Bioinformatic approaches to regulatory genomics and epigenomics

376-1347-00L - 2022 | week 03

Pierre-Luc Germain

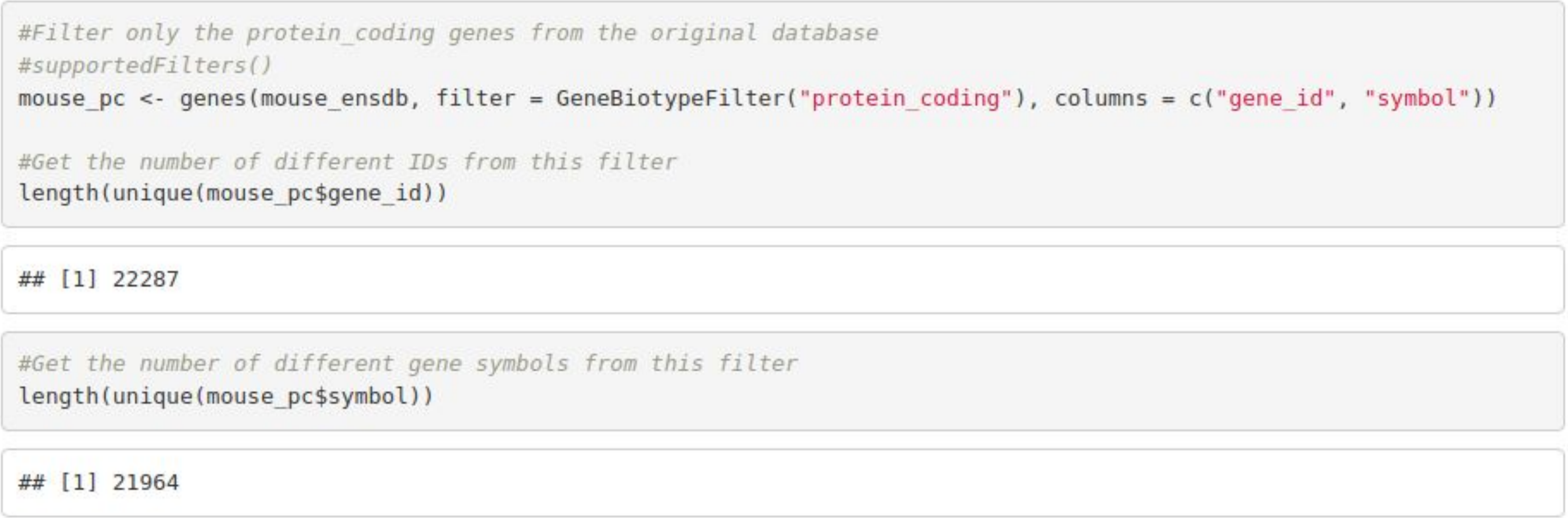
ETH Zürich

Plan for today

- Debriefing on the assignments

- Overview of NGS basic analysis pipelines, file formats, etc.

How many protein-coding gene IDs, and how many gene symbols, does the mouse annotation have?



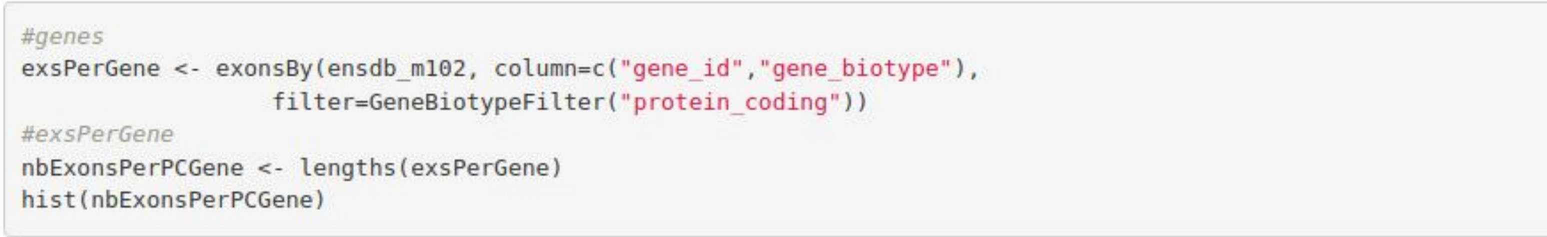
```r
#Filter only the protein_coding genes from the original database
#supportedFilters()
mouse_pc <- genes(mouse_ensdb, filter = GeneBiotypeFilter("protein_coding"), columns = c("gene_id", "symbol"))

#Get the number of different IDs from this filter
length(unique(mouse_pc$gene_id))
```

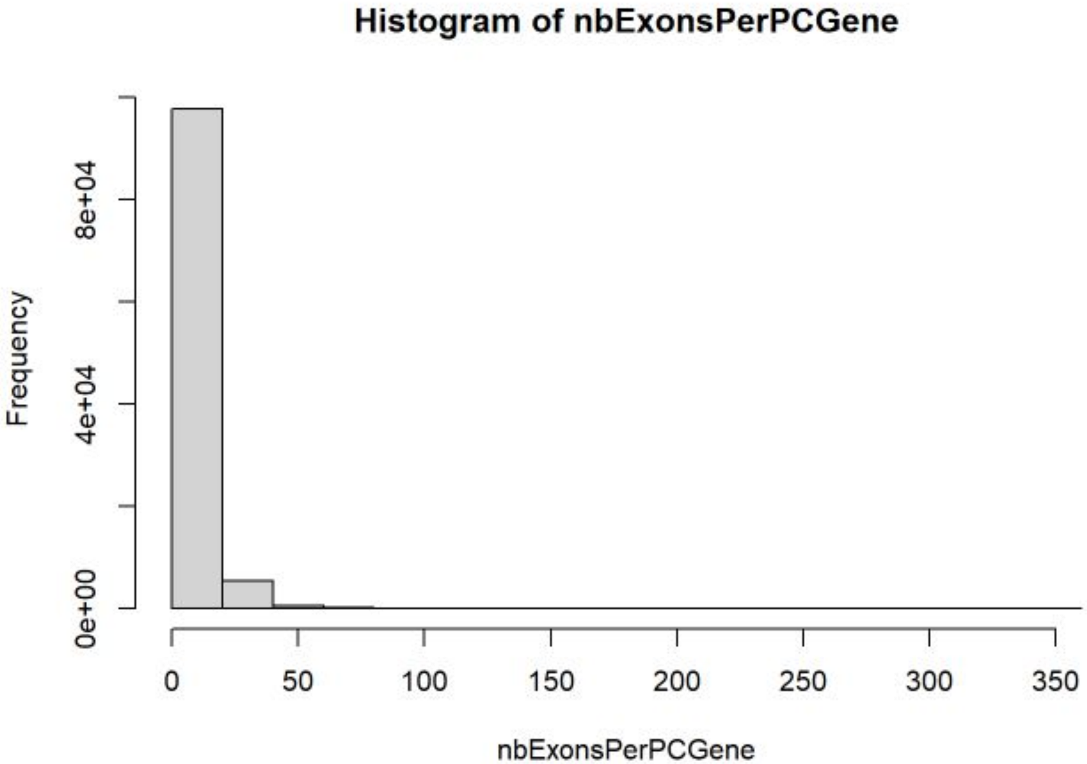
```
## [1] 22287
```

```r
#Get the number of different gene symbols from this filter
length(unique(mouse_pc$symbol))
```

```
## [1] 21964
```

