Казачкова Анна, БЭК 181

Данные:

организм (human/mouse, mm10)	DNA_structure	histone_mark	Тип клеток	гистон. метка (.bed файл 1)	гистон. метка (.bed файл 2)
Human (hg19)	ZDNA_DeepZ	H3K36me3	H9	ENCFF830UKK	ENCFF267AKB

Скачивание данных

Используя кластер, скачиваем данные по выбранным параметрам для двух экспериментов.

Создаем рабочую папку и переходим в нее

mkdir -p ~/_my/DataLog/0513.HSE.minor

cd ~/_my/DataLog/0513.HSE.minor

Скачиваем данные с помощью команды wget, ссылки на свои данные берем из общей гугл-таблицы

Данные изначально были скачаны сборки h38, поэтому надо будет воспользоваться программой liftOver

mkdir ~/bin/

cd ~/bin/

wget http://hgdownload.soe.ucsc.edu/admin/exe/linux.x86_64/liftOver

chmod a+x liftOver

source ~/.profile

liftOver H3K36me3_H9.ENCFF830UKK.h38.bed map.chain H3K36me3_H9.ENCFF830UKK.h19.bed unmapped

И то же самое для второго файла

Chain-file из wget https://hgdownload.cse.ucsc.edu/goldenpath/hg38/liftOver/hg38ToHg19.over.chain.gz

В итоге получили файлы для дальнейшей работы: H3K36me3_H9.ENCFF267AKB.h38.bed и H3K36me3_H9.ENCFF830UKK.h38.bed

Визуализация распределения длин в R

Смотрим, какие длины получаются в файлах

Код для гистограммы длин пиков:

```
library(ggplot2)
library(dplyr)
# library(tidyr)   # replace_na
# library(tibble)  # column_to_rownames

###

#NAME <- 'H3K36me3_H9.intersect_with_DeepZ'
#NAME <- 'DeepZ'
#NAME <- 'H3K36me3_H9.ENCFF267AKB.h19 '
#NAME <- 'H3K36me3_H9.ENCFF830UKK.h19’
#BED_FN <- paste0(NAME, '.bed')
OUT_DIR <- 'Results/'

###

bed_df <- read.delim(paste0('data/', NAME, '.bed'), as.is = TRUE, header = FALSE)
colnames(bed_df) <- c('chrom', 'start', 'end', 'name', 'score')
bed_df$len <- bed_df$end - bed_df$start
head(bed_df)

# hist(bed_df$len)

ggplot(bed_df) +
  aes(x = len) +
  geom_histogram() +
  ggtitle(NAME, subtitle = sprintf('Number of peaks = %s', nrow(bed_df))) +
  theme_bw()
ggsave(paste0('len_hist.', NAME, '.pdf'), path = OUT_DIR)
```

И чтобы отфильтровать слишком длинные участки при наличии и тоже визуализировать:

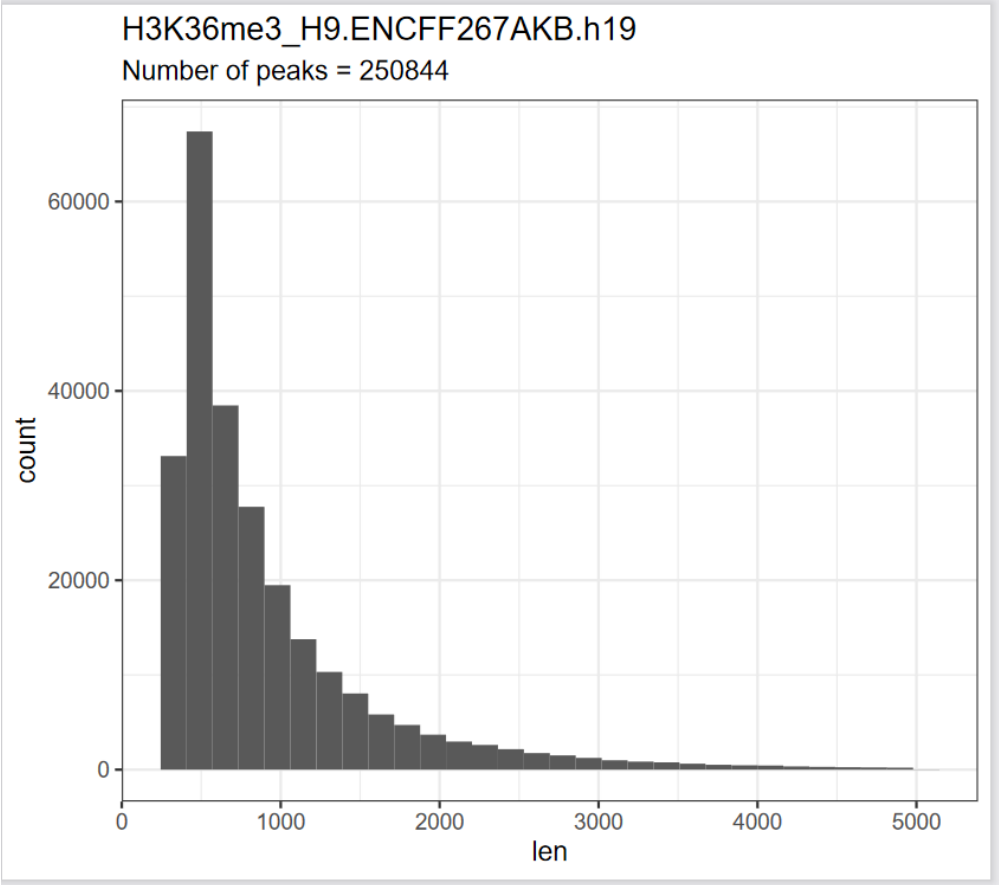
```
install.packages("dplyr")


library(ggplot2)
library(dplyr)
# library(tidyr)   # replace_na
# library(tibble)  # column_to_rownames

###

#NAME <- 'H3K36me3_H9.intersect_with_DeepZ'
#NAME <- 'DeepZ'
#NAME <- 'H3K36me3_H9.ENCFF267AKB.h19 '
#NAME <- 'H3K36me3_H9.ENCFF830UKK.h19’
#BED_FN <- paste0(NAME, '.bed')
OUT_DIR <- 'Results/'

###

bed_df <- read.delim(paste0('data/', NAME, '.bed'), as.is = TRUE, header = FALSE)
colnames(bed_df) <- c('chrom', 'start', 'end', 'name', 'score')
bed_df$len <- bed_df$end - bed_df$start
```

```
head(bed_df)

bed_df <- bed_df %>%
  arrange(-len) %>%
  filter(len < 50000) порог длины 50000 примерно выбран по изначальным графикам длин

ggplot(bed_df) +
  aes(x = len) +
  geom_histogram() +
  ggtitle(NAME, subtitle = sprintf('Number of peaks = %s', nrow(bed_df))) +
  theme_bw()
ggsave(paste0('len_hist.', NAME, '.filtered.pdf'), path = OUT_DIR)
```

