ChIP-seq analytics, Practical 2

Created by Jun Wang, 2023/2024, CAN7031/131

This practical covers ChIP-seq downstream analysis after peak calling, including peak annotation, comparison between samples and transcription factor (TF) binding analysis. We will use the R package 'ChIPseeker' for peak annotation and sample comparison, and use MEME suite for TF binding motif analysis

(1) First open Rstudio, install and load R packages, 'ChIPseeker', gene annotation library

Use Bioconductor biocManager to install required R packages

```
if (!require("BiocManager", quietly = TRUE))
install.packages("BiocManager")
BiocManager::install("ChIPseeker")

if (!require("BiocManager", quietly = TRUE))
   install.packages("BiocManager")
BiocManager::install("TxDb.Hsapiens.UCSC.hg19.knownGene")
```

This installs "TxDb.Hsapiens.UCSC.hg19.knownGene" database in your R. If you sequencing data is analysed using hg38, install "TxDb.Hsapiens.UCSC.hg38.knownGene"

Now load "ChIPseeker" and "TxDb.Hsapiens.UCSC.hg19.knownGene"

```
> library(ChIPseeker)
> library(TxDb.Hsapiens.UCSC.hg19.knownGene)
```

(2) Download the zipped file from QMplus CAN7031/131 module page under ChIP-seq tab, and unzip in your local working directory. There are three peak BED files. Use R to read the peak files and ChIPseeker to load peak files. Here we only use peak calls from hg19 chromosome 12 for this practical to save time.

In Rstudio, set you working directory

```
> setwd("your_working_directory")
```

e.g., setwd("/Users/wang02/Work/MSc genomics and data sciences/Teaching materials/Module CAN7031 Omics analytics/ChIP-seq/")

```
> files <-
list("GSM1574235_H3K27ac.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed",
"GSM1574242_H3K4me1.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed",
"GSM1574256_H3K4me3.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed")
```

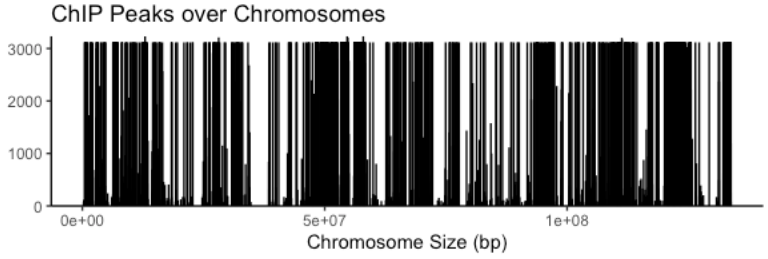
common error here is using “. This is the wrong double quote in R. You should use ". You can just copy this " in R to use

```
> files
[[1]]
[1] "GSM1574235_H3K27ac.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed"

[[2]]
[1] "GSM1574242_H3K4me1.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed"

[[3]]
[1] "GSM1574256_H3K4me3.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed"

> files[[3]]
[1] "GSM1574256_H3K4me3.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed"

> peak <- readPeakFile(files[[3]], header=F)
> peak
GRanges object with 1507 ranges and 2 metadata columns:
        seqnames            ranges strand |          V4        V5
           <Rle>         <IRanges>  <Rle> |    <character> <numeric>
    [1]    chr12       253452-254110      * | MACS_peak_5799     51.82
    [2]    chr12       297877-298674      * | MACS_peak_5800    100.90
    [3]    chr12       495461-499966      * | MACS_peak_5801   3100.00
    [4]    chr12       509687-512362      * | MACS_peak_5802   3100.00
    [5]    chr12       568181-570851      * | MACS_peak_5803   3100.00
    ...      ...                 ...    ... .          ...       ...
 [1503]    chr12 133756569-133762618      * | MACS_peak_7301   3100.00
 [1504]    chr12 133762727-133765402      * | MACS_peak_7302    227.64
 [1505]    chr12 133765929-133772514      * | MACS_peak_7303   1174.17
 [1506]    chr12 133777056-133784094      * | MACS_peak_7304   3100.00
 [1507]    chr12 133786907-133787946      * | MACS_peak_7305     63.83
 -------
 seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

now you have “peak” as the ChIP peaks of H3K4me3 CAPAN1_vs_INPUT data from chromosome 12, in total of 1507 peaks

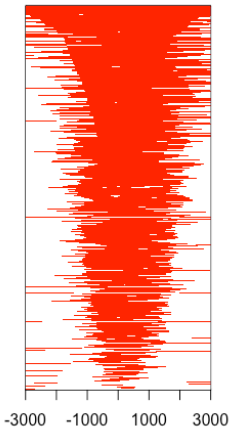
(3) Create a chromosome plot to display the ChIP peaks over chromosomes

```
> covplot(peak, weightCol="V5", chrs="chr12")
```



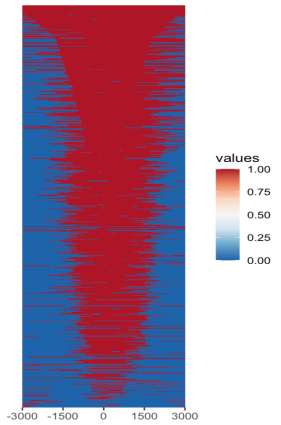
(4) Profile of ChIP peaks binding to transcription starting site (TSS) regions

```
> txdb19 <- TxDb.Hsapiens.UCSC.hg19.knownGene

> library(clusterProfiler)

## install clusterProfiler package as below if you don't have it in your computer

if (!require("BiocManager", quietly = TRUE))
   install.packages("BiocManager")
BiocManager::install("clusterProfiler")

