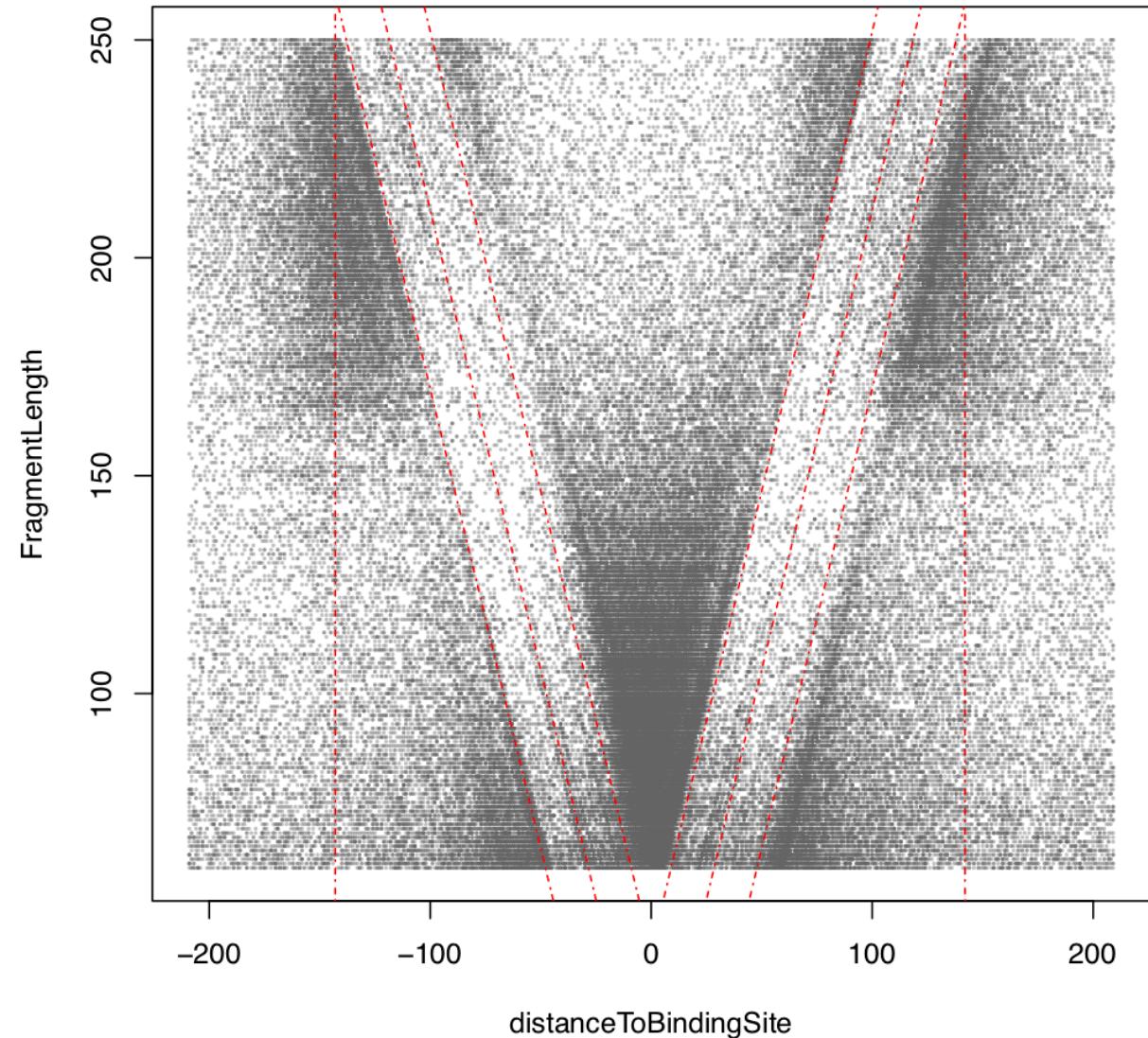
vPLOT

V-plot is a plot to visualize fragment midpoint vs length for a given transcription factors.

```
vp <- vPlot(shiftedBamfile, pfm=CTCF[[1]],  
            genome=genome, min.score="90%",  
            seqlev=seqlev,  
            upstream=200, downstream=200,  
            ylim=c(30, 250), bandwidth=c(2, 1))
```



