USC Department Of Translational Genomics
Keck Medicine of USC

TRGN 599: Applied Data Science and Bioinformatics

UNIT VI. Enrichment Analysis, Linear Regression

Week 13 - Lecture 1

Enrique I. Velazquez Villarreal, M.D., Ph.D., M.P.H., M.S. | **Assistant Professor**

Dept. of Translational Genomics

USC | Keck School of Medicine | Norris Comprehensive Cancer Center

Leader of the USC Bioinformatics Core – *USC CaRE2 Health Equity Center*

David W. Craig, Ph.D. | **Professor and Vice Chair**

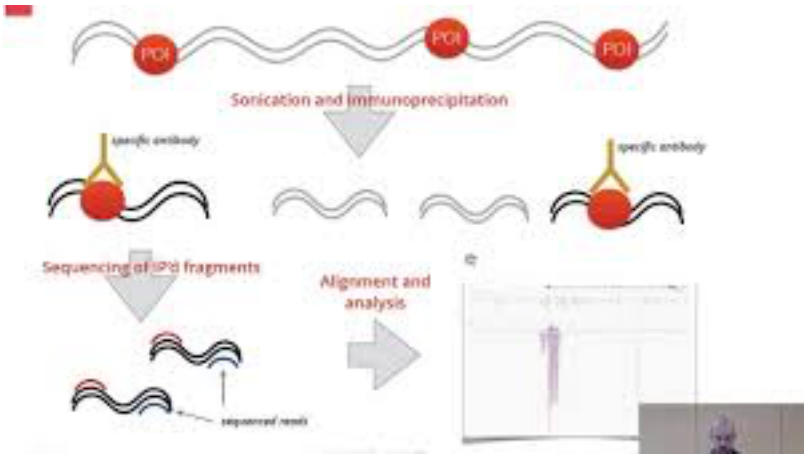
Dept. of Translational Genomics

USC | Keck School of Medicine | Norris Comprehensive Cancer Center

Co-Director, Institute of Translational Genomics

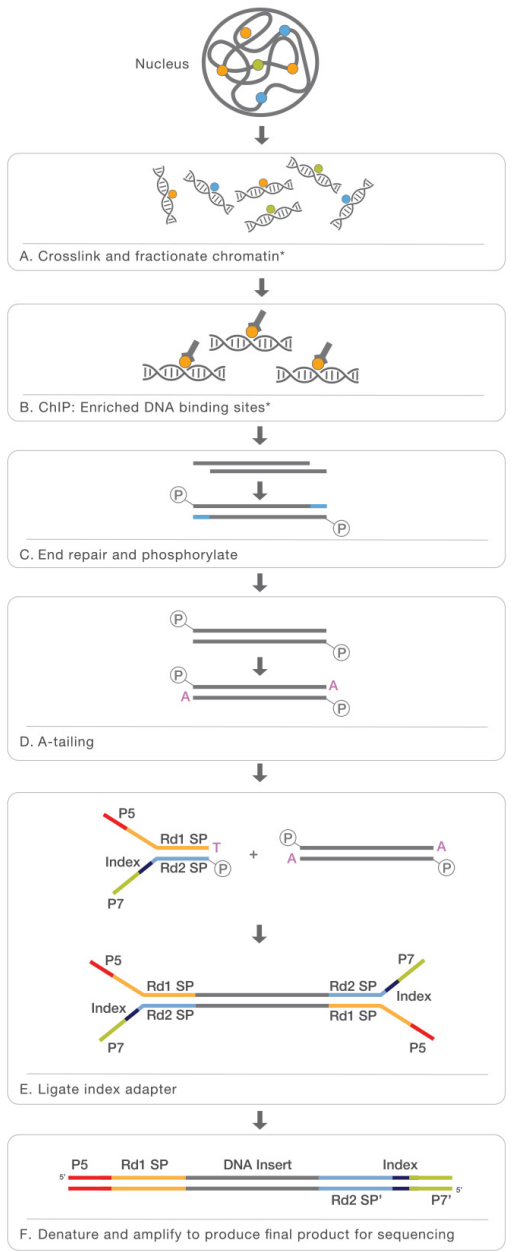
Topics

- Chromatin Immunoprecipitation (ChIP) - ChIPseq.



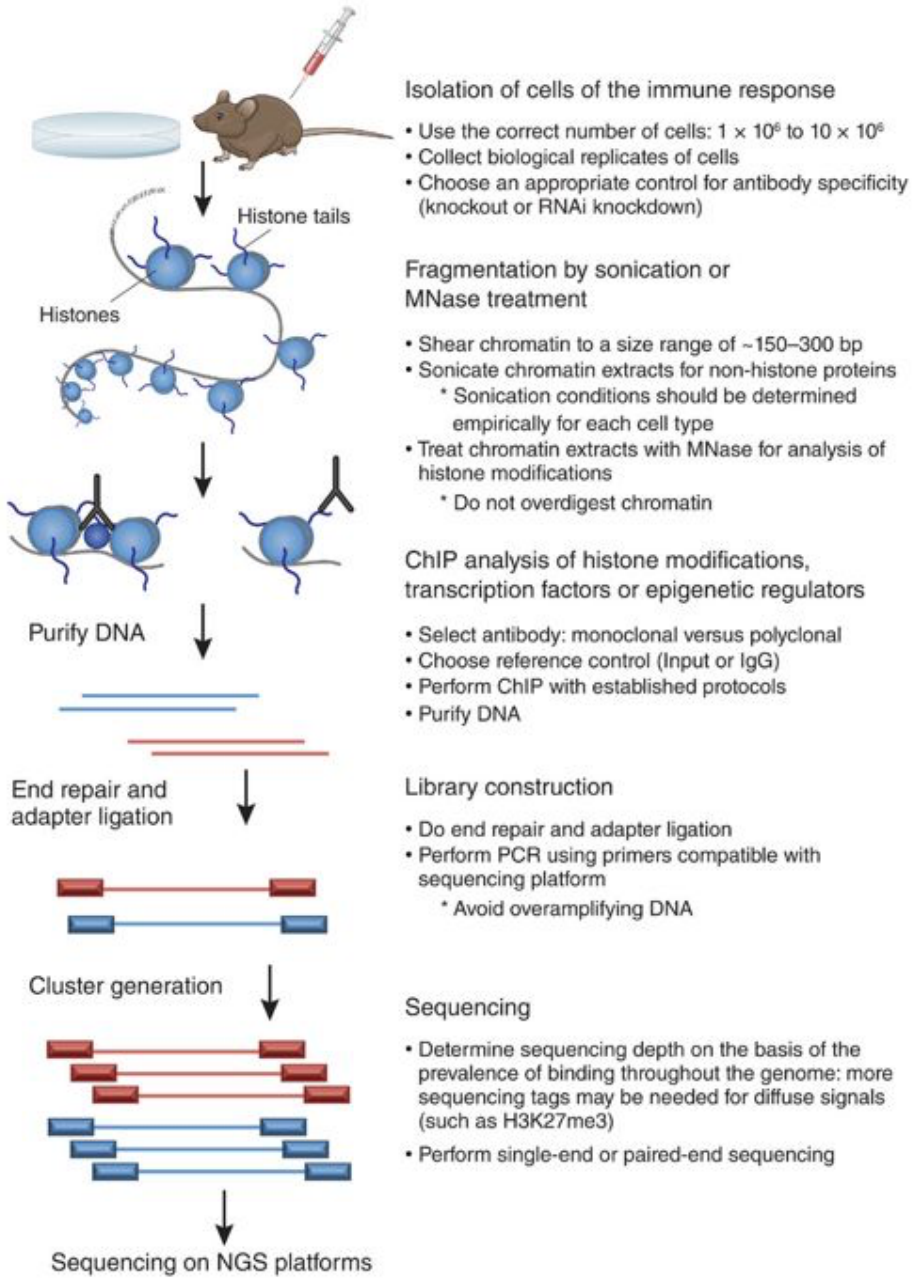
Chromatin Immunoprecipitation (ChIP)

- ChIP is a method for studying DNA regions that interact with proteins.
- These include transcription factor binding sites, histone binding regions, transcription initiation sites (TGBS), and others.
- Using antibodies against DNA interacting proteins is a useful method for the identification of genomic sequences interacting with them.
- After fishing up from the whole genome, individual binding sites can be cloned and sequenced, or they can be investigated in a high-throughput manner.



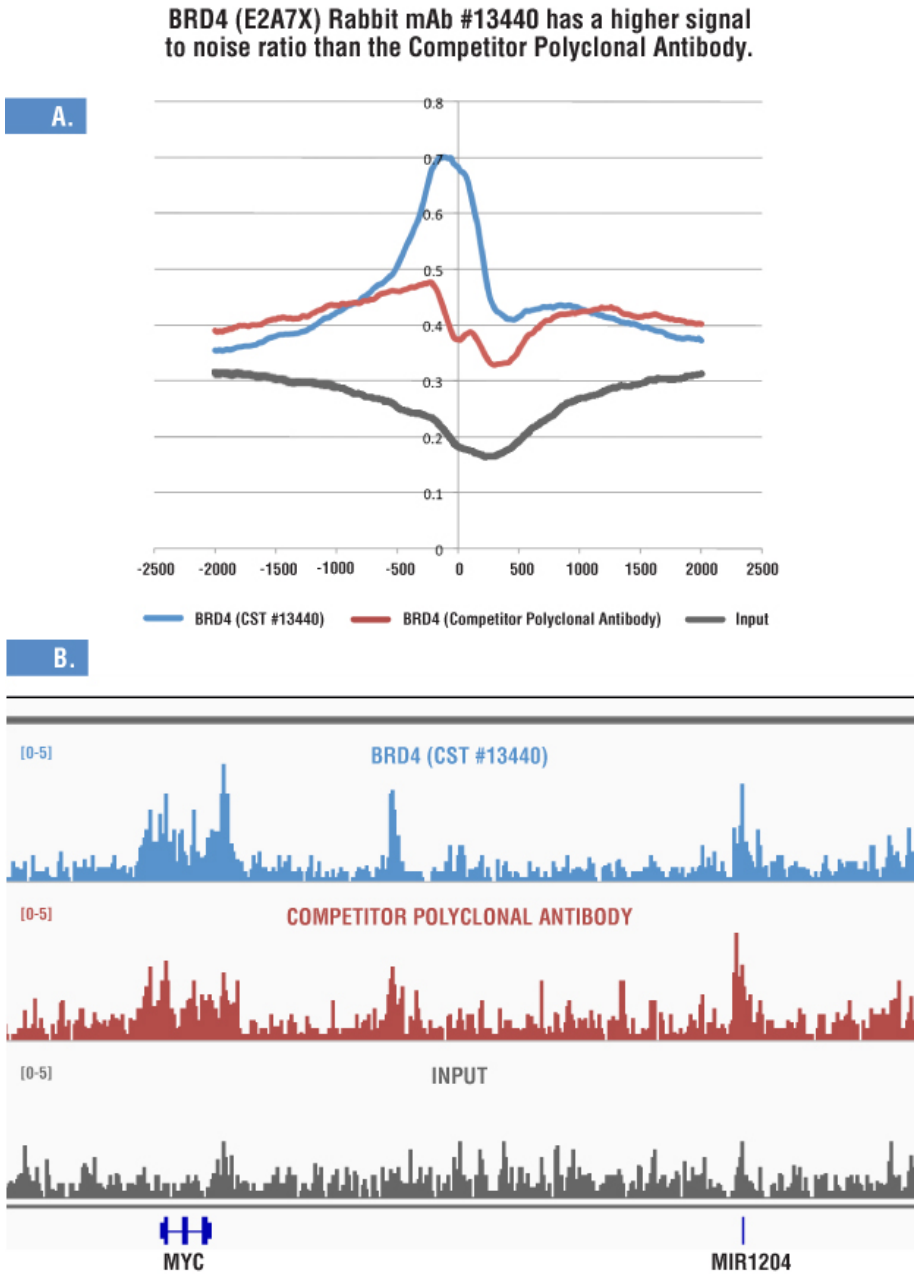
Chromatin Immunoprecipitation (ChIP)

- Antibodies generated against particular proteins are frequently used tools in molecular biology experiments.
- They selectively attach to target molecules and offer a way to isolate them from biological samples.
- While monoclonal antibodies recognize a single region (epitope) of the protein of interest, polyclonal antibodies are a mixture of antibodies against multiple epitopes of the target protein.
- ChIP studies, antibodies are used to isolate DNA-binding proteins together with the genomic DNA fragment (chromatin in ChIP terminology) of the protein of interest, polyclonal antibodies are a mixture of antibodies against multiple epitopes of the target protein.