## first prepare the TSS regions
## expand the region to upstream 3000bp and downstream 3000bp

> promoter <- getPromoters(TxDb=txdb19, upstream=3000, downstream=3000)
> tagMatrix <- getTagMatrix(peak, windows=promoter)
###>> preparing start_site regions by ... 2022-10-14 10:34:15
###>> preparing tag matrix...  2022-10-14 10:34:15

##Heatmap of ChIP binding to TSS regions
> tagHeatmap(tagMatrix, xlim=c(-3000, 3000), color="red")
```

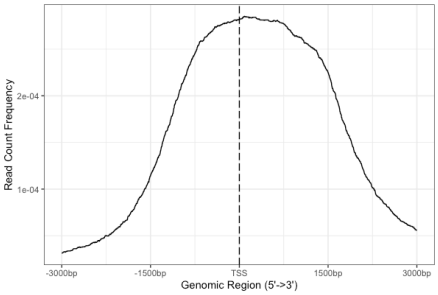


Or just try

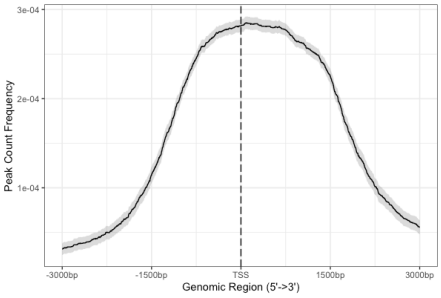
```
> tagHeatmap(tagMatrix)
```



##Average Profile of ChIP peaks binding to TSS region
> plotAvgProf(tagMatrix, xlim=c(-3000, 3000),
xlab="Genomic Region (5'->3')", ylab = "Read Count Frequency")
###>> plotting figure... 2022-10-14 10:35:37



##Confidence interval estimated by bootstrap method
> plotAvgProf(tagMatrix, xlim=c(-3000, 3000), conf = 0.95, resample = 1000)
###>> plotting figure... 2022-10-13 17:03:45



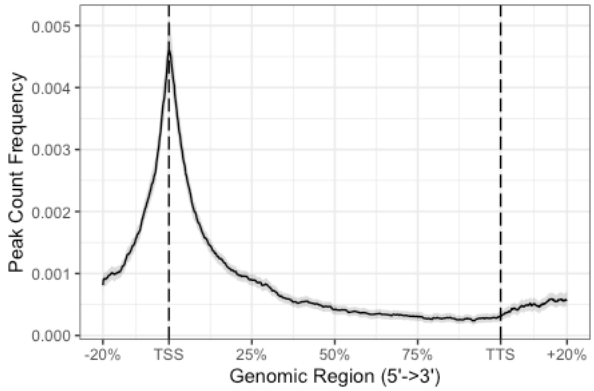
be mindful if you do this for the whole chromosome, as this resampling to calculate 95% confidence interval will take lots of your computing resources and time.

(5) Profile of ChIP peaks binding to body regions

##Here uses `plotPeakProf2` to do all things in one step.
##Binning method for profile of ChIP peaks
the ignore_strand is FALSE in default. We put here to emphasize that.

> plotPeakProf2(peak = peak, upstream = rel(0.2), downstream = rel(0.2),
conf = 0.95, by = "gene", type = "body", nbin = 800,

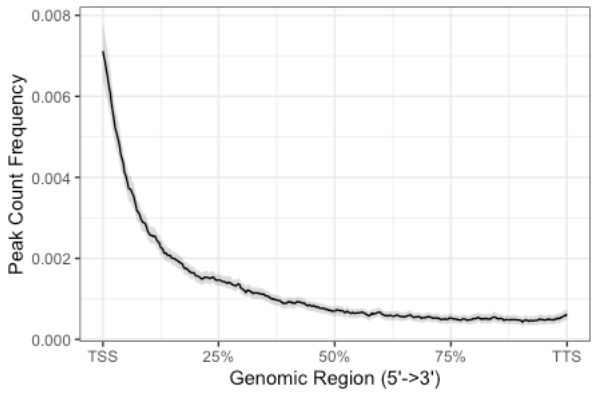
TxDb = txdb19, weightCol = "V5",ignore_strand = F)
##>> binning method is used…2022-10-14 10:48:08
##>> preparing body regions by gene... 2022-10-14 10:48:08
##>> preparing tag matrix by binning... 2022-10-14 10:48:08
##>> preparing matrix with extension from (TSS-20%)~(TTS+20%)... 2022-10-14 10:48:08
##>> 12 peaks(1.423488%), having lengths smaller than 800bp, are filtered... 2022-10-14 10:48:10



we can also use getBioRegion(), getTagMatrix() and plotPeakProf() to plot in three steps.

> genebody <- getBioRegion(TxDb = txdb19, by = "gene", type = "body")
> matrix_no_flankextension <- getTagMatrix(peak,windows = genebody, nbin = 800)
##>> binning method is used...2022-10-14 10:49:54
##>> preparing body regions by … 2022-10-14 10:49:54
##>> preparing tag matrix by binning... 2022-10-14 10:49:54
##>> preparing matrix for body region with no flank extension... 2022-10-14 10:49:54
##>> 12 peaks(1.494396%), having lengths smaller than 800bp, are filtered... 2022-10-14 10:49:54

> plotPeakProf(matrix_no_flankextension,conf = 0.95)
##>> Running bootstrapping for tag matrix... 2022-10-14 10:50:12

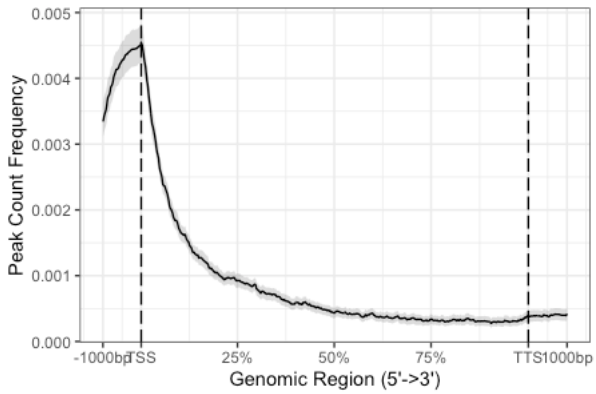


you can also use getTagMatrix() and plotPeakProf() to plot in two steps

> matrix_actual_extension <- getTagMatrix(peak,windows = genebody, nbin = 800, upstream = 1000,downstream = 1000)

##>> binning method is used...2022-10-14 10:51:39
##>> preparing body regions by ... 2022-10-14 10:51:39
##>> preparing tag matrix by binning... 2022-10-14 10:51:39
##>> preparing matrix with flank extension from (TSS-1000bp)~(TTS+1000bp)... 2022-10-14 10:51:39
##>> 15 peaks(1.838235%), having lengths smaller than 800bp, are filtered... 2022-10-14 10:51:39

> plotPeakProf(matrix_actual_extension,conf = 0.95)
##>> Running bootstrapping for tag matrix... 2022-10-14 10:51:55



see the manual to plot against other regions

(6) Peak annotation

Since some annotation may overlap, ChIPseeker adopted the following priority in genomic annotation: Promoter, 5' UTR, 3' UTR, Exon, Intron, Downstream, Intergenic

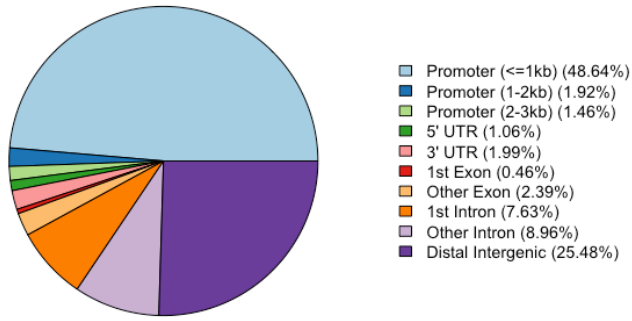
install Bioconductor R package ("org.Hs.eg.db"

