ChIP-Seq Practical 2

Jovan Xavier

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

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

```
## Bioconductor version '3.19' is out-of-date; the current release version '3.20'
##   is available with R version '4.4'; see https://bioconductor.org/install
```

```
BiocManager::install("ChIPseeker")
```

```
## Bioconductor version 3.19 (BiocManager 1.30.25), R 4.4.1 (2024-06-14)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'ChIPseeker'
```

```
## Old packages: 'clue', 'curl', 'dendextend', 'reticulate'
```

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

```
## Bioconductor version 3.19 (BiocManager 1.30.25), R 4.4.1 (2024-06-14)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'TxDb.Hsapiens.UCSC.hg19.knownGene'
```

```
## Old packages: 'clue', 'curl', 'dendextend', 'reticulate'
```

```
library(ChIPseeker)
```

```
##
```

```
## ChIPseeker v1.40.0
##
## If you use ChIPseeker in published research, please cite:
## Qianwen Wang, Ming Li, Tianzhi Wu, Li Zhan, Lin Li, Meijun Chen, Wenqin Xie, Zijing Xie, Erqiang Hu,
```

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

```
## Loading required package: GenomicFeatures
```

```
## Loading required package: BiocGenerics
```

```
##
## Attaching package: 'BiocGenerics'
```

```
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
```

```
## The following objects are masked from 'package:base':
##
##     anyDuplicated, aperm, append, as.data.frame, basename, cbind,
##     colnames, dirname, do.call, duplicated, eval, evalq, Filter, Find,
##     get, grep, grepl, intersect, is.unsorted, lapply, Map, mapply,
##     match, mget, order, paste, pmax, pmax.int, pmin, pmin.int,
##     Position, rank, rbind, Reduce, rownames, sapply, setdiff, table,
##     tapply, union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

## Loading required package: stats4

##
## Attaching package: 'S4Vectors'

## The following object is masked from 'package:utils':
##
##     findMatches

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

## Loading required package: GenomeInfoDb

## Loading required package: GenomicRanges

## Loading required package: AnnotationDbi

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.
```

(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.

```
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"
)
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]]
```

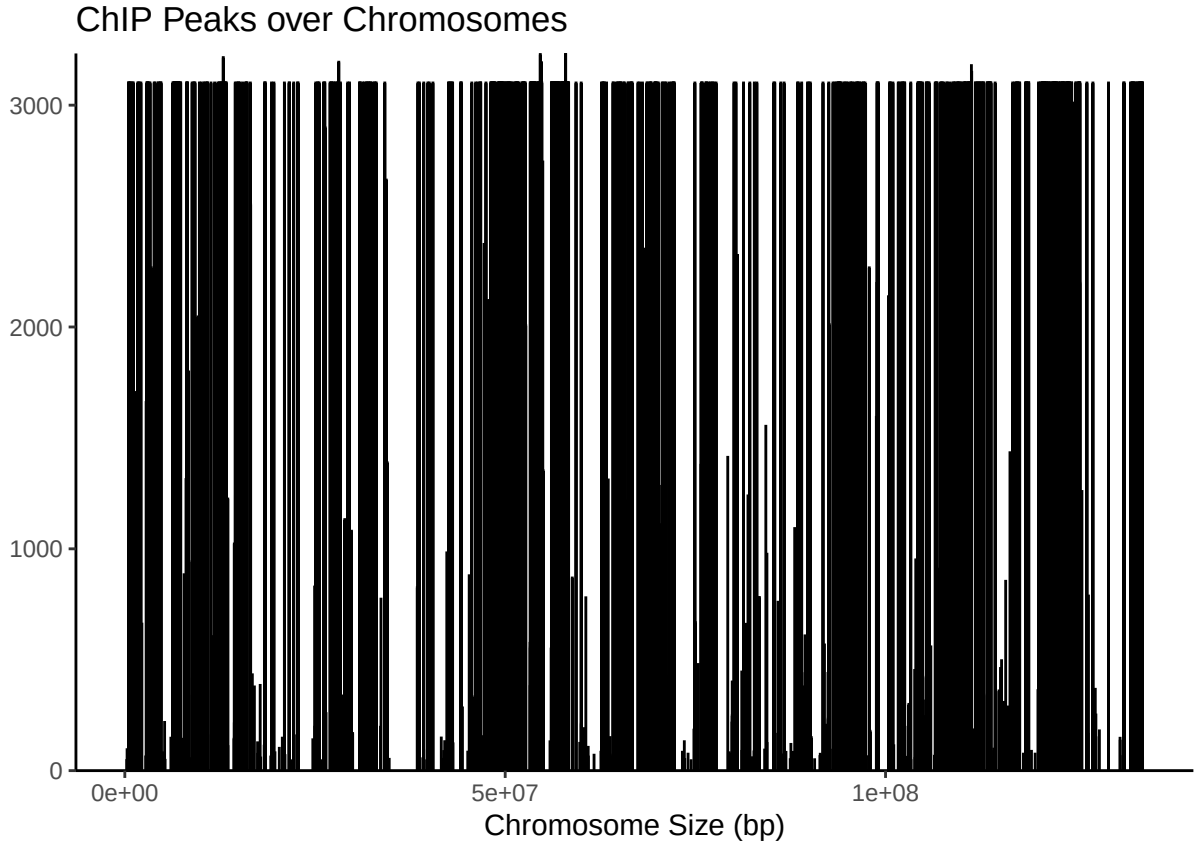
```
## [1] "GSM1574256_H3K4me3.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed"
```

```
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
```

(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

## 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")
```

```
## Bioconductor version 3.19 (BiocManager 1.30.25), R 4.4.1 (2024-06-14)

## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'clusterProfiler'

## Old packages: 'clue', 'curl', 'dendextend', 'reticulate'
```

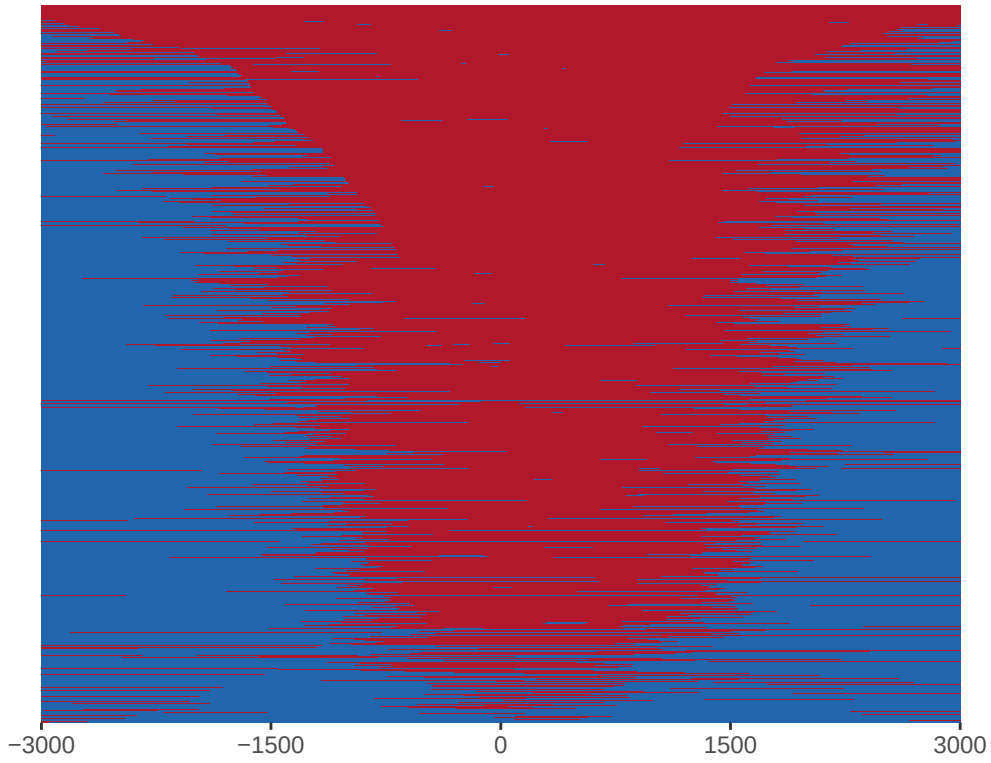
```
library(clusterProfiler)
```