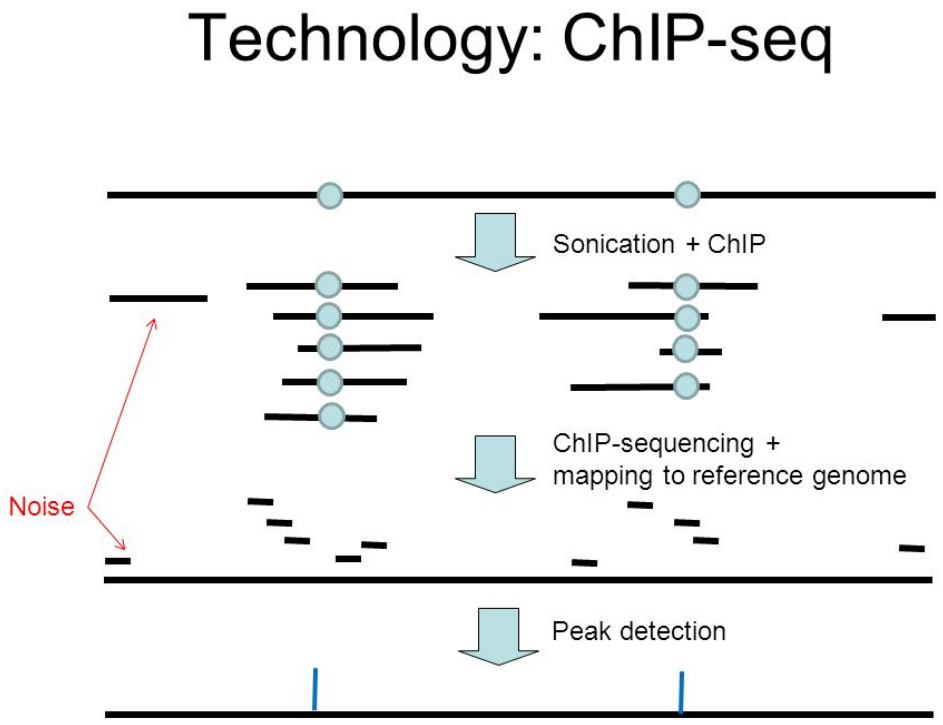
Chromatin Immunoprecipitation (ChIP)

- In ChIP studies, antibodies are used to isolate DNA-binding proteins together with the genomic DNA fragment (chromatin in ChIP terminology) to which they are attached.
- In ChIP experiments, the focus is on the identificantion of the DNA segments.
- There are two classes of these experiments: one is focused on a selected TF, while the other is interested in nucleosomes.
- In both cases, researchers are looking for tissue, cell-state, or disease-specific differences in protein-DNA relations.



Chromatin Immunoprecipitation (ChIP)

- TFs bind to the promoter regions of genes, usually enhancing the gene expression in eukaryotic cells.
- Accordingly, the immunoprecipitation of a target TF will pinpoint the activated genes in biological samples.
- In addition, we can also study the sequence motifs to which the RF is binding.

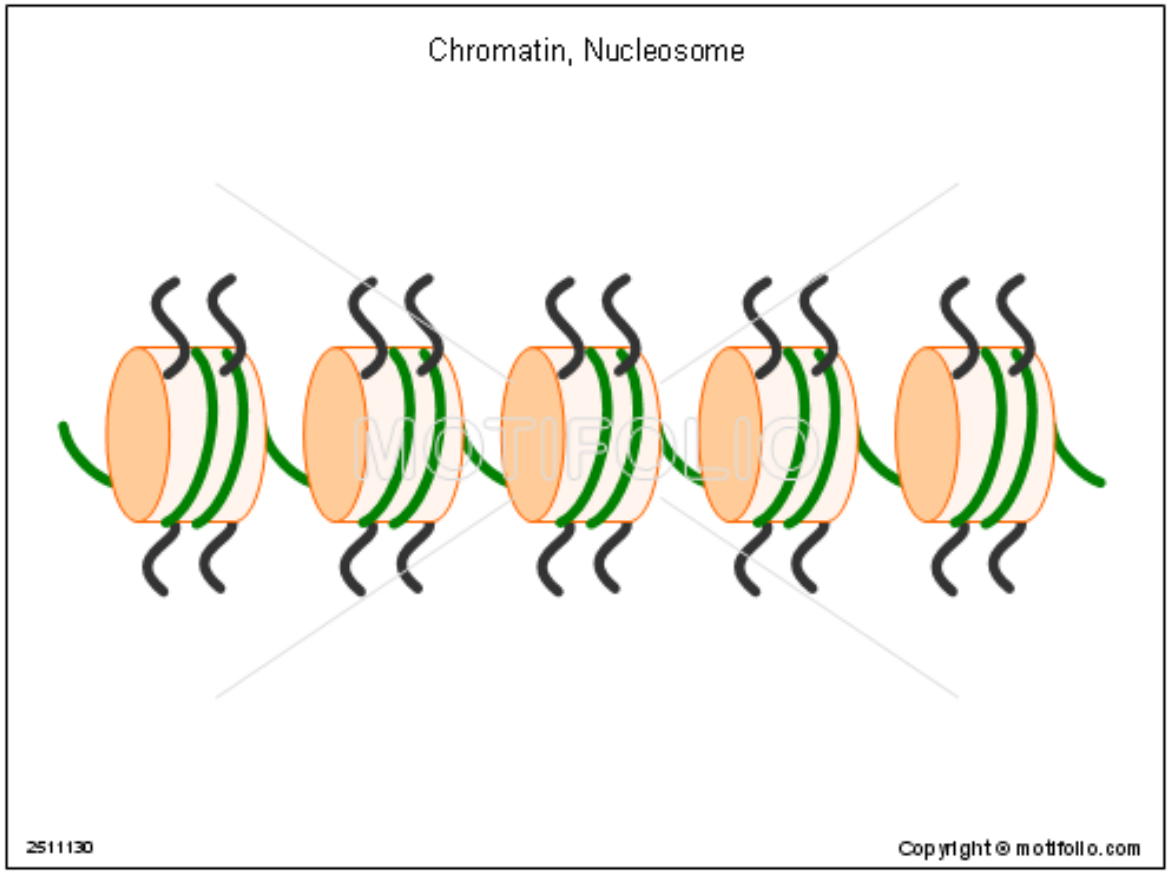


Chromatin Immunoprecipitation (ChIP)

- Nucleosomes are complex structures containing histone protein complexes around which a segment of DNA is wrapped.
- Their role is to pack the genomic DNA, and via this they repress gene expression.
- Covalent modifications of histone proteins can hold epigenetic signals essential for tissue development, organ formation, and diseases.
- These issues are actively studied in ChIP experiments since antibodies can be developed against specific histone modifications.

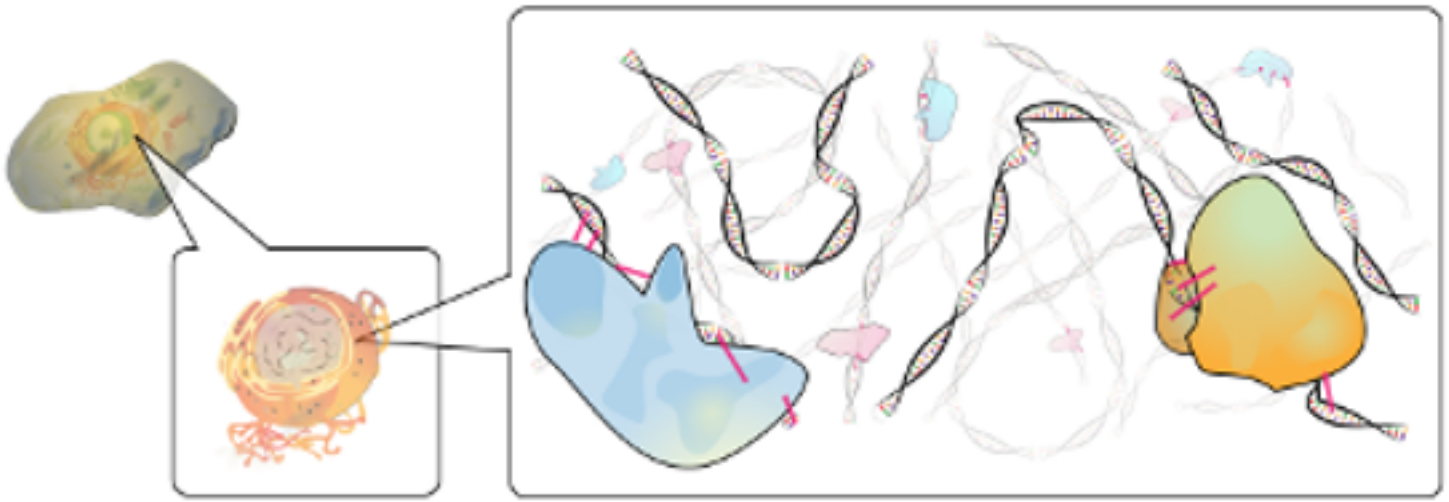
Chromatin Immunoprecipitation (ChIP)

- In both types of these experiments, the targeted protein-associated DNA fragments are isolated and submitted for further studies.
- They can be mapped to the genome either by using tiling microarrays (ChIP-Chip) or can be directly sequenced using next-generation sequencing (NGS) methods (ChIP-seq).



Experimental Background

- The wet-lab part of ChIP experiments relies on very high-quality antibodies.
- They have to fulfill special requirements like high specificity and low sensitivity to inhibitory factors present in the input chromatin samples.
- For these reasons, antibody providers distinguish ChIP-grade antibodies as a unique product group.

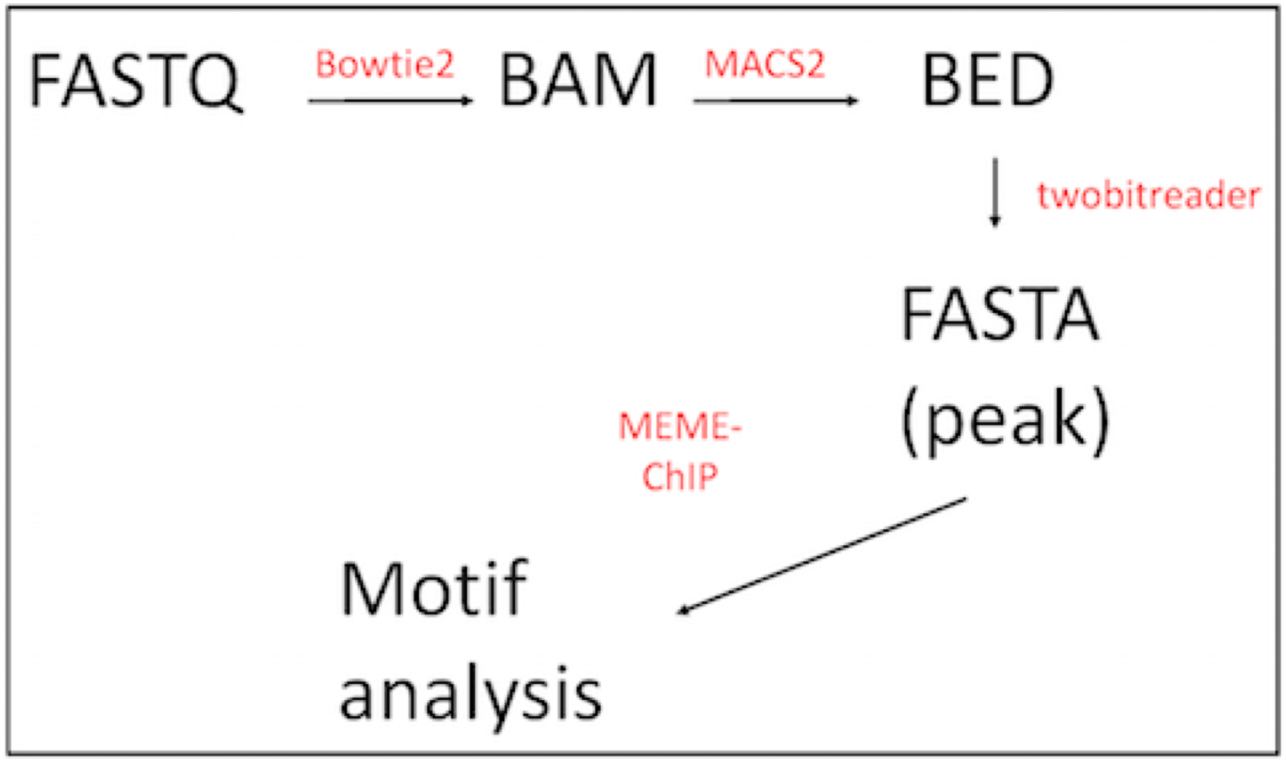


Experimental Background

- Some experimental procedures are different in cases when the target protein is a TF or a nucleosome protein (typically histone modification).
- While genomic DNA is naturally linked to nucleosomes as it is wrapped tightly around the protein complex, promoter regions should be artificially cross-linked to TFs using formaldehyde or ultraviolet light.
- Native nucleosome-wrapped DNA is digested with microccal nuclease enzyme that cuts the DNA between the nucleosomes providing 200 base-pair (bp)-long DNA fragments for further analysis.

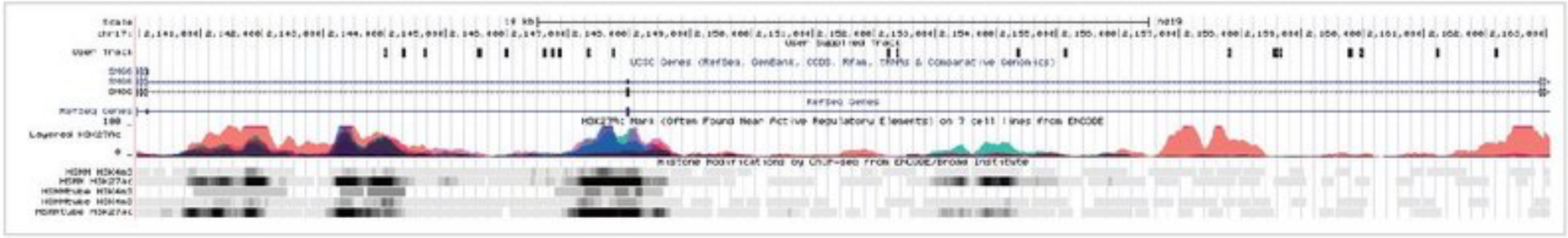
Experimental Background

- On the other hand, DNA artificiallyu cross-linked to TFs is fragmented (sheared) by sonication.
- This method results in a less-defined distribution of fragment sizes with a broad peak between 300 to 1000 bp, depending on the amount of energy used and many other factors.
- Therefore, in data analysis, it is important to know how the DNA fragments were isolated.



Fragment analysis

- The two dominant ways to identify genome segments linked to the target proteins are to use microarray or high-throughput sequencing.
- Tiling arrays used in ChIP-chip (or sometimes also called ChIP-on-Chip) experiments are special microarrays containing a high number of oligunucleotides.
- These probes, usually called “reporters” in these experiments, are designed so that they cover entire genomic regions in a continuous way.
- The dimensions of these array types are very different.
  - Affymetrix covering the entire human genome consist of seven arrays having 45 million perfect match probes, each 25bp long.