```
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("org.Hs.eg.db")
```

> library(org.Hs.eg.db)

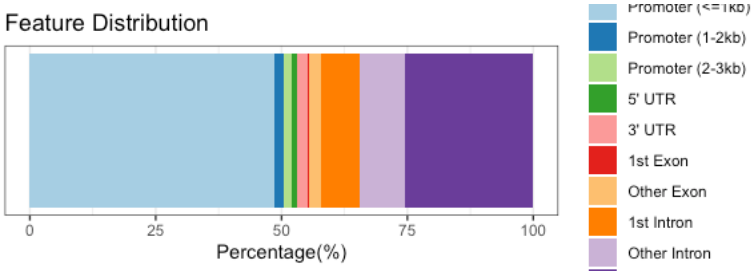
```
> peakAnno <- annotatePeak(peak, tssRegion=c(-3000, 3000),
                           TxDb=txdb19, annoDb="org.Hs.eg.db")
##>> preparing features information...                2022-10-14 10:54:10
##>> identifying nearest features...                  2022-10-14 10:54:11
##>> calculating distance from peak to TSS...         2022-10-14 10:54:12
##>> assigning genomic annotation...                  2022-10-14 10:54:12
##>> assigning chromosome lengths                     2022-10-14 10:54:31
##>> done...                                          2022-10-14 10:54:31
```

> plotAnnoPie(peakAnno)



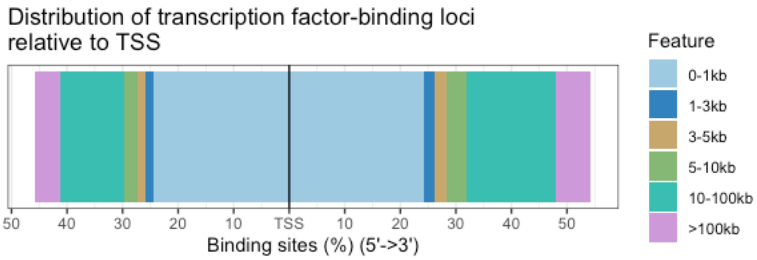
Note that we use H3K4me3 peaks to annotate in the example, and H3K4me3 marks the active promoter regions, this confirms that H3K4me3 marked regions largely reside with promoter regions in this annotation.
This pie chart is very different from the one in the lecture slides where H3K27ac peaks were used, which marks active enhancer regions.

> plotAnnoBar(peakAnno)



plotDistToTSS to calculate the percentage of binding sites upstream and downstream from the TSS of the nearest genes, and visualize the distribution.

> plotDistToTSS(peakAnno,
title="Distribution of transcription factor-binding loci\nrelative to TSS")



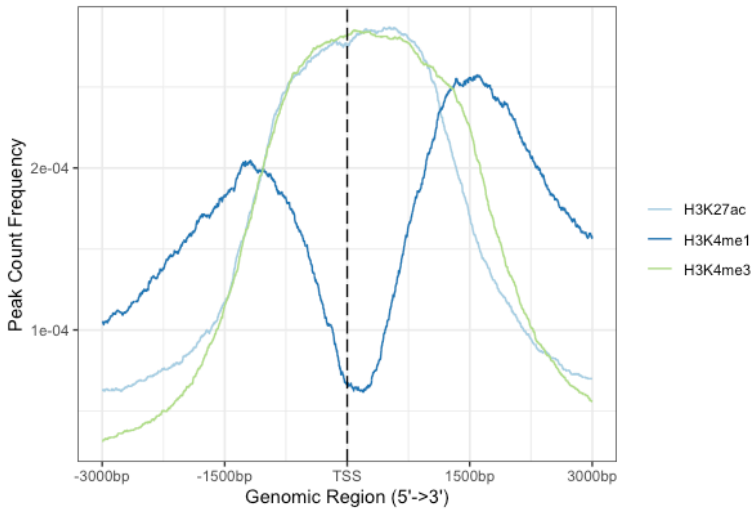
again you can see H3K4me3 peaks were largely around promoter regions

(7) Compare multiple peaks using ChIPseeker

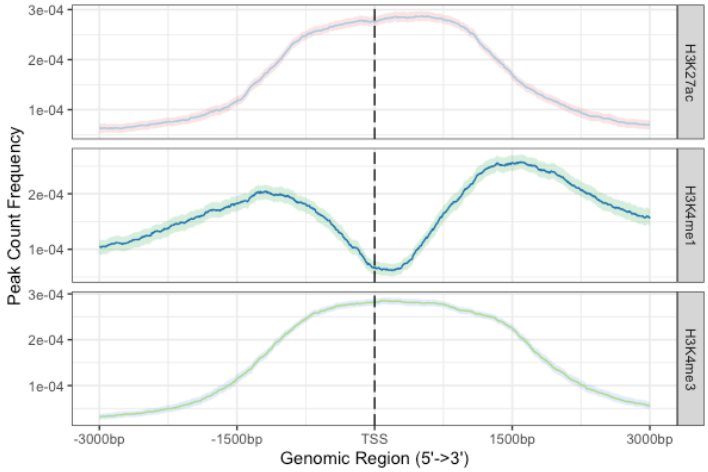
Now we use the three peak files "GSM1574235_H3K27ac.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed", "GSM1574242_H3K4me1.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed", "GSM1574256_H3K4me3.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed", and compare peaks between them

Note that, enhancers are marked by H3K4me1, and promoters are marked by H3K4me3, and H3K27ac marks open chromatin and enchancers