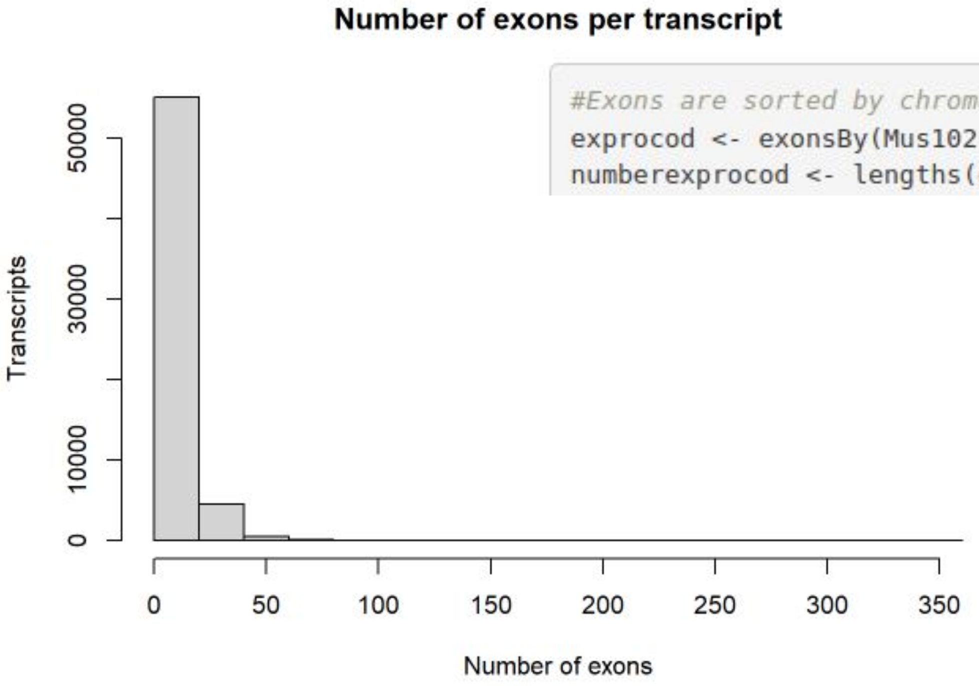


```
#genes
exsPerGene <- exonsBy(ensdb_m102, column=c("gene_id","gene_biotype"),
                  filter=GeneBiotypeFilter("protein_coding"))
#exsPerGene
nbExonsPerPCGene <- lengths(exsPerGene)
hist(nbExonsPerPCGene)
```

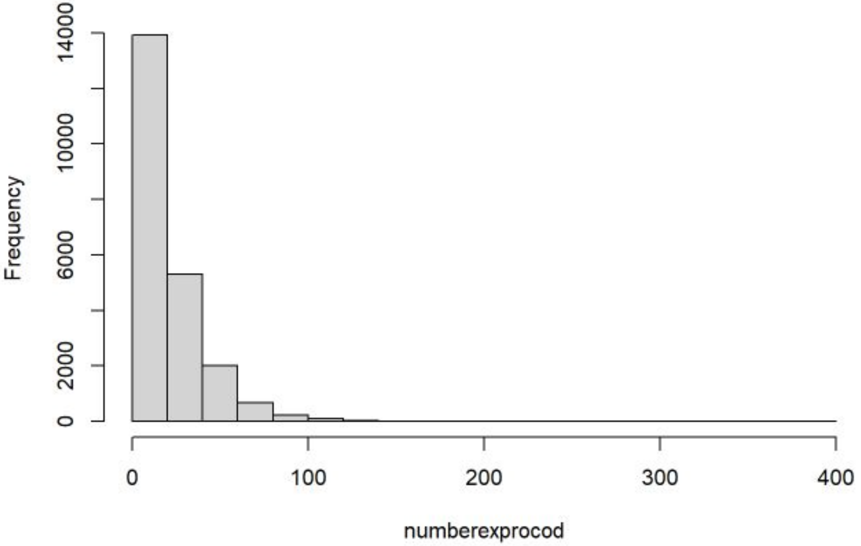


Plot the distribution of the number of exons for protein-coding genes

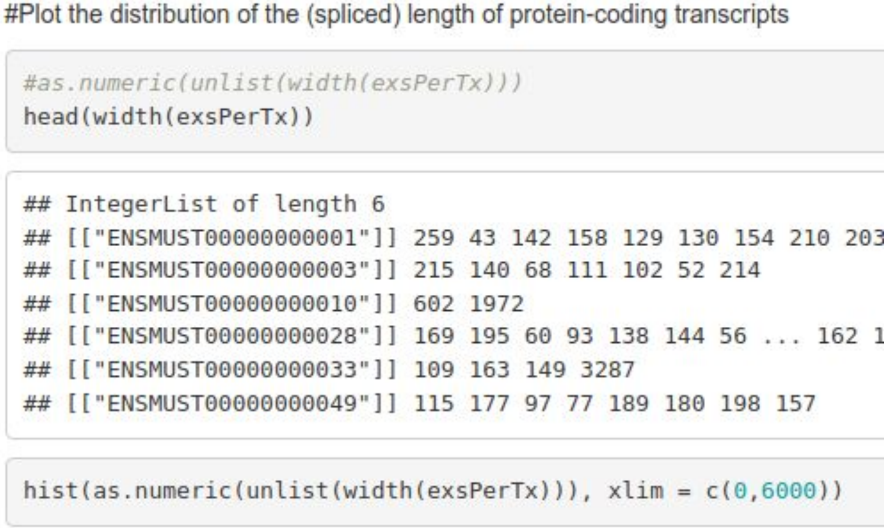
```
exsPerTx <- exonsBy(ensdb, column=c("tx_id","tx_biotype"),
                    filter=TxBiotypeFilter("protein_coding"))
nbExonsPerPCtx <- lengths(exsPerTx)
hist(nbExonsPerPCtx, main="Number of exons per transcript",
     xlab="Number of exons", ylab="Transcripts")
```



```
#Exons are sorted by chromosome, strand, start and end values by using by = gene
exprocod <- exonsBy(Mus102, by = "gene", filter=GeneBiotypeFilter("protein_coding"))
numberexprocod <- lengths(exprocod)
```



Plot the distribution of the (spliced) length of protein-coding transcripts

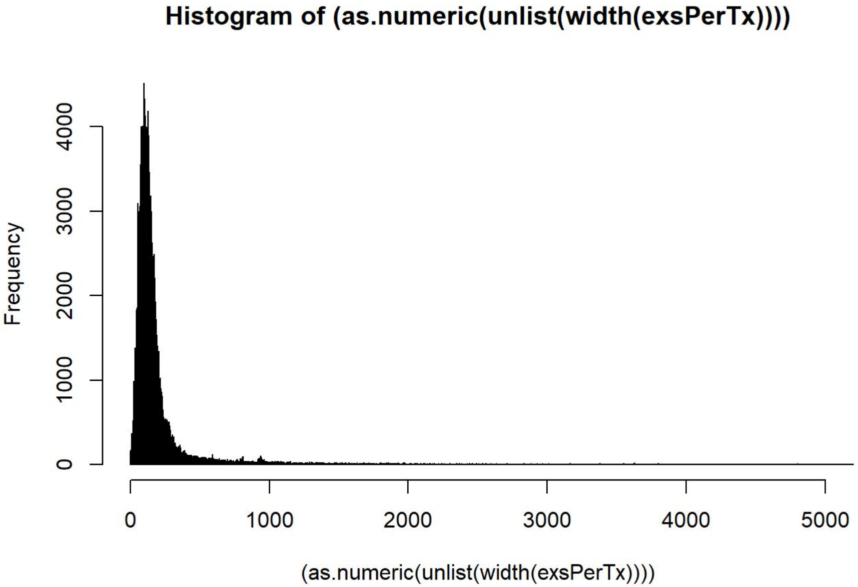


#Plot the distribution of the (spliced) length of protein-coding transcripts

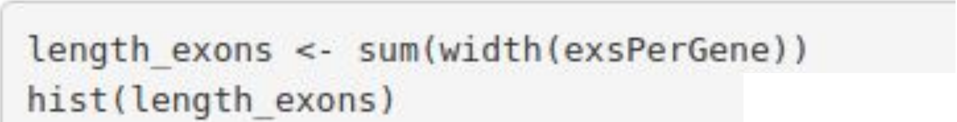
```
#as.numeric(unlist(width(exsPerTx)))
head(width(exsPerTx))
```

```
## IntegerList of length 6
## [["ENSMUST00000000001"]] 259 43 142 158 129 130 154 210 203
## [["ENSMUST00000000003"]] 215 140 68 111 102 52 214
## [["ENSMUST00000000010"]] 602 1972
## [["ENSMUST00000000028"]] 169 195 60 93 138 144 56 ... 162 1
## [["ENSMUST00000000033"]] 109 163 149 3287
## [["ENSMUST00000000049"]] 115 177 97 77 189 180 198 157
```

