Analyzing ChIP-Seq data using Bioconductor

Ghada Sharif

Isoform-specific function of AIB1 during breast cancer progression

- AIB1: Amplified in Breast Cancer 1, transcription coactivator
- High grade tumors, Poor prognosis, Therapy Resistance
- Two isoforms:

1) AIB1-FL
2) AIB1-D4

- Expression of both isoforms goes up during progression

Experimental Design

- CRISPR engineered breast cancer cell lines that express either one of AIB1 isoforms

- ChIP-Seq: crosslink and pull down AIB1 with DNA, then DNA sequencing to identify difference and similarities.

- Used Bioconductor packages: ChIPseeker and ChIPpeakAnno.

```{r setup, include=TRUE}
knitr::opts_chunk$set(echo = FALSE, message=FALSE, warning=FALSE)

# Installing packages from Bioconductor, showing an example below

# if (!requireNamespace("BiocManager", quietly = TRUE))
# install.packages("BiocManager")
# BiocManager::install("ChIPseeker")

library(AnnotationDbi)
library(BiocManager)
library(Biostrings)
library(BSgenome)
library(plyr)
library(dplyr)
library(GenomicAlignments)
library(GenomicFeatures)
library(ggplot2)
library(knitr)
library(limma)
library(org.Hs.eg.db)
library(seqRFLP)
library(tibble)
library(wesanderson)
library(ChIPseeker)
library(vennplot)
library(clusterProfiler)
library(TxDb.Hsapiens.UCSC.hg38.knownGene)
library(ChIPpeakAnno)
library(tidyverse)
library(EnsDb.Hsapiens.v86)


# save human genome version hg38 with a shorter name
txdb <- TxDb.Hsapiens.UCSC.hg38.knownGene

# Converting genes into genomic ranges
columns(EnsDb.Hsapiens.v86)
TxGR<-toGRanges(EnsDb.Hsapiens.v86, feature='gene')

```

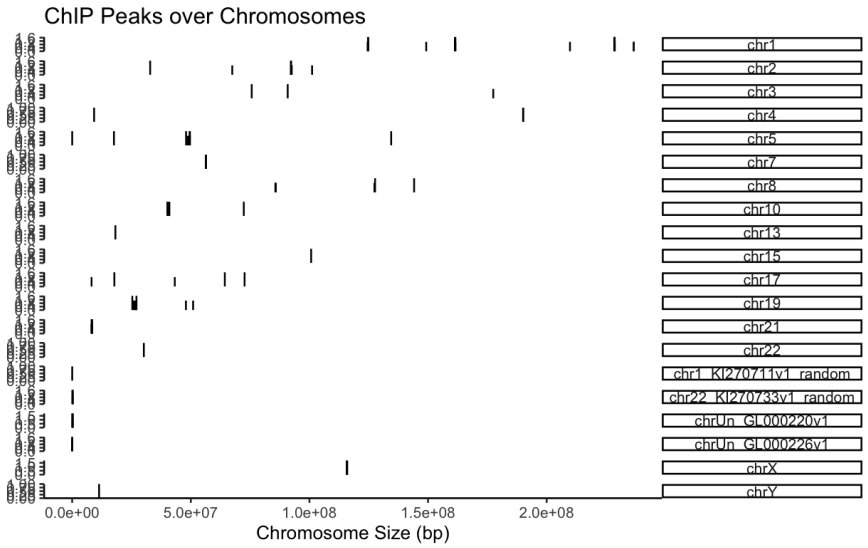
DNA sequences were aligned to the genome then peaks called using MACS2

```
# After algining to the genome, sequencing peaks are shortened to 250bp

aDCIS_all<-("./aDCIS_all.bed")
aDCIS_all_peak<-readPeakFile(aDCIS_all)
aDCIS_all_peak_250<-resize(aDCIS_all_peak,250,fix="center")
export.bed(aDCIS_all_peak_250,"./aDCIS_peak_250")

aD10_all<-("./aD10_all.bed")
aD10_all_peak<-readPeakFile(aD10_all)
aD10_all_peak_250<-resize(aD10_all_peak,250,fix="center")
export.bed(aD10_all_peak_250,"./aD10_peak_250")


