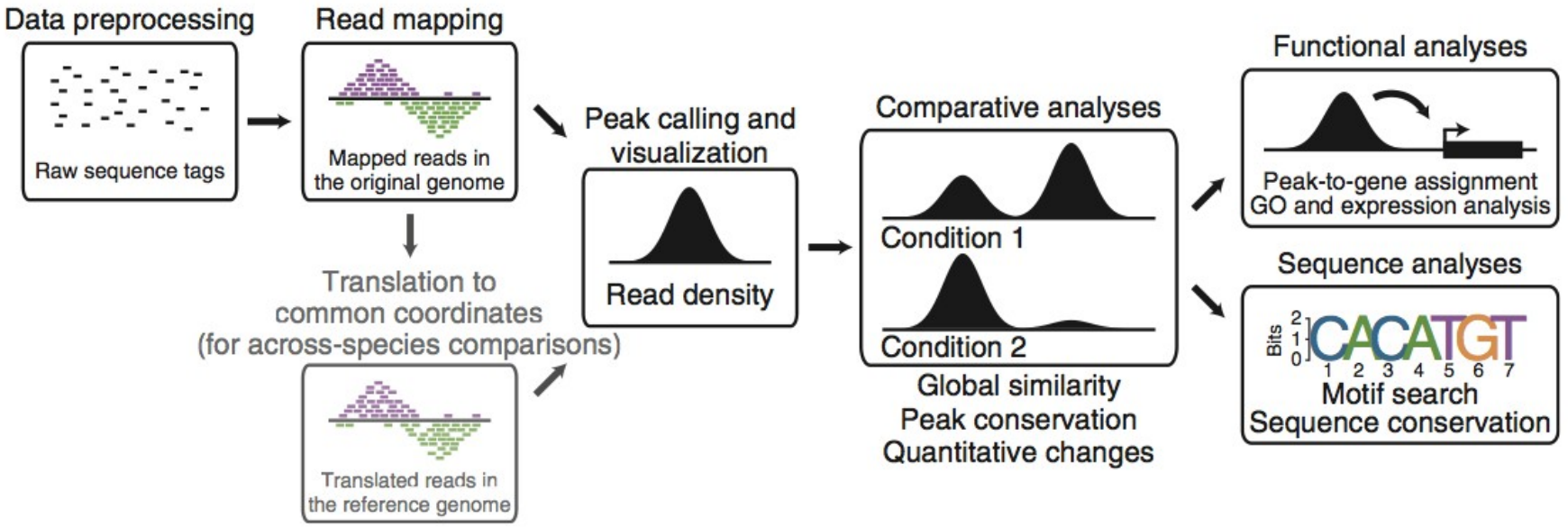
ChIP-seq

Slides from
Brian Parker & David Gresham

ChIP-seq pipeline



*A. Bardet, Q. He, J. Zeitlinger & A. Stark. "A computational pipeline for comparative ChIP-seq analyses. Nature Protocols 7, 45–61 (2012)

Outline

- In these lectures we will focus on the following stages of a typical chip-seq analysis:

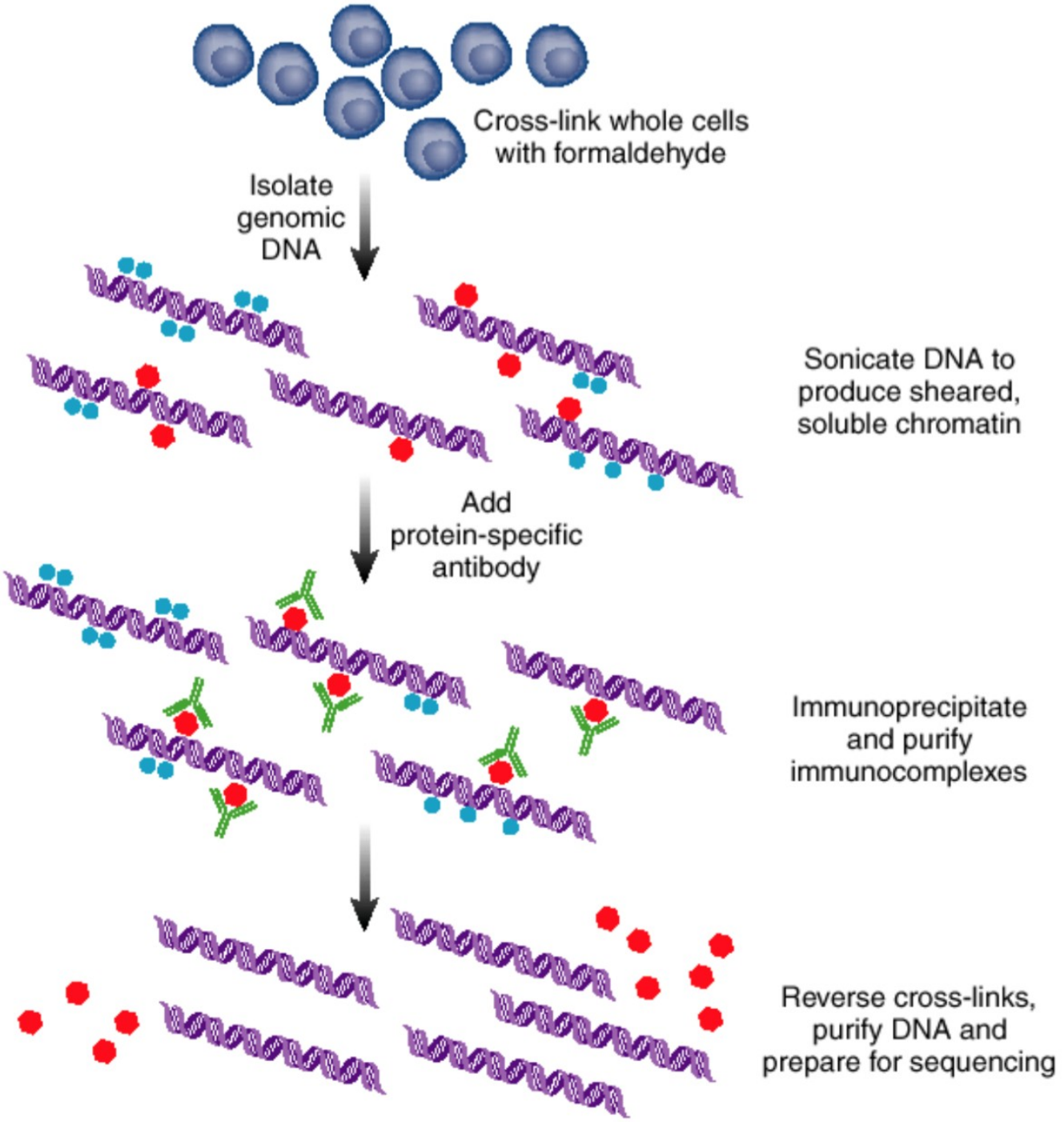
  (1) Summarizing and visualizing chip-seq data