distanceNucl=285bp, CTCFbindingWidth=41bp

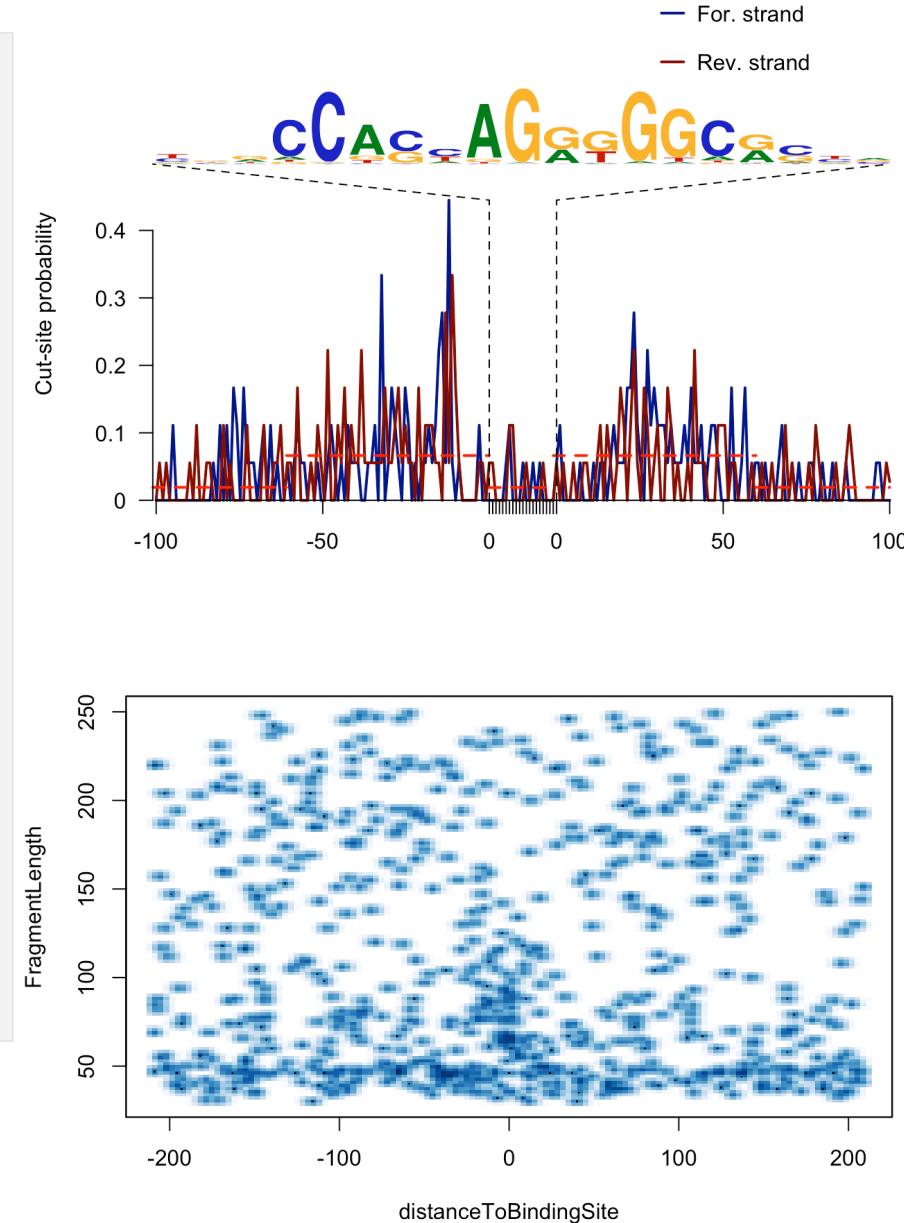


Ref: Henikoff et.al., 2011. doi: 10.1073/pnas.1110731108

FOOTPRINTS

```
## foot prints
library(MotifDb)
CTCF <- query(MotifDb, c("CTCF")) CTCF <- as.list(CTCF)
sigs <- factorFootprints(shiftedBamfile, pfm=CTCF[[1]],
                           genome=genome, min.score="90%",
                           seqlev=seqlev,
                           upstream=100, downstream=100)
```

```
vp <- vPlot(shiftedBamfile, pfm=CTCF[[1]],
              genome=genome, min.score="90%", seqlev=seqlev,
              upstream=200, downstream=200,
              ylim=c(30, 250), bandwidth=c(2, 1))
```



ATACseqQC CAN ...