Fragment analysis

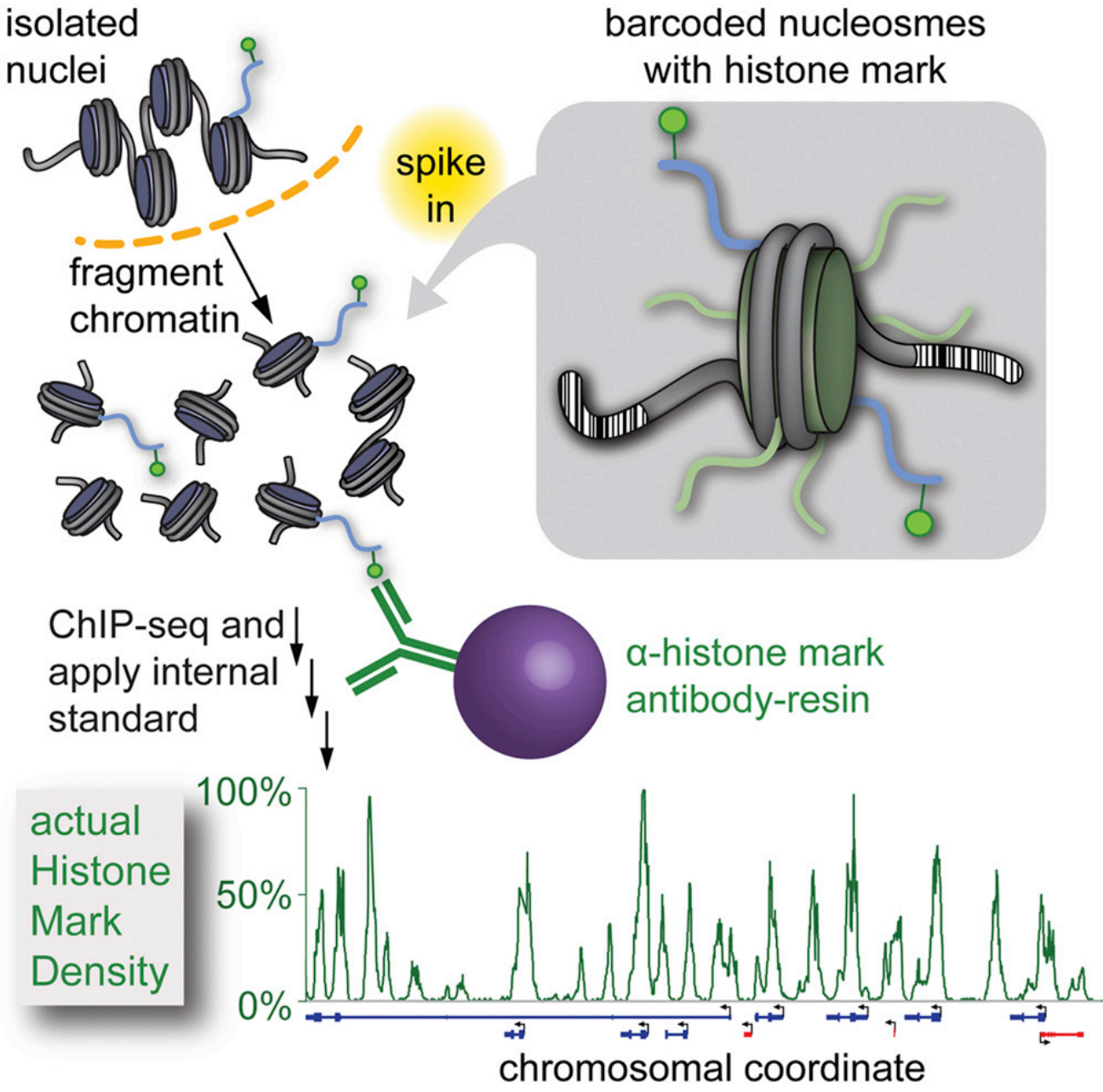
- Recently, ChIP-seq analysis replaced the array-based ChIP experiments completely for several reasons.
- While the availability of commercial arrays limited the experiments to a handful of model organisms.
- NGS does not have a similar limitation, at leas not in the laboratory part.
- Also, the precision of ChIP-seq is not restricted by the tiling of proves, and generally, they offer better resolution than arrays.
- Also, read densities can quantify the DNA-binding affinity of proteins, taking ChIP studies one step further from qualitative-to quantitative-type research.

ChIP data in ENDOCE

- A systematic mapping of TF-binding patterns in different issues and developmental stages has a clear potential in assembling a complex picture of gene expression regulation in various physiological stages and biological processes.

- Such a project would clearly be an enormous research attempt, comparable to the Human Genome Project of its time.

- The Encyclopedia of DNA Elements (ENCODE) Consortium, or its more known name the ENCODE project (ENCODE Project consortium 2012), aims to build a comprenhensive list of parts of functional elements in the human genome.



ChIP data in ENDOCE

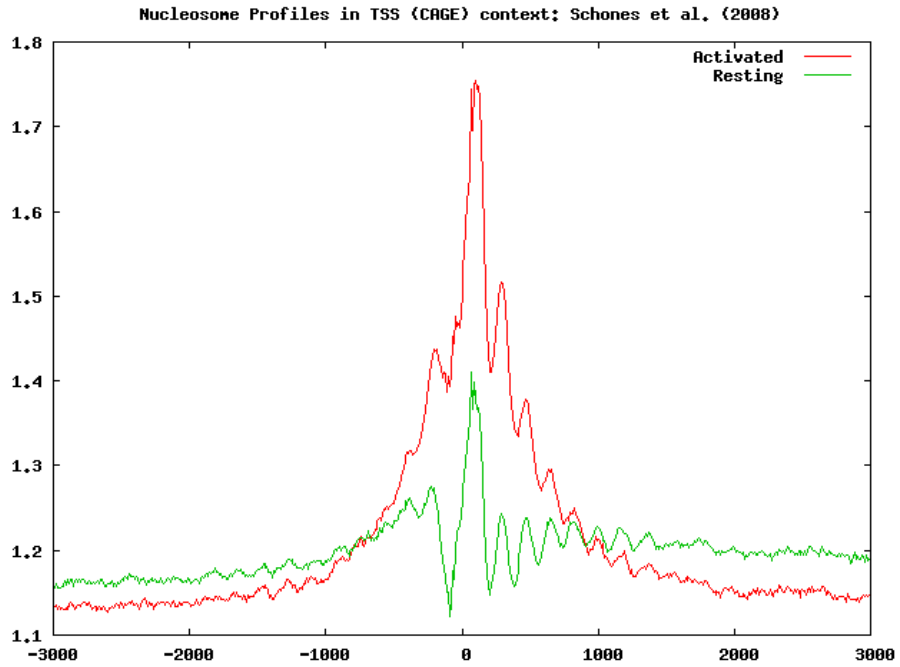
- One component of this project is using ChIP followed by high-throughput sequencing (ChIP-seq) to map DNA-protein interactions.
- Even such grand project cannot map all 1800 known RFs in all the 147 investigated cell lines.
- Instead, they mapped 119 factors in a limited number of cell lines, in addition to 13 histone modifications.
- The results of this project are available from both UCSC and Ensembl genome browsers.

ChIP data in ENDOCE

- ENCODE also produces data for mouse genome. Further, modENCODE, a sister project of ENCODE, has similar goals (modENCODE Consortium et al. 2010)
- It aims for elucidating protein binding sites in Drosophila melanogaster and Caenorhabditis elegans genomes.
- All these publicly available data are excellent starting point for gene expression regulation studies.
- Although their results should be handled with special care, as most experiments were executed as a single replicate.
- It means that despite ENCODE data, the doors are open for more thorough analysis of binding patterns of many chromatin binding proteins in many tissues and organisms.

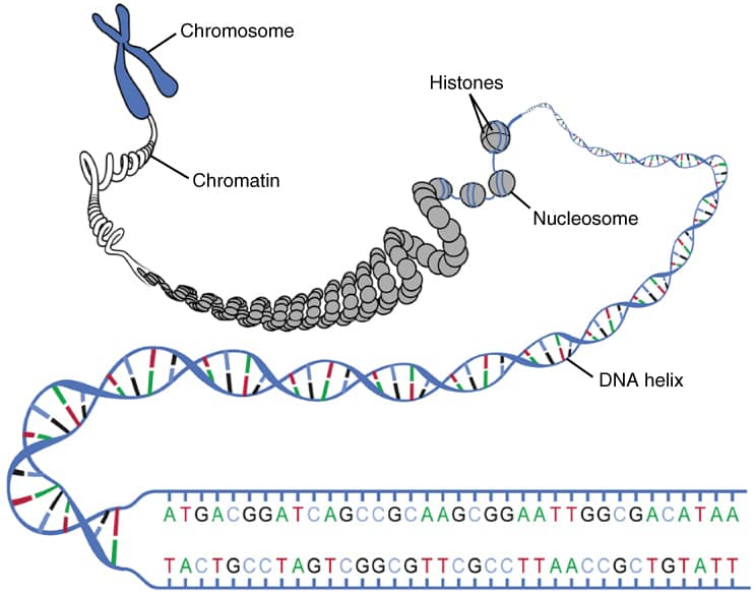
ChIP with tiling microarrays

- ChIP-on-Chip experiments were widely used for a decade to identify chromatin structure and TFBS.
- ArrayExpress and Gene Expression Omnibus (GEO) databases contain a large amount of data produced by these experiments.
- Most of the data in public databases are available in processed format.
- These data originate form ChIP that was followed by DNA fragment identification experiments.



ChIP with tiling microarrays

- The processed data files are dimple tab-delimited text files with reporters and the corresponding normalized enrichment ratios, meaning that no special libraries are needed to access their content.
- In these two-color experiments, DNA fragments coming from an immuno-precipitation experiment are hybridized to one channel of the tiling array, while the other channel is occupied by the full genomic isolate (called the “input”).
- Scanning the array provides the enrichment ratio of the reporters that tile those genomic regions which are covered by target protein (“immunoprecipitated”).
- Since enrichment ratio can have a wide numerical range. it is usual to provide its logarithm.



ChIP with tiling microarrays

- High ratio values represent genomic regions where the investigated proteins binds.
- Since enrichment ratio can have a wide numerical range, usually the result table contains the logarithm values of the enrichment scores.
- Also, single-color tiling arrays are available for similar applications.
- After normalization, the values of different reporters can be directly compared in the different biological samples.

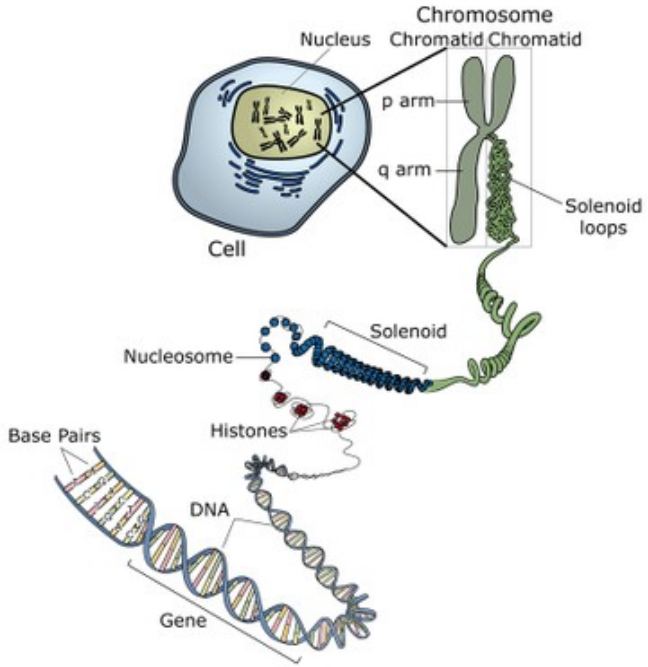


Image adapted from: National Human Genome Research Institute.

ChIP with tiling microarrays

- The example dataset in the E-GEOD-31301_example.zip file contains three data tables originating from Agilent two-color arrays from ArrayExpress.

- Since both the data files and the array definition file containing the genomic coordinates of the reporters are in a simple text format, accessing their content happens by the basic string handling capabilities of R.

```
---
title: "Rmarkdown_Week_13_Lecture_1"
author: "Enrique I. Velazquez Villarreal, MD, PhD, MPH, MS"
date: "4/15/2019"
output: html_document
---

## Uploading the file "trgn599.clinical.tsv":

```{R}
#data.dir <- '/Users/enriquevelazquez/Documents/R_working_directory/E-GEOD-31301'

setwd('/Users/enriquevelazquez/Documents/R_working_directory/E-GEOD-31301')

my.dat<-read.delim("GSM775491_sample_table.txt",row.names=1)

adf <- read.delim('A-GEOD-13972.adf.txt',skip=15,row.names=1)
names(adf) <- c("Coordinate","Sequence")

```
```

ChIP with tiling microarrays

````
```{R}

head(my.dat)

```
````

```
         VALUE
4 -0.145817615
5 -0.006284506
6  0.116073348
7  0.784466893
8 -0.143302852
9 -0.102563881
```

````
```{R}

head(adf)
```
````

	Coordinate	Sequence	chr	pos	start	end
4	chr4:000179775-000179834	ATTCCCCAATTAGGATACTTAGTAACCATTAGCGTCTTAATGGAACCTTTATTGGATGGC	chr4	000179775-000179834	179775	179834
5	chr3:000071144-000071192	CAAGTTCCTACCTGGGCAATAACTCCAATTCAAATTCGAGTTATGGGGG	chr3	000071144-000071192	71144	71192
6	chr3:000239622-000239681	AACTTCTGGTCTAGATCAACTAATGGATGCTATTCAATACACAGAAACAGACGAATATGG	chr3	000239622-000239681	239622	239681
7	chr14:000285414-000285473	CCATTAGTTAAACTTTTGGAAGTTGCAGAATATAAAACTAAAAAAGAAGCTTGTTGGGCT	chr14	000285414-000285473	285414	285473
8	chr7:000835520-000835578	AGGATACTCTCATCTTCAAAAACAATATTGAATCCCGCTATTCTGCCAGGTAAATCCTC	chr7	000835520-000835578	835520	835578
9	chr10:000492976-000493035	AAATCCATATCATGCACTTGCTTGCAAAGAACGTTGACACAATCTTTTAGAGAGTCAGTT	chr10	000492976-000493035	492976	493035

ChIP with tiling microarrays

- In the created data structures, row names are reported identifiers (IDs) that are similar to probe IDs in gene expression microarray experiments.
- In the adf variable, the “Coordinate” column holds the information we need for correlating genomic regions with high immunoprecipitation ratios.
- For control reporters, this field is empty, but for the rest it contains the chromosome ID, the start and end positions of the reporter on the chromosome.
- Let us use the apply() function to separate these three data items into discrete columns.