```
## clusterProfiler v4.12.6 Learn more at https://yulab-smu.top/contribution-knowledge-mining/
##
## Please cite:
##
## G Yu. Thirteen years of clusterProfiler. The Innovation. 2024,
## 5(6):100722

##
## Attaching package: 'clusterProfiler'

## The following object is masked from 'package:AnnotationDbi':
##
##     select

## The following object is masked from 'package:IRanges':
##
##     slice

## The following object is masked from 'package:S4Vectors':
##
##     rename

## The following object is masked from 'package:stats':
##
##     filter
```

```
## 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)
```

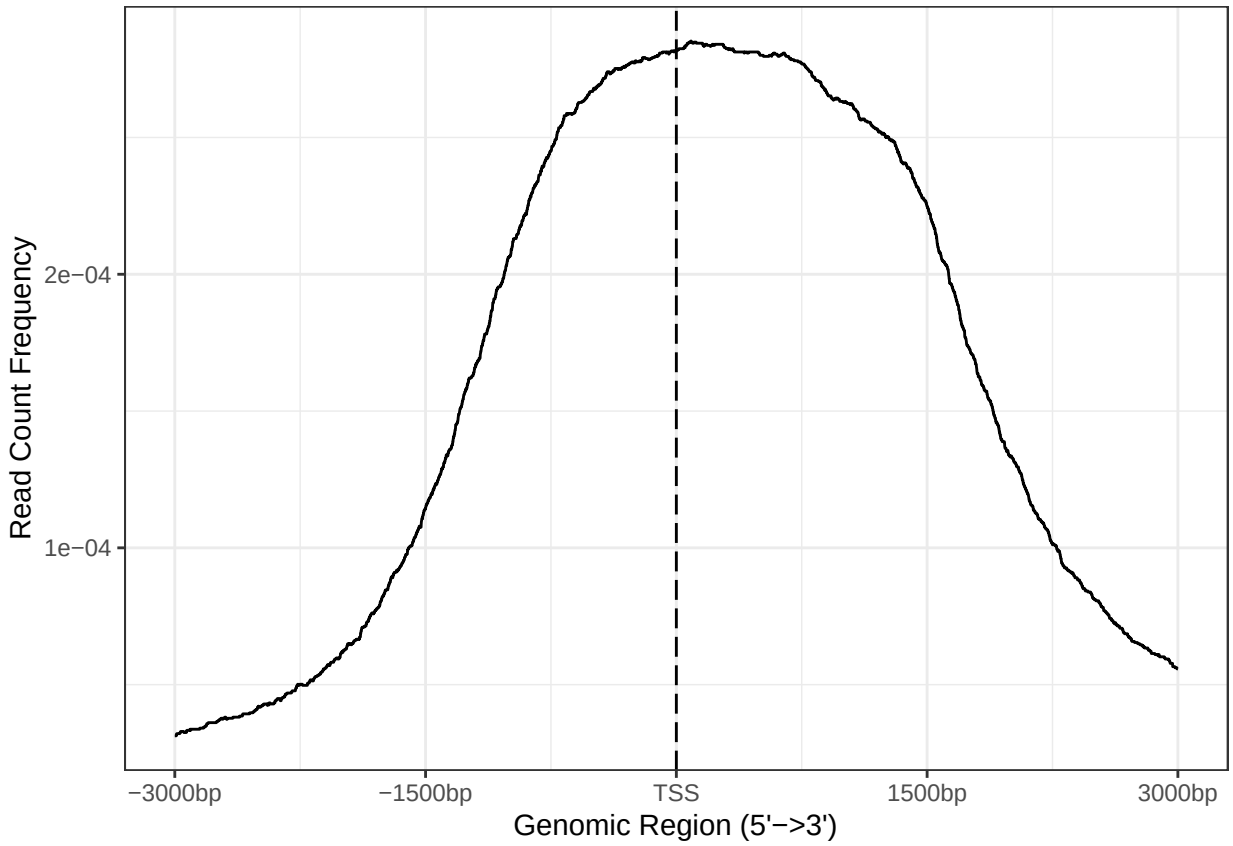
```
## >> preparing start_site regions by gene... 2024-11-15 13:15:52
## >> preparing tag matrix...  2024-11-15 13:15:52
```

```
##Heatmap of ChIP binding to TSS regions
tagHeatmap(tagMatrix)
```



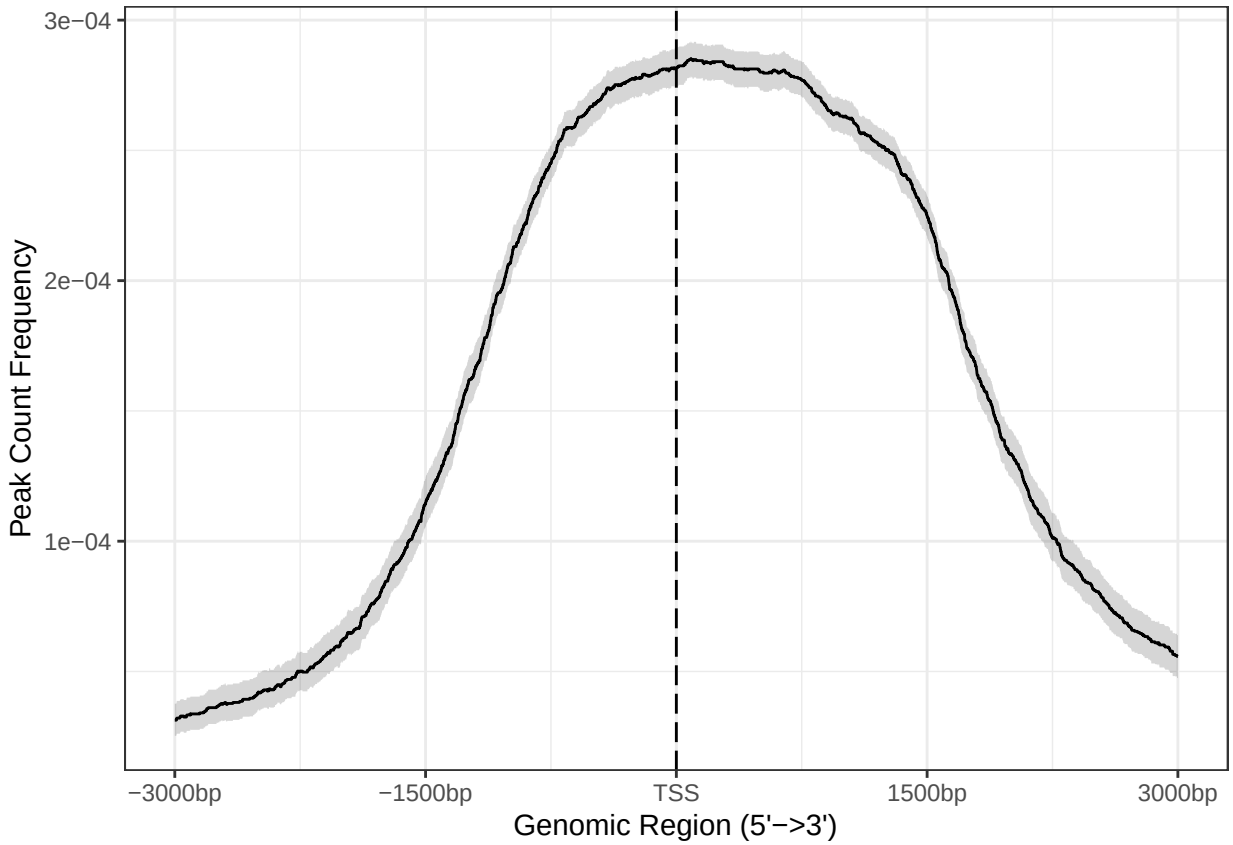

```
##Average Profile of ChIP peaks binding to TSS region
plotAvgProf(tagMatrix, xlim=c(-3000, 3000), xlab="Genomic Region (5'->3')", ylab = "Read Count Frequenc
```

```
## >> plotting figure...              2024-11-15 13:16:07
```




```
##Confidence interval estimated by bootstrap method
plotAvgProf(tagMatrix, xlim=c(-3000, 3000), conf = 0.95, resample = 1000)
```

