Practical 4: ChIP-seq Peak calling

Shamith Samarajiiwa, Dora Bihary

September 2017

Contents

1 Calling ChIP-seq peaks using MACS2

1.1 Assess the quality of the aligned datasets

- strand cross-correlation analysis.

A very useful ChIP-seq quality metric that is independent of peak calling is strand cross-correlation. It is based on the fact that a high-quality ChIP-seq experiment produces significant clustering of enriched DNA sequence tags at locations bound by the protein of interest, and that the sequence tag density accumulates on forward and reverse strands centered around the binding site. The cross-correlation metric is computed as the Pearson's linear correlation between the Crick strand and the Watson strand, after shifting Watson by k base pairs. This typically produces two peaks when cross-correlation is plotted against the shift value

The spp R package can be used for strand cross-correlation (not run in this practical). Two cross-correlation peaks are usually observed in a ChIP experiment, one corresponding to the read length ("phantom" peak) and one to the average fragment length of the library.

see ***Landt et al., 2012, Genome Res. "ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia"*** for more details.

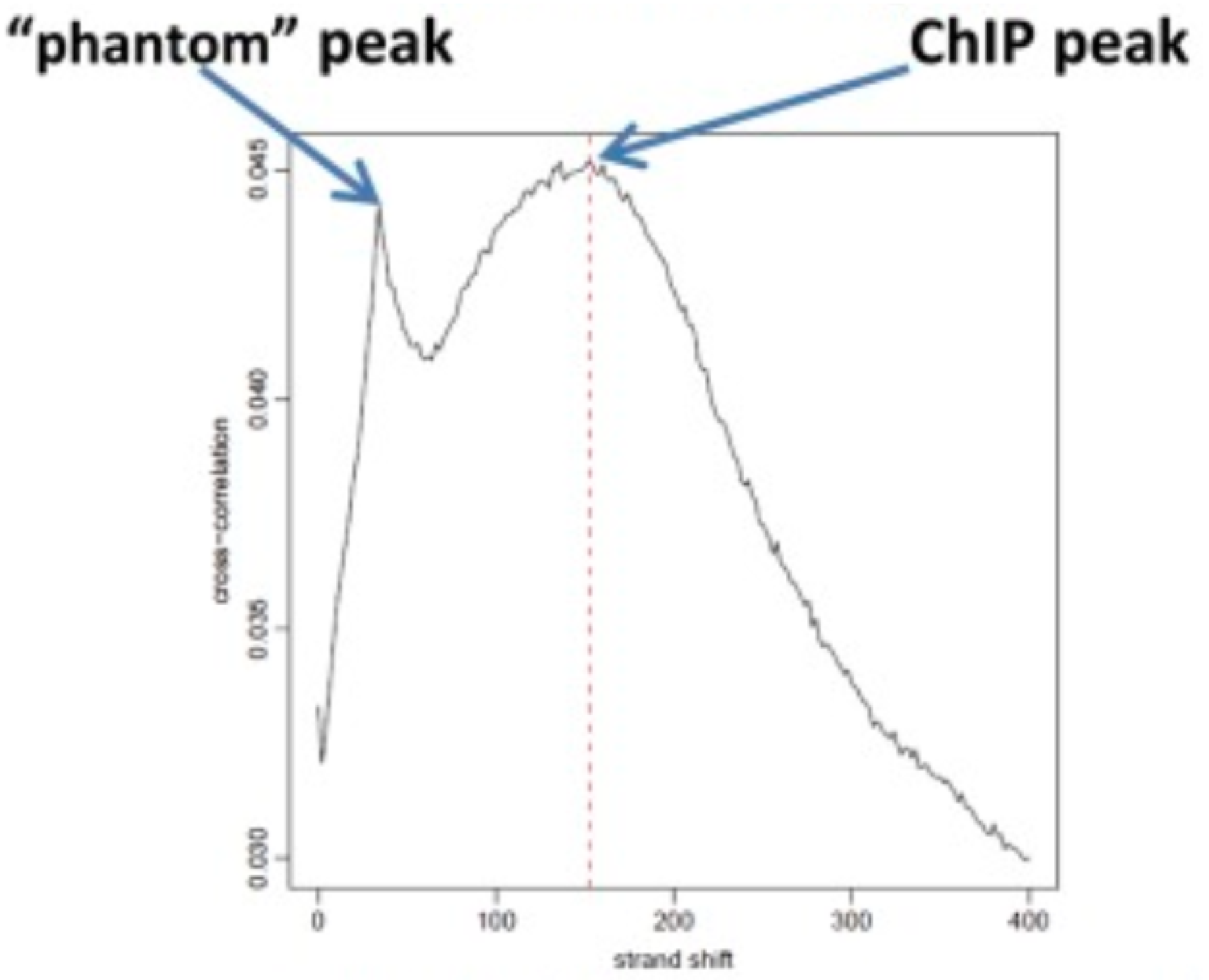

Figure 1:

The absolute and relative height of the two peaks are useful determinants of the success of a ChIP-seq experiment. A high-quality immunoprecipitation is characterized by a ChIP peak that is much higher than the "phantom" peak, while often very small or no such peak is seen in failed experiments.