ChIP with tiling microarrays

```
33 ```{R}
34 adf$chr <- apply(matrix(adf$Coordinate,ncol=1),1,function(coord){strsplit(coord,':')[[1]][1]})
35 adf$pos <- apply(matrix(adf$Coordinate,ncol=1),1,function(coord){strsplit(coord,':')[[1]][2]})
36 adf$start <- apply(matrix(adf$pos,ncol=1),1,function(coord){strsplit(coord,'-')[[1]][1]})
37 adf$end <- apply(matrix(adf$pos,ncol=1),1,function(coord){strsplit(coord,'-')[[1]][2]})
38 ```
```

- The rows for control reporters can be dropped from the table as they are not helpful after normalization.

```
41 ```{R}
42 adf <- adf[!is.na(adf$chr),]
43 ```
```

ChIP with tiling microarrays

- Since the Coordinate column is interpreted as containing character strings, all further columns derived from it are handled as characters.

- Therefore, the columns holding the start and end coordinates of the reporters should be converted to numerical format.

- Additionally, we might want to have chromosome information as a factor.

```
45 ```{R}
46 adf$start <- as.numeric(adf$start)
47 adf$end <- as.numeric(adf$end)
48 adf$chr <- as.factor(adf$chr)
49
50 head(adf)
51
52 ```
```

	Coordinate	Sequence	chr	pos	start	end
4	chr4:000179775-000179834	ATTCCCCAATTAGGATACTTAGTAACCATTAGCGTCTTAATGGAACCTTTATTGGATGGC	chr4	000179775-000179834	179775	179834
5	chr3:000071144-000071192	CAAGTTCCTACCTGGGCAATAACTCCAATTCAAATTCGAGTTATGGGGG	chr3	000071144-000071192	71144	71192
6	chr3:000239622-000239681	AACTTCTGGTCTAGATCAACTAATGGATGCTATTCAATACACAGAAACAGACGAATATGG	chr3	000239622-000239681	239622	239681
7	chr14:000285414-000285473	CCATTAGTTAAACTTTTGGAAGTTGCAGAATATAAAACTAAAAAAGAAGCTTGTTGGGCT	chr14	000285414-000285473	285414	285473
8	chr7:000835520-000835578	AGGATACTCTCATCTTCAAAAACAATATTGAATCCCGCTATTCTGCCAGGTAAATCCTC	chr7	000835520-000835578	835520	835578
9	chr10:000492976-000493035	AAATCCATATCATGCACTTGCTTGCAAAGAACGTTGACACAATCTTTTAGAGAGTCAGTT	chr10	000492976-000493035	492976	493035

ChIP with tiling microarrays

- Now the adf structure can be merged to the data table based on the common row names.

```
54 ```{R}
55 my.dat <- cbind(my.dat,adf[,c(3,5,6)])
56 head(my.dat)
57
58 ```
```

	VALUE	chr	start	end	chr	start	end
4	-0.145817615	chr4	179775	179834	chr4	179775	179834
5	-0.006284506	chr3	71144	71192	chr3	71144	71192
6	0.116073348	chr3	239622	239681	chr3	239622	239681
7	0.784466893	chr14	285414	285473	chr14	285414	285473
8	-0.143302852	chr7	835520	835578	chr7	835520	835578
9	-0.102563881	chr10	492976	493035	chr10	492976	493035

ChIP with tiling microarrays

- This table provides some basic overview of the available data.

- For example, one can count the length of the reporters on the basis of the start and end positions, and prepare a simple frequency table.

```
60 ```{R}
61
62 table(my.dat$end-my.dat$start+1)
63
64 ```
```

```
   45    46    47    48    49    50    51    52    53    54    55    56    57    58    59    60
  935   288   280   370   421   490   530   621   797   922  1229  1762  2672  4289  6710 38331
```

ChIP with tiling microarrays

- The frequency table reveals that most of the probes are 60 nucleotides (nt) long, while there are some with any length between 45 and 59 nt.

- We can check the number of reporters on the different yeast chromosomes in a very similar way.

```{R}

table(my.dat$chr)

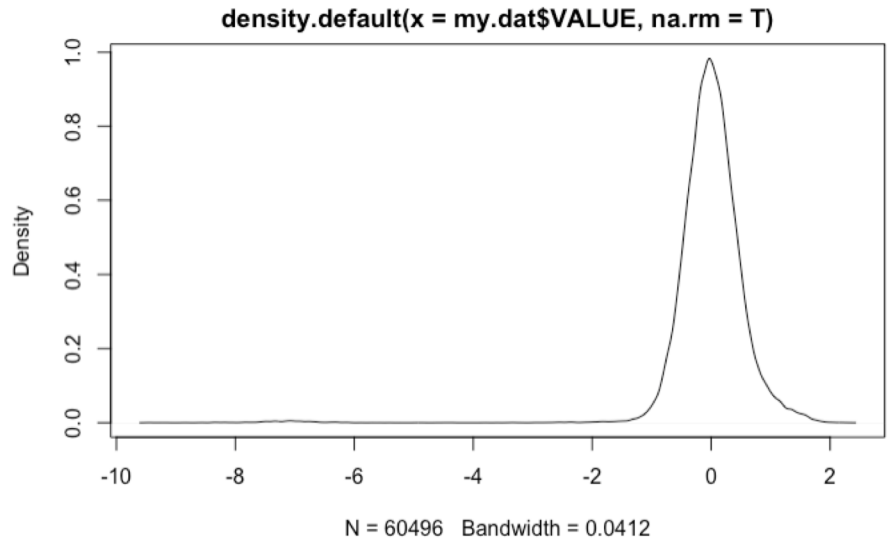
```

```
 chr1 chr10 chr11 chr12 chr13 chr14 chr15 chr16  chr2  chr3  chr4  chr5  chr6  chr7  chr8  chr9  chrM
 1015  3685  3435  5284  4715  3929  5497  4760  4114  1528  7657  2852  1320  5475  2760  2171   450
```

ChIP with tiling microarrays

- These steps still focus on the annotation information.
- To gain an overview of what this distribution of the immunoprecipitation ratios is, the density() function can be employed.
- It is not very surprising that we can see a normal distribution centered on 0.

```
72 ```{R}
73
74 plot(density(my.dat$VALUE,na.rm=T))
75
76 ```
```



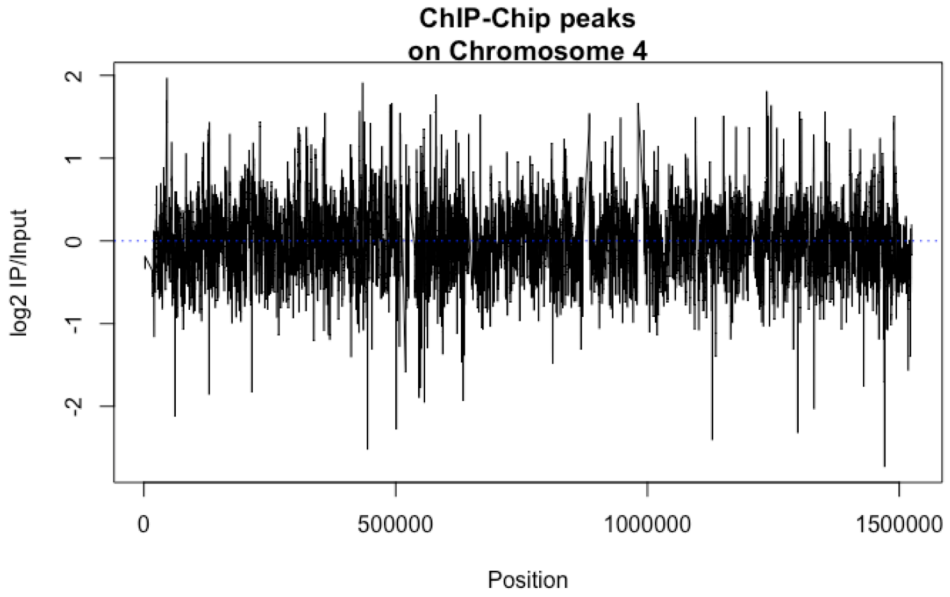
Distribution of immunoprecipitation ratios in an Investigated samples. Since measured values show actually the logarithm of the ratios, a value close to zero means equal measured binding in the treatment and control conditions.

ChIP with tiling microarrays

- Let us study chromosome 4 as this one has the highest number of reporters on the array.

- We create a slice of the original data table and order the rows according to their chromosomal position.

```
78 ```{R}
79 my.dat.c4 <- my.dat[my.dat$chr=='chr4',]
80 my.dat.c4 <- my.dat.c4[order(my.dat.c4$start),]
81
82 w.start <- min(my.dat.c4$start)
83 w.end <- max(my.dat.c4$end)
84
85 plot(my.dat.c4$VALUE ~ my.dat.c4$start,t='l',xlim=c(w.start,w.end),main="ChIP-Chip peaks\non Chromosome 4",xlab="Position",ylab="log2 IP/Input")
86 abline(h=0,col="blue",lty=3)
87
88 ```
```



ChIP with tiling microarrays

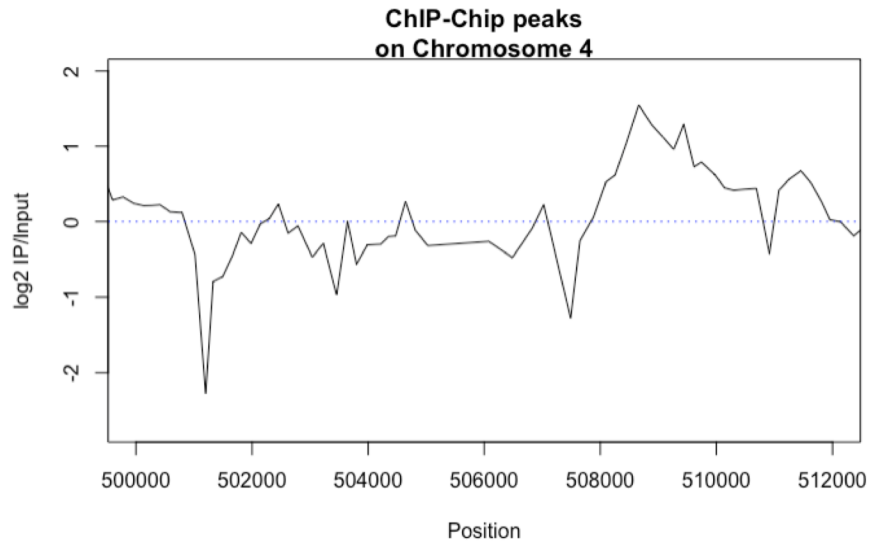
- The plot for the entire chromosome 4 contains too much information.
- However, any smaller segments can be investigated by zooming.
- For this purpose, one should specify the start and end coordinates of the desired segment.

```
```{R}

w.start <- 500000
w.end <- 512000

plot(my.dat.c4$VALUE ~ my.dat.c4$start,t='l',xlim=c(w.start,w.end),main="ChIP-Chip peaks\non Chromosome 4",xlab="Position",ylab="log2 IP/Input")
abline(h=0,col="blue",lty=3)

```
```