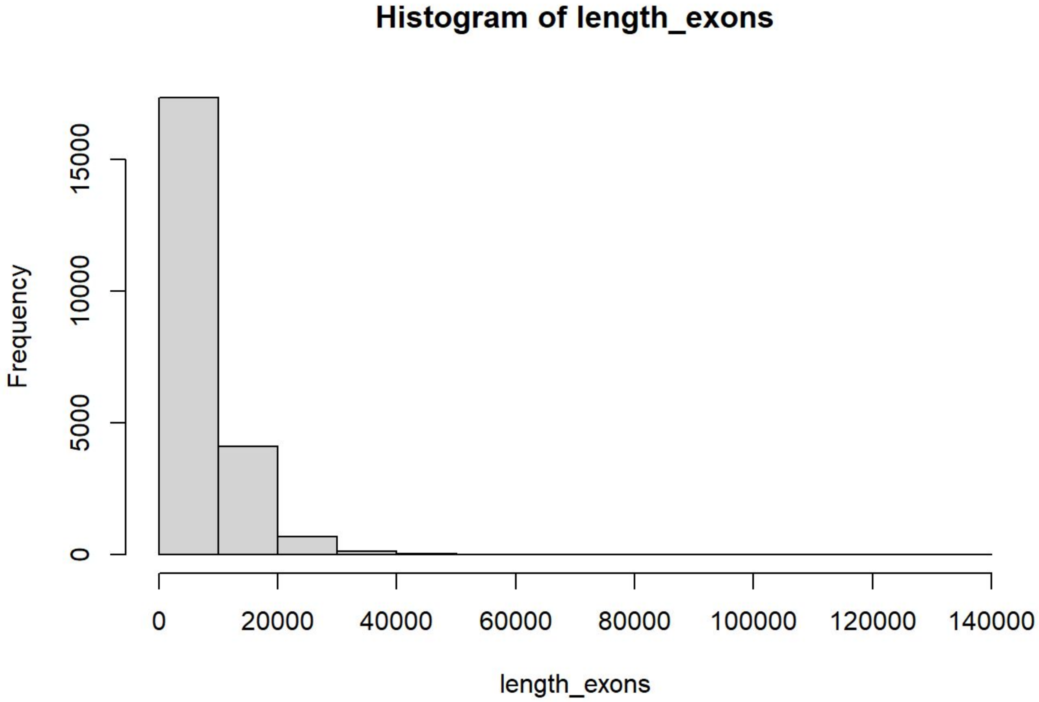
```
hist(as.numeric(unlist(width(exsPerTx))), xlim = c(0,6000))
```



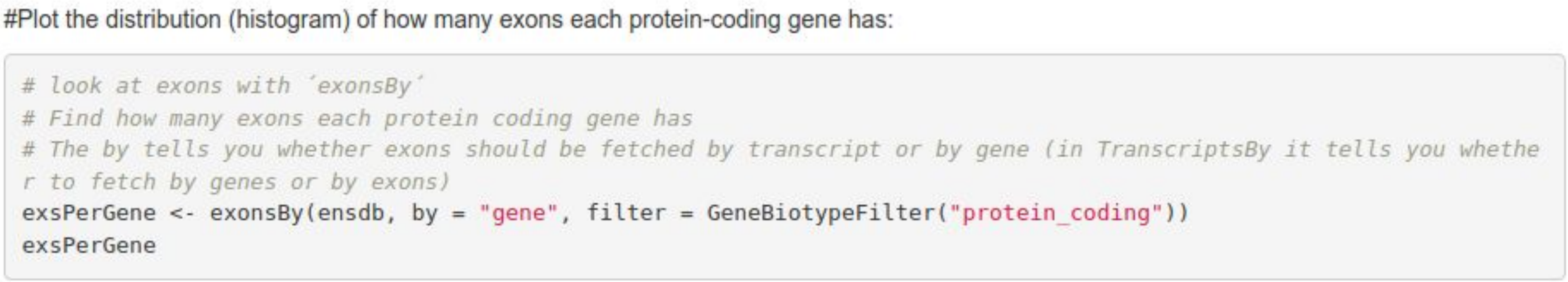
Plot the distribution of the (spliced) length of protein-coding transcripts



```
length_exons <- sum(width(exsPerGene))
hist(length_exons)
```



misc…

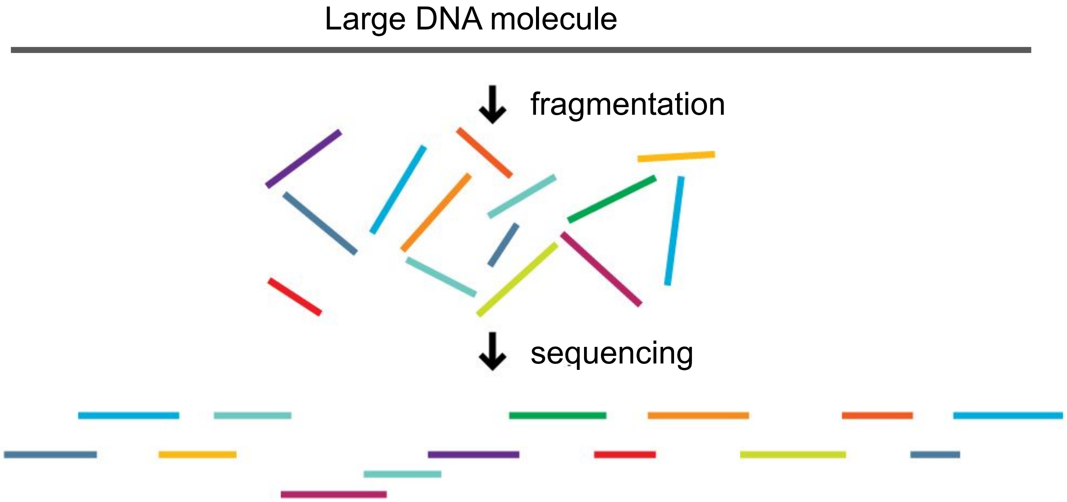


#Plot the distribution (histogram) of how many exons each protein-coding gene has:

```
# look at exons with ´exonsBy´
# Find how many exons each protein coding gene has
# The by tells you whether exons should be fetched by transcript or by gene (in TranscriptsBy it tells you whethe
r to fetch by genes or by exons)
exsPerGene <- exonsBy(ensdb, by = "gene", filter = GeneBiotypeFilter("protein_coding"))
exsPerGene
```

Next Generation Sequencing (NGS)

Shotgun sequencing:



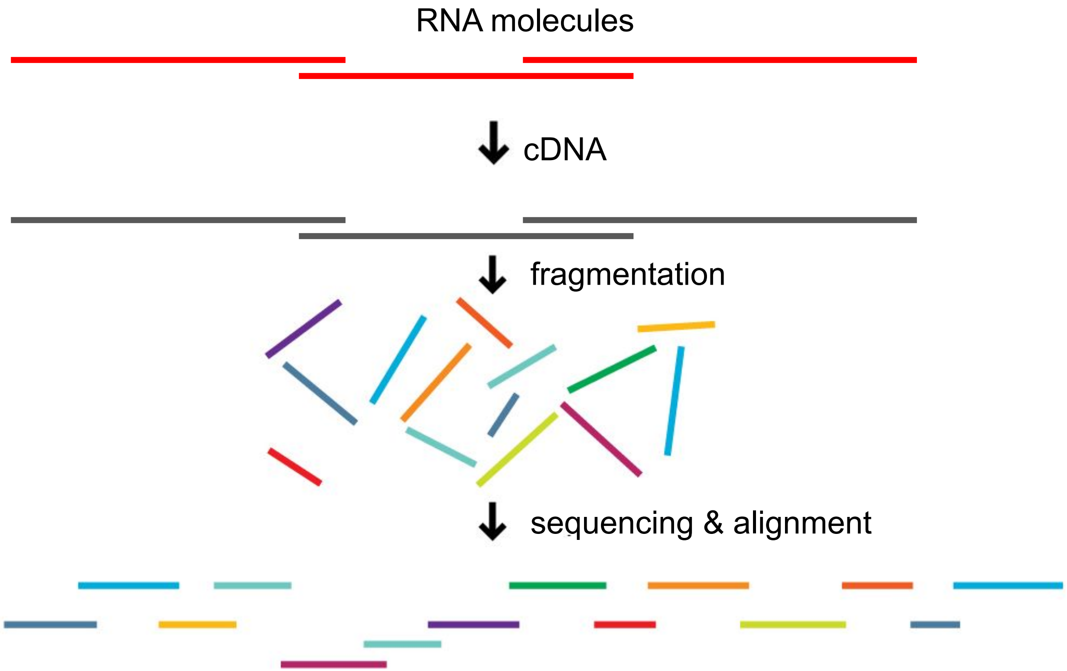
Large DNA molecule

fragmentation

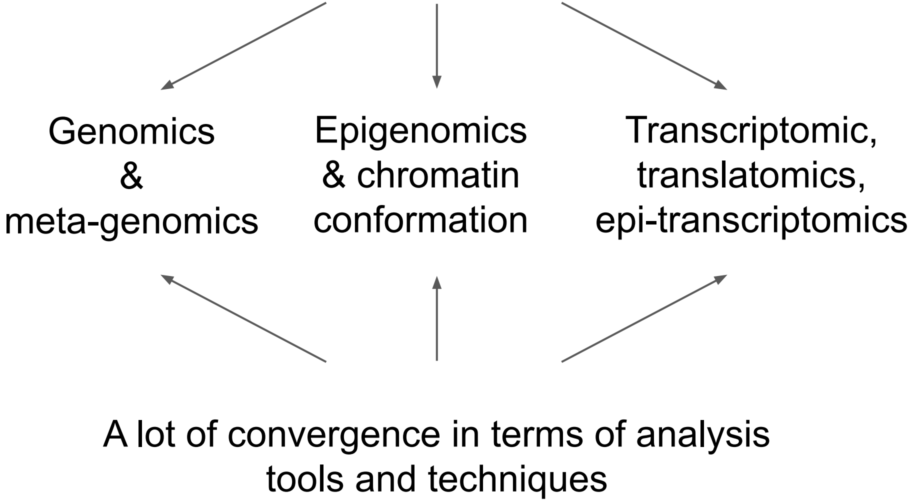
sequencing

Next Generation Sequencing (NGS)

RNA sequencing:



Next Generation Sequencing:
one technology to rule them all

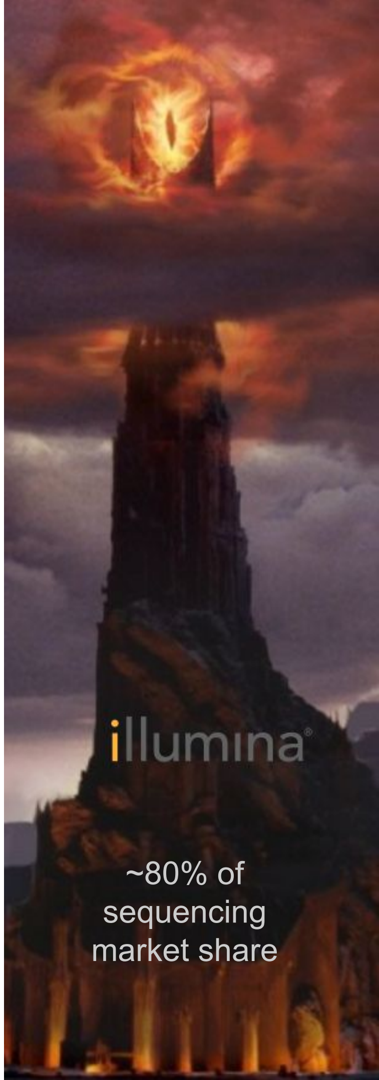


Genomics
&
meta-genomics

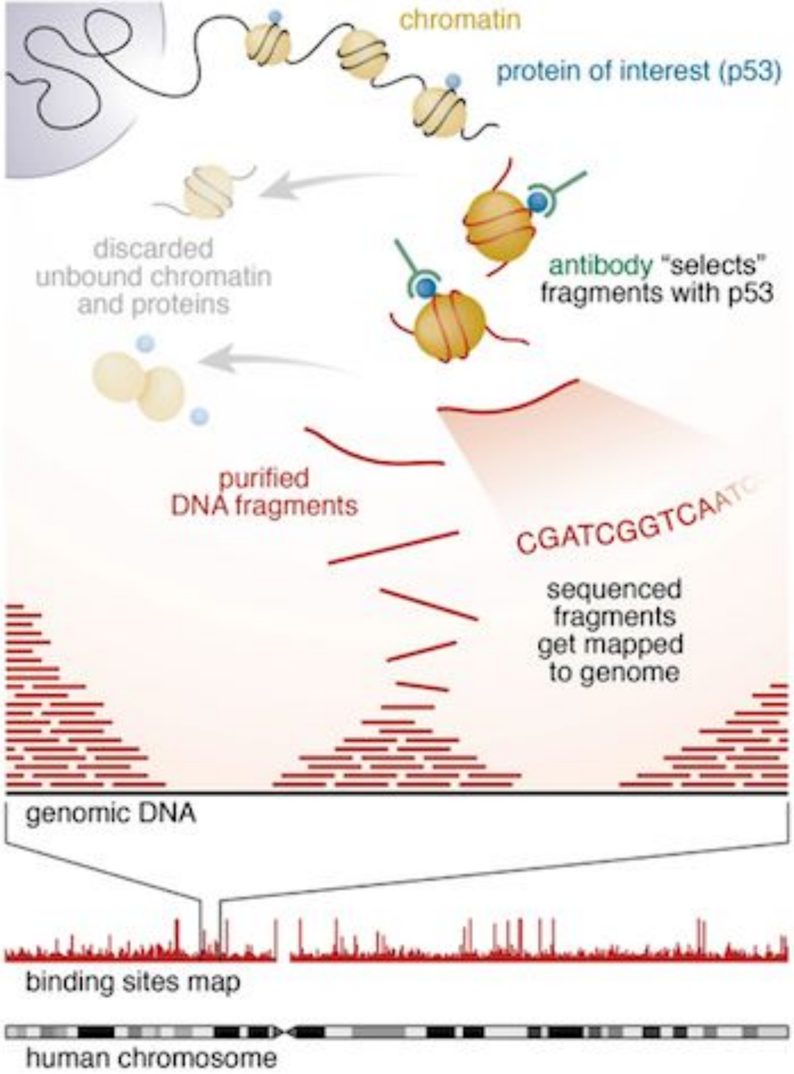
Epigenomics
& chromatin
conformation

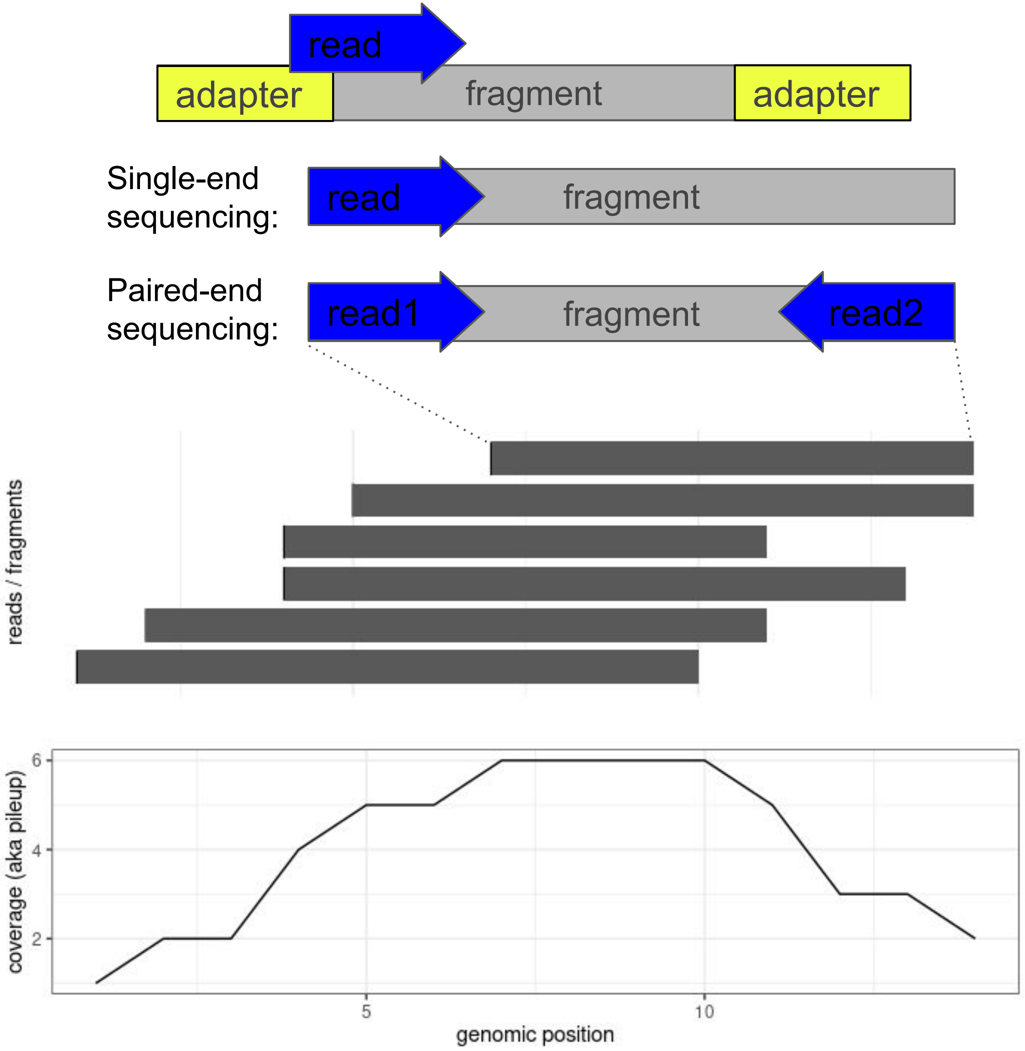
Transcriptomic,
translatomics,
epi-transcriptomics

A lot of convergence in terms of analysis
tools and techniques

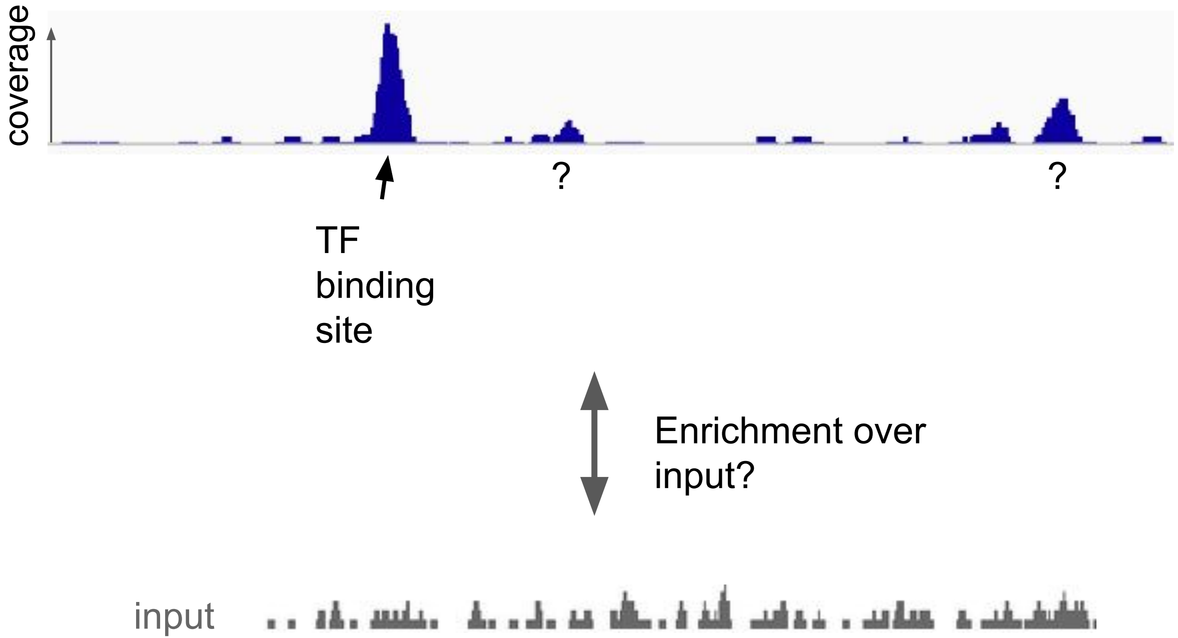


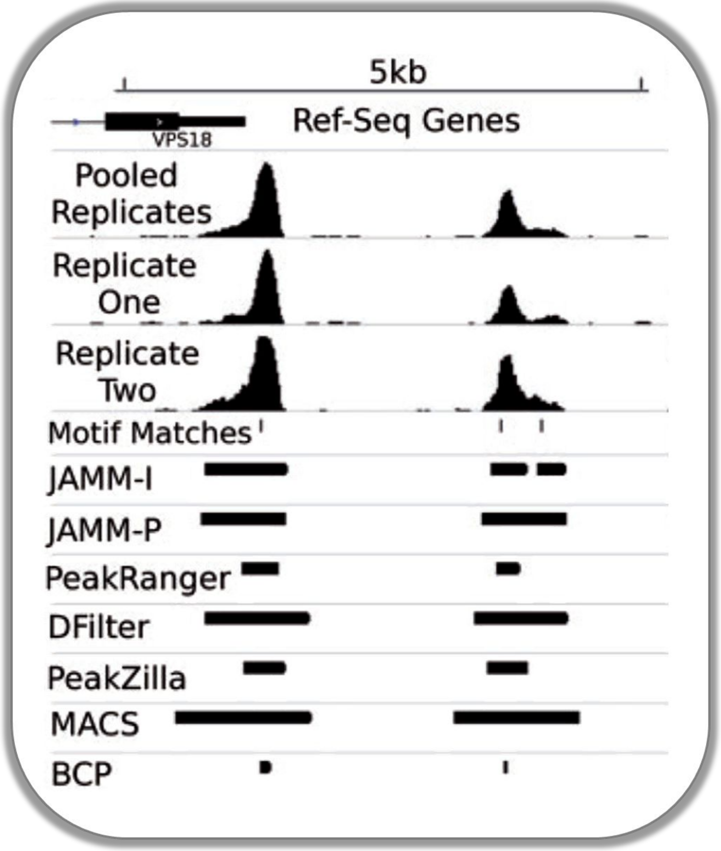
~80% of
sequencing
market share


chromatin
protein of interest (p53)
discarded unbound chromatin and proteins
antibody "selects" fragments with p53
purified DNA fragments
CGATCGGTCAATC
sequenced fragments get mapped to genome
genomic DNA
binding sites map
human chromosome