```
## >> plotting figure...              2024-11-15 13:16:08
## >> Running bootstrapping for tag matrix...        2024-11-15 13:16:13
```

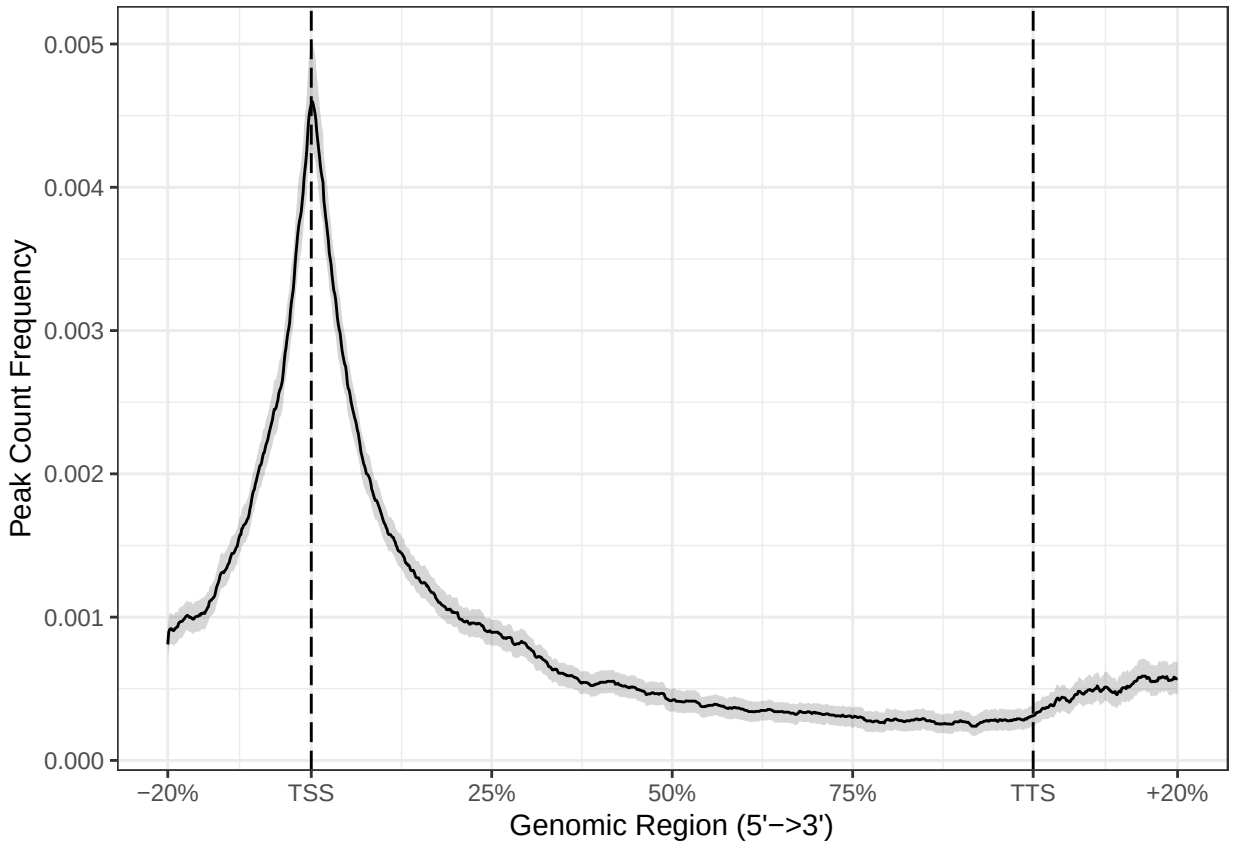



```
## be mindful if you do this for the whole chromosome, as this resampling to calculate 95% confidence i
```

(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 =
```

```
## >> binning method is used...2024-11-15 13:16:32
## >> preparing body regions by gene... 2024-11-15 13:16:32
## >> preparing tag matrix by binning...  2024-11-15 13:16:32
## >> preparing matrix with extension from (TSS-20%)~(TTS+20%)... 2024-11-15 13:16:32
## >> 12 peaks(1.423488%), having lengths smaller than 800bp, are filtered... 2024-11-15 13:16:33
## >> Running bootstrapping for tag matrix...        2024-11-15 13:16:45
```

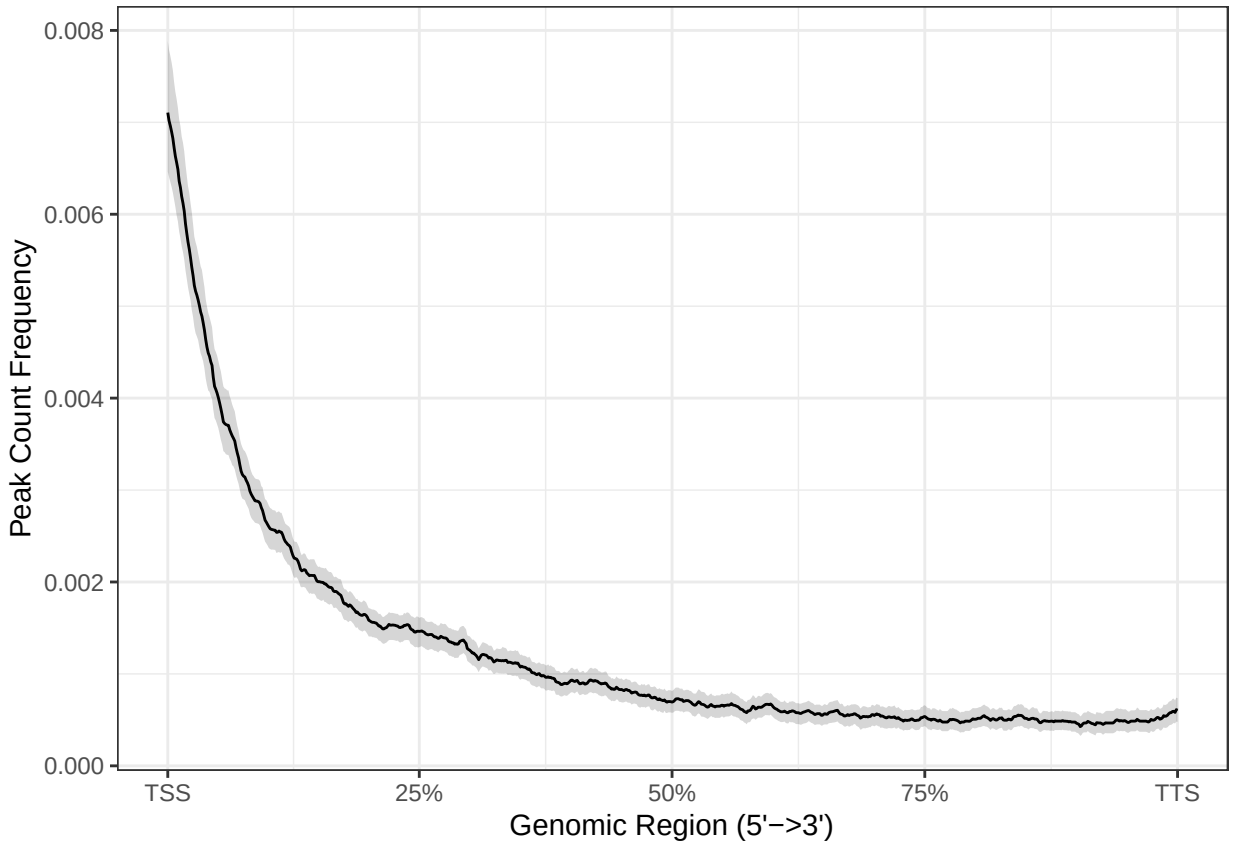



```
## we can also use getBioRegion(), getTagMatrix() and plotPeakProf() to plot in three steps.
genebody <- getBioRegion(TxDb = txdb19, by = "gene", type = "body")
matrix_no_flankextension <- getTagMatrix(peak,windows = genebody, nbin = 800)
```

```
## >> binning method is used...2024-11-15 13:16:45
## >> preparing body regions by gene... 2024-11-15 13:16:45
## >> preparing tag matrix by binning...  2024-11-15 13:16:45
## >> preparing matrix for body region with no flank extension... 2024-11-15 13:16:45
## >> 12 peaks(1.494396%), having lengths smaller than 800bp, are filtered... 2024-11-15 13:16:45
```

```
plotPeakProf(matrix_no_flankextension,conf = 0.95)
```

```
## >> Running bootstrapping for tag matrix...        2024-11-15 13:16:53
```