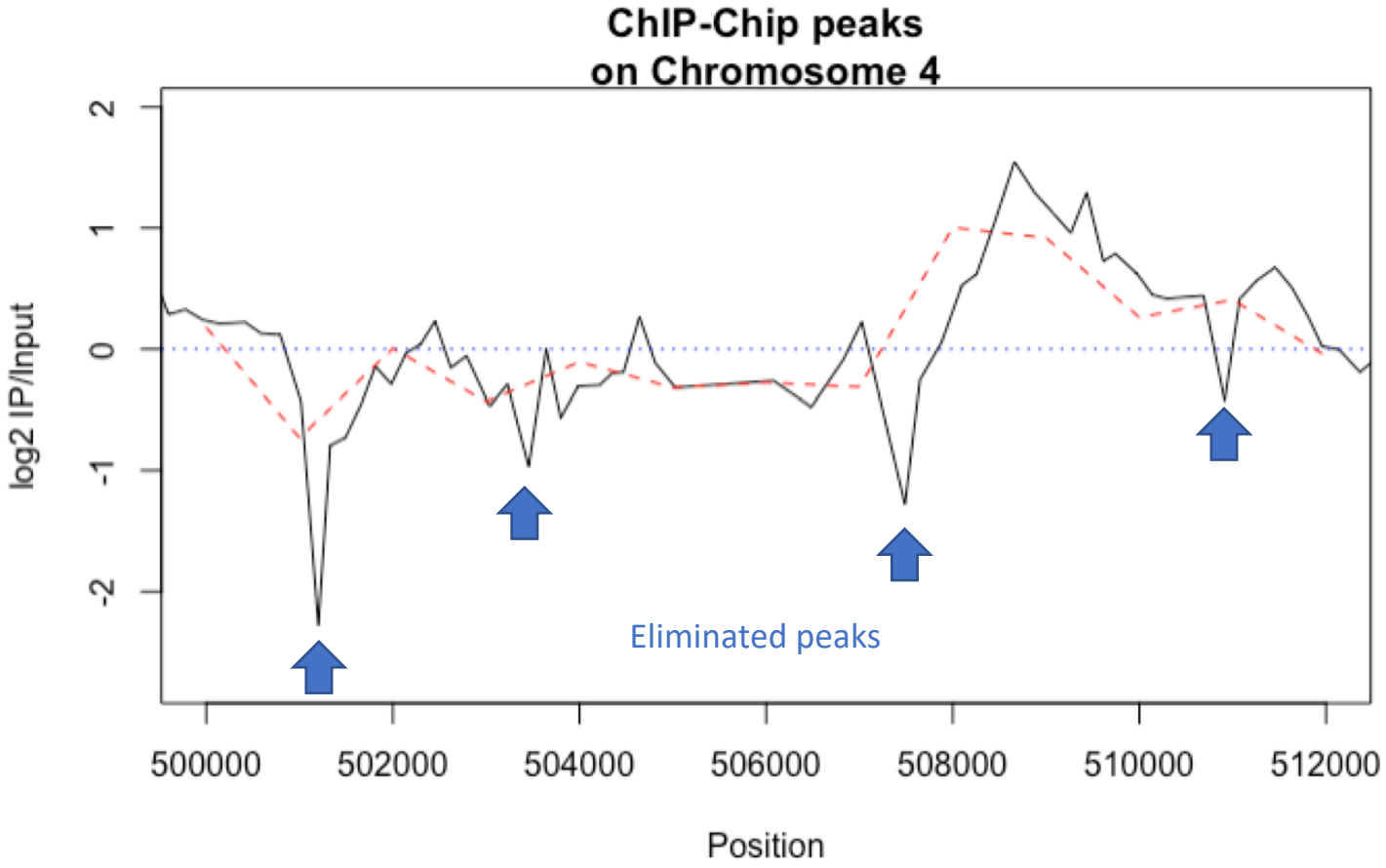
Binding peaks of Isw1 protein in yeast investigated by two-color tiling array.
The plot shows a segment of Chromosome 4.
Horizontal bars show regions that could be subjects of further investigation.

ChIP with tiling microarrays

- These peaks are very sharp, especially the negative peak at 501,000. A single reporter can show high or low ratio.
- For example, because of experimental failures.
- False peaks can appear in any region
- A sliding window approach is employed next to smooth sharp changes in the plot to eliminate false peaks coming from single reporters.

```
100 ```{R}
101 w.size <- 1000
102
103 w.pos <- rep(NA,(w.end-w.start)/w.size)
104 w.smooth <- w.pos
105
106 i <- 1
107
108 for(pos in seq(w.start,w.end+w.size,w.size)){
109   vals <- my.dat.c4$VALUE[my.dat.c4$start>pos &  my.dat.c4$start<(pos+w.size)]
110   w.pos[i] <- pos
111   w.smooth[i] <- mean(vals)
112   i <- i+1
113 }
114
115 plot(my.dat.c4$VALUE ~ my.dat.c4$start,t='l',xlim=c(w.start,w.end),main="ChIP-Chip peaks\non Chromosome 4",xlab="Position",ylab="log2 IP/Input")
116 abline(h=0,col="blue",lty=3)
117 lines(w.pos,w.smooth,col="red",lty=2)
118 ```
```

ChIP with tiling microarrays



Using sliding window approach to smooth
The immunoprecipitation peaks.
The arrows show the most prominent peaks
that are probably coming from single reporters.
The smoothed curve represented by dashed
line can be carried further in the analysis.

ChIP with tiling microarrays

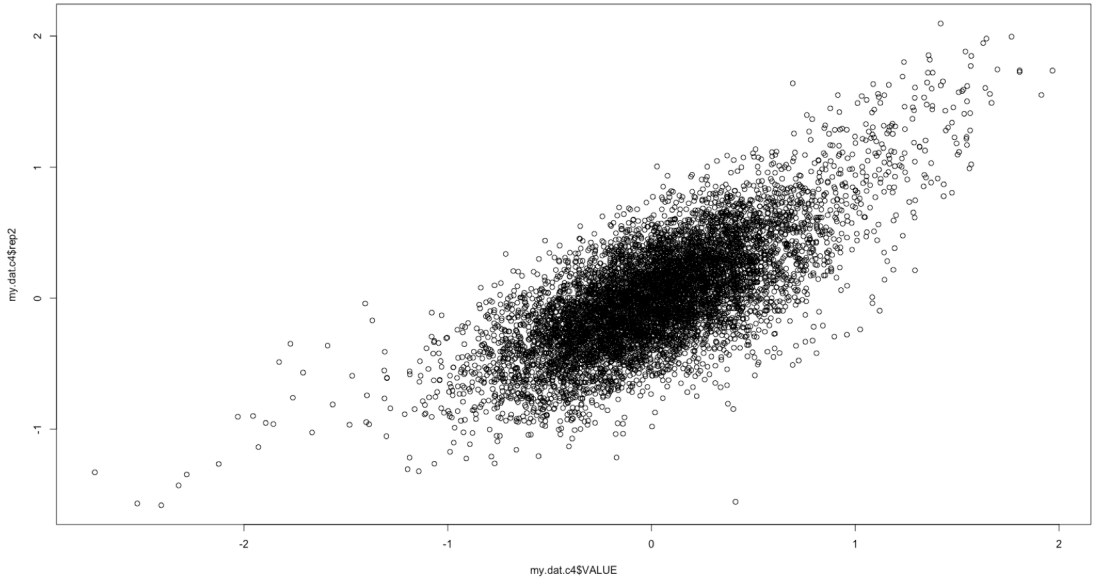
- As a result, the plot shows a single region on the right half that represents a well-defined binding region of Isw1 protein.
- Another possibility to confirm real binding regions is to use biological replicates.
- Fortunately, the example dataset contains three independent samples.
- Let us read in the data tables of the two further samples and bind their content to the data frame holding chromosome 4 data using the reporter IDs there.

```
120 ```{R}
121 ## replicates
122 
123 reporters <- row.names(my.dat.c4)
124 
125 my.dat.2<-read.delim("GSM775492_sample_table.txt",row.names=1)
126 my.dat.3<-read.delim("GSM775493_sample_table.txt",row.names=1)
127 
128 my.dat.c4$rep2 <- my.dat.2[reporters,]
129 my.dat.c4$rep3 <- my.dat.3[reporters,]
```

ChIP with tiling microarrays

- The first task is to check the general agreement among the biological replicates.

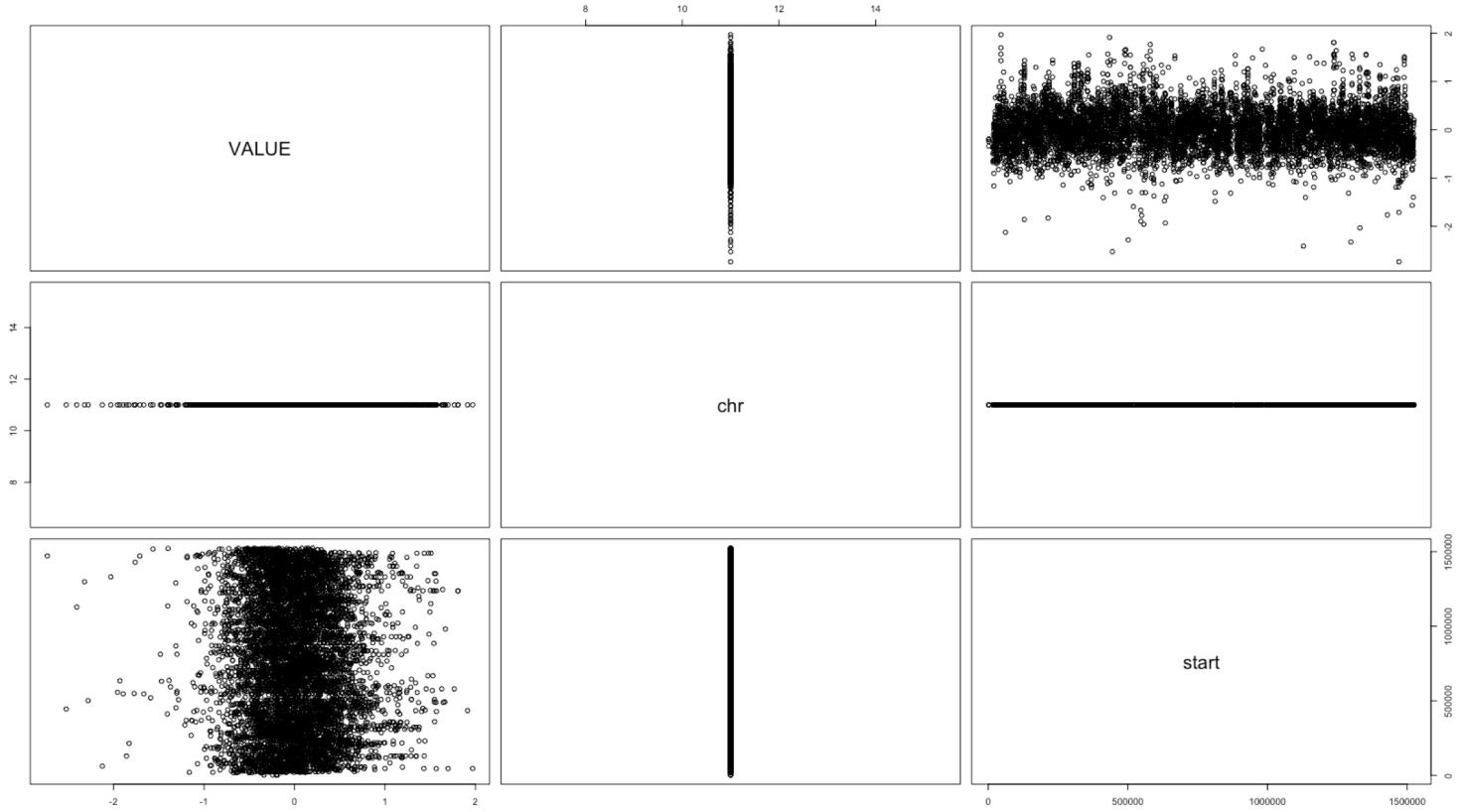
- We can use simple scatter plots to test the correlation between sample pairs.



ChIP with tiling microarrays

- Or the pairs() function can be applied that produces pairwise correlograms:

```
137 ```{R}
138
139 pairs(my.dat.c4[,c(1,5,6)])
140
141 ```
```



ChIP with tiling microarrays

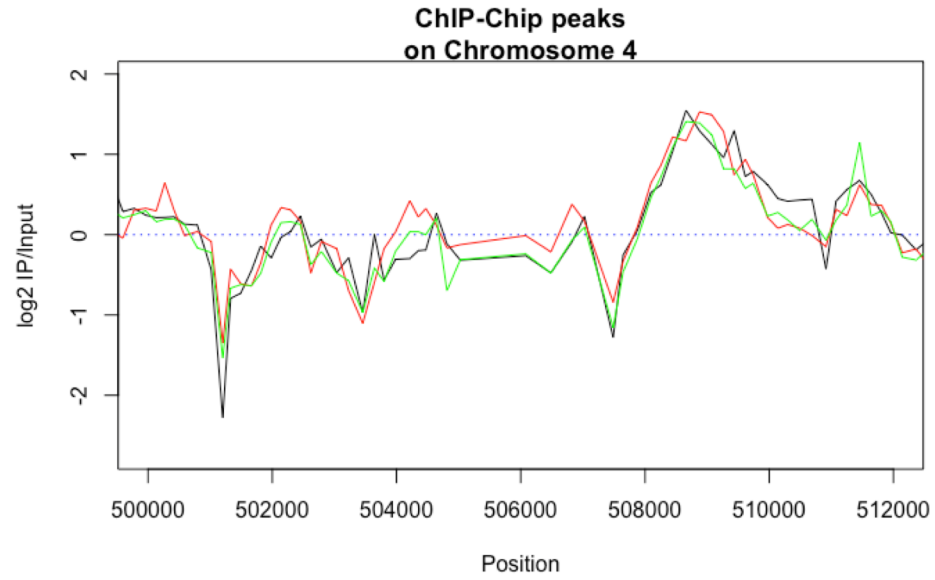
- This function offers much more than one can observe from the first sight.
- Let us mark the running median of the plot and the correlation coefficients in the upper and lower triangles of the correlogram.

```
```{R}

panel.cor <- function(x, y, digits = 2, prefix = "", cex.cor, ...){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- abs(cor(x, y,use="pairwise.complete.obs"))
  txt <- format(c(r, 0.123456789), digits = digits)[1]
  txt <- paste0(prefix, txt)
  if(missing(cex.cor)) cex.cor <- 0.8/strwidth(txt)
  text(0.5, 0.5, txt, cex = cex.cor * r)
}
pairs(my.dat.c4[,c(1,5,6)],upper.panel=panel.smooth,lower.panel=panel.cor)

plot(my.dat.c4$VALUE ~ my.dat.c4$start,t='l',xlim=c(w.start,w.end),main="ChIP-Chip peaks\non Chromosome 4",xlab="Position",ylab="log2 IP/Input")
abline(h=0,col="blue",lty=3)
lines(my.dat.c4$start,my.dat.c4$rep2,col="red")
lines(my.dat.c4$start,my.dat.c4$rep3,col="green")

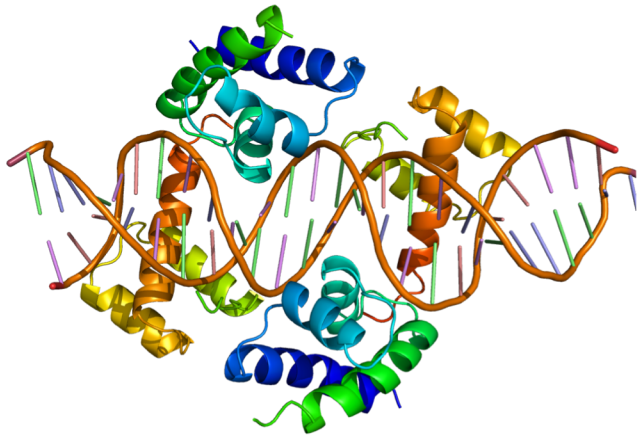
```
```



Comparison of the binding peaks of Isw1 Protein in three independent biological samples.