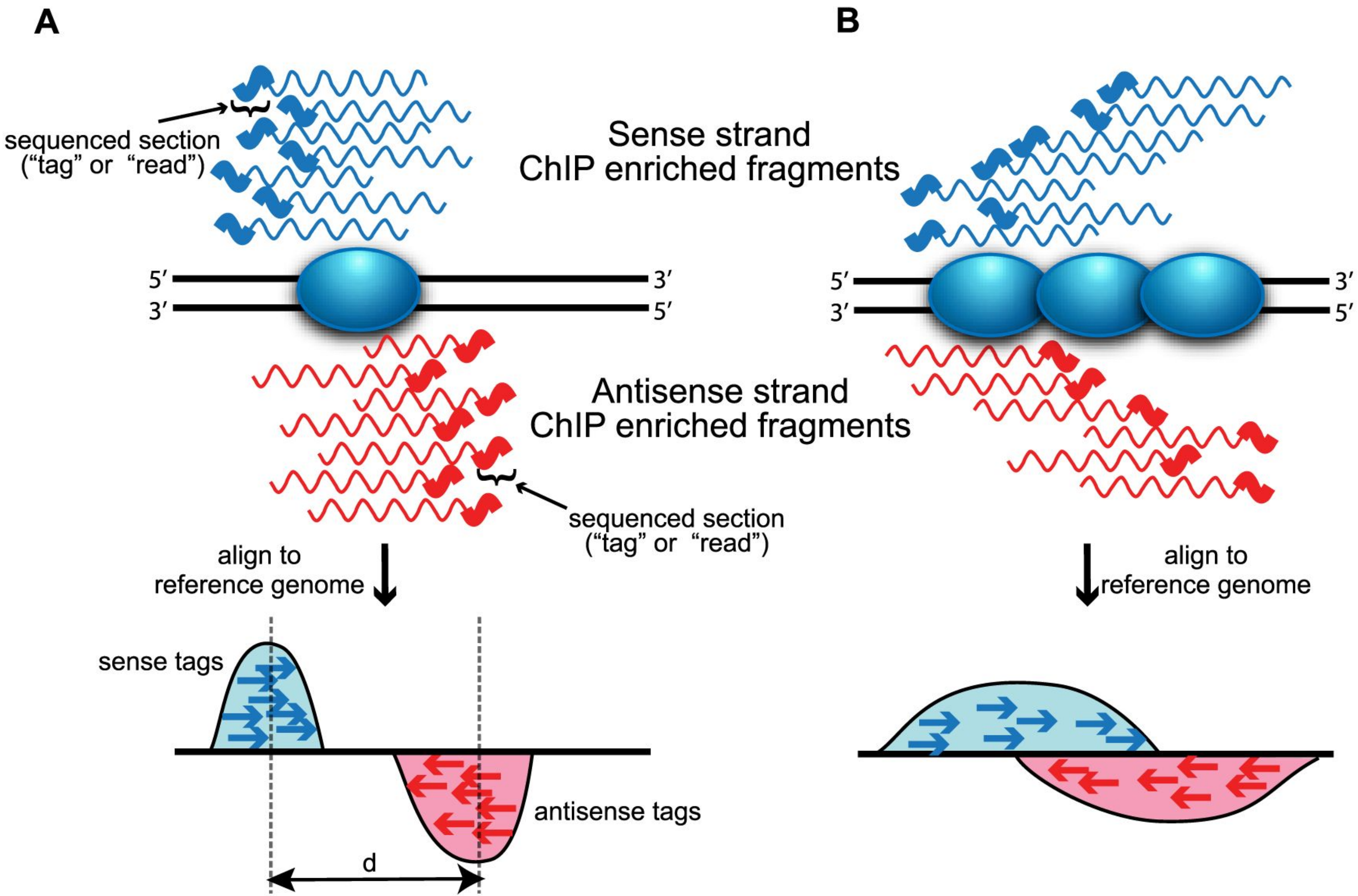

read
adapter
fragment
adapter
Single-end sequencing:
read
fragment
Paired-end sequencing:
read1
fragment
read2
reads / fragments
coverage (aka pileup)
6
4
2
5
10
genomic position

Peak calling



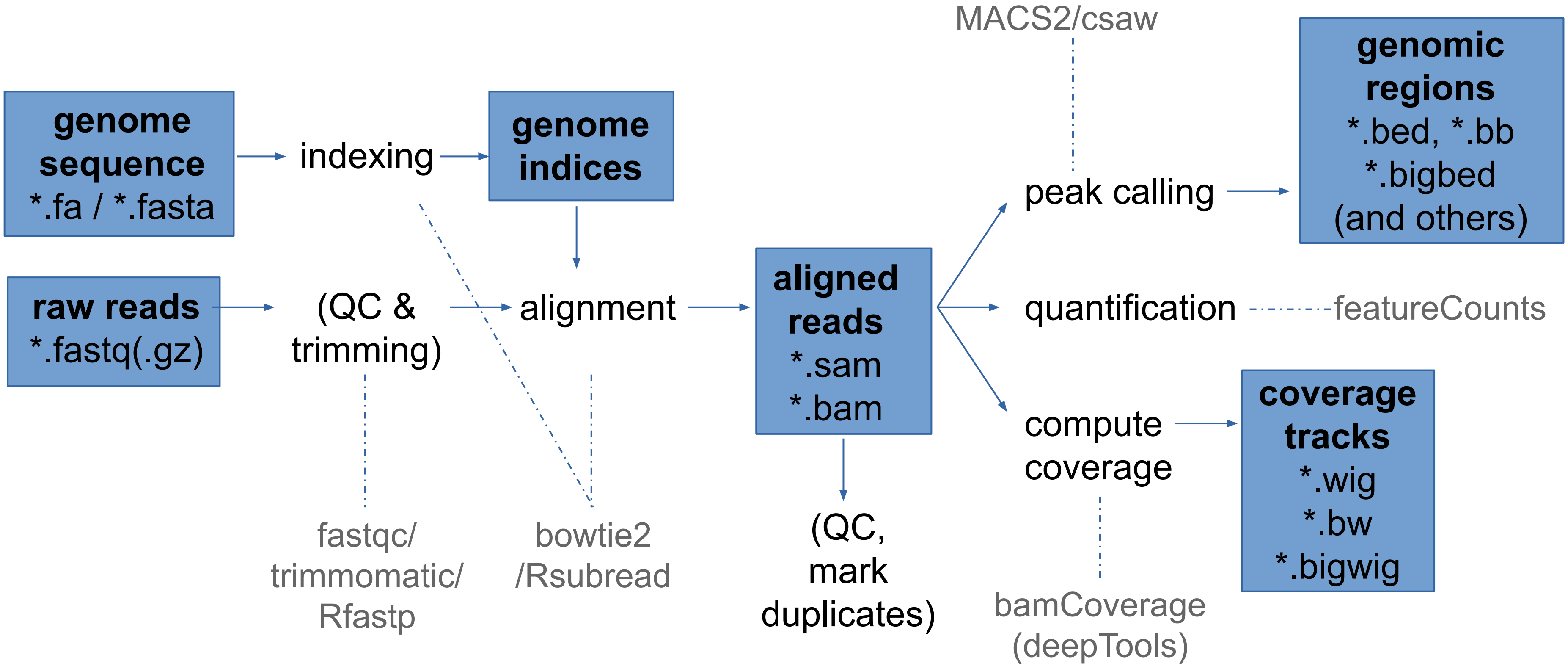


(Ibrahim et al., NAR 2014)