```
> files
[[1]]
[1] "GSM1574235_H3K27ac.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed"

[[2]]
[1] "GSM1574242_H3K4me1.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed"

[[3]]
[1] "GSM1574256_H3K4me3.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed"

> names(files) <- c("H3K27ac", "H3K4me1", "H3K4me3")

> library(TxDb.Hsapiens.UCSC.hg19.knownGene)
> txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene
> library(clusterProfiler)
> promoter <- getPromoters(TxDb=txdb, upstream=3000, downstream=3000)
> tagMatrixList <- lapply(files, getTagMatrix, windows=promoter)
##>> preparing start_site regions by ... 2022-10-14 11:12:19
##>> preparing tag matrix...  2022-10-14 11:12:19
##>> preparing start_site regions by ... 2022-10-14 11:12:21
##>> preparing tag matrix...  2022-10-14 11:12:21
##>> preparing start_site regions by ... 2022-10-14 11:12:21
##>> preparing tag matrix...  2022-10-14 11:12:21

## we will compare the peaks among the three files in relation to promoters

## average profile of the 3 bed files against promoters
> plotAvgProf(tagMatrixList, xlim=c(-3000, 3000))
##>> plotting figure...                    2022-10-14 11:18:40
```



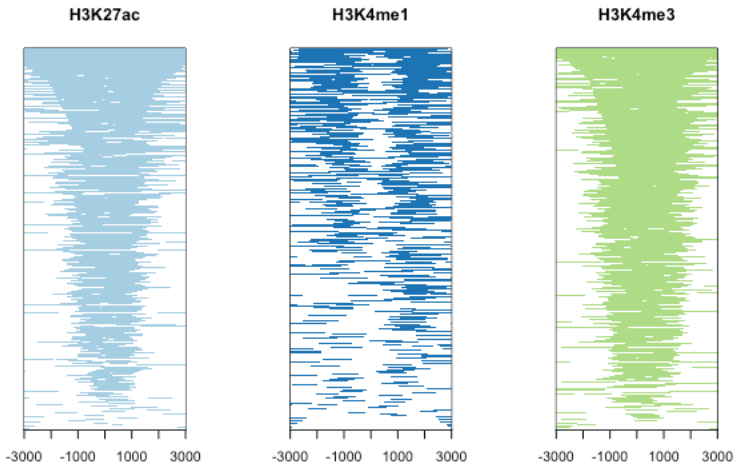