High-throughput sequencing of ChIP fragments

- Recently, the focus of the identification of DNA fragments originating from ChIP experiments shifted from microarrays to NGS methods.
- From data analysis point of view, ChIP-seq experiments are very similar to RNA-seq.
- Preprocessing regarding quality checks and filtering are employed the same way, and filtered reads are mapped to reference genomes.
- The downstream analysis starts from alignment files holding information about the genomic mapping individual reads.



High-throughput sequencing of ChIP fragments

- In the following example workflow, a dataset is employed that is available from the EatonEtAlChIPseq Bioconductor package.
- Installing this package will offer access several files in mapview and BED formats.
- The read alignments are in two files in the extdata folder of the package.
- They read alignments are in two files in the extdata folder of the package.
- They can be accessed as pre-built R data structures, but here they are handled as any other mapview file downloaded from a database or coming directly from MAQ software performing unmapped alignment of short reads.

````
# High-throughput sequencing of ChIP fragments
```{R}

#if (!requireNamespace("BiocManager", quietly = TRUE))
#    install.packages("BiocManager")
#BiocManager::install("EatonEtAlChIPseq", version = "3.8")

library(EatonEtAlChIPseq)

#data.path <- paste0(system.file(package = "EatonEtAlChIPseq"),"/extdata")
data.path <- dir(system.file(package = "EatonEtAlChIPseq"),pattern='extdata',full.names=T)
map.files <- list.files(data.path,pattern=".gz")
map.files

```
````

```
[1] "GSM424494_wt_G2_orc_chip_rep1_S288C_14.mapview.txt.gz" "GSM424494_wt_G2_orc_chip_rep2_S288C_14.mapview.txt.gz"
```

High-throughput sequencing of ChIP fragments

- The first command uses the system.file() function to have the path to the directory where the package is installed.

- It provides the path to the extdata directory in an OS-independent way.

- There are two mapview files in the directory, both of them compressed with Gzip algorithm (observe the “.gz” suffix).

- These files should not be uncompressed, as R has built in Gzip library to handle these compressed files the very same way, as normal text files.

High-throughput sequencing of ChIP fragments

- In case of high-throughput data, a considerable amount of disk space can be saved by storing analysis results in compressed formats.
- We will use only on of the available replicates in this example session.
- Aligned reads in mapview format can be read using the readAligned() function of ShortRead package. It can read a large variety of file formats, including those coming from Solexa Genome Alignment software, MAQ, Bowtie, and SOAP, or other software in BAM format.
- This offers an excellet opportunity to merge different data analysis pipelines at this point.
- The used file format should be specified with the typr parameter of the function.

```
179 ```{R}
180 #/Users/enriquevelazquez/Documents/R_working_directory/
181 #data.path <- '/home/ortutay/tmp/HTDA/T6_data/chipseq'
182 #data.path <- '/Users/enriquevelazquez/Documents/R_working_directory/'
183 library(ShortRead)
184
185 chip.aln <- readAligned(data.path, pattern="GSM424494_wt_G2_orc_chip_rep1_S", type="MAQMapview")
186
187 chip.aln
188
189 ```
```

```
class: AlignedRead
length: 478774 reads; width: 39 cycles
chromosome: S288C_14 S288C_14 ... S288C_14 S288C_14
position: 2 4 ... 784295 784295
strand: + - ... + +
alignQuality: IntegerQuality
alignData varLabels: nMismatchBestHit mismatchQuality nExactMatch24 nOneMismatch24
```

High-throughput sequencing of ChIP fragments

- The created data structure has 478774 reads, all of which are 39 nt long.
- The chromosome information where each of these reads mapped can be accessed with the chromosome() function.

````
190
191 ```{R}
192
193 head(chromosome(chip.aln))
194
195 ```
````

```
[1] S288C_14 S288C_14 S288C_14 S288C_14 S288C_14 S288C_14
Levels: S288C_14
```

````
196
197 ```{R}
198 levels(chromosome(chip.aln))
199 ```
````

```
[1] "S288C_14"
```

High-throughput sequencing of ChIP fragments

- All the reads in the dataset are mapped to chromosome 14 of Saccharomyces cerevisiae.
- Unfortunately, it is recorded with a nonstandard notation.
- We will update the notation to avoid conflicts when annotation data are attached.
- The renew() function allows the modification of data in different slots in AlignedRead data structures.

````
```{R}

new.chr <- chromosome(chip.aln)
levels(new.chr) <- "chrXIV"
chip.aln <- renew(chip.aln,chromosome=new.chr)
head(chromosome(chip.aln))

```
````

```
[1] chrXIV chrXIV chrXIV chrXIV chrXIV chrXIV
Levels: chrXIV
```

High-throughput sequencing of ChIP fragments

- The chipseq package offers basic calculation tools for analyzing ChIP-seq data, and many other packages use different operations from chipseq.
- The key functions of this package require aligned read data in Granges format.

````
```{R}

chip.ranges <- as(chip.aln,"GRanges")
ranges(chip.ranges)
mean(width(ranges(chip.ranges)))
```
````

```
IRanges object with 478774 ranges and 0 metadata columns:
                start       end     width
            <integer> <integer> <integer>
        [1]         2        40        39
        [2]         4        42        39
        [3]         5        43        39
        [4]         6        44        39
        [5]         6        44        39
        ...       ...       ...       ...
   [478770]    784295    784333        39
   [478771]    784295    784333        39
   [478772]    784295    784333        39
   [478773]    784295    784333        39
   [478774]    784295    784333        39
[1] 39
```

High-throughput sequencing of ChIP fragments

- The reads and some information about them are in a familiar format after the conversion.
- Now the data are prepared for the further analysis with the chipseq package.
- The first step of the analysis itself is to identify those fragments to which the target proteins were bound.
- It is very important to remember that the different fragmentation techniques produce 150-1000 nt long fragments in ChIP experiments, but NGS yields only short reads from the end of those fragments.
- The actual binding sites tend to be in the middle of the fragments, but the reads are located at the ends.
- For example, the reads in this example are uniformly 39 nt long, buth what might be the length of the fragments they present?
- The estimate.mean.fraglen() functions implement different algorithms for estimating the mean of fragment lengths in an experiment.

High-throughput sequencing of ChIP fragments

```
217 ```{R}
218
219 #if (!requireNamespace("BiocManager", quietly = TRUE))
220 #    install.packages("BiocManager")
221 #BiocManager::install("chipseq", version = "3.8")
222
223 library(chipseq)
224
225 estimate.mean.fraglen(chip.ranges,method ="coverage")
226 estimate.mean.fraglen(chip.ranges,method ="correlation")
227 ```
```

```
chrXIV
   110
chrXIV
   190
```

High-throughput sequencing of ChIP fragments

- These two approaches provide estimates between 100 and 200 nt. Based on this information, it is safe to assume that the genomic region beginning at 200 nt from the standard point of a read contains the actual binding site.
- To emulate this, the resize() function can be used to extend the read ranges from 39 nt to longer regions.

```
228
229 ```{R}
230 chip.ext.ranges <- resize(chip.ranges,width=200)
231
232 chip.ext.ranges
233 ```
```

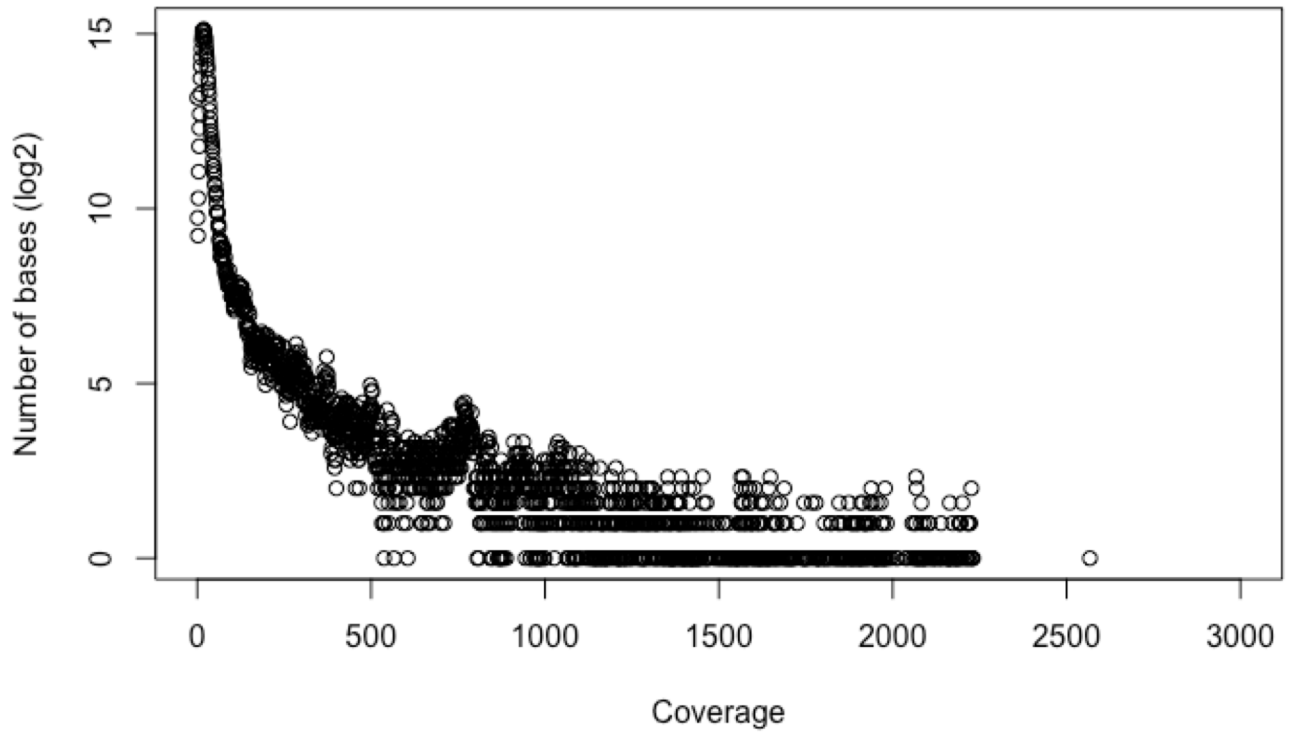
```
GRanges object with 478774 ranges and 5 metadata columns:
            seqnames        ranges strand |                      id nMismatchBestHit mismatchQuality nExactMatch24 nOneMismatch24
               <Rle>     <IRanges>  <Rle> |            <BStringSet>       <integer>       <integer>     <integer>      <integer>
       [1]    chrXIV         2-201      + | X8193_200:5:175:690:668               0               0             5              0
       [2]    chrXIV       -157-42      - |  X8193_200:5:62:612:145               4              19             5              0
       [3]    chrXIV         5-204      + | X8193_200:5:206:446:786               0               0             5              0
       [4]    chrXIV         6-205      + |  X8193_200:5:12:950:859               1               4             6              0
       [5]    chrXIV       -155-44      - | X8193_200:5:230:400:822               4              56             0              6
       ...       ...           ...    ... .                     ...             ...             ...           ...            ...
  [478770]    chrXIV 784295-784494      + |  X8193_200:5:81:727:781               3              26             3              1
  [478771]    chrXIV 784295-784494      + | X8193_200:5:111:213:419               1              11             3              1
  [478772]    chrXIV 784295-784494      + | X8193_200:5:114:891:790               3              34             3              1
  [478773]    chrXIV 784295-784494      + | X8193_200:5:151:159:782               6              35             0              1
  [478774]    chrXIV 784295-784494      + |  X8193_200:5:192:90:704               1               9             3              1
  -------
  seqinfo: 1 sequence from an unspecified genome; no seqlengths
```

High-throughput sequencing of ChIP fragments

- Observe that the first few reads on the negative stand of the genome are extended to negative chromosome coordinates.
- Fortunately, the subsequent analysis functions are prepared to handle this issue.
- The analysis procedure continues with calculating the coverage of each nucleotide on chromosome 14.
- The coverage of an nt represents the number o reads covering a given chromosomal position.
- It will help to identify long genomic ranges with high coverage representing putative binding sites.