1.2 Peak Calling

Some parameters to consider: * Adjust the sequence tags to better represent the original DNA fragment (by 'shifting tags in the 3 prime direction' or by 'extending tags' to the estimated length of the original fragment length) * Background model used * Use of strand dependent bimodality * fragment length, read length, replicates, duplication, total read count

1.2.1 Filter duplicates

```
# change directory
cd /home/participant/Course_Materials/ChIPSeq/Materials/Practicals
```

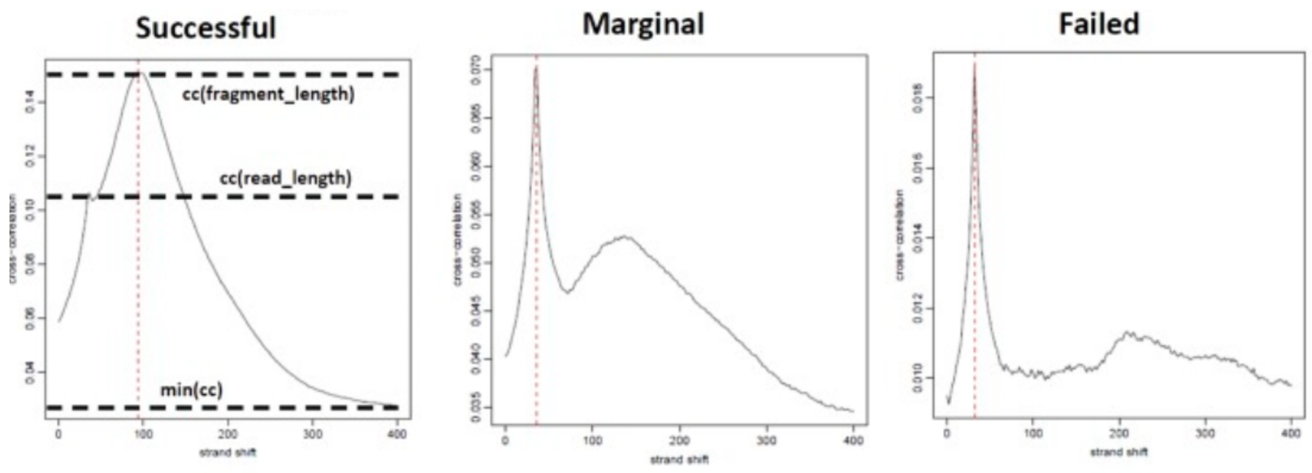

Figure 2:

```
mkdir Macs2
cd Macs2

# Filter duplicates
# Write down the count of reads for each sample after duplicate removal

macs2 filterdup \
-i "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/tp53_r2.fastq_trimmed.fastq_sorted.bam" \
-f BAM -g hs --keep-dup=1  --verbose=3 -o "tp53_r2.fastq_trimmed.fastq_sorted_filterdup.bed"

macs2 filterdup \
-i "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/TAp73beta_r2.fastq_trimmed.fastq_sorted.bam" \
-f BAM -g hs --keep-dup=1  --verbose=3 -o "TAp73beta_r2.fastq_trimmed.fastq_sorted_filterdup.bed"

macs2 filterdup \
-i "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/input.fastq_trimmed.fastq_sorted.bam" \
-f BAM -g hs --keep-dup=1  --verbose=3 -o "input.fastq_trimmed.fastq_sorted_filterdup.bed"
```

- Scaling factor = ChIP read count / Input readcount

1.2.2 Predict fragment length

```

# predict fragment length
# write down the fragment length for each sample

macs2 predictd -i tp53_r2.fastq_trimmed.fastq_sorted_filterdup.bed -g hs -m 5 20
macs2 predictd -i TAp73beta_r2.fastq_trimmed.fastq_sorted_filterdup.bed -g hs -m 5 20
macs2 predictd -i input.fastq_trimmed.fastq_sorted_filterdup.bed -g hs -m 5 20
```

- try tweaking the mfold parameter

1.2.3 MACS2 options

```
# MACS2 callpeak options
macs2 callpeak -h

# -t sample -c control -g effective genome size needs to be empirically computed using
# a hg38.fa genome file for
# hg38 but for this practical use 'hs' which is = 2.6e9, the value for hg19
# -f filetype --bdg generate bedgraph
```

1.2.4 Standard MACS2 run

```
cd /home/participant/Course_Materials/ChIPSeq/Materials/Practicals


macs2 callpeak \
-t "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/tp53_r2.fastq_trimmed.fastq_sorted.bam" \
-c  "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/input.fastq_trimmed.fastq_sorted.bam" \
-g hs -n tp53_r2.fastq_trimmed.fastq_sorted_standard -f BAM --keep-dup auto --bdg

macs2 callpeak \
-t "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/TAp73beta_r2.fastq_trimmed.fastq_sorted.bam" \
-c "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/input.fastq_trimmed.fastq_sorted.bam" \
-g hs -n TAp73beta_r2.fastq_trimmed.fastq_sorted_standard -f BAM --keep-dup auto --bdg
```