```
> plotAvgProf(tagMatrixList, xlim=c(-3000, 3000),
conf=0.95,resample=500, facet="row")
##>> plotting figure...                    2022-10-14 11:20:43
##>> Running bootstrapping for tag matrix...          2022-10-14 11:20:57
##>> Running bootstrapping for tag matrix...          2022-10-14 11:21:14
##>> Running bootstrapping for tag matrix...          2022-10-14 11:21:31
```



peak heatmap across samples

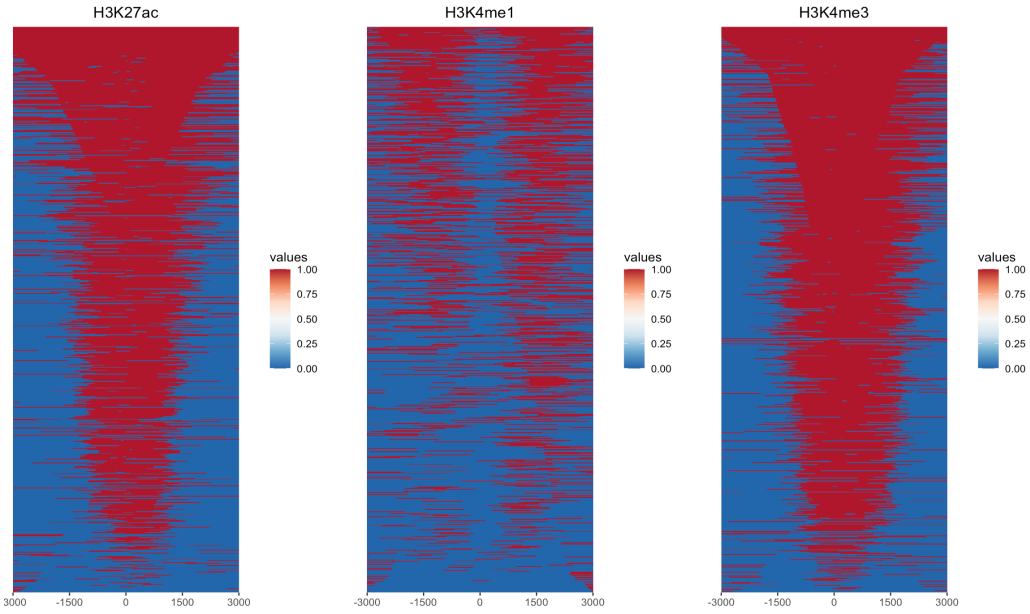
you can clear all previous plot using the following command
> dev.off()

> tagHeatmap(tagMatrixList, xlim=c(-3000, 3000), color=NULL)



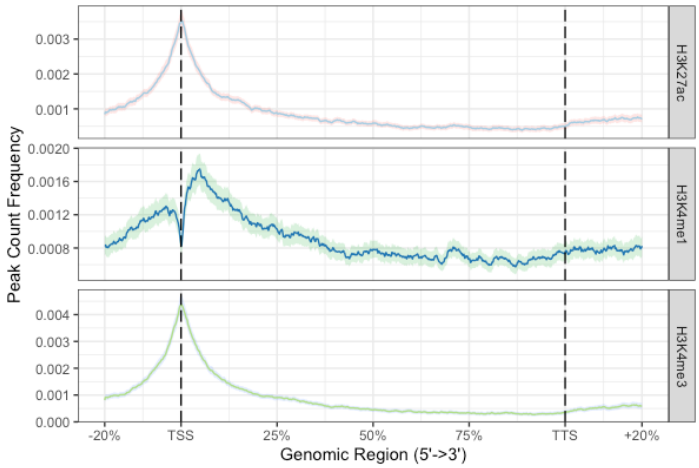
Or just try

> tagHeatmap(tagMatrixList)



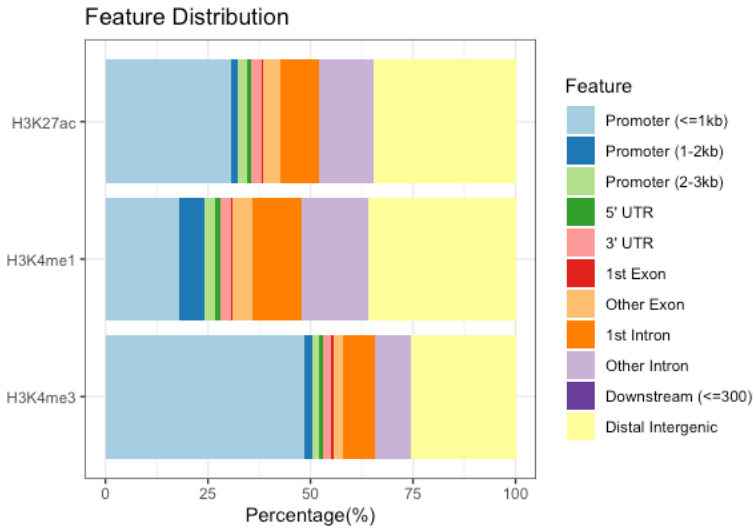
Profile of the three ChIP peak files binding to body region

```
> plotPeakProf2(files, upstream = rel(0.2), downstream = rel(0.2),
              conf = 0.95, by = "gene", type = "body",
              TxDb = txdb, facet = "row", nbin = 800)
```

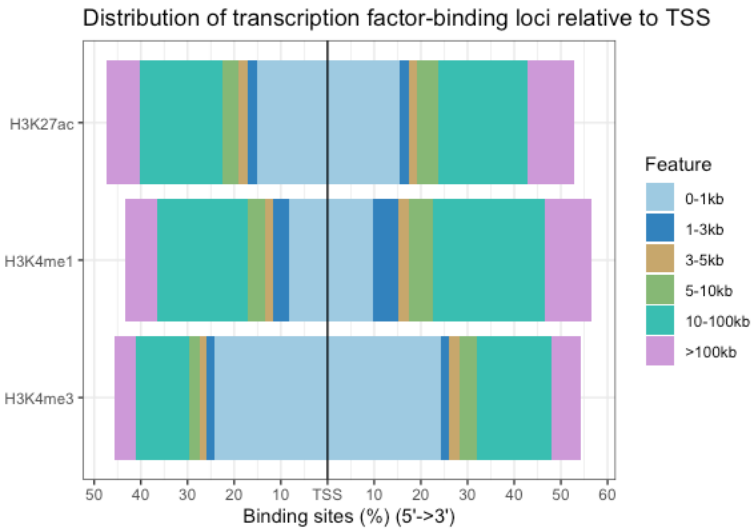


ChIP peak annotation comparison

```
> peakAnnoList <- lapply(files, annotatePeak, TxDb=txdb,
                         tssRegion=c(-3000, 3000), verbose=FALSE)

> ## use plotAnnoBar to comparing their genomic annotation.
> plotAnnoBar(peakAnnoList)
```