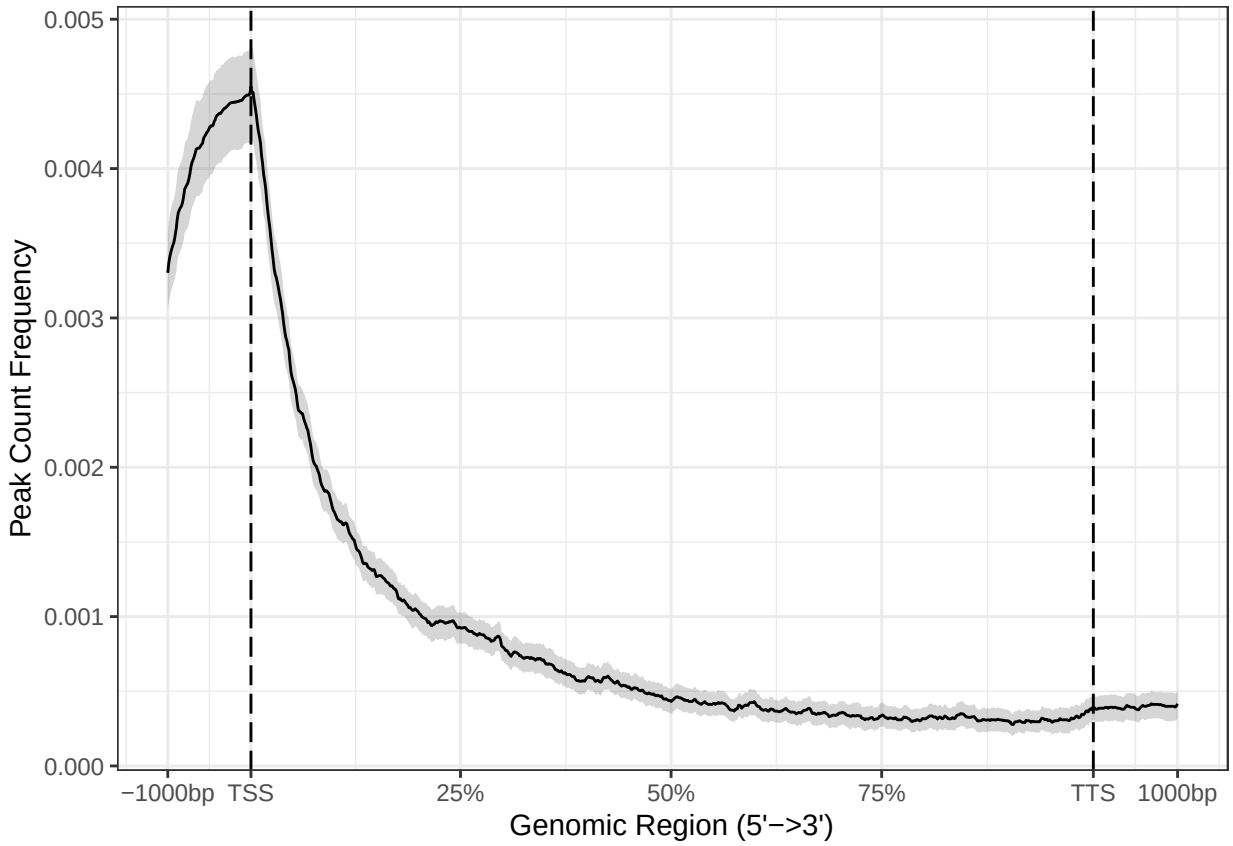



```
#### you can also use getTagMatrix() and plotPeakProf() to plot in two steps
matrix_actual_extension <- getTagMatrix(peak,windows = genebody, nbin = 800, upstream = 1000,downstream
```

```
## >> binning method is used...2024-11-15 13:16:53
## >> preparing body regions by gene... 2024-11-15 13:16:53
## >> preparing tag matrix by binning...  2024-11-15 13:16:53
## >> preparing matrix with flank extension from (TSS-1000bp)~(TTS+1000bp)... 2024-11-15 13:16:53
## >> 15 peaks(1.838235%), having lengths smaller than 800bp, are filtered... 2024-11-15 13:16:53
```

```
plotPeakProf(matrix_actual_extension,conf = 0.95)
```

```
## >> Running bootstrapping for tag matrix...        2024-11-15 13:17:02
```




```
## see the manual to plot against other regions
```

(6) Peak annotation

```
## Since some annotation may overlap, ChIPseeker adopted the following priority in genomic annotation: P
## install Bioconductor R package ("org.Hs.eg.db")


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

```
## Bioconductor version 3.19 (BiocManager 1.30.25), R 4.4.1 (2024-06-14)
```

```
## Warning: package(s) not installed when version(s) same as or greater than current; use
##   `force = TRUE` to re-install: 'org.Hs.eg.db'
```

```
## Old packages: 'clue', 'curl', 'dendextend', 'reticulate'
```

```
library(org.Hs.eg.db)
```

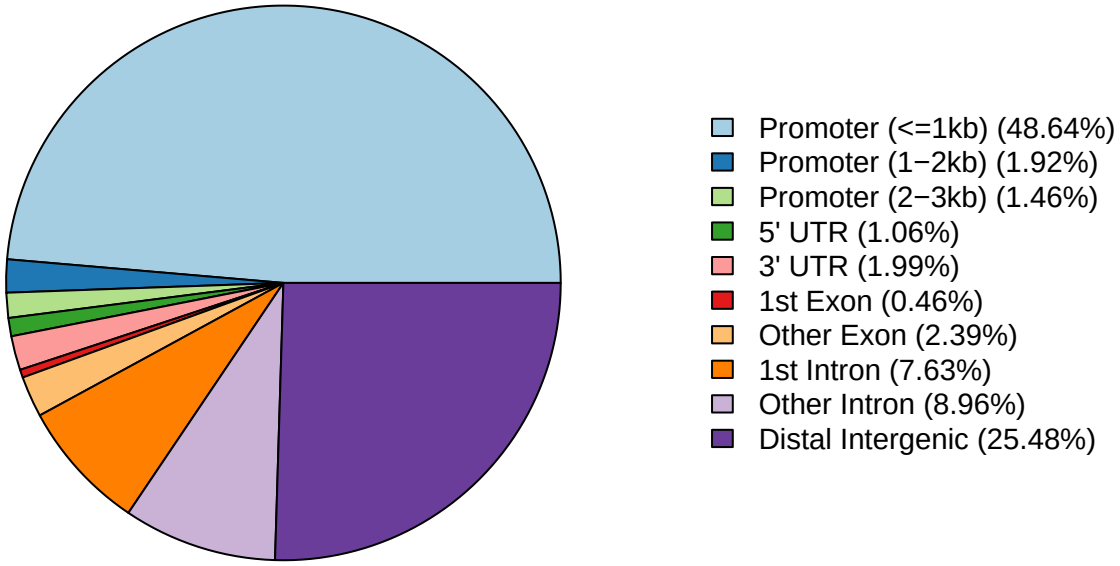
```
##
```