```

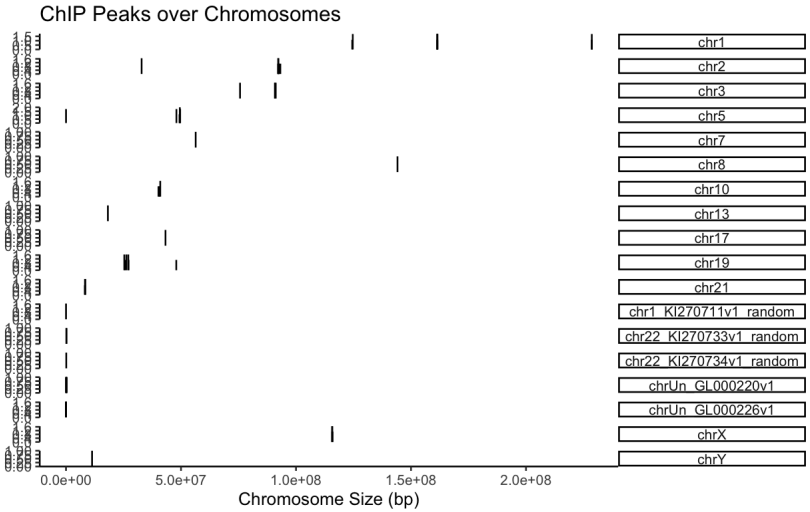
To visualize peaks on chromosomes

```{r}

ChIPseeker::covplot(aDCIS_all_peak)

ChIPseeker::covplot(aD10_all_peak)