```
## plotDistToTSS to compare distance to TSS profiles among ChIPseq data.
> plotDistToTSS(peakAnnoList)
```

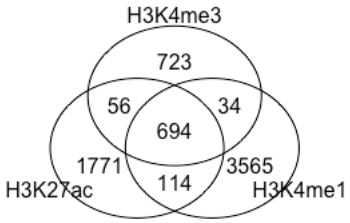


(8) Overlap of peaks and annotated genes

compared annotated genes across the 3 ChIP-seq peak files
> genes= lapply(peakAnnoList, function(i) as.data.frame(i)$geneId)
> vennplot(genes)

it is likely your plot looks very thin, so you can clear all previous plots and then re-plot

> dev.off()
> vennplot(genes)



Finally, we can also perform peak overlap enrichment analysis between these three peak files in the practical.

Shuffle genome coordination
> p <- GRanges(seqnames="chr12",
ranges=IRanges(start=c(1, 100), end=c(50, 130)))
> shuffle(p, TxDb=txdb)

> enrichPeakOverlap(queryPeak = files[[3]], targetPeak = unlist(files[1:2]), TxDb = txdb, pAdjustMethod = "BH", nShuffle = 50, chainFile = NULL, verbose = FALSE)

this command compares H3K4me3 peals against H3K27ac and H3K4me1 peaks, and determine if there is a significant overlap between gene annotation

```
> enrichPeakOverlap(queryPeak = files[[3]], targetPeak = unlist(files[1:2]), TxDb = txdb, pAd
justMethod = "BH", nShuffle = 50, chainFile = NULL, verbose = FALSE)
                                                        qSample
H3K27ac GSM1574256_H3K4me3.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed
H3K4me1 GSM1574256_H3K4me3.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed
                                                        tSample qLen tLen N_OL
H3K27ac GSM1574235_H3K27ac.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed 1507 2635 1266
H3K4me1 GSM1574242_H3K4me1.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed 1507 4407 1370
            pvalue   p.adjust
H3K27ac 0.01960784 0.01960784
H3K4me1 0.01960784 0.01960784
```

There seems to be a significant overlap of gene annotation between the three gene annotation.

(9) Finally, let's do some practise using MEME suite for TF binding analysis

As this is a very computational intense analysis, let's randomly select 10 peaks from the H3K4me3 peaks and perform the motif discovery and enrichment analysis for this practical.

> peaks_H3K4me3 <- read.delim("GSM1574256_H3K4me3.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed", header=F)

randomly select 10 peaks from the file
> set.seed(12345) # Set seed for reproducibility

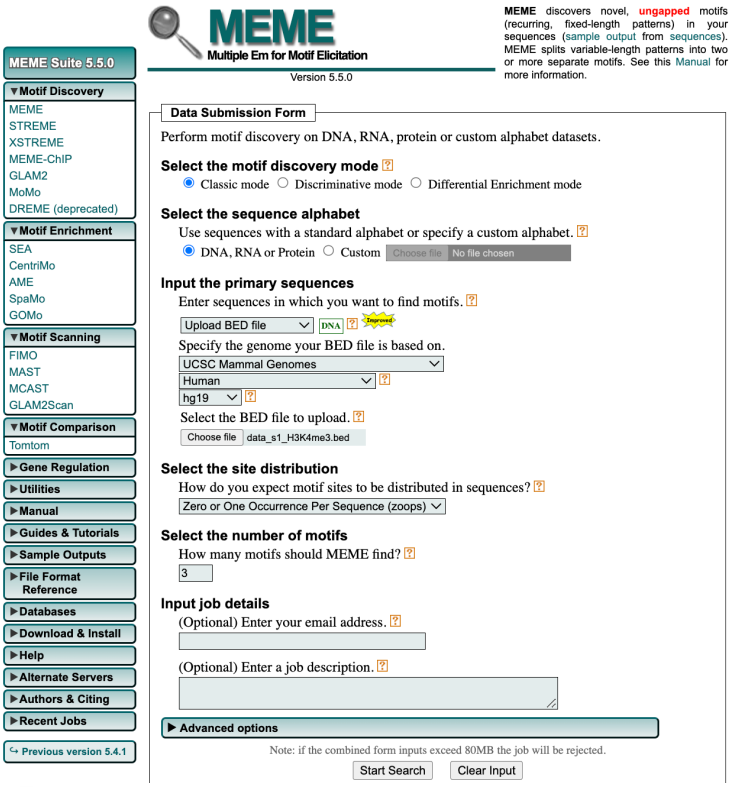
Sample rows of data with Base R
> data_s1 <- peaks_H3K4me3[sample(1:nrow(peaks_H3K4me3), 10),]

Print sampled data
> data_s1

write this into a new BED file, ready for the analysis using MEME suite
> write.table(data_s1, file="data_s1_H3K4me3.bed", quote=F, sep="\t", row.names=F, col.names = F)

(9.1) Now, we are going to first use MEME suite to perform TF motif discovery

MEME discovers novel, ungapped motifs (recurring, fixed-length patterns) in the query sequences



Please wait. Your MEME job is now running. Further details may be available below. You may bookmark this page or use the **Recent Jobs** menu at the left to access your job's results.

Job Details ...

Results

- (Primary) Uploaded BED file
- (Primary) Sequences from BED file

Status Messages

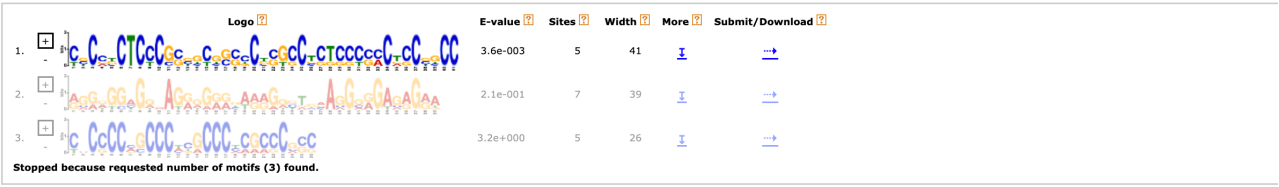
- Parsing arguments
- Arguments ok
- Starting bed2fasta