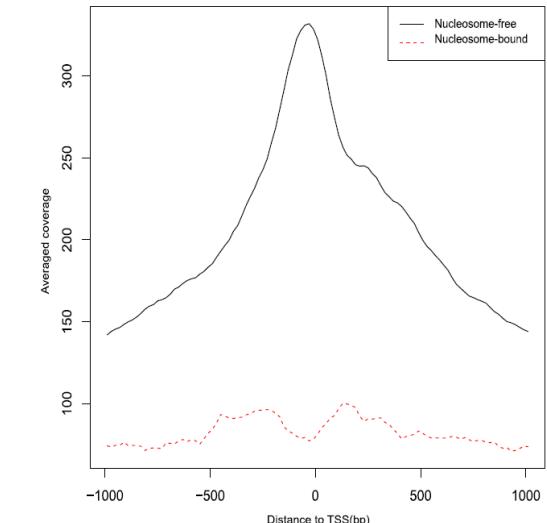
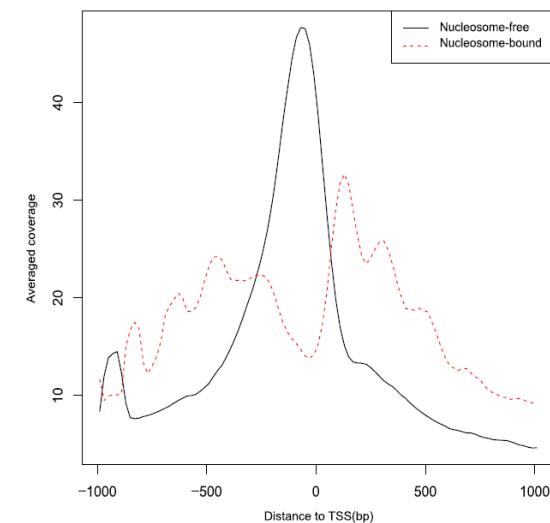
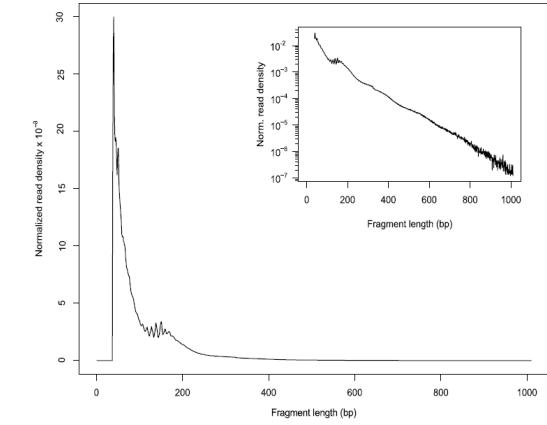
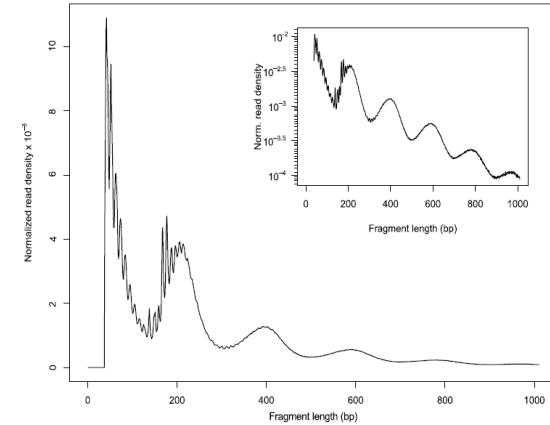
help researcher to assess the quality of the ATAC library preparation by simple plots.