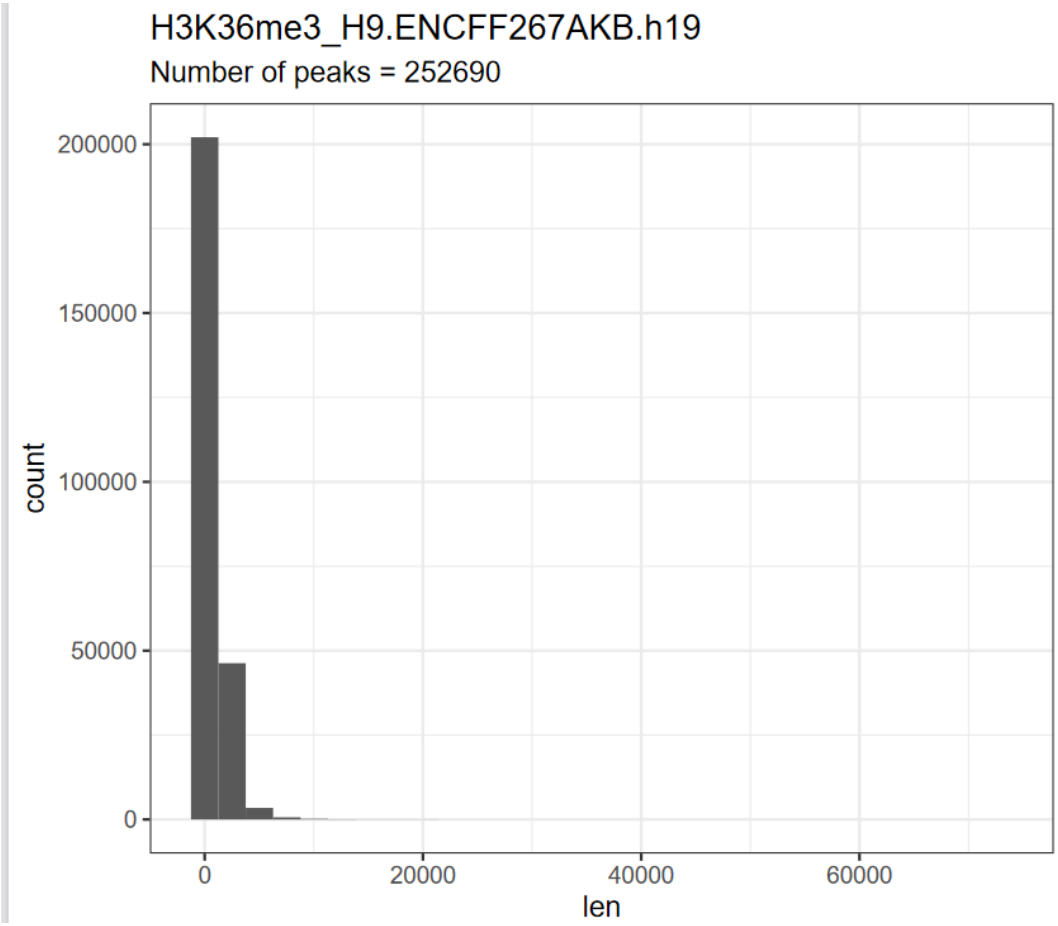
Число пиков

- В файлах до пересечения и фильтрации: 252690 и 297807 соответственно
- В файле ZDNA: 19394