  (2) Peak-calling (using MACS)

  (3) Motif enrichment analysis (using meme) and logo display

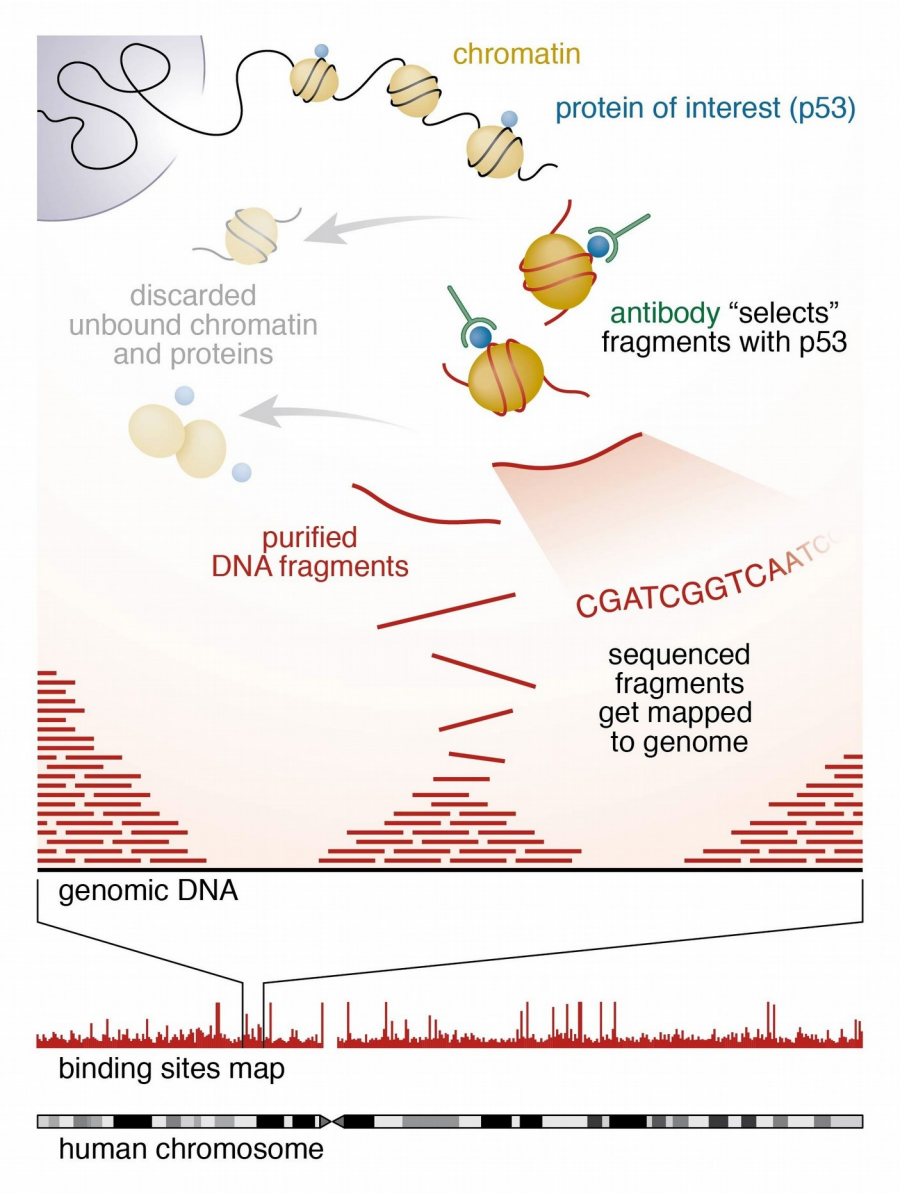
ChIP-seq overview

- Chromatin ImmunoPrecipitation (ChIP) captures DNA bound proteins.
- Used for:
  - Transcription factor binding sites.
  - Histone modifications.
  - Histone variants (paralogs of core histones H2A/H2B/H3/H4).
  - Polymerase machinery
  - Other