- -bdg create bedgraph output files (similar to wiggle files)
- -f Input format (typically BAM for single-end reads and BAMPE for paired-end reads). In BAMPE mode, MACS2 will only use properly-paired read alignments.
- –nomodel whether or not to build the shifting model. If True, MACS will not build model. By default it means shifting size = 100.
- –extsize When nomodel is true, MACS will use this value as fragment size to extend each read towards 3 prime end
- –keep-dup How should MACS handle duplicate tags at the same location? The default is to keep just one. The **auto** option (recommended for high-coverage samples) has MACS calculate the expected maximum tags at the same location (based on the binomal distribution).
- The ChIPQC or SPP Bioconductor packages (or the MACS2 predictd) can be used to determine the fragment size used as input for (–extsize)

1.2.5 set the –extsize based on MACS2 predictd fragment length

```
#use d calculated above for --extsize

macs2 callpeak \
-t "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/tp53_r2.fastq_trimmed.fastq_sorted.bam" \
-c "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/input.fastq_trimmed.fastq_sorted.bam" \
-g hs -n tp53_r2.fastq_trimmed.fastq_sorted_extended -f BAM --keep-dup auto --bdg --nomodel --extsize

macs2 callpeak \
-t "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/TAp73beta_r2.fastq_trimmed.fastq_sorted.bam" \
-c "~/Course_Materials/ChIPSeq/Preprocessed/Alignment_BWA/input.fastq_trimmed.fastq_sorted.bam" \
-g hs -n TAp73beta_r2.fastq_trimmed.fastq_sorted_extended -f BAM --keep-dup auto --bdg --nomodel --extsize
```

1.3 Working with peaks

MACS2 peak calls are written to the user specified output directory. the files have an "_peaks.xls" extension.

In addition this file contains information on samples and parameters used.

```r
peakfile <- "tp53_r2.fastq_trimmed.fastq_sorted_standard_peaks.xls"
peaks_DF <- read.delim(peakfile)
peaks_DF[1:25,]
```

importing peaks into R

```r
peakfile <- "tp53_r2.fastq_trimmed.fastq_sorted_standard_peaks.xls"
peaks_DF <- read.delim2(peakfile, comment.char="#")
peaks_DF[1:3,]
```

Create GRanges object made of chromosome names and intervals stored as IRanges.

```r
library(GenomicRanges)
peaks_GR <- GRanges(
 seqnames=peaks_DF[,"chr"],
 IRanges(peaks_DF[,"start"],
         peaks_DF[,"end"]
 )
)



peaks_GR
```

```r
seqnames(peaks_GR)
```

```r
ranges(peaks_GR)
```

Compare peak sets

```r
peakfile1 <- "tp53_r2.fastq_trimmed.fastq_sorted_extended_peaks.xls"
peaks_DF1 <- read.delim2(peakfile1, comment.char="#")

  # load into GenomicRanges
  peaks_GR1 <- GRanges (
  seqnames=peaks_DF1[,"chr"],
  IRanges(peaks_DF1[,"start"],
  peaks_DF1[,"end"]
  )
)



peakfile2 <- "TAp73beta_r2.fastq_trimmed.fastq_sorted_extended_peaks.xls"
peaks_DF2 <- read.delim(peakfile2, comment.char="#")

  # load into GenomicRanges
  peaks_GR2 <- GRanges(
  seqnames=peaks_DF2[,"chr"],
  IRanges(peaks_DF2[,"start"],
  peaks_DF2[,"end"]
  )
)
```

The number of peaks in first replicate overlapping those in second is different from number of second replicate peaks overlapping first. This is because 2 peaks from one replicate may overlap 1 peak in the other replicate (`Transitive` property of Venn operations between genomic ranges).

```
OnlyfirstPeakSet <- peaks_GR1[!peaks_GR1 %over% peaks_GR2]
OnlySecondPeakSet <- peaks_GR2[!peaks_GR2 %over% peaks_GR1]

firstANDsecondPeakSets <- peaks_GR1[peaks_GR1 %over% peaks_GR2]
secondANDfirstPeakSets <- peaks_GR2[peaks_GR2 %over% peaks_GR1]

length(OnlyfirstPeakSet)
length(OnlySecondPeakSet)

length (firstANDsecondPeakSets)
length (secondANDfirstPeakSets)
```

Finally convert Granges to bed

```
df <- data.frame(seqnames=seqnames(peaks_GR1),
                 starts=start(peaks_GR1)-1,
                 ends=end(peaks_GR1))

write.table(df, file="peaks_GR1_converted.bed", quote=F, sep="\t", row.names=F, col.names=F)
```

Acknowledgments: This practical is modified from ChIPseq course given by Tom Carroll (MRC CSC)