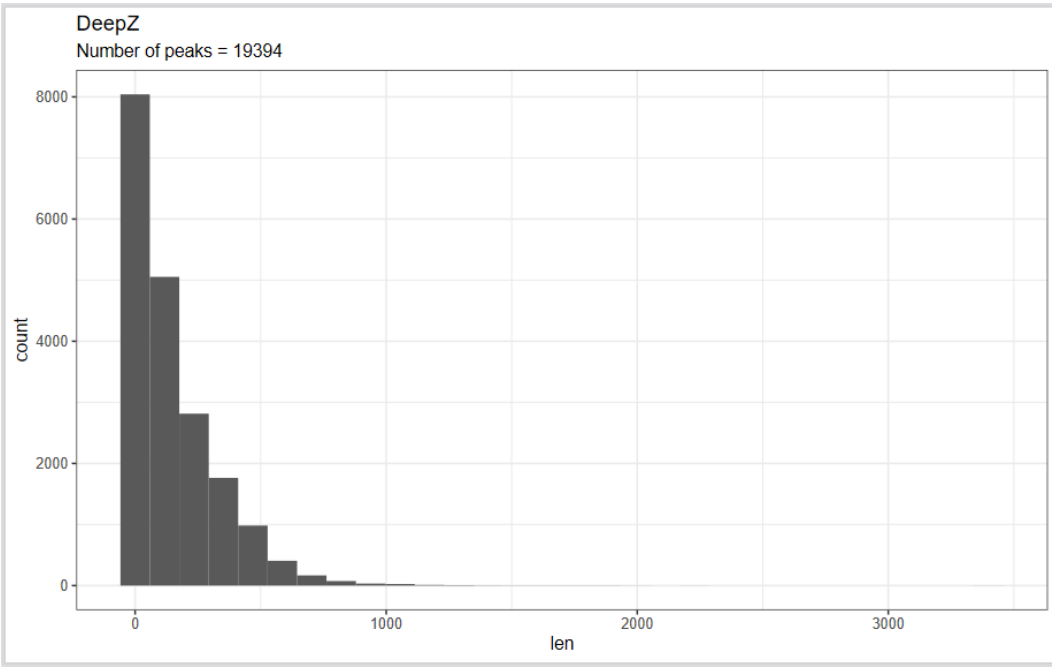
Отфильтрованные данные:



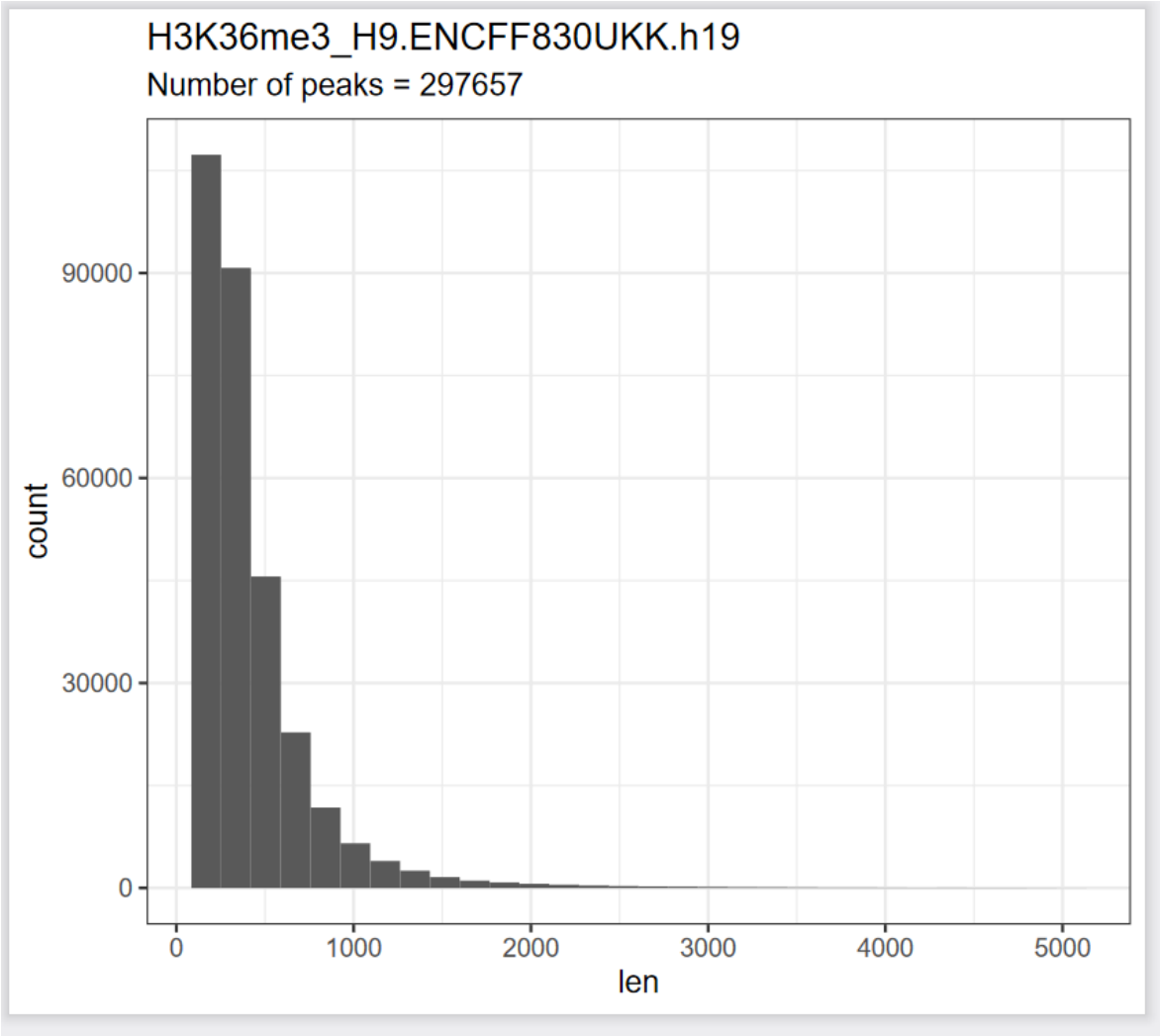
Изначальные данные:



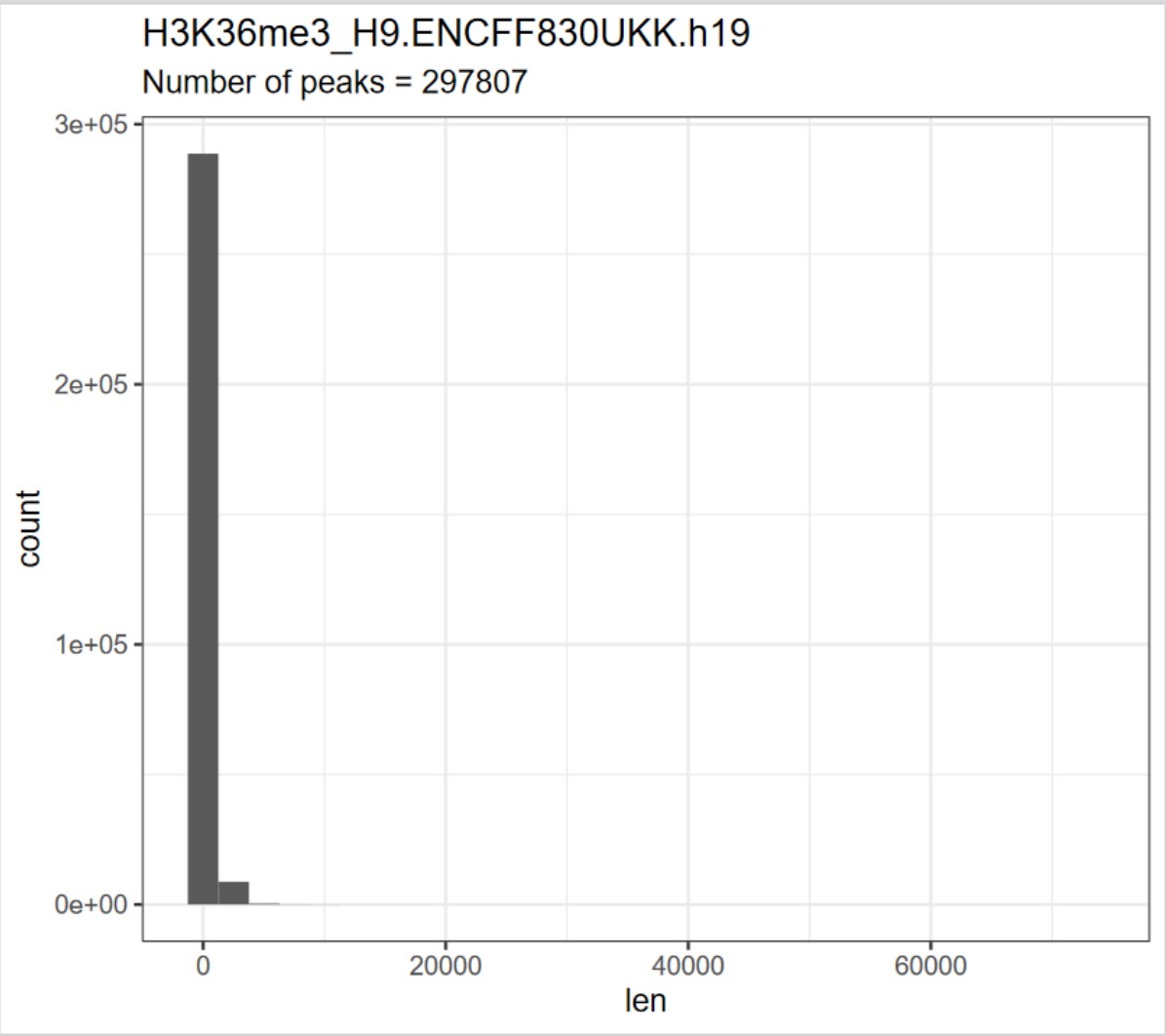
Данные DeepZ:



Отфильтрованные данные:



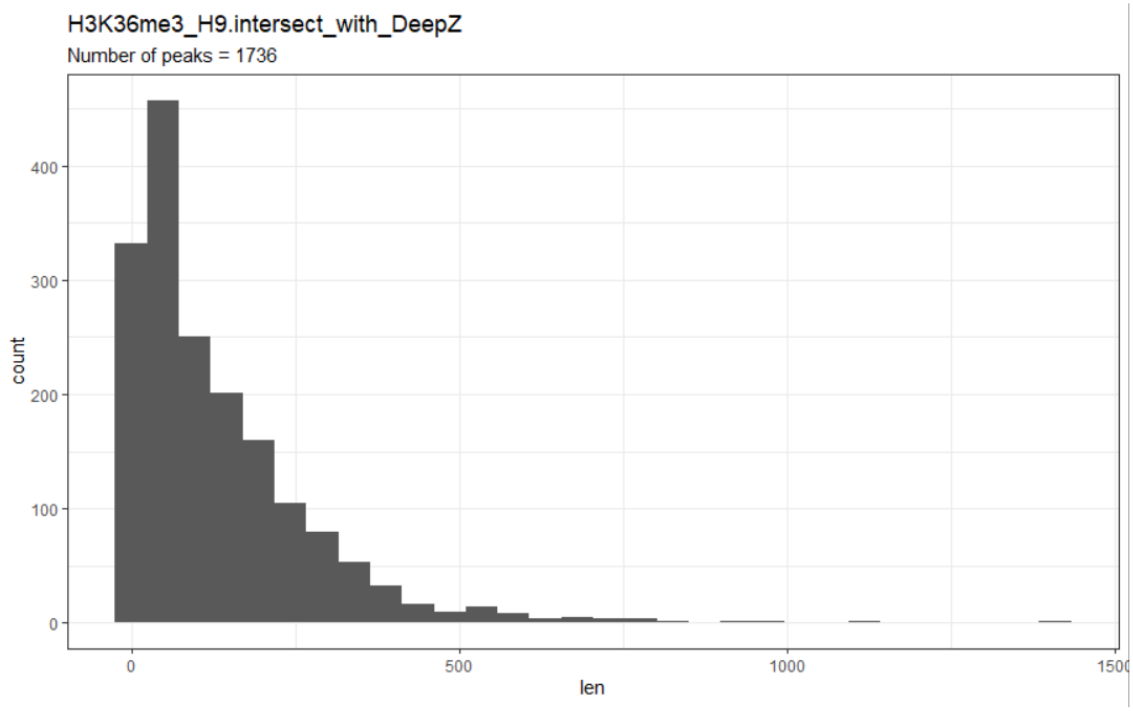
Изначальные данные:



Объединяем файлы

```
cat *.filtered.bed | sort -k1,1 -k2,2n | bedtools merge > H3K36me3_H9.merge.hg19.bed
```

Получаем новое распределение длин (код приведен ранее):



Геномный браузер

Визуализируем все треки:

track visibility=dense name=" ENCFF267AKB " description="H3K36me3_H9.ENCFF267AKB.h19.filtered.bed "
https://raw.githubusercontent.com/anyakazachkova/hse21_H3K36me3_ZDNA_DeepZ_human/main/data/H3K36me3_H9.ENCFF267AKB.h19.filtered.bed

track visibility=dense name="ENCFF830UKK" description=" H3K36me3_H9.ENCFF830UKK.h19.filtered.bed "
https://raw.githubusercontent.com/anyakazachkova/hse21_H3K36me3_ZDNA_DeepZ_human/main/data/H3K36me3_H9.ENCFF830UKK.h19.filtered.bed

track visibility=dense name=" DeepZ " color=50,50,200   description=" DeepZ "
https://raw.githubusercontent.com/anyakazachkova/hse21_H3K36me3_ZDNA_DeepZ_human/main/data/DeepZ.bed

track visibility=dense name="ChIP_merge" color=50,50,200   description="H3K36me3_H9.merge.hg19.bed"
https://raw.githubusercontent.com/anyakazachkova/hse21_H3K36me3_ZDNA_DeepZ_human/main/data/H3K36me3_H9.merge.h19.bed

(скриншот будет приведет далее)

Визуализация аннотаций в R

После скачивания файла строим пай-чарт для того, чтобы понять, где располагаются участки ДНК относительно аннотированных генов