```

AIB1-FL

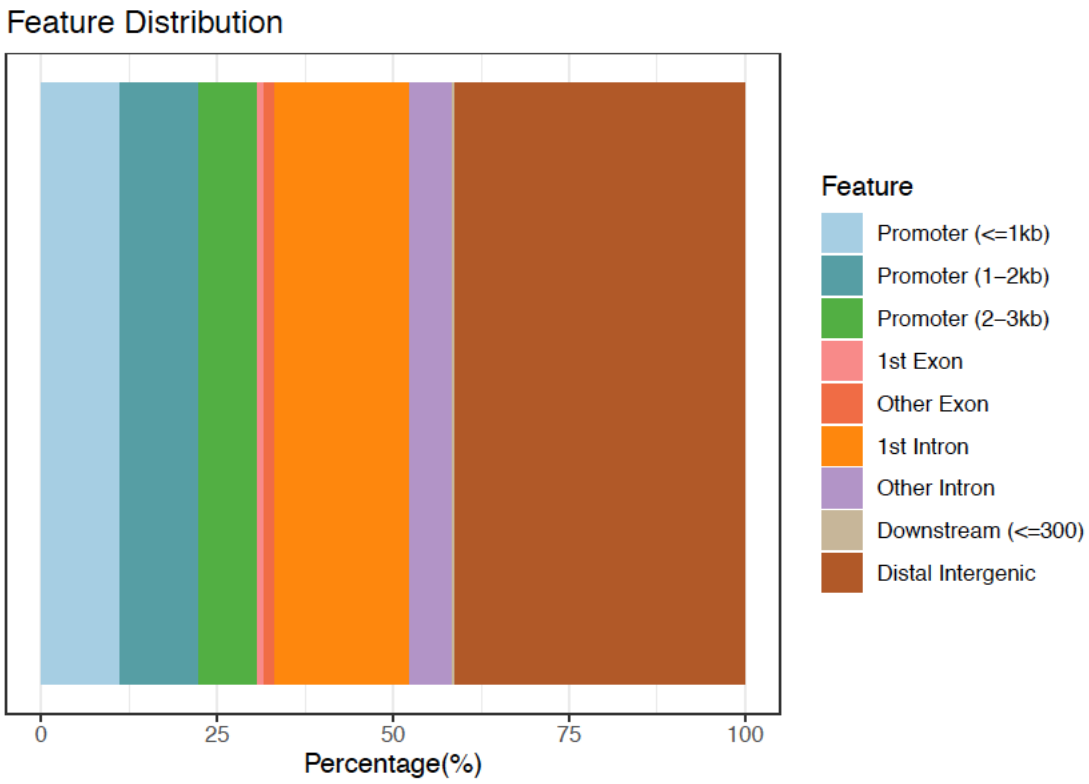


AIB1-D4



Distribution of peaks relative to gene features

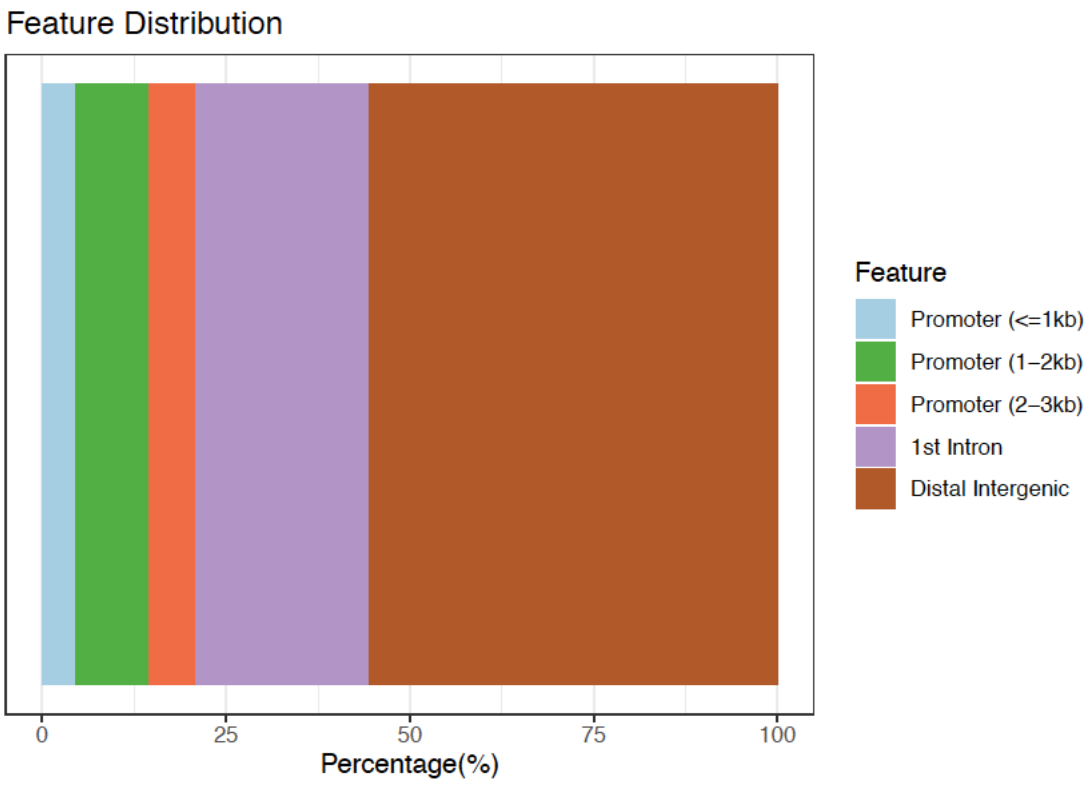
```
peakAnno_DCIS <- annotatePeak(aDCIS_all_peak, tssRegion=c(-3000, 3000),
                              TxDb=txdb, annoDb="org.Hs.eg.db")
plotAnnoBar(peakAnno_DCIS)
```