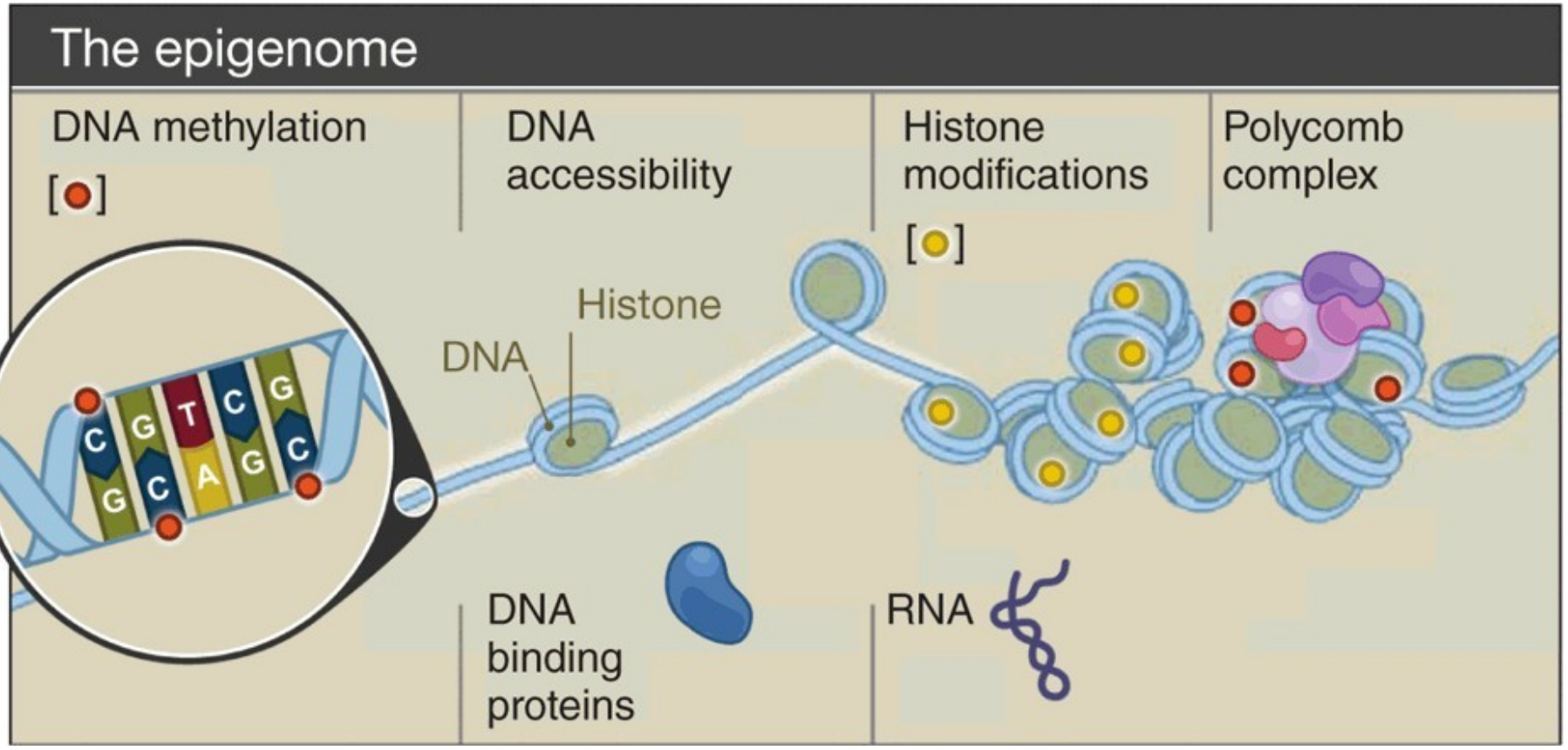


Mardis. Nature Methods 2007.