#source('lib.R')

###

if (!requireNamespace("BiocManager", quietly = TRUE))

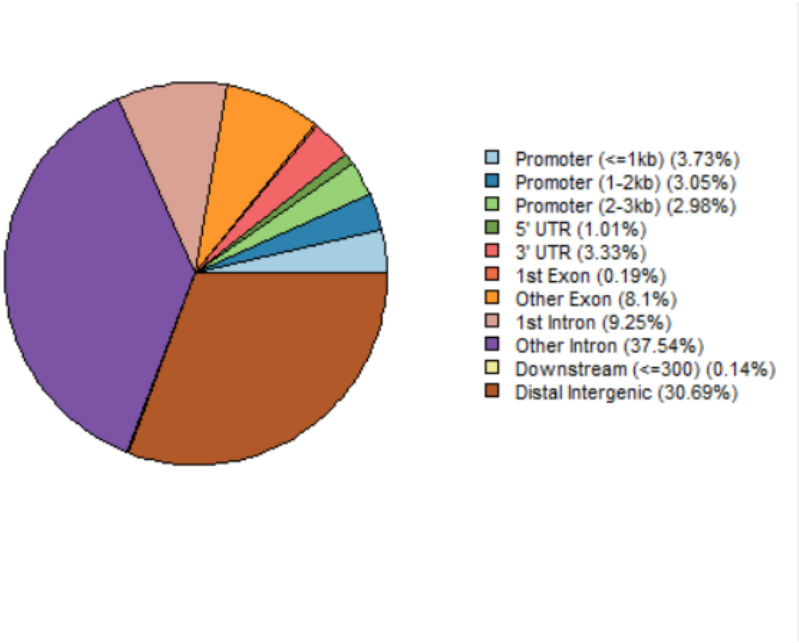
install.packages("BiocManager")

```
# BiocManager::install("TxDb.Hsapiens.UCSC.hg19.knownGene")
# BiocManager::install("TxDb.Mmusculus.UCSC.mm10.knownGene")

library(ChIPseeker)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
#library(TxDb.Mmusculus.UCSC.mm10.knownGene)
library(clusterProfiler)

###

NAME <- 'H3K36me3_H9.intersect_with_DeepZ'
#NAME <- 'DeepZ'
#NAME <- 'H3K36me3_H9.ENCFF267AKB.h19.filtered'
#NAME <- 'H3K36me3_H9.ENCFF830UKK.h19.filtered'
BED_FN <- paste0(NAME, '.bed')

###

txdb <- TxDb.Hsapiens.UCSC.hg19.knownGene

peakAnno <- annotatePeak(BED_FN, tssRegion=c(-3000, 3000), TxDb=txdb, annoDb="org.Hs.eg.db")

#pdf(paste0(OUT_DIR, 'chip_seeker.', NAME, '.plotAnnoPie.pdf'))
png(paste0('chip_seeker.', NAME, '.plotAnnoPie.png'))
plotAnnoPie(peakAnno)
dev.off()

# peak <- readPeakFile(BED_FN)
# pdf(paste0(OUT_DIR, 'chip_seeker.', NAME, '.covplot.pdf'))
# covplot(peak, weightCol="V5")
# dev.off()
#
```

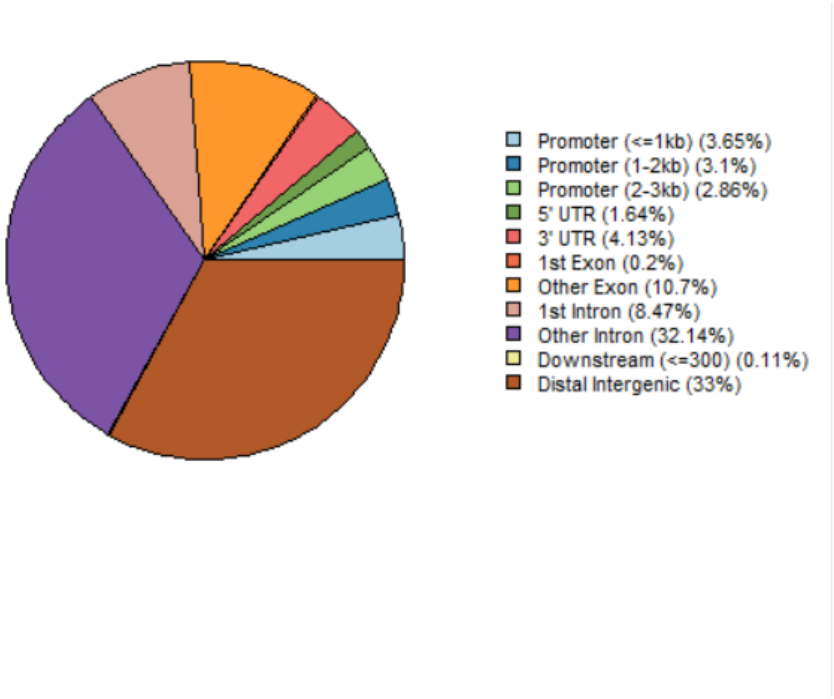
Результаты:

Для пиков гистоновой метки относительно аннотированных генов уже после фильтрации слишком длинных пиков

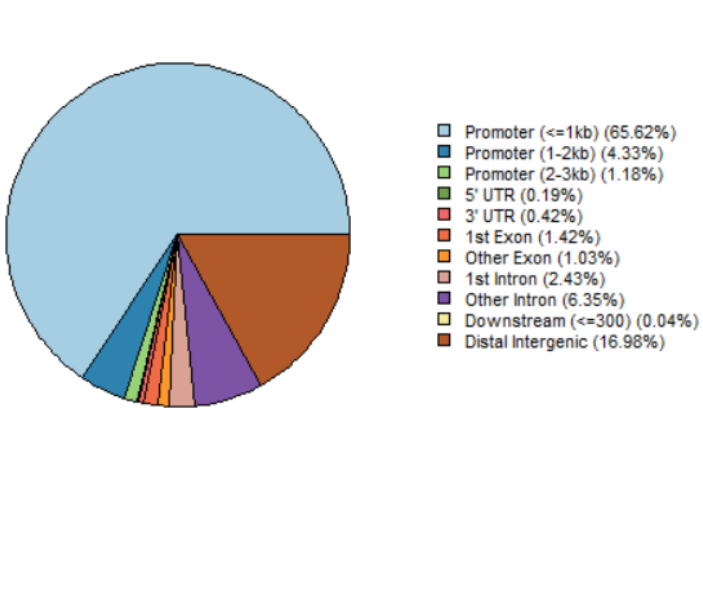
ENCFF830UKK.h19.filtered



ENCFF267AKB.h19.filtered



DeepZ



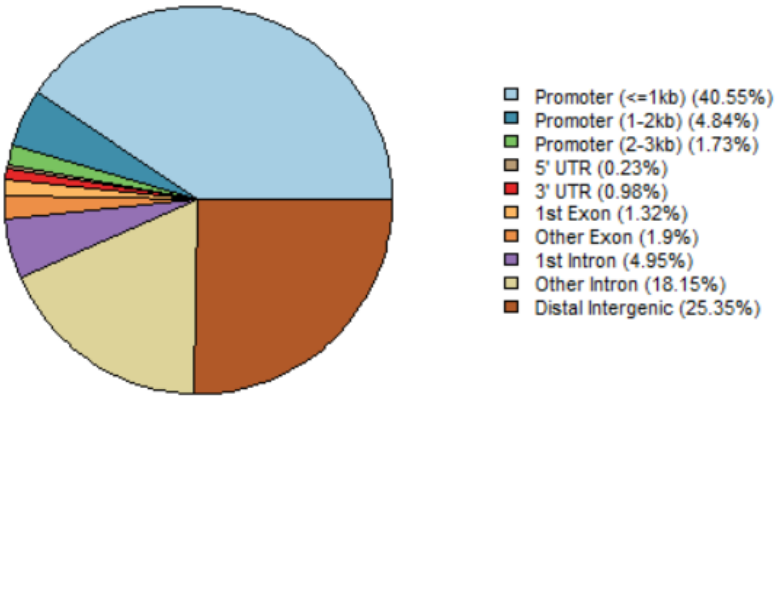
intersect

Находим пересечение гистоновой метки и структур ДНК

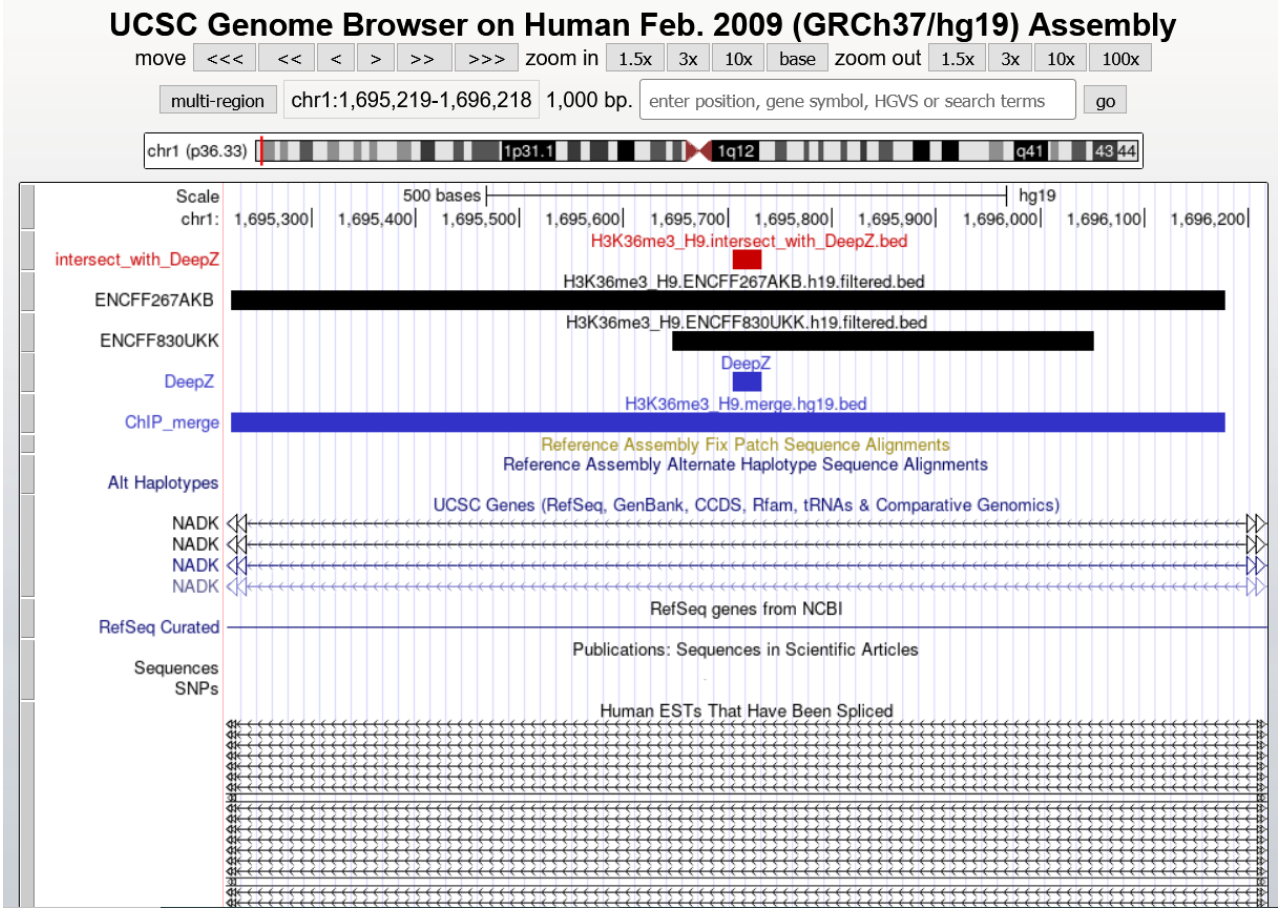
bedtools intersect -a DeepZ.bed -b H3K36me3_H9.merge.hg19.bed > H3K36me3_H9.intersect_with_DeepZ.bed

Добавляем соответствующие треки в геномном браузере (команда для DeepZ уже была приведена выше)

track visibility=dense name="intersect_with_DeepZ" color=200,0,0 description=" H3K36me3_H9.intersect_with_DeepZ.bed"
https://raw.githubusercontent.com/anyakazachkova/hse21_H3K36me3_ZDNA_DeepZ_human/main/data/H3K36me3_H9.intersect_with_DeepZ.bed



Результат:



На скриншоте можно видеть пересечение гистоновой метки со структурой ДНК (координаты chr1:1,695,712-1,695,733)

Теперь делаем ассоциирование полученных пересечений с ближайшими генами

#source('lib.R')

###

#https://bioconductor.org/packages/release/bioc/vignettes/ChIPpeakAnno/inst/doc/quickStart.html

BiocManager::install("ChIPpeakAnno")

BiocManager::install("org.Hs.eg.db")

BiocManager::install("org.Mm.eg.db")

library(ChIPpeakAnno)

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