```
bed2fasta -s -both -o data_s1_H3K4me3.bed.fa data_s1_H3K4me3.bed /data/apache-tomcat/instance/meme/work/meme-data-new/fasta_databases/UCSCMammal/hg19.fna
```

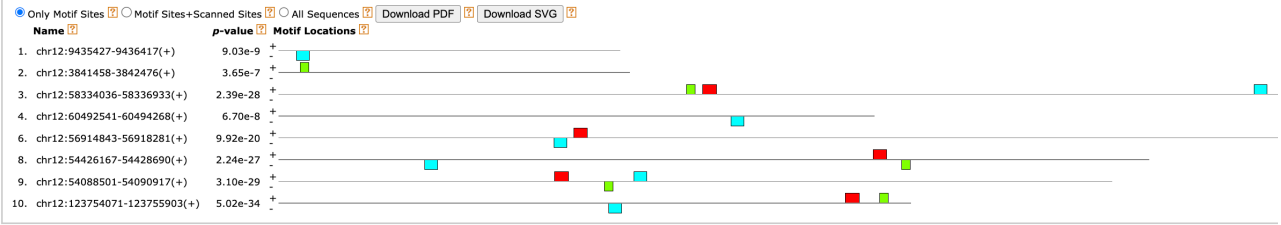
- bed2fasta ran successfully in 0 seconds
- Starting meme