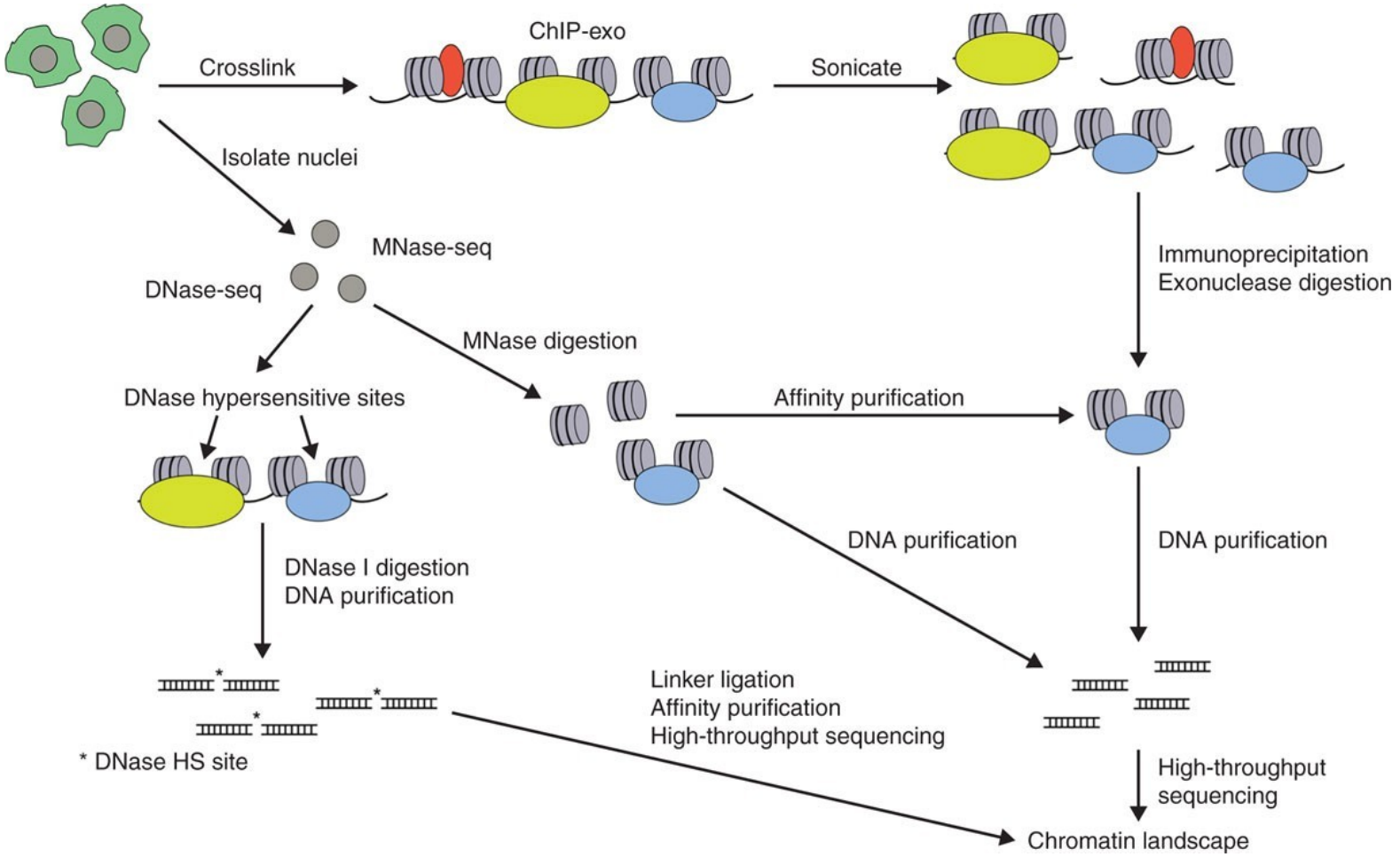
ChIP-–seq overview



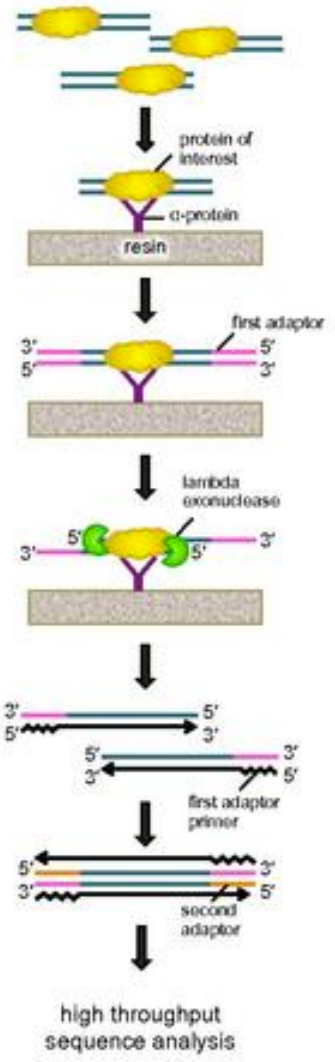
Interrogating the epigenome



Methods for studying the epigenome



Chip-Seq exo may increase resolution of binding sites



Crosslink proteins to DNA, shear DNA

Chromatin-immunoprecipitation

Ligated first adaptor, fill in ends

Exonuclease digestion

LM-PCR with primers to first adaptor

Ligate second adaptor, amplify DNA with LM-PCR

ChIP-seq mapping issues

- Use a non-splice aware mapper such as Bowtie2.
- Pull down is often low concentration and so PCR duplication artefacts can be an issue, so may need to remove duplicates e.g.