library(org.Hs.eg.db)

```
#library(TxDb.Mmusculus.UCSC.mm10.knownGene)
#library(org.Mm.eg.db)

###



peaks <- toGRanges(paste0('H3K36me3_H9.intersect_with_DeepZ.bed'), format="BED")
peaks[1:2]

annoData <- toGRanges(TxDb.Hsapiens.UCSC.hg19.knownGene)
annoData[1:2]



anno <- annotatePeakInBatch(peaks, AnnotationData=annoData,
                        output="overlapping",
                        FeatureLocForDistance="TSS",
                        bindingRegion=c(-2000, 300))
data.frame(anno) %>% head()

anno$symbol <- xget(anno$feature, org.Hs.egSYMBOL)
data.frame(anno) %>% head()

anno_df <- data.frame(anno)
write.table(anno_df, file=paste0('H3K36me3_H9.intersect_with_DeepZ.bed.genes.txt'),
          col.names = TRUE, row.names = FALSE, sep = '\t', quote = FALSE)

uniq_genes_df <- unique(anno_df['symbol'])
write.table(uniq_genes_df, file=paste0('H3K36me3_H9.intersect_with_DeepZ.bed.genes_uniq.txt'),
          col.names = FALSE, row.names = FALSE, sep = '\t', quote = FALSE)
```

В итоге получаем два файла: файл ассоциации пиков с генами и список уникальных генов.

Число пиков, которые удалось проассоциировать: 661. Число уникальных генов: 526.

GO-анализ

Далее проводим анализ на основе полученных данных, результат сохраняем в текстовом формате