```
meme data_s1_H3K4me3.bed.fa -dna -oc . -nostatus -time 14400 -mod zoops -nmotifs 3 -minw 6 -maxw 50 -objfun classic -revcomp -markov_order 0
```

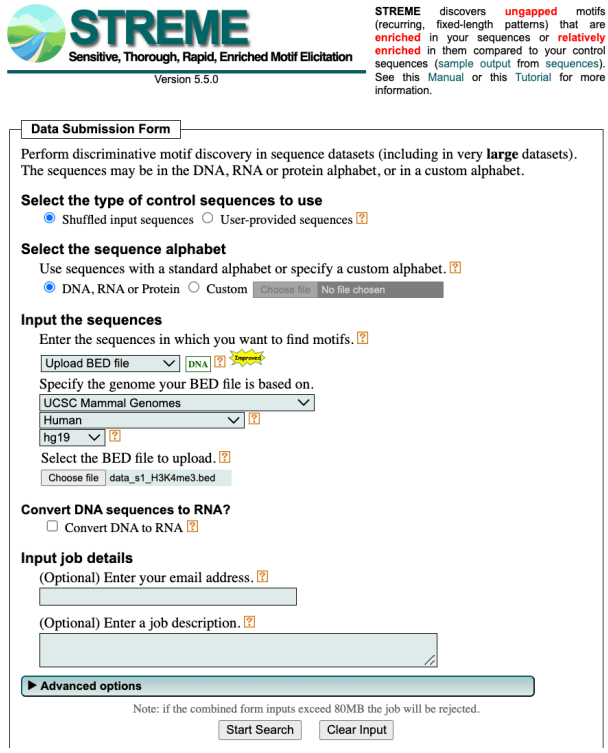
Discovered Motifs



Motif Locations



(9.2) use STREME to perform discriminative motif discovery in sequence datasets. STREME discovers ungapped motifs (recurring, fixed-length patterns) that are enriched in your sequences or relatively enriched in them compared to your control sequences.



Your STREME job is complete. The results should be displayed below.

Job Details ...

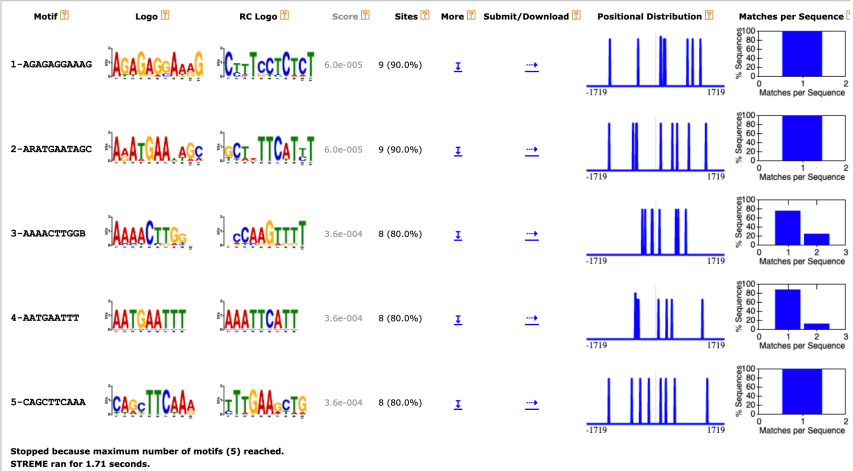
Results

- STREME HTML output
- STREME text output (motifs in MEME format)
- STREME XML output
- (Primary) Uploaded BED file
- (Primary) Sequences from BED file
- Warning/Status Messages

Status Messages

- Parsing arguments
- Arguments ok
- Starting bed2fasta
  `bed2fasta -s -both -o data_s1_H3K4me3.bed.fa data_s1_H3K4me3.bed /data/apache-tomcat/instance/meme/work/meme-data-new/fasta_databases/UCSCMammal/hg19.fna`
- bed2fasta ran successfully in 0 seconds
- Starting streme
  `streme --verbosity 1 --oc . --dna --totallength 4000000 --time 14400 --minw 8 --maxw 15 --thresh 0.05 --align center --p data_s1_H3K4me3.bed.fa`
- streme ran successfully in 1.87 seconds
- Done

Discovered Motifs



(9.3) use SEA to Perform motif enrichment analysis in sequence datasets.
SEA identifies known or user-provided motifs that are relatively enriched in your sequences compared with shuffled sequences or your control sequences

SEA
Simple Enrichment Analysis
Version 5.5.0

SEA identifies **known** or **user-provided** motifs that are **relatively** enriched in your sequences compared with shuffled sequences or your control sequences (sample output from sequences). See this Manual for more information.

Data Submission Form

Perform motif enrichment analysis in sequence datasets. The sequences may be in the DNA, RNA or protein alphabet, or in a custom alphabet.

Select the type of control sequences to use

Shuffled input sequences User-provided sequences

Select the sequence alphabet

Use sequences with a standard alphabet or specify a custom alphabet.

DNA, RNA or Protein Custom Choose file No file chosen

Input the sequences

Enter the sequences in which you want to measure motif enrichment.

Upload BED file DNA Improved

Specify the genome your BED file is based on.

UCSC Mammal Genomes

Human

hg19

Select the BED file to upload.

Choose file data_s1_H3K4me3.bed

Input the motifs

Select a motif database or enter the motifs you wish to test for enrichment.

Eukaryote DNA DNA

Human and Mouse (HOCOMOCO v11 FULL)

Job Details ...

Results

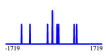
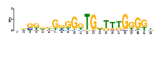
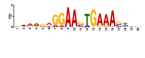
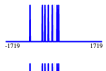
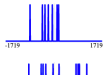
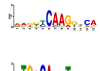
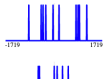
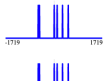
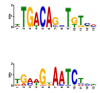
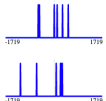
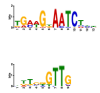
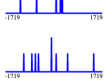
- SEA HTML output
- SEA TSV output
- SEA passing sequences
- (Primary) Uploaded BED file
- (Primary) Sequences from BED file
- Warning/Status Messages

Status Messages

- Parsing arguments
- Arguments ok
- Starting bed2fasta
  `bed2fasta -s -both -o data_s1_H3K4me3.bed.fa data_s1_H3K4me3.bed genome_db/UCSCMammal/hg19.fna`
- bed2fasta ran successfully in 0 seconds
- Starting sea
  `sea --verbosity 1 --oc . --thresh 10.0 --align center --p data_s1_H3K4me3.bed.fa --m motif_db/MOUSE/HOCOMOCOv11_full_MOUSE_mono_meme_format.meme --m motif_db/HUMAN/HOCOMOCOv11_full_HUMAN_mono_meme_format.meme`
- sea ran successfully in 13.24 seconds
- Done

ENRICHED MOTIFS

Found 12 motifs with *E*-values ≤ 10.

Logo	Database	ID	Alt ID	*P*-value	*E*-value	Q-value	TP	FP	Enrichment Ratio	Score Threshold	Positional Distribution	Matches per Sequence
	HOCOMOCOv11 full MOUSE mono meme format	IRF4_MOUSE.H11MO.0.A		1.09e-3	1.42e0	2.73e-1	9 / 10 (90.0%)	1 / 10 (10.0%)	5.00	9.58		
	HOCOMOCOv11 full HUMAN mono meme format	RREB1_HUMAN.H11MO.0.D		1.09e-3	1.42e0	2.73e-1	9 / 10 (90.0%)	1 / 10 (10.0%)	5.00	6.71		
	HOCOMOCOv11 full MOUSE mono meme format	RREB1_MOUSE.H11MO.0.D		1.09e-3	1.42e0	2.73e-1	9 / 10 (90.0%)	1 / 10 (10.0%)	5.00	6.71		
	HOCOMOCOv11 full HUMAN mono meme format	IRF8_HUMAN.H11MO.0.B		1.55e-3	2.01e0	2.73e-1	7 / 10 (70.0%)	0 / 10 (0.0%)	8.00	10.53		
	HOCOMOCOv11 full MOUSE mono meme format	IRF8_MOUSE.H11MO.0.A		1.55e-3	2.01e0	2.73e-1	7 / 10 (70.0%)	0 / 10 (0.0%)	8.00	10.53		
	HOCOMOCOv11 full MOUSE mono meme format	NR6A1_MOUSE.H11MO.0.D		4.64e-3	6.03e0	3.69e-1	10 / 10 (100.0%)	3 / 10 (30.0%)	2.75	9.07		
	HOCOMOCOv11 full HUMAN mono meme format	MEIS2_HUMAN.H11MO.0.B		5.42e-3	7.04e0	3.69e-1	6 / 10 (60.0%)	0 / 10 (0.0%)	7.00	8.58		
	HOCOMOCOv11 full MOUSE mono meme format	MEIS2_MOUSE.H11MO.0.B		5.42e-3	7.04e0	3.69e-1	6 / 10 (60.0%)	0 / 10 (0.0%)	7.00	8.58		
	HOCOMOCOv11 full HUMAN mono meme format	ZNF85_HUMAN.H11MO.1.C		5.42e-3	7.04e0	3.69e-1	6 / 10 (60.0%)	0 / 10 (0.0%)	7.00	10.32		
	HOCOMOCOv11 full HUMAN mono meme format	BRCA1_HUMAN.H11MO.0.D		5.47e-3	7.11e0	3.69e-1	9 / 10 (90.0%)	2 / 10 (20.0%)	3.33	9.89		