  # get good MAPQ reads

  samtools view -bS -q 30 input.sam > input_bestAlignment.bam

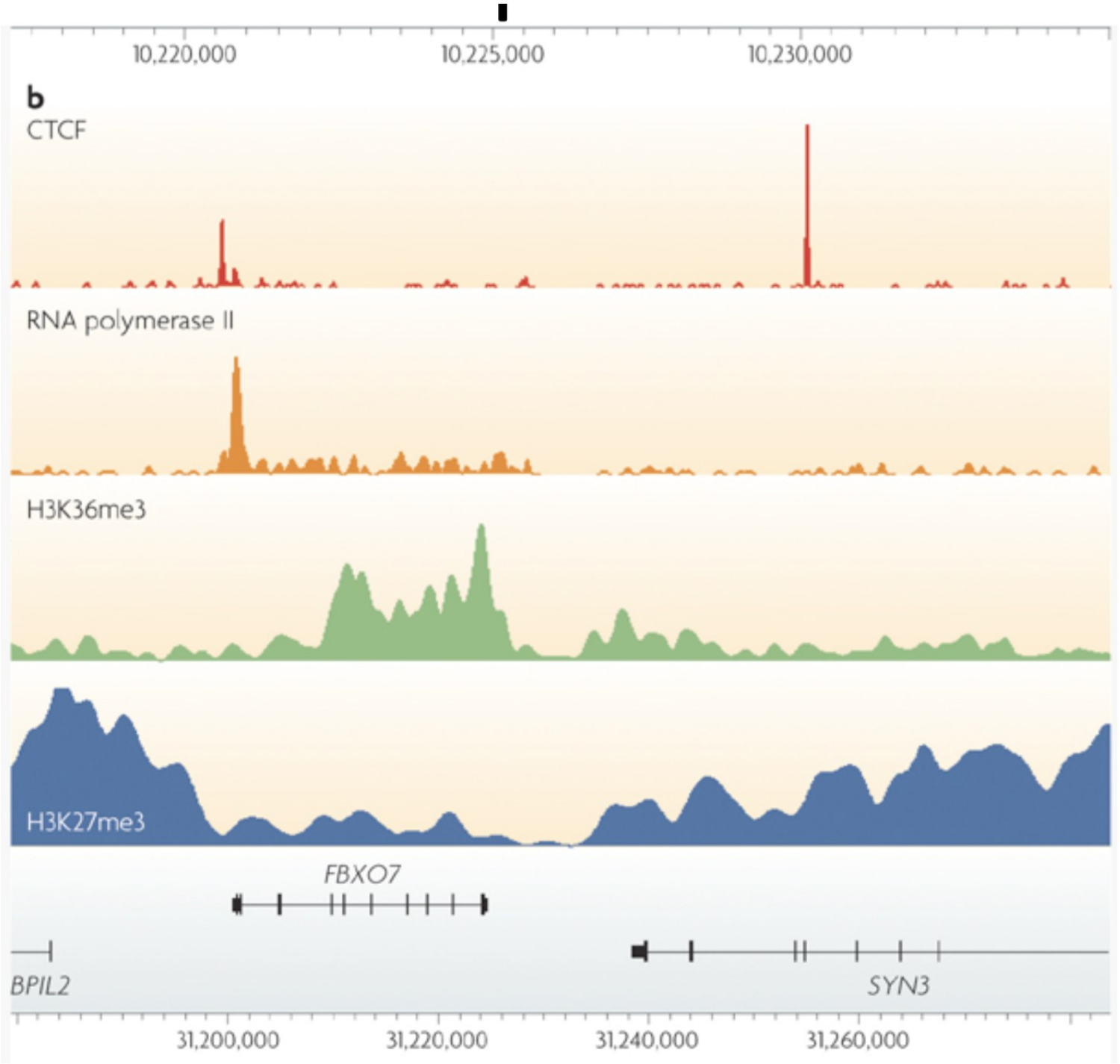
  # remove duplicates (note Picard MarkDuplicates is more accurate across chromosomes so use it in preference)

  samtools rmdup -s input_bestAlignment.bam input_filtered.bam

- See "ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia" Genome Research 2012
- ENCODE's goal is to obtain >10 million uniquely mapping reads per replicate experiment for mammalian genomes, and requires at least 2 replicate samples with a high concordance.