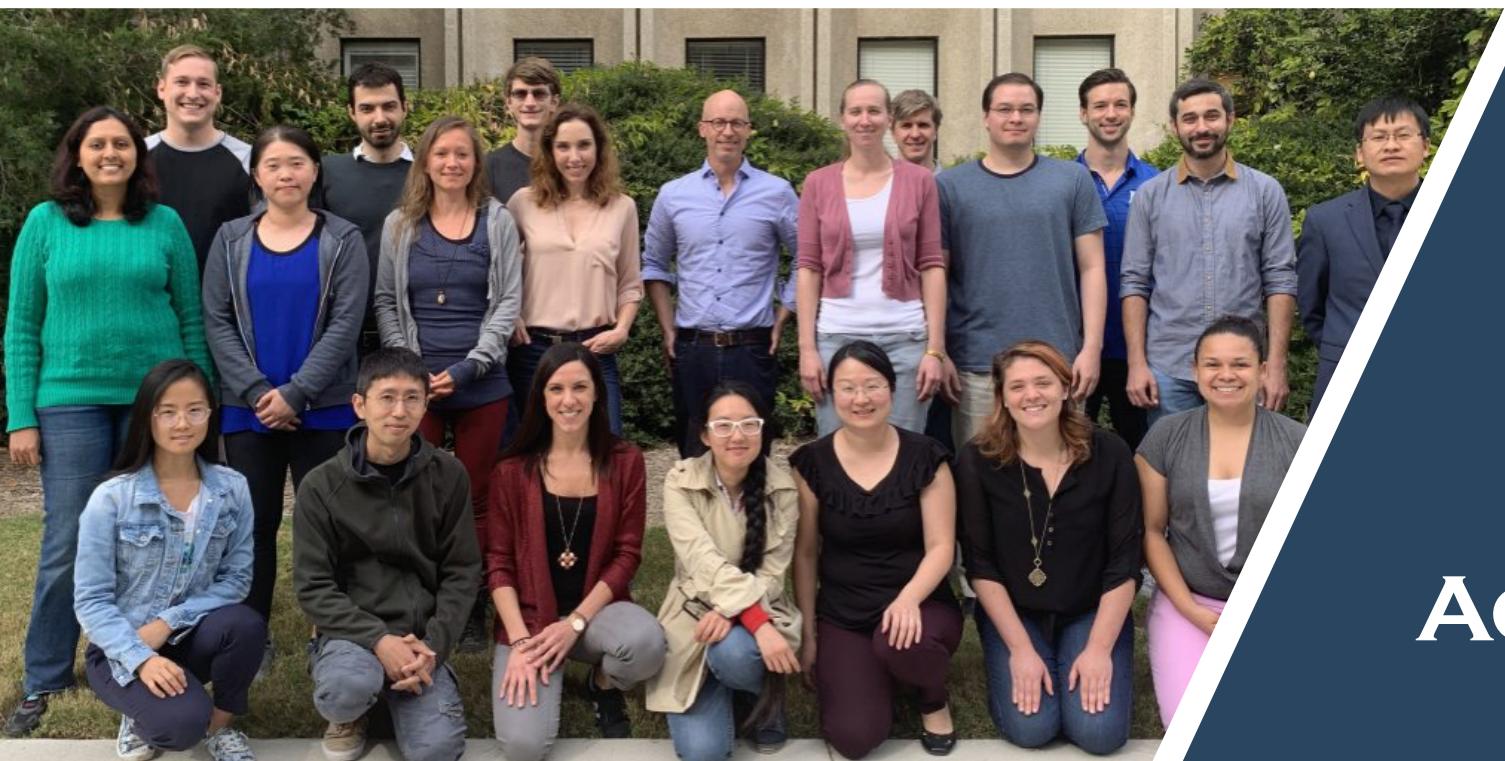


split bam files for downstream analysis and validation.



- ❖ Shifted aligned reads
- ❖ Splitting BAM files





ACKNOWLEDGEMENT