Distribution of peaks relative to gene features

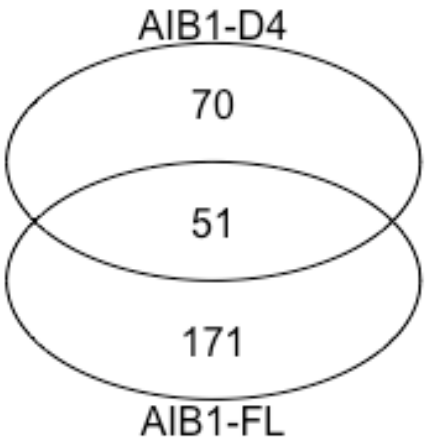
```
peakAnno_D10 <- annotatePeak(aD10_all_peak, tssRegion=c(-3000, 3000),
                             TxDb=txdb, annoDb="org.Hs.eg.db")
```

```
plotAnnoBar(peakAnno_D10)
```



To determine the overlap of peaks between the two isoforms

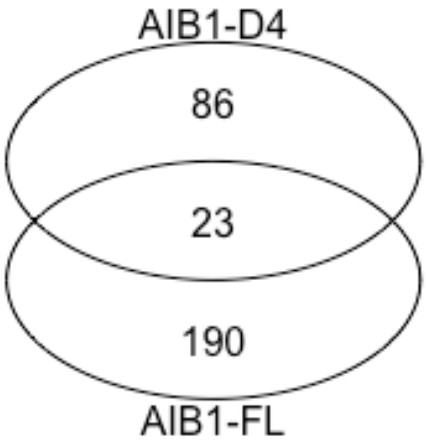
```{r}

# Determine the overlap of peaks between the two isoforms

aDCIS_short<-("./aDCIS_peak_250")
aDCIS_short_peak<-readPeakFile(aDCIS_short)
aD10_short<-("./aD10_peak_250")
aD10_short_peak<-readPeakFile(aD10_short)
aDCIS_comp_list<-list(aDCIS_short_peak,aD10_short_peak)
names(aDCIS_comp_list)<-c("AIB1-FL","AIB1-D4")
ChIPseeker::vennplot(aDCIS_comp_list)

# Add statistics to the overlap

enrichPeakOverlap(queryPeak= aDCIS_short,targetPeak= aD10_short, TxDb= txdb,pAdjustMethod = "BH",nShuffle= 1000,chainFile= NULL,verbose= FALSE)

```




```
##        qSample      tSample qLen tLen N_OL     pvalue    p.adjust
## 1 aDCIS_peak_250 aD10_peak_250  222  121   51 0.000999001 0.000999001
```

To annotate identified peaks to genes and check overlap

```{r}

peakAnnoList <- lapply(aDCIS_comp_list, annotatePeak, TxDb=txdb,tssRegion=c(-3000, 3000), verbose=FALSE)
genes= lapply(peakAnnoList, function(i) as.data.frame(i)$geneId)
ChIPseeker::vennplot(genes)
```



Converting peak files to Genomic Ranges, overlap and determine distribution from TSS

```
GR_aDCIS<-toGRanges(data = '../Projects/aDCIS_1.macs2_peaks.narrowPeak')
GR_aD10<-toGRanges(data = '../Projects/aD10_1.macs2_peaks.narrowPeak')
GR_aDCIS %>% head
```

```
overlap <- findOverlapsOfPeaks(GR_aDCIS, GR_aD10)
```

```
overlap <- addMetadata(overlap, colNames="score", FUN=mean)

pk_anno<-annotatePeakInBatch(myPeakList = overlap$peaklist[['GR_aDCIS///GR_aD10']],
                             AnnotationData = TxGR,
                             PeakLocForDistance = 'middle',
                             FeatureLocForDistance = 'start')
pk_anno$'.mcols[, colNames]' %>% which.max()
```

```
ggplot(data = tibble(dist = pk_anno$distancetoFeature), aes(x = dist))+
  theme_bw()+
  scale_x_continuous(name = 'Distance from TSS (bp)', limits = c(-5e3, 5e3))+
  scale_y_continuous(name = 'Density')+
  geom_vline(xintercept = 0, color = 'red')+
  geom_density(alpha = 0.1, fill = 'blue')
```