(Wilbanks et al., PLoS One 2010)

Overview of a primary analysis pipeline (ChIP-seq and the likes)



genome sequence *.fa / *.fasta

indexing

genome indices

raw reads *.fastq(.gz)

(QC & trimming)

alignment

aligned reads *.sam *.bam

(QC, mark duplicates)

peak calling

quantification

compute coverage

genomic regions *.bed, *.bb *.bigbed (and others)

coverage tracks *.wig *.bw *.bigwig

fastqc/ trimmomatic/ Rfastp

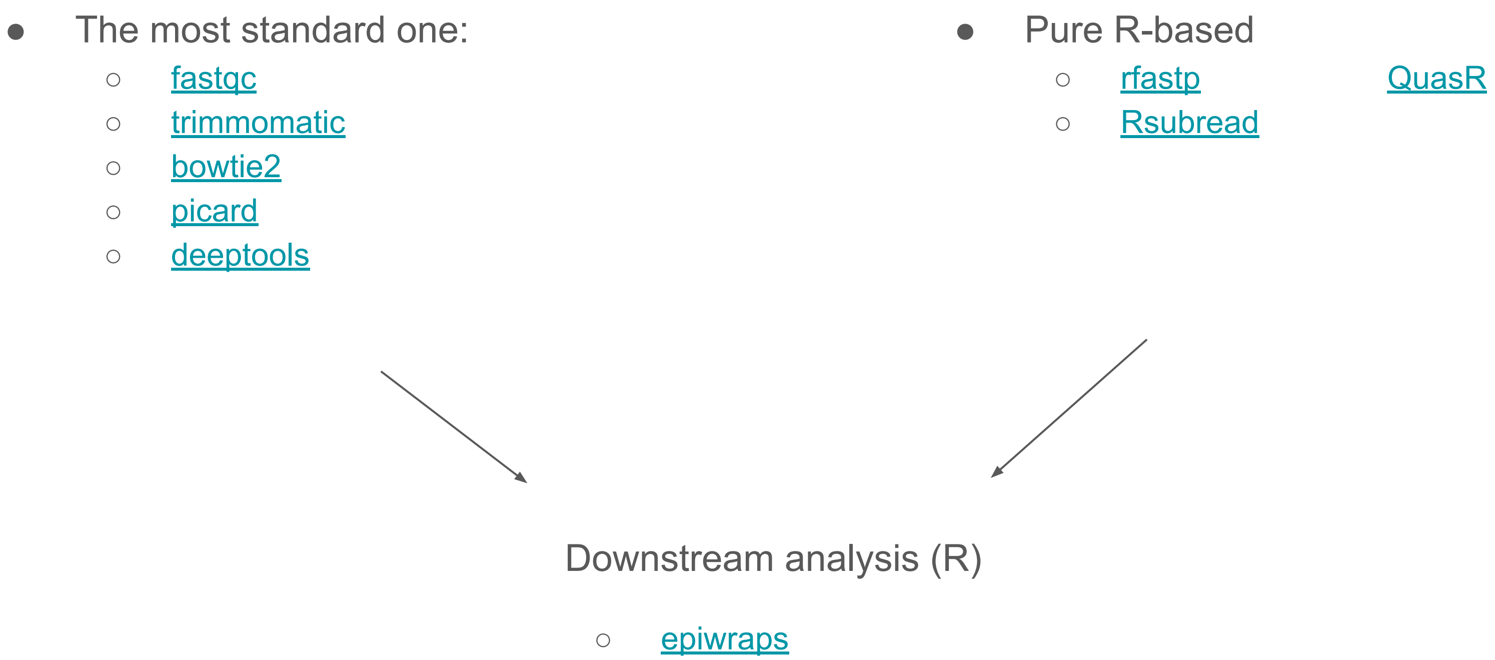
bowtie2 /Rsubread

MACS2/csaw

featureCounts

bamCoverage (deepTools)

Alternative toolsets for (DNA) primary analysis



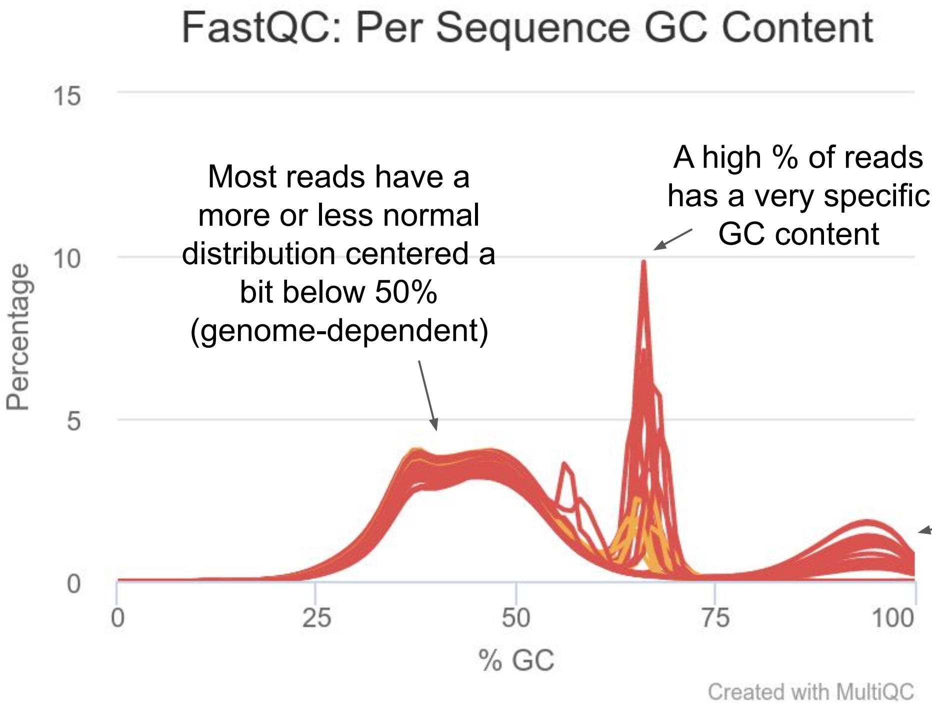
- The most standard one:
  - fastqc
  - trimmomatic
  - bowtie2
  - picard
  - deeptools

- Pure R-based
  - rfastp
  - Rsubread
  - QuasR

Downstream analysis (R)