High-throughput sequencing of ChIP fragments

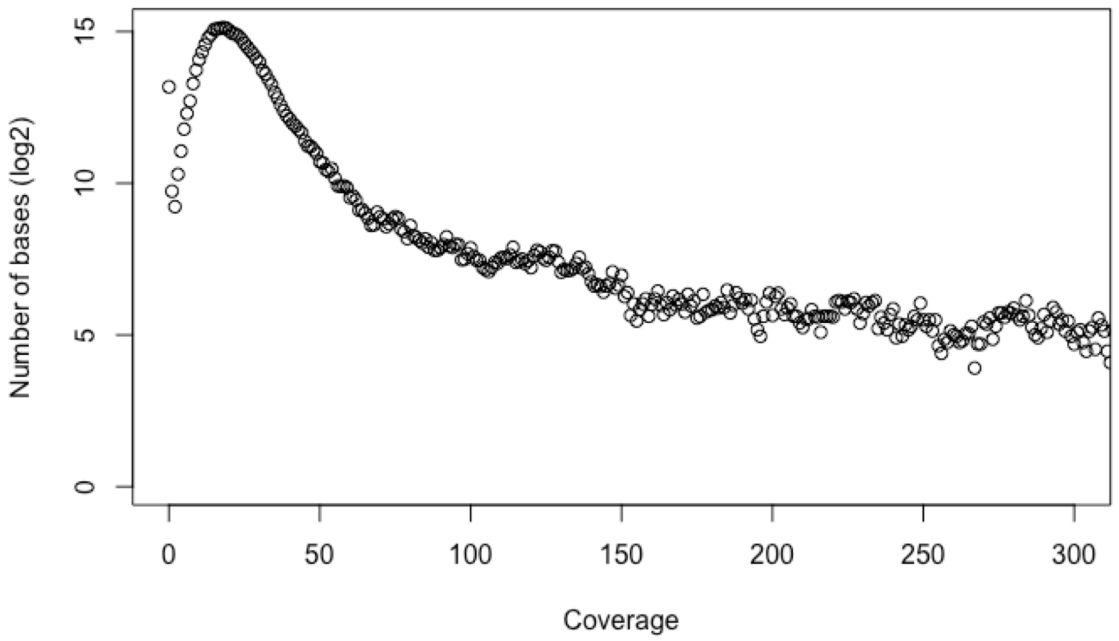
```
235 ```{R}
236 chip.cov <- coverage(chip.ext.ranges)
237
238 plot(as.numeric(dimnames(table(chip.cov))[[2]]),log2(as.numeric(table(chip.cov))),xlim=c(0,3000),xlab="Coverage",ylab="Number of bases (log2)")
239
240 ```
```



High-throughput sequencing of ChIP fragments

- The resulted coverage list can be employed for finding the high coverage regions.
- The slice() function can calculate their regions with a given lower threshold of coverage.

```
242 ```{R}
243
244 plot(as.numeric(dimnames(table(chip.cov))[[2]]),log2(as.numeric(table(chip.cov))),xlim=c(0,300),xlab="Coverage",ylab="Number of bases (log2)")
245 ```
```



High-throughput sequencing of ChIP fragments

- Almost the entire chromosome is covered by at least one read, so we have to use a more strict condition to identify a region as a peak.
- The peakCutoff() function helps us to estimate a reasonable lower threshold coverage to a specified false-discovery rate.

````
247 ```{R}
248 peakCutoff(chip.cov, fdr = 0.003)
249 ```
````

```
[1] 7.374084
```

High-throughput sequencing of ChIP fragments

- Since the peakCutoff() suggest a suitable cutoff between 7 and 8 coverage in this demonstration.
- Let us use the eight-times coverage as a minimum to define a region as IP peak.
- Generating the list of peaks.

```
```{R}
chip.peaks <- slice(chip.cov,lower=8)

chip.peaks$chrXIV

```
```

```
Views on a 784494-length Rle subject

views:
       start    end width
  [1]      1  24415 24415 [178 182 183 184 185 188 188 189 190 193 193 196 196 198 199 200 202 204 204 205 207 208 208 209 211 214 215 215 215 215 218 220 222 222 225 225 229 231 231 233 234 236 236 237 237 236 236 237 238 239 239 240 240 240 240 241 245 245 247 247 250 ...]
  [2]  24475  24481     7 [8 8 8 8 8 8 8]
  [3]  24485  29543  5059 [ 8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  9  9  9  9  9  9  8  9  9  9  9  9  8  8  8  8  8  8  8  8  8 10 11 11 11 11 11 11 11 11 11 11 12 13 13 13 13 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 15 15 16 16 ...]
  [4]  29565  29566     2 [8 8]
  [5]  29571  30229   659 [ 8  8  9  9  9  9  9  9  9 10 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 11 11 11 12 12 12 12 13 14 14 14 14 14 14 14 15 ...]
  [6]  30239  30250    12 [8 8 8 8 8 8 8 8 8 8 8 8]
  [7]  30319  31512  1194 [ 8  8  8  8  8  8  8  8  8  9  9  9  9  9  8  9  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 13 15 15 15 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 16 15 15 15 14 14 15 15 14 15 15 15 15 15 15 15 15 15 15 16 16 16 16 16 16 ...]
  [8]  31515  34757  3243 [ 8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  9  9  9  9  9  9 10 10 12 12 12 12 12 12 13 13 13 13 13 13 14 14 14 14 14 14 14 14 14 14 13 13 13 13 13 14 14 14 14 15 15 15 15 15 15 16 16 16 16 16 ...]
  [9]  34767  34982   216 [ 8  8  8  9  9 10 10 10 10 10 11 11 11 11 11 11 13 13 13 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 13 13 13 13 13 13 13 13 13 12 12 12 11 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 ...]
  ...    ...    ...   ... ...
[442] 755445 756678  1234 [ 8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  9 10 10 10 10  9 10 11 11 11 11 11 11 11 11 11 12 12 12 12 12 13 13 13 13 13 13 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 16 15 15 14 14 14 14 14 14 14 14 ...]
[443] 756789 765534  8746 [11 12 11 13 13 13 13 13 13 13 13 14 15 17 17 17 17 17 17 17 17 17 17 17 17 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 18 19 19 19 19 19 19 19 19 19 20 20 20 21 21 21 21 21 21 21 21 21 21 21 21 21 21 21 ...]
[444] 772617 772819   203 [ 9 10 11 12 12 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 14 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 14 14 14 14 14 ...]
[445] 773133 773290   158 [ 8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8 ...]
[446] 773927 774126   200 [ 8  8  9  9  9  9  9  9  9 10 10 10 10 11 11 11 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 13 12 ...]
[447] 775069 775084    16 [8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8]
[448] 775090 775769   680 [ 8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8  8 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11 12 12 12 12 12 12 12 12 13 13 13 13 13 13 13 13 13 13 13 13 ...]
[449] 775799 775863    65 [8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8]
[450] 778951 784492  5542 [ 8  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9  9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 11 ...]
```

High-throughput sequencing of ChIP fragments

- There are 450 peak candidates, ech of them contains nucleotides with at least 8x coverage.
- To generate some further statistics about these DNA segments, we can employ the peakSummary() function.

```
258 ```{R}
259 peak.ranges <- peakSummary(chip.peaks)
260 peak.ranges
261
262 ```
```

```
RangedData with 450 rows and 3 value columns across 1 space
       space        ranges   |       max    maxpos       sum
    <factor>     <IRanges>   | <integer> <integer> <integer>
1     chrXIV       1-24415   |      1983      7172   2331050
2     chrXIV   24475-24481   |         8     24478        56
3     chrXIV   24485-29543   |       113     28860    176994
4     chrXIV   29565-29566   |         8     29565        16
5     chrXIV   29571-30229   |        27     29836     12057
6     chrXIV   30239-30250   |         8     30244        96
7     chrXIV   30319-31512   |        36     30765     22206
8     chrXIV   31515-34757   |        34     33726     61711
9     chrXIV   34767-34982   |        14     34803      2645
...      ...           ... ...       ...       ...       ...
442   chrXIV 755445-756678   |        34    756016     28728
443   chrXIV 756789-765534   |       378    764123    490823
444   chrXIV 772617-772819   |        16    772763      2831
445   chrXIV 773133-773290   |        10    773267      1349
446   chrXIV 773927-774126   |        13    773987      2254
447   chrXIV 775069-775084   |         8    775076       128
448   chrXIV 775090-775769   |        26    775325     10493
449   chrXIV 775799-775863   |         8    775831       520
450   chrXIV 778951-784492   |      2231    784190   1849017
```

High-throughput sequencing of ChIP fragments

- Table 8.1 contains the maximum coverage, the position of the maximum coverage, and accumulated coverage sum for these regions.
- For some data inspection and visualization approaches, it is much easier to store this information in a simple data frame, and not in a complex RangeData object.
- For example, in this format one can immediately observe that there is a region in this dataset that distorts all subsequent statistics, as it has an unexpectedly high coverage.

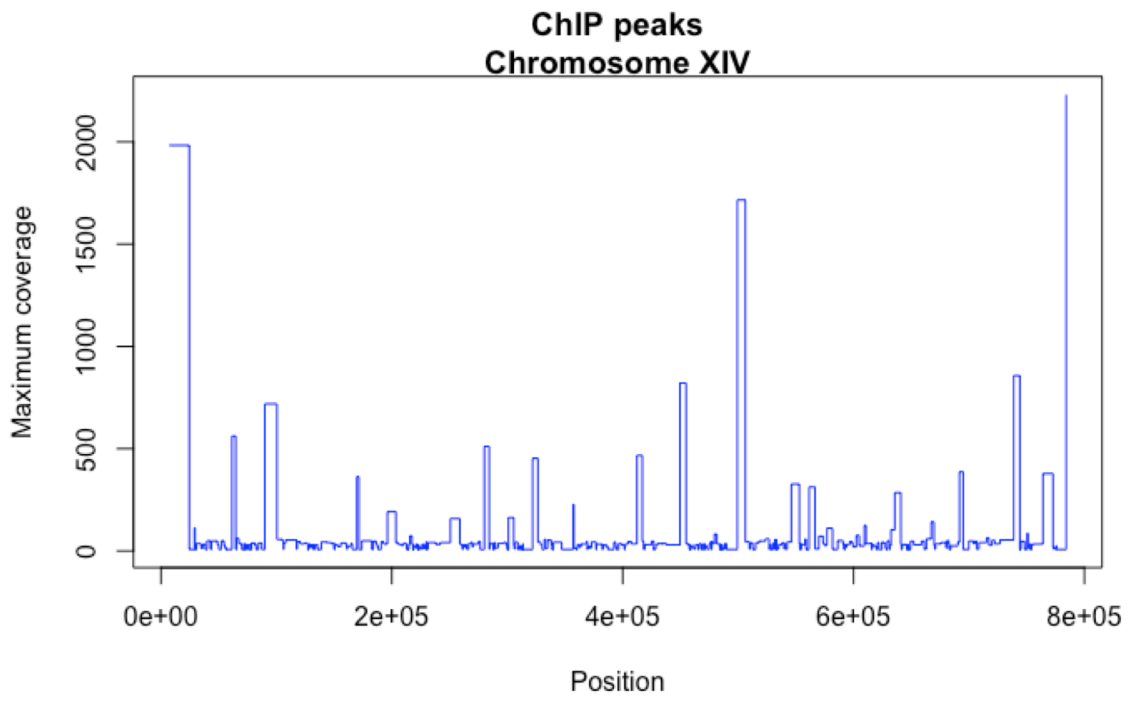
````
264 ```{R}
265 peak.ranges.df <- as.data.frame(peak.ranges)
266 max(peak.ranges.df$max)
267
268 ```
````

```
[1] 350399
```

High-throughput sequencing of ChIP fragments

- It might be an artifact, or there might be other reasons for this outstanding coverage.

```
270 ```{R}
271 rougepeak<-which(max(peak.ranges.df$max)==peak.ranges.df$max)
272 peak.ranges.df <- peak.ranges.df[-rougepeak,]
273
274 plot(peak.ranges.df$maxpos,peak.ranges.df$max,type="s",main="ChIP peaks\nChromosome XIV",xlab="Position",ylab="Maximum coverage",col="blue")
275
276 ```
```



High-throughput sequencing of ChIP fragments

- Now we have an excellent overview of the peaks along chromosome 14.
- There are some more and many less dominant peaks.
- One of the next possible steps could be the investigation of the individual peaks to spot artifacts.
- Reads should come from the positive and the negative genomic DNA strand with equal probability, and they should have a roughly similar distribution along the region.
- One can easily investigate these aspects by separating the + and – strand reads and plot them in individual peak regions.

~~~
```{R}
cov.pos <- coverage(chip.ext.ranges[strand(chip.ext.ranges)=='+'])
cov.neg <- coverage(chip.ext.ranges[strand(chip.ext.ranges)=='-'])

peaks.pos <- Views(cov.pos, ranges(chip.peaks))
peaks.neg <- Views(cov.neg, ranges(chip.peaks))

peaks.pos$chrXIV

```
~~~

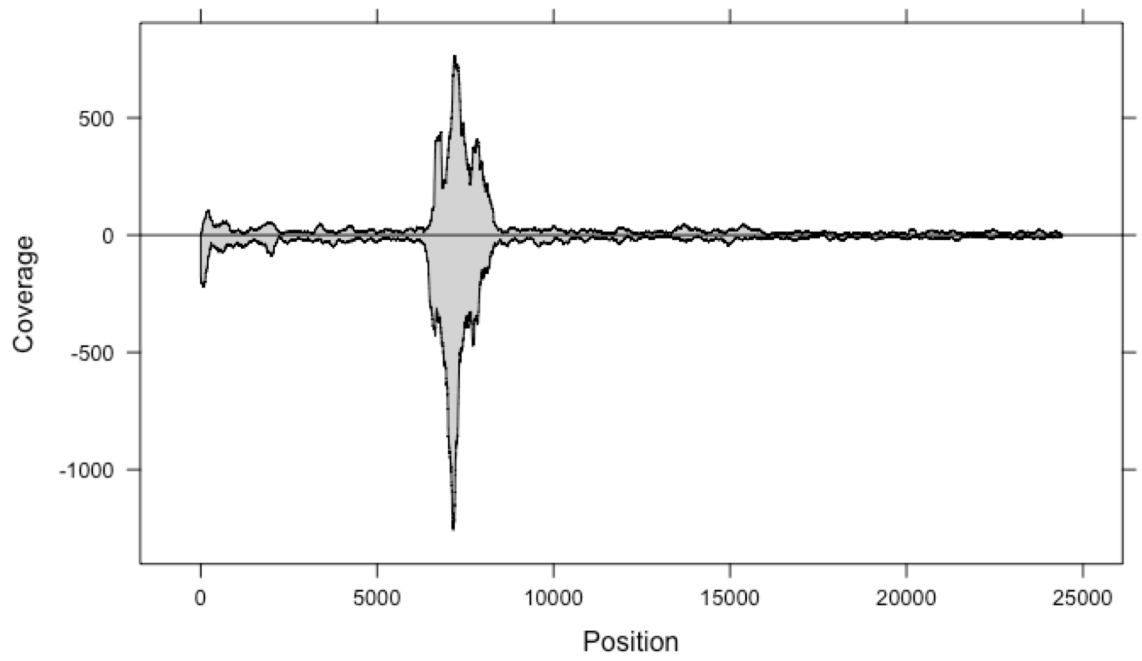
```
Views on a 784494-length Rle subject

views:
        start    end width
  [1]       1  24415 24415 [ 0 1 1 1 2 3 3 3 3 3 3 4 4 4 4 4 5 5 5 5 7 8 8 8 8 11 12 12 12 12 14 15 16 16 19 19 20 22 22 23 24 26 27 28 29 29 30 31 33 33 34 35 35 35 35 35 35 35 37 39 41 ...]
  [2]   24475  24481     7 [1 1 1 1 1 1 1]
  [3]   24485  29543  5059 [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 ...]
  [4]   29565  29566     2 [3 3]
  [5]   29571  30229   659 [ 4 4 5 5 5 5 5 5 5 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 9 9 9 9 9 9 9 10 ...]
  [6]   30239  30250    12 [7 7 7 7 7 7 7 7 7 7 7 7]
  [7]   30319  31512  1194 [ 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 9 11 11 11 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 12 11 11 11 10 10 11 11 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 ...]
  [8]   31515  34757  3243 [ 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 8 8 8 8 8 8 9 9 9 9 9 ...]
  [9]   34767  34982   216 [4 4 4 5 5 6 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 6 6 6 6 6 6 6 6 6 6 5 5 5 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 ...]
  ...     ...    ...   ... ...
[442]  755445 756678  1234 [2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...]
[443]  756789 765534  8746 [ 3 4 4 4 4 4 4 4 4 4 4 5 6 6 6 6 6 6 6 6 6 6 6 6 6 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 8 ...]
[444]  772617 772819   203 [ 7 8 9 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 10 ...]
[445]  773133 773290   158 [4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 6 6 6 6 6 6 6 6 6 ...]
[446]  773927 774126   200 [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ...]
[447]  775069 775084    16 [5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5]
[448]  775090 775769   680 [ 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 5 ...]
[449]  775799 775863    65 [7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7 7]
[450]  778951 784492  5542 [ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 ...]
```