```
peakAnno <- annotatePeak(peak, tssRegion=c(-3000, 3000), TxDb=txdb19, annoDb="org.Hs.eg.db")
```

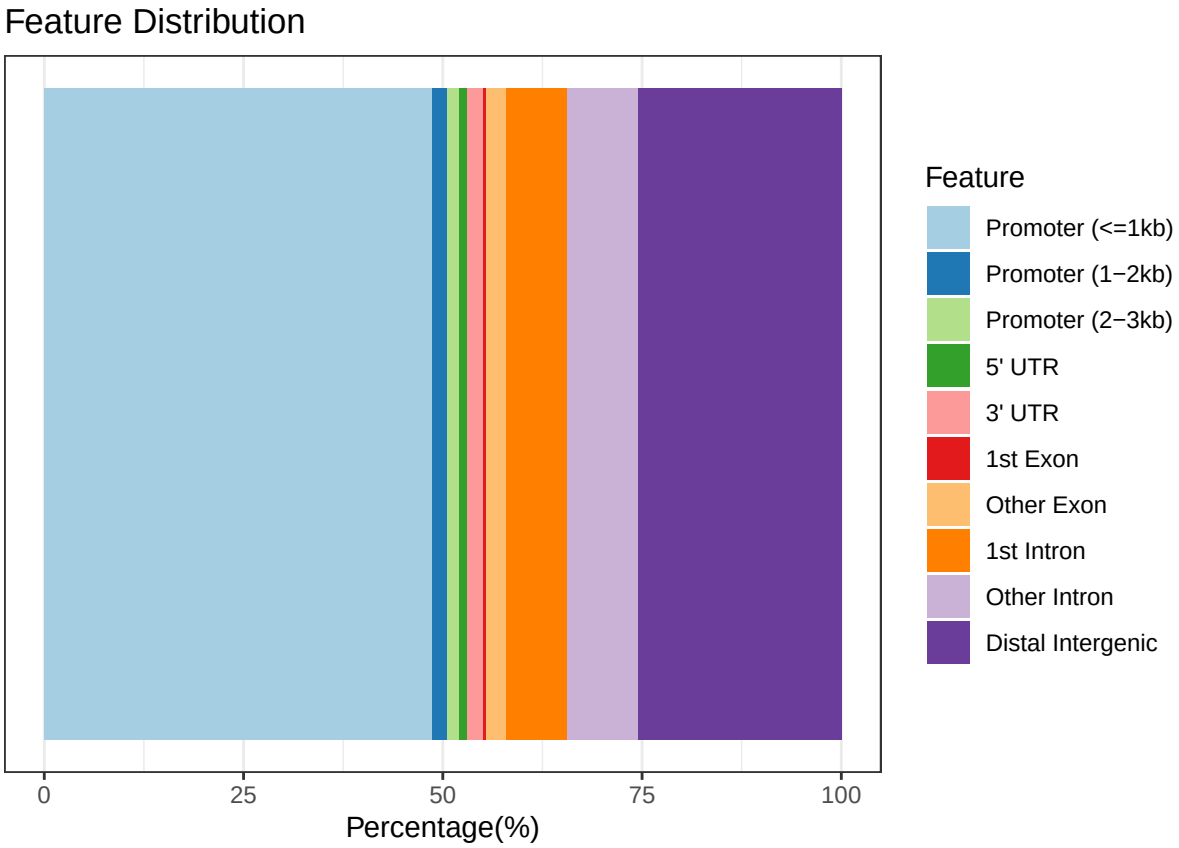
```
## >> preparing features information...      2024-11-15 13:17:04
## >> identifying nearest features...        2024-11-15 13:17:04
## >> calculating distance from peak to TSS...   2024-11-15 13:17:04
## >> assigning genomic annotation...        2024-11-15 13:17:04
```

```
## >> adding gene annotation...           2024-11-15 13:17:13
## 'select()' returned 1:many mapping between keys and columns
## >> assigning chromosome lengths           2024-11-15 13:17:13
## >> done...                    2024-11-15 13:17:13
```

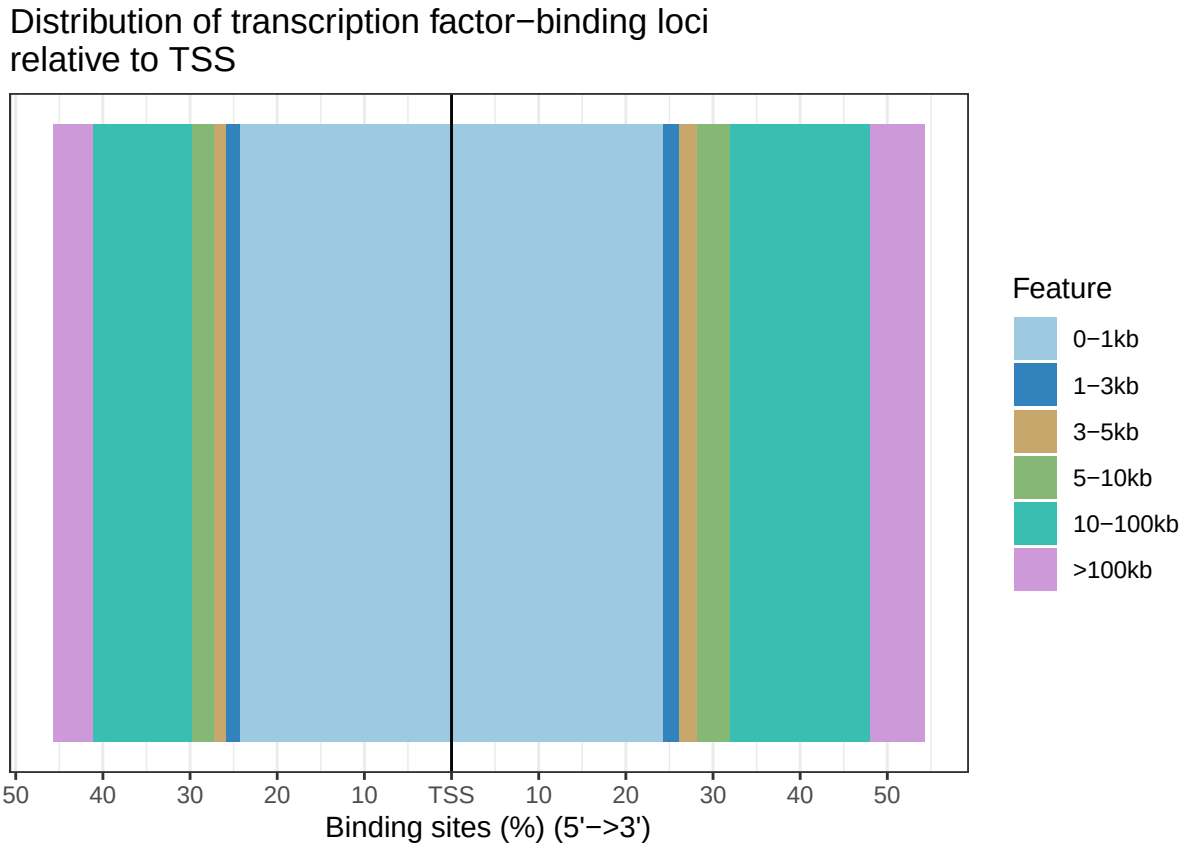
```
plotAnnoPie(peakAnno)
```




```
## Note that we use H3K4me3 peaks to annotate in the example, and H3K4me3 marks the active promoter reg
## This pie chart is very different from the one in the lecture slides where H3K27ac peaks were used, w

plotAnnoBar(peakAnno)
```




```
## plotDistToTSS to calculate the percentage of binding sites upstream and downstream from the TSS of t