ChIP-seq peak finding

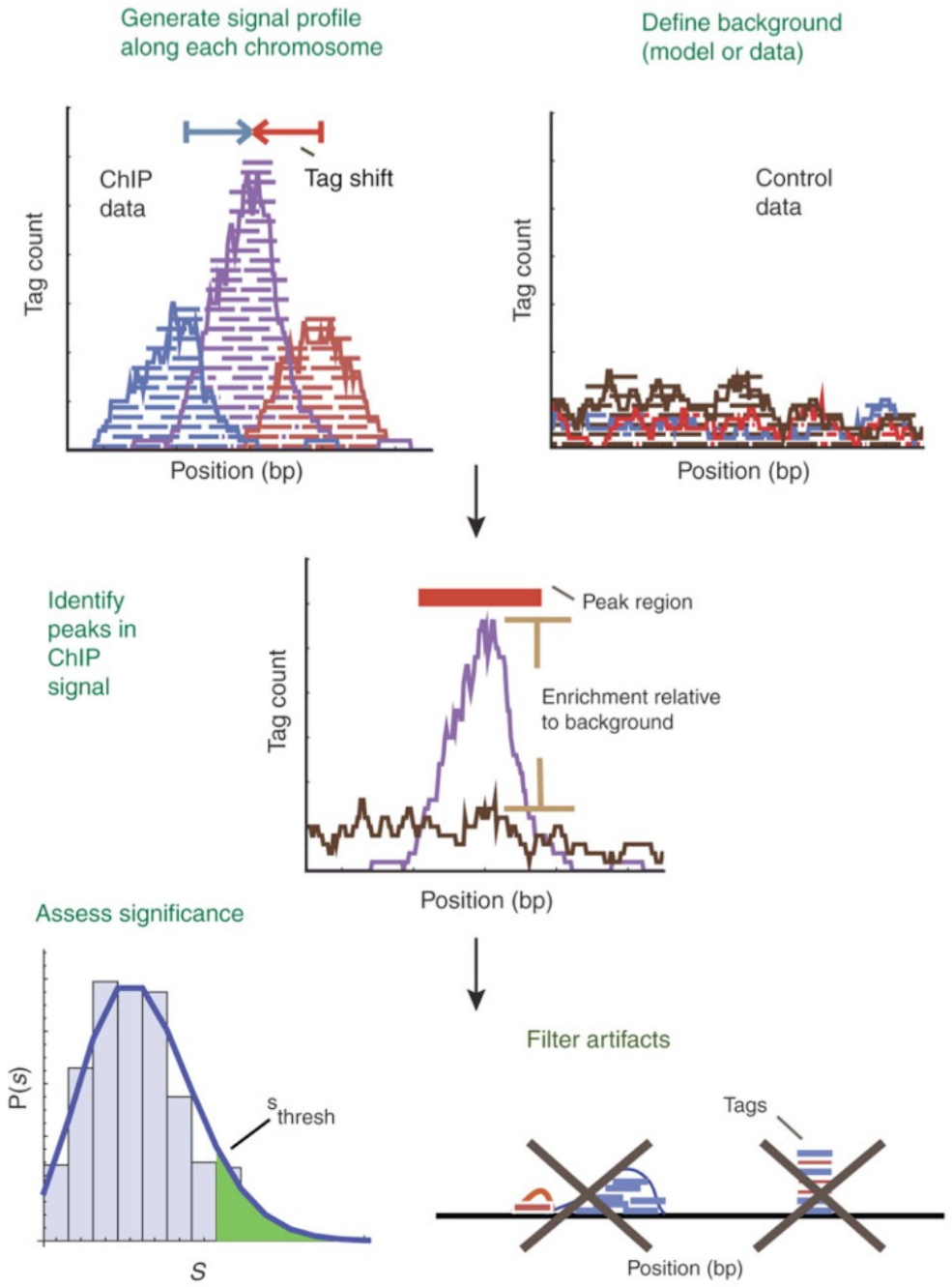
- Some regions of open chromatin are preferentially represented in the fragmented DNA, and there are platform-specific sequencing efficiency biases that contribute to nonuniformity
- To correct for this a control sample should be used. Either input sample (non-pulled down DNA) or a non-specific antibody pulldown.
  - (1) DNA is isolated from cells that have been cross-linked and fragmented under the same conditions as the immunoprecipitated DNA (“Input” DNA); or
  - (2) a “mock” ChIP reaction is performed using a control antibody that reacts with an irrelevant, non-nuclear antigen (“IgG” control).



Park J, Nature Reviews Genetics, 2009

ChIP-seq peak finding

- Transcription factors tend to give a sharp peak
- Histone modifications and histone variants tend to give broad peaks, or cover larger regions.
- MACS2 and Homer are two common ChIP-seq peak finders (see lab session).
- They both have options to detect either sharp or broad peaks (Sicer is another program focussed on broad peaks).
- As another option, limma can be used in a sliding-window mode to find differential regions without a need for peak calling, when replicates are available (see csaw).



Pepke et al. Nature Methods, 2009

ChIP-seq peak differential analysis stage

- ChIP-seq data should have sufficient biological replicates in which case limma could be used for differential analysis, using the same methods we have seen for RNA-seq

  (See Lun,A.T. and Smyth,G.K. (2014) De novo detection of differentially bound regions for ChIP-seq data using peaks and windows: controlling error rates correctly. Nucleic Acids Res., 42.)