- epiwraps

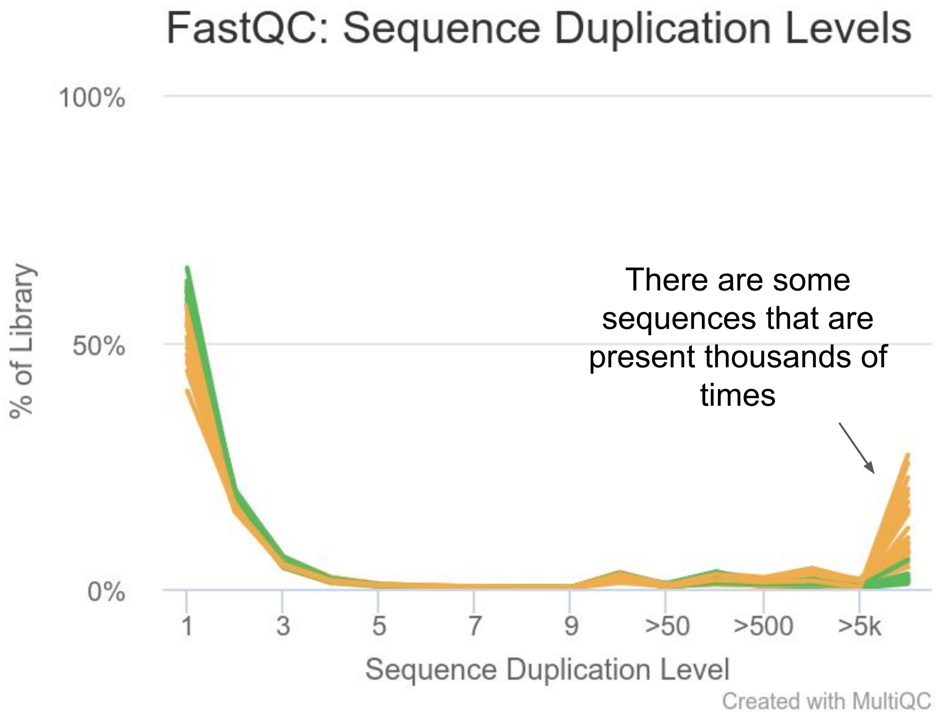
Example (rather extreme) QC problems



FastQC: Per Sequence GC Content

Most reads have a more or less normal distribution centered a bit below 50% (genome-dependent)

A high % of reads has a very specific GC content



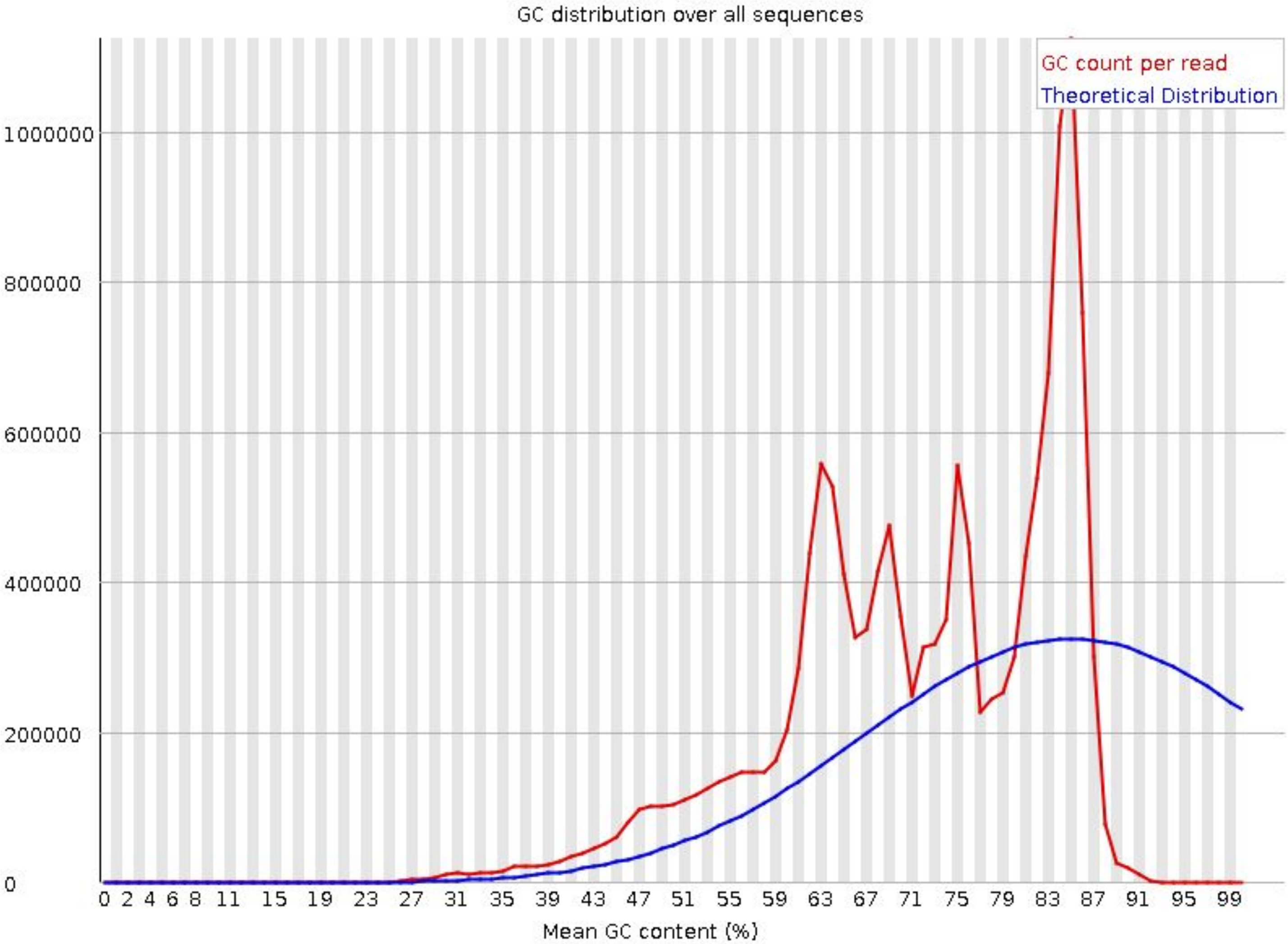
FastQC: Sequence Duplication Levels

There are some sequences that are present thousands of times

A certain % of the reads has an extremely high GC content

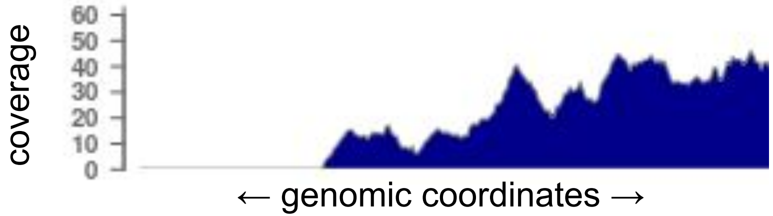
Example (rather extreme) QC problems:

Bias from overamplification



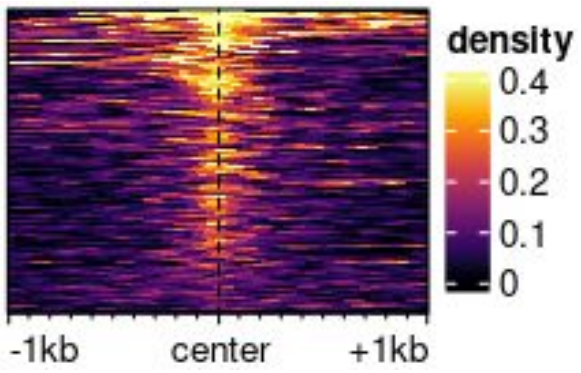
Visualizations available in *epiwraps*

- Signal across one genomic region: `plotSignalTracks`



(Based on the *Gviz* R package)

- Signal across several genomic regions: `signal2Matrix` → `plotEnrichedHeatmaps`



(Mainly based on the EnrichedHeatmap R package, itself based on ComplexHeatmap)

Assignment

- Download a mouse ChIPseq dataset
- Download and process it from the raw data, obtaining:
  - bam file, along with number and percentage of mapped reads
  - bigwig file
  - peaks
- How many peaks do you find?
- Plot the signal round one of the peaks

- Please make sure that you name your final file `assignment.html` !!

- Suggested dataset:
  - p300 in mESC:

    https://www.encodeproject.org/files/ENCFF001LJN/@@download/ENCFF001LJN.fastq.gz

  - the corresponding input control would be:

    https://www.encodeproject.org/files/ENCFF001KEU/@@download/ENCFF001KEU.fastq.gz