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

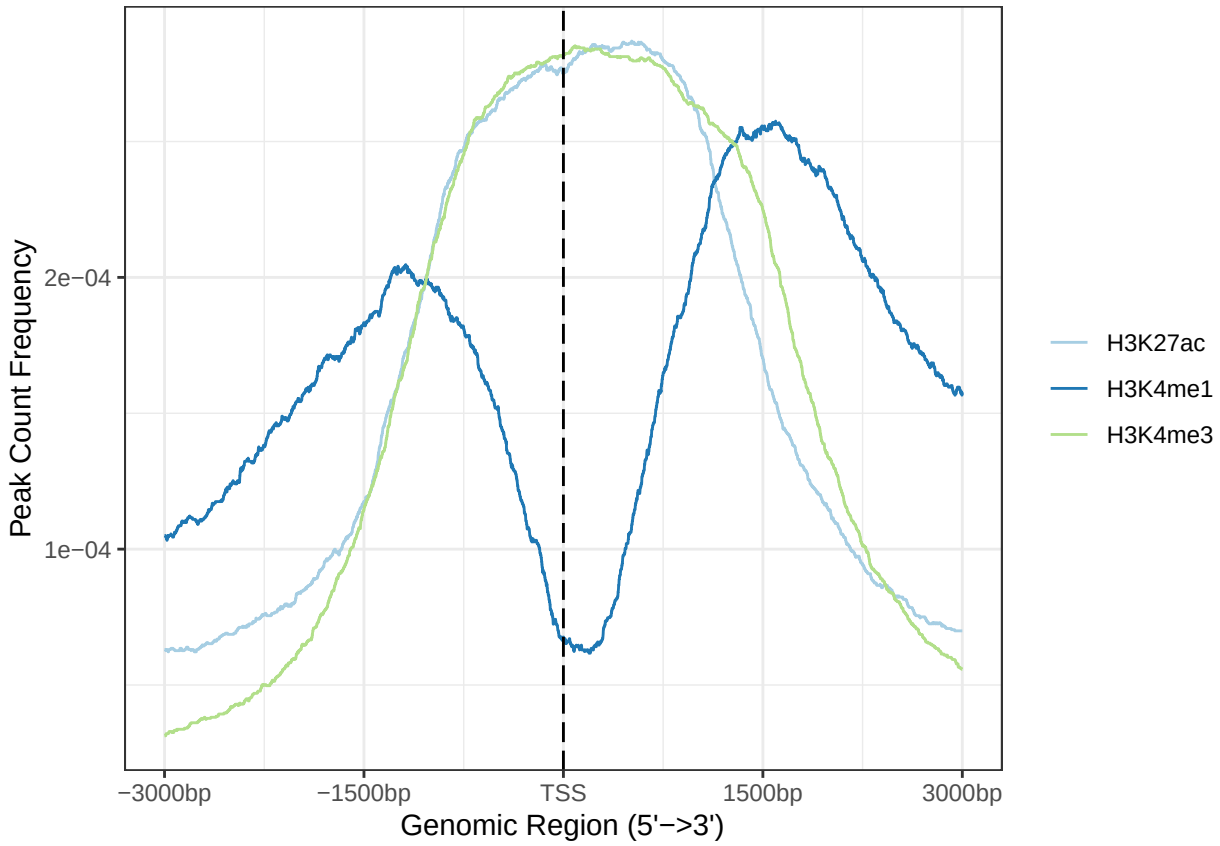
Now we use the three peak files "GSM1574235_H3K27ac.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed", "GSM1574242_H3K4me1.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed", "GSM1574256_H3K4me3.CAPAN1_vs_ and compare peaks between them

```
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 gene... 2024-11-15 13:17:14
## >> preparing tag matrix...  2024-11-15 13:17:14
## >> preparing start_site regions by gene... 2024-11-15 13:17:15
## >> preparing tag matrix...  2024-11-15 13:17:15
## >> preparing start_site regions by gene... 2024-11-15 13:17:15
## >> preparing tag matrix...  2024-11-15 13:17:15
```

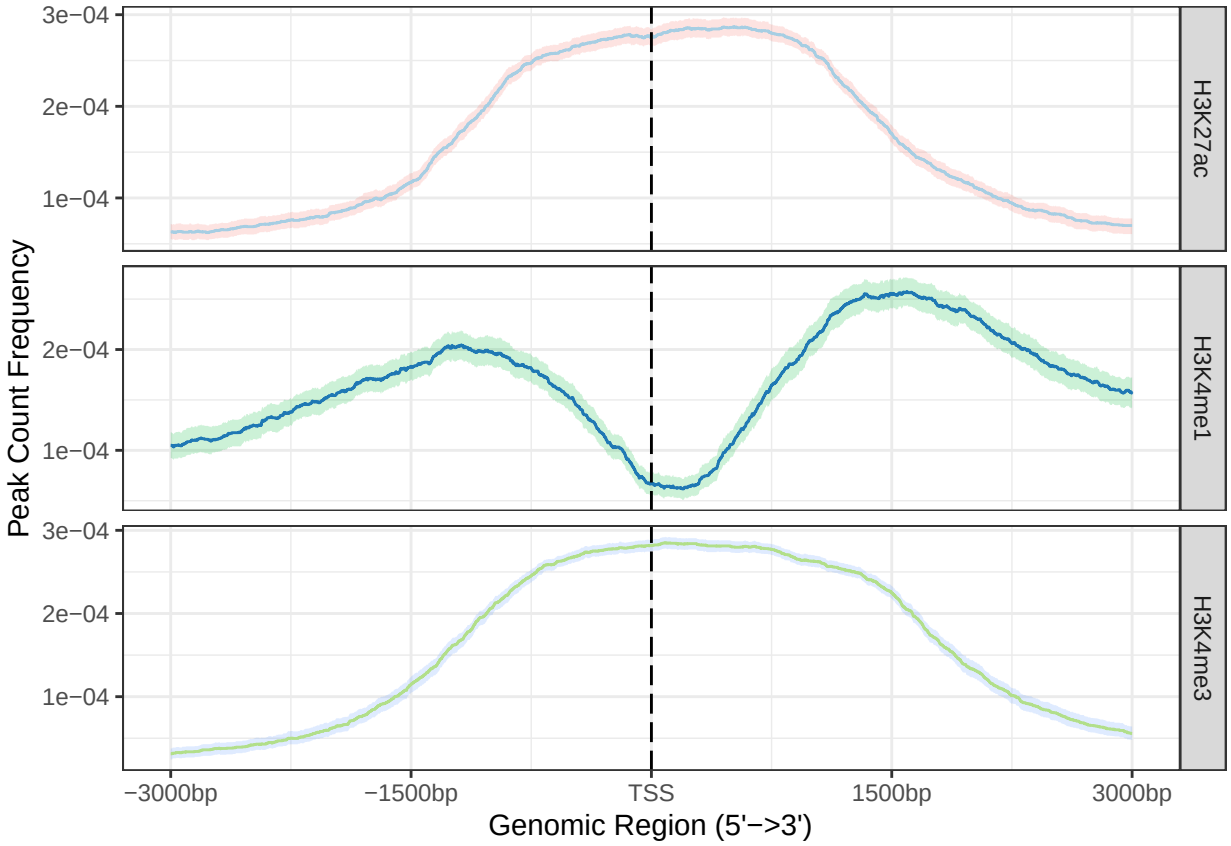
```
## we will compare the peaks among the three files in relation to promoters
## average profile of the 3 bed files against promoters
plotAvgProf(tagMatrixList, xlim=c(-3000, 3000))
```

```
## >> plotting figure...              2024-11-15 13:17:15
```

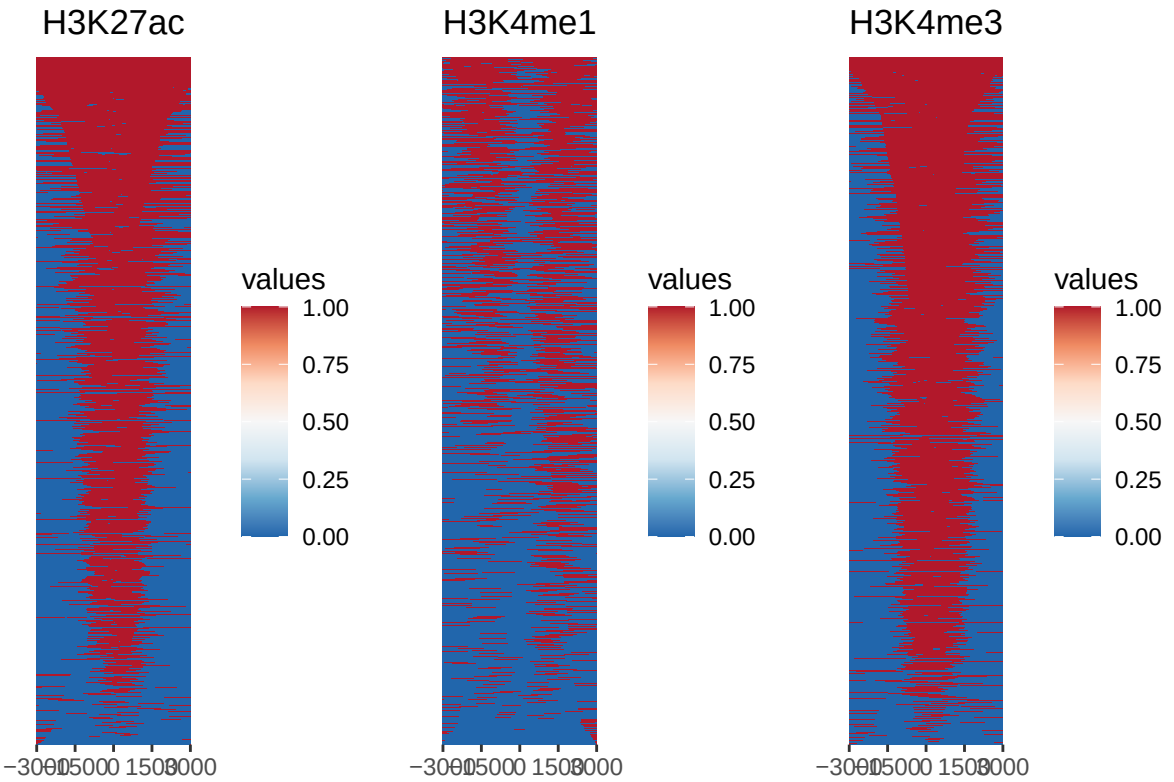



```
plotAvgProf(tagMatrixList, xlim=c(-3000, 3000), conf=0.95,resample=500, facet="row")
```

```
## >> plotting figure...              2024-11-15 13:17:15
## >> Running bootstrapping for tag matrix...        2024-11-15 13:17:18
## >> Running bootstrapping for tag matrix...        2024-11-15 13:17:23
## >> Running bootstrapping for tag matrix...        2024-11-15 13:17:26
```