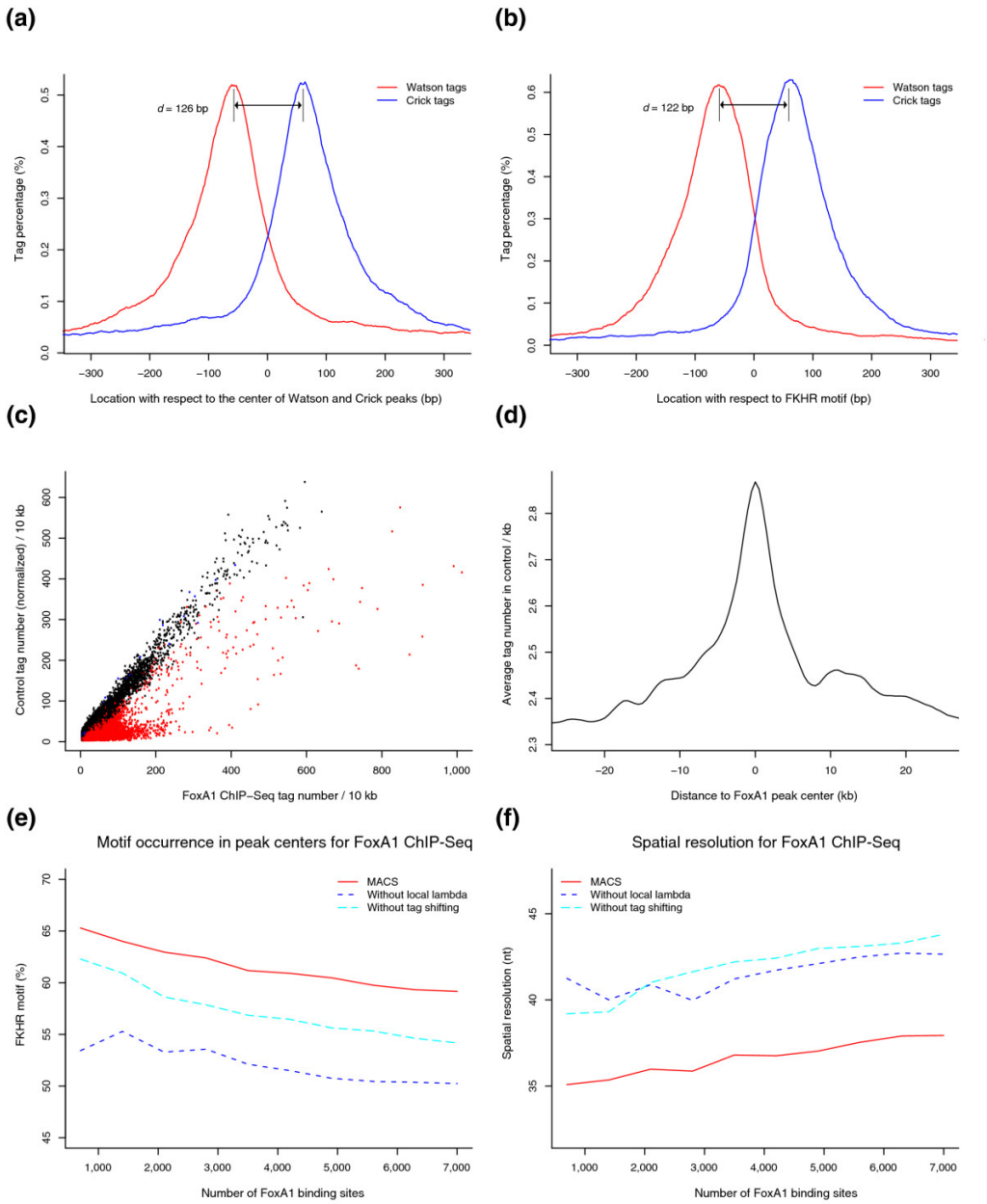
- Although differential analysis is also offered by macs2 and other peak-callers directly (and they can handle unreplicated samples)

ChIP-seq quality check

- In addition to the quality checks using fastqc that were discussed previously, we can do a specific quality plot to check that the antibody and immuno-precipitation (IP) worked ok.
- We use the fingerprint plot from deepTools
- See http://deeptools.readthedocs.io/en/latest/content/tools/plotFingerprint.html