High-throughput sequencing of ChIP fragments

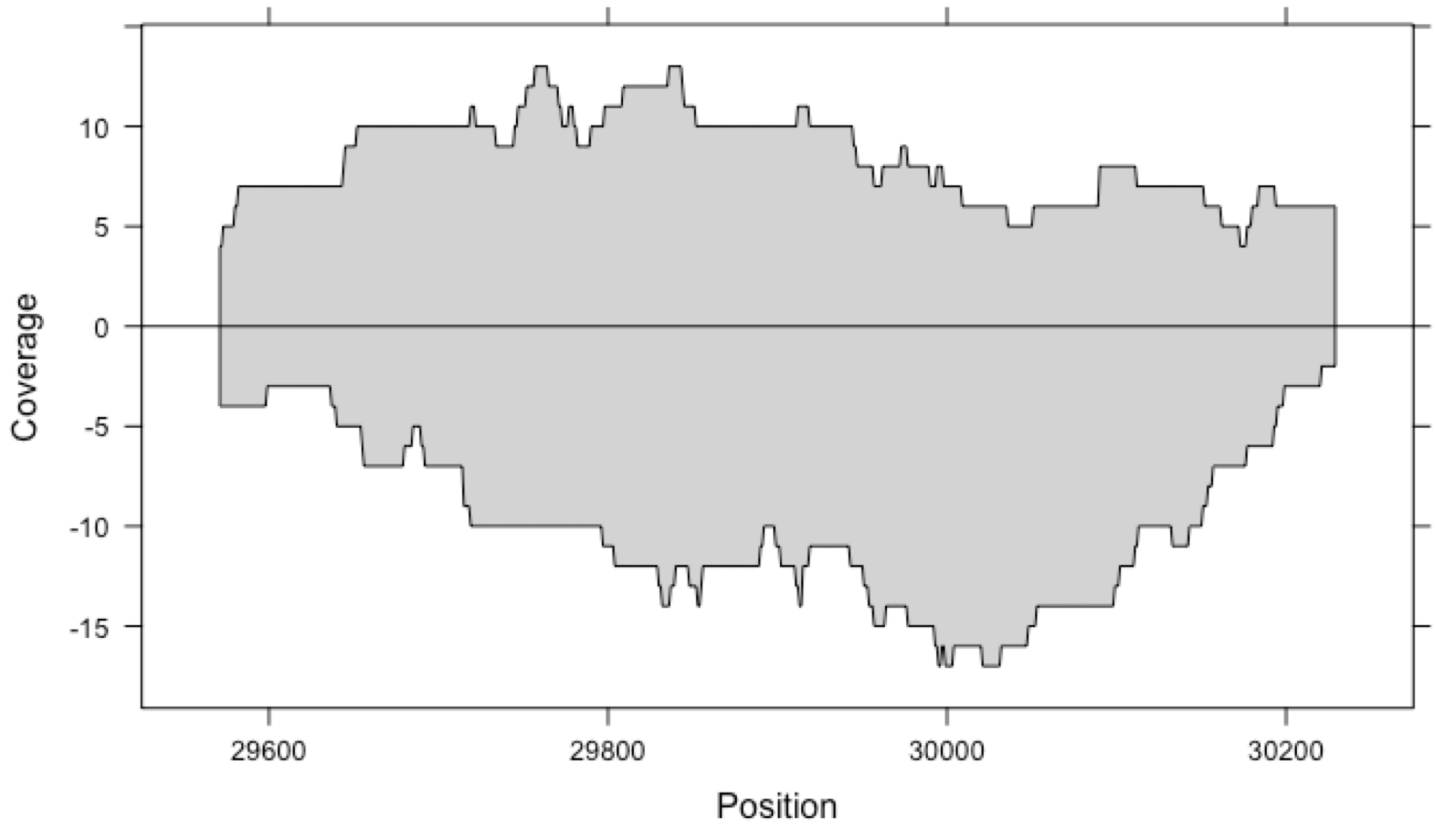
- Basically the previous procedure is replicated here on the reads of the two separate stands.
- The coverageplot() function can be employed to draw individual peaks and the coverage histograms so that the reads of the + stand are represented on the upper half of the plot, while reads originating from the – strand are on the bottom part.

```
289 ```{R}
290
291 coverageplot(peaks.pos$chrXIV[1],peaks.neg$chrXIV[1])
292
293 ```
```



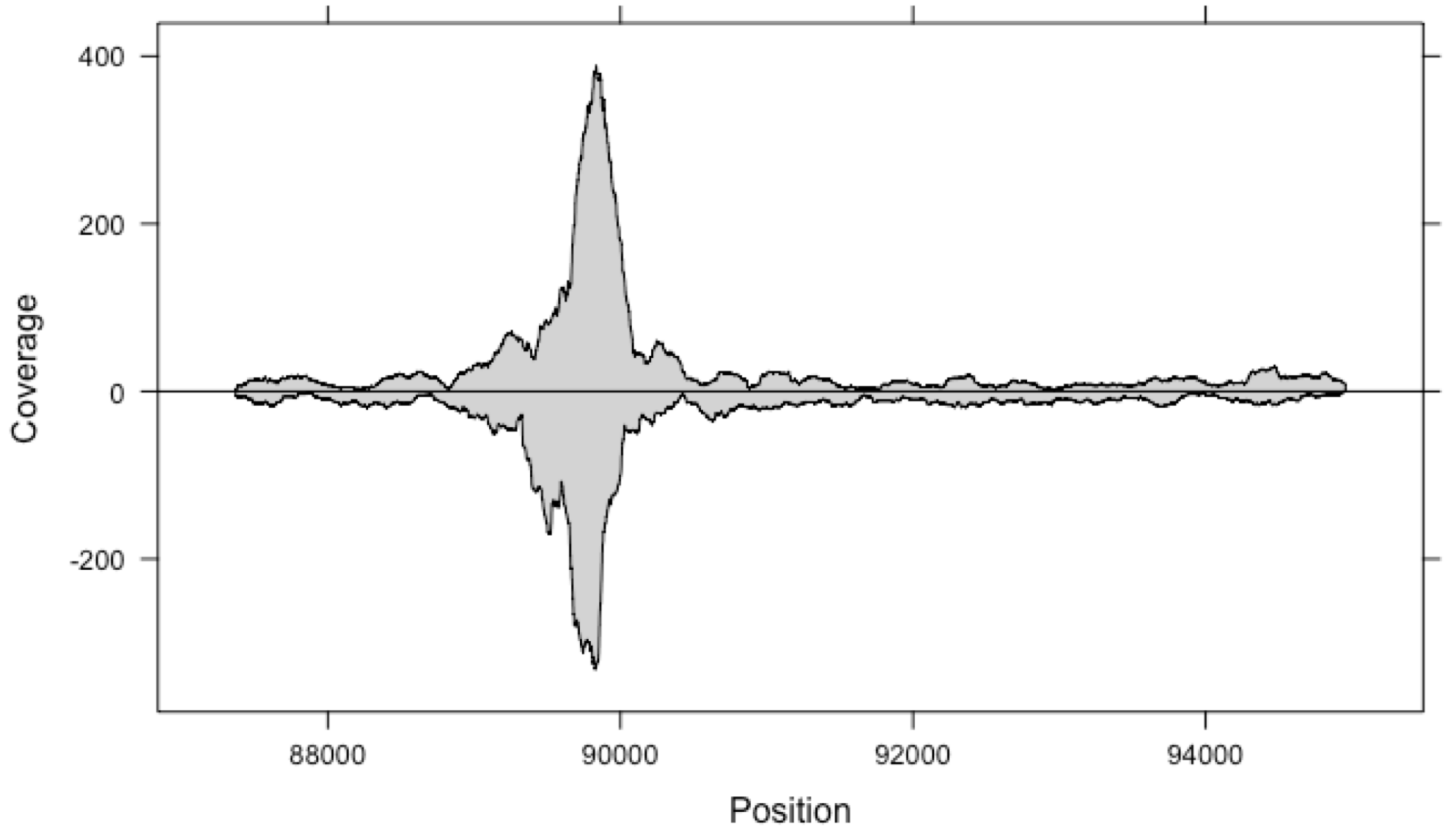
High-throughput sequencing of ChIP fragments

```
295 ```{R}
296
297 coverageplot(peaks.pos$chrXIV[5],peaks.neg$chrXIV[5])
298 ```
```



High-throughput sequencing of ChIP fragments

```
300 ```{R}
301
302 coverageplot(peaks.pos$chrXIV[47],peaks.neg$chrXIV[47])
303 ```
```



High-throughput sequencing of ChIP fragments

- The figure shows that most of the peaks are nice and symmetric; however, some of them are represented by only a handful of reads.
- In the subsequent analysis, we want to focus only on those, which have tall preaks with high read numbers.
- By employing the lower parameter of the slice() function, one can find a limit that produces a desired number of reads.

```
310 ```{R}
311
312 chip.high.peaks <- slice(chip.cov,lower=150)
313 ```
314
315 ```{R}
316 chip.sel.peaks <- peakSummary(chip.high.peaks$chrXIV[width(peakSummary(chip.high.peaks))>100])
317
318 chip.sel.peaks
319 ```
```

```
RangedData with 26 rows and 3 value columns across 1 space
       space        ranges |       max    maxpos       sum
    <factor>     <IRanges> | <integer> <integer> <integer>
1          1         1-261 |       285        96     60429
2          1     6452-8295 |      1983      7172   1454805
3          1   61278-61891 |       562     61522    230848
4          1   89394-90052 |       719     89834    248001
5          1 169521-169928 |       363    169711    108088
6          1 196049-196160 |       192    196104     20107
7          1 279578-280348 |       511    280092    232479
8          1 300972-301130 |       163    301073     25026
9          1 321577-322158 |       453    321920    174458
...      ...          ... ...       ...       ...       ...
18         1 545923-546424 |       327    546121    125455
19         1 560747-560868 |       188    560808     21008
20         1 561231-561792 |       313    561452    121825
21         1 635495-635721 |       285    635645     55198
22         1 635769-635976 |       229    635935     39015
23         1 691546-691970 |       387    691715    121786
24         1 737874-739387 |       857    738740    691885
25         1 763761-764625 |       378    764123    206480
26         1 782526-784469 |      2231    784190   1666690
```

High-throughput sequencing of ChIP fragments

- Additionally, let us keep only those peaks with a length of at least 100 nt.
- Since the estimated fragment length is 200 nt, a length of 100 is a safe lowest size for a real fragment.
- We can assume that the regions covered by the peaks are really coming from immunoprecipitated genomic fragments.
- This list contains 26 peaks, which is an appropriate size for further statistics.
- This list will be now converted to a Granges object for subsequent analysis.

```
RangedData with 26 rows and 3 value columns across 1 space
       space        ranges   |       max    maxpos       sum
    <factor>     <IRanges>   | <integer> <integer> <integer>
1          1         1-261   |       285        96     60429
2          1     6452-8295   |      1983      7172   1454805
3          1   61278-61891   |       562     61522    230848
4          1   89394-90052   |       719     89834    248001
5          1 169521-169928   |       363    169711    108088
6          1 196049-196160   |       192    196104     20107
7          1 279578-280348   |       511    280092    232479
8          1 300972-301130   |       163    301073     25026
9          1 321577-322158   |       453    321920    174458
...      ...           ... ...       ...       ...       ...
18         1 545923-546424   |       327    546121    125455
19         1 560747-560868   |       188    560808     21008
20         1 561231-561792   |       313    561452    121825
21         1 635495-635721   |       285    635645     55198
22         1 635769-635976   |       229    635935     39015
23         1 691546-691970   |       387    691715    121786
24         1 737874-739387   |       857    738740    691885
25         1 763761-764625   |       378    764123    206480
26         1 782526-784469   |      2231    784190   1666690
```

```
321 ```{R}
322 bind.sites <- GRanges(seqnames='chrXIV',ranges=unlist(ranges(chip.sel.peaks)),strand='*')
323 ```
```