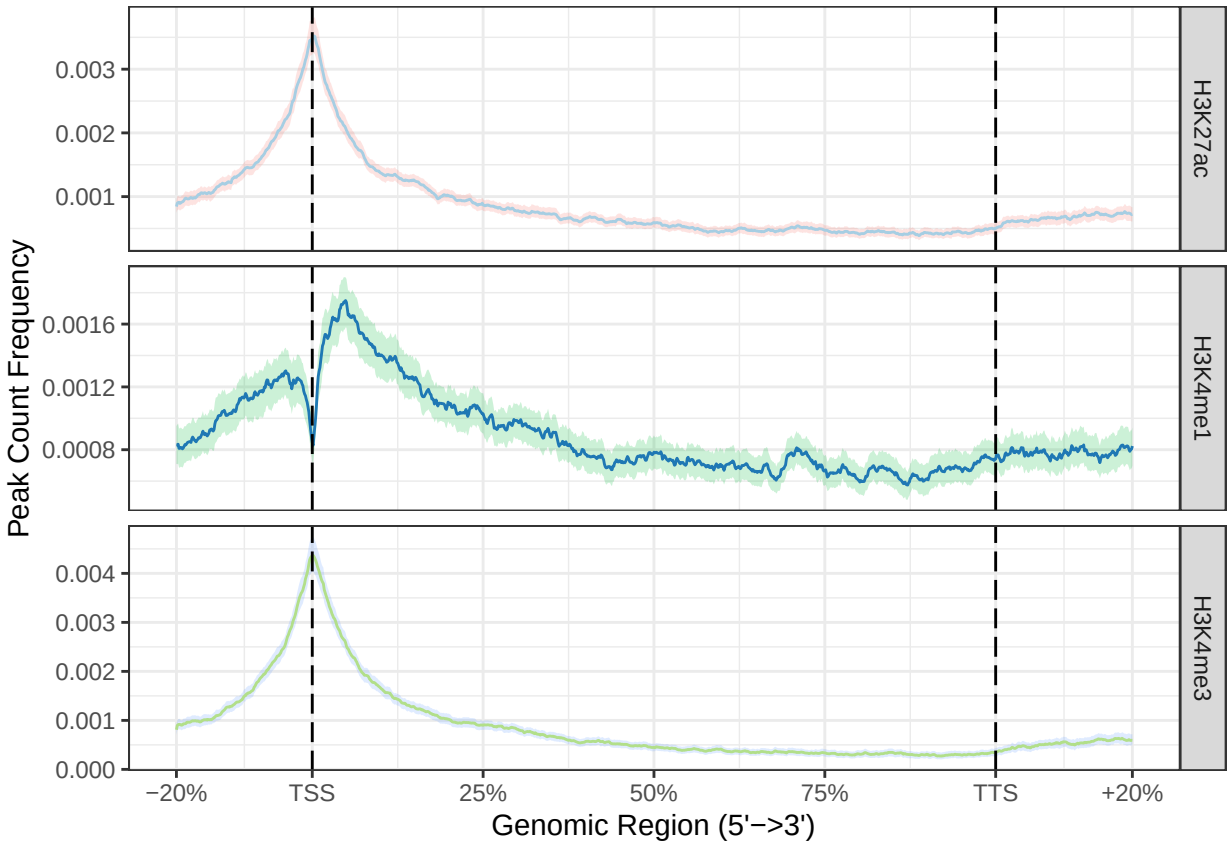



```
### peak heatmap across samples
## you can clear all previous plot using the following command
# dev.off()
tagHeatmap(tagMatrixList)
```

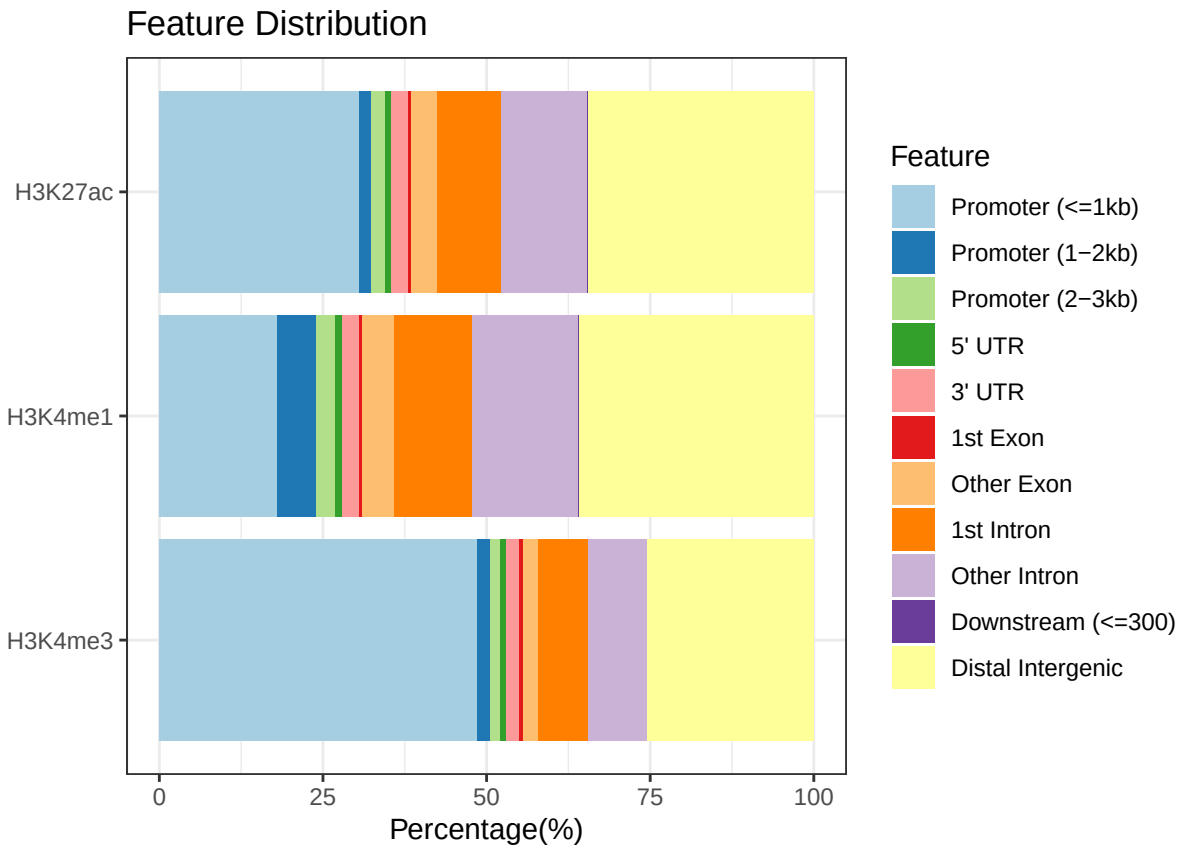



```
### Profile of the three ChIP peak files binding to body region
plotPeakProf2(files, upstream = rel(0.2), downstream = rel(0.2), conf = 0.95, by = "gene", type = "body
```

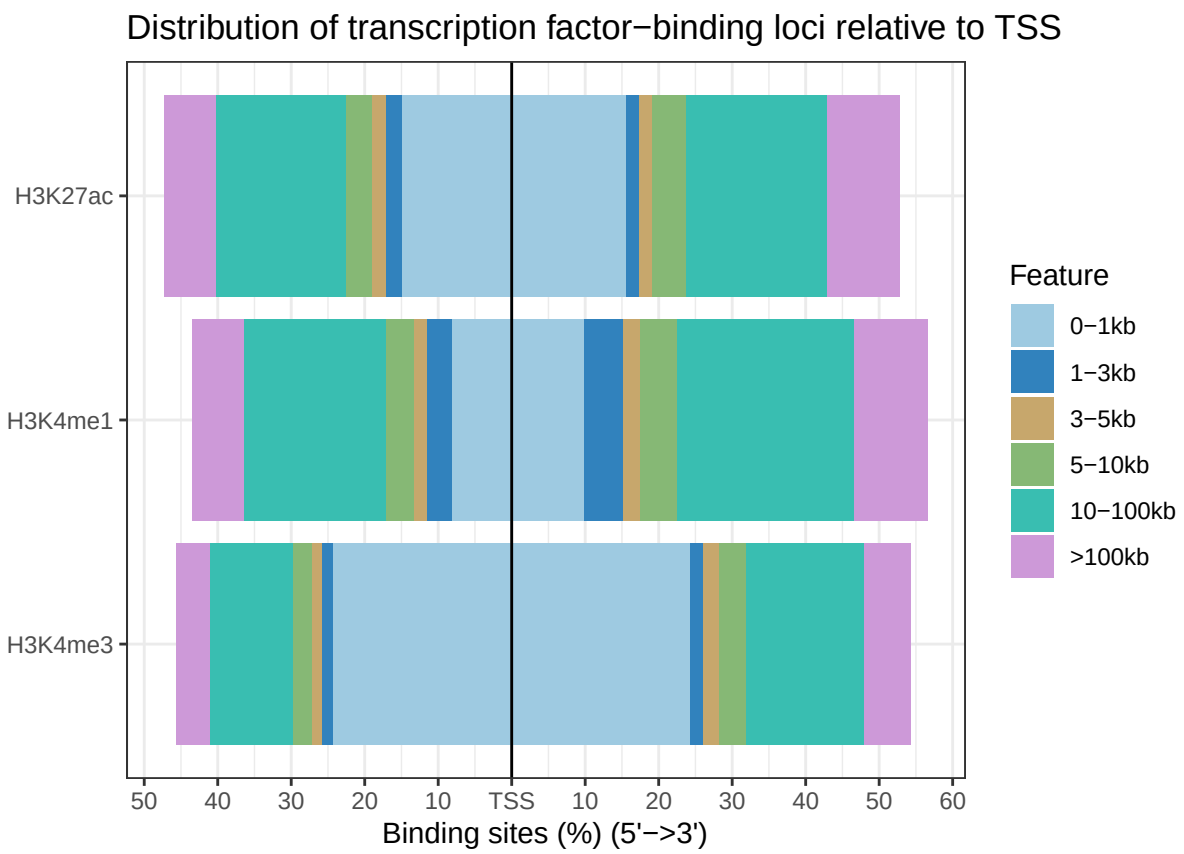
```
## >> binning method is used...2024-11-15 13:17:56
## >> preparing body regions by gene... 2024-11-15 13:17:56
## >> preparing tag matrix by binning...  2024-11-15 13:17:56
## >> preparing matrix with extension from (TSS-20%)~(TTS+20%)... 2024-11-15 13:17:56
## >> 17 peaks(1.958525%), having lengths smaller than 800bp, are filtered... 2024-11-15 13:17:57
## >> binning method is used...2024-11-15 13:18:08
## >> preparing body regions by gene... 2024-11-15 13:18:08
## >> preparing tag matrix by binning...  2024-11-15 13:18:08
## >> preparing matrix with extension from (TSS-20%)~(TTS+20%)... 2024-11-15 13:18:08
## >> 14 peaks(1.605505%), having lengths smaller than 800bp, are filtered... 2024-11-15 13:18:08
## >> binning method is used...2024-11-15 13:18:19
## >> preparing body regions by gene... 2024-11-15 13:18:19
## >> preparing tag matrix by binning...  2024-11-15 13:18:19
## >> preparing matrix with extension from (TSS-20%)~(TTS+20%)... 2024-11-15 13:18:19
## >> 12 peaks(1.423488%), having lengths smaller than 800bp, are filtered... 2024-11-15 13:18:19
## >> Running bootstrapping for tag matrix...        2024-11-15 13:18:32
## >> Running bootstrapping for tag matrix...        2024-11-15 13:18:33
## >> Running bootstrapping for tag matrix...        2024-11-15 13:18:34
```




```
### 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)
```

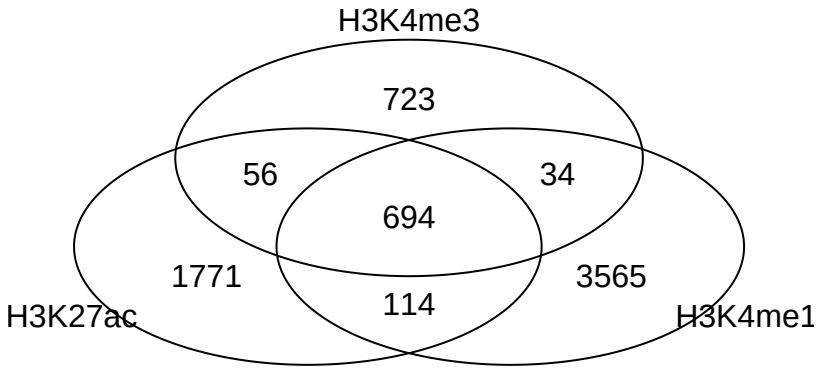



```
## plotDistToTSS to compare distance to TSS profiles among ChIPseq data.
plotDistToTSS(peakAnnoList)
```



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

```
## GRanges object with 2 ranges and 0 metadata columns:
```

```
##       seqnames            ranges strand
##          <Rle>         <IRanges>  <Rle>
##   [1]    chr12 98217245-98217294      *
##   [2]    chr12 93690257-93690287      *
##   -------
##   seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

```
enrichPeakOverlap(queryPeak = files[[3]], targetPeak = unlist(files[1:2]), TxDb = txdb, pAdjustMethod =
```

```
##                                                                  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
## H3K27ac GSM1574235_H3K27ac.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed 1507
## H3K4me1 GSM1574242_H3K4me1.CAPAN1_vs_INPUT.CAPAN1_peaks_hg19_chr12.bed 1507
##         tLen N_OL     pvalue   p.adjust
## H3K27ac 2635 1266 0.01960784 0.01960784
## H3K4me1 4407 1370 0.01960784 0.01960784
```

```
## this command compares H3K4me3 peals against H3K27ac and H3K4me1 peaks, and determine if there is a s
```

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
```

```
##         V1        V2        V3             V4      V5
## 142  chr12   9435427   9436417 MACS_peak_5940  308.03
## 51   chr12   3841458   3842476 MACS_peak_5849   87.23
## 720  chr12  58334036  58336933 MACS_peak_6518 3100.00
## 730  chr12  60492541  60494268 MACS_peak_6528  173.20
## 1244 chr12 116068173 116069338 MACS_peak_7042  139.94
## 664  chr12  56914843  56918281 MACS_peak_6462 3100.00
## 826  chr12  69890963  69891924 MACS_peak_6624   69.35
## 605  chr12  54426167  54428690 MACS_peak_6403 3100.00
## 587  chr12  54088501  54090917 MACS_peak_6385 1627.15
## 1376 chr12 123754071 123755903 MACS_peak_7174 3100.00
```

```
## 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

(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.

(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