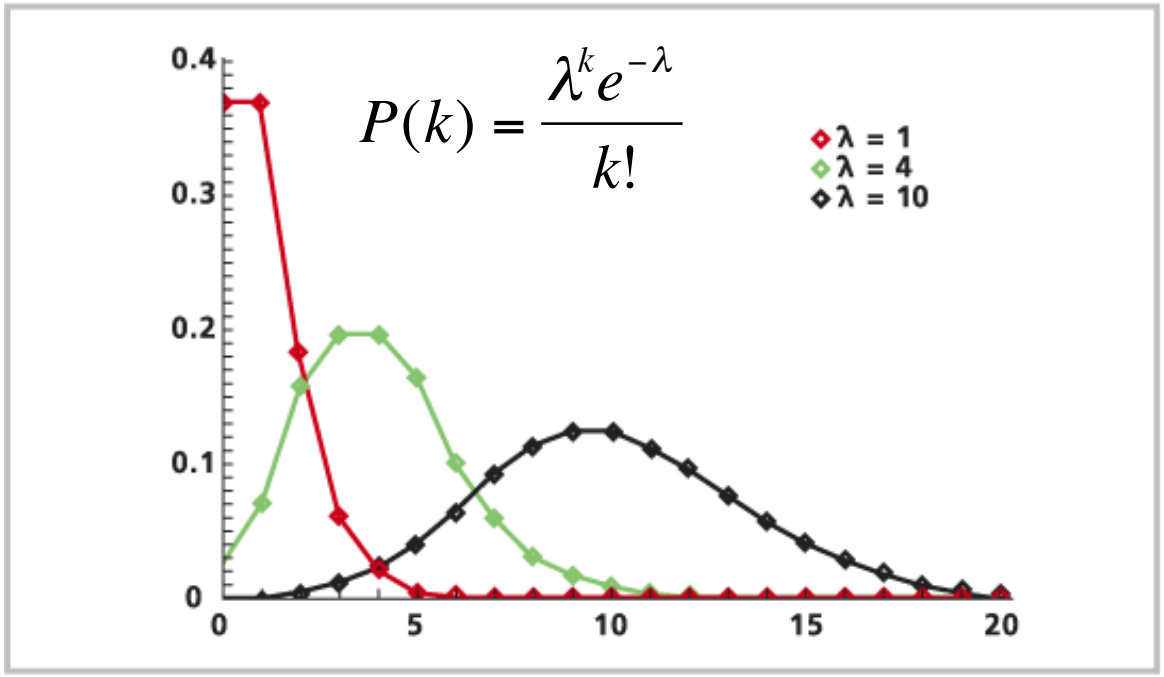
The MACS algorithm

- Makes use of features of the data
  - ChIP-Seq tags are often 3' of the protein-DNA interaction site
  - Therefore, the tag density around a site should be bimodal
  - Define 'd' as the distance between the Watson and Crick peak.
  - MACS defines d/2 as the most likely site of DNA-protein interaction


(a)
Watson tags
Crick tags
d = 126 bp
Tag percentage (%)
Location with respect to the center of Watson and Crick peaks (bp)
(b)
Watson tags
Crick tags
d = 122 bp
Tag percentage (%)
Location with respect to FKHR motif (bp)
(c)
Control tag number (normalized) / 10 kb
FoxA1 ChIP−Seq tag number / 10 kb
(d)
Average tag number in control / kb
Distance to FoxA1 peak center (kb)
(e)
Motif occurrence in peak centers for FoxA1 ChIP-Seq
MACS
Without local lambda
Without tag shifting
FKHR motif (%)
Number of FoxA1 binding sites
(f)
Spatial resolution for FoxA1 ChIP-Seq
MACS
Without local lambda
Without tag shifting
Spatial resolution (nt)
Number of FoxA1 binding sites

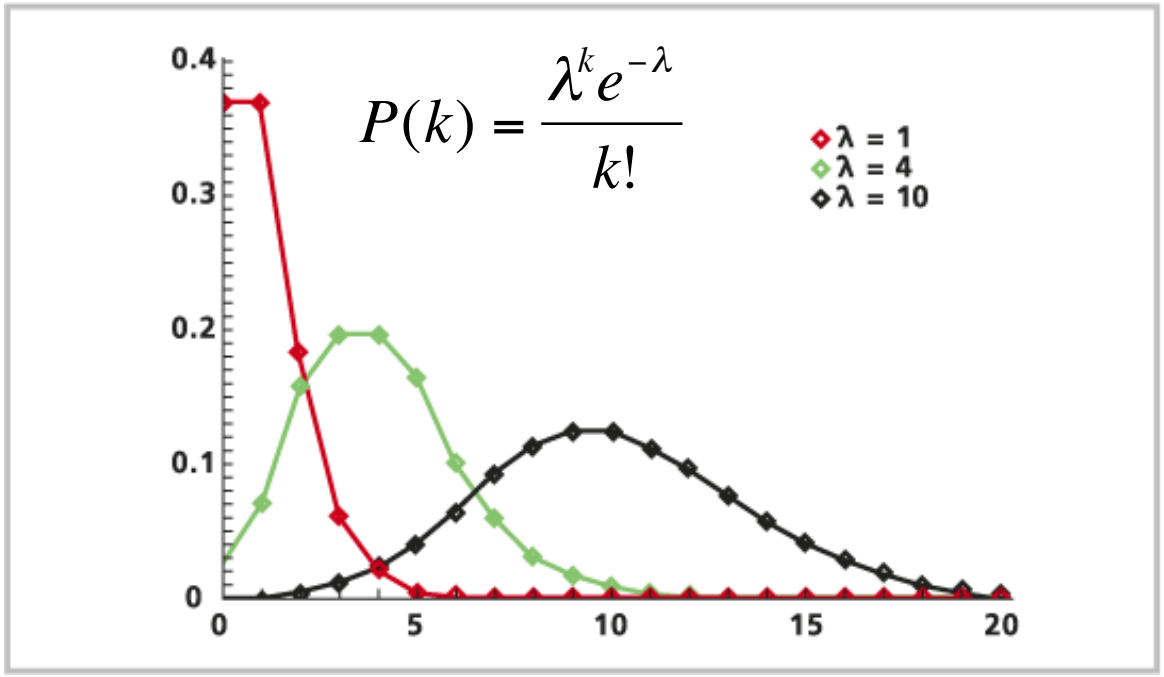
The MACS algorithm

- Tag distribution along chromosome is modeled as a Poisson distribution
  - i.e. a random variable with mean λ



The MACS algorithm

- How “surprised” we are to see a particular tag count depends on λ – this is quantified as the probability of seeing that value given λ



Estimation of λ

$$\lambda_{\text{local}} = \max(\lambda_{\text{BG}}, [\lambda_{1\text{k}},] \lambda_{5\text{k}}, \lambda_{10\text{k}})$$

Lab session: ChIP-seq peak-finding using MACS2

Dataset we will use is from
https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSM409307

ChIP-Seq Analysis of H3K4me1 in hESC H1 (CDI-01) Cells

On prince at

Run quality_plot.sh on prince to generate a cdf-based quality plot to check chip-seq and input datasets for adequate antibody.

Run run_macs2.sh on prince to do peak calling

Lab session: ChIP-seq peak-finding using MACS2

peak calling with MACS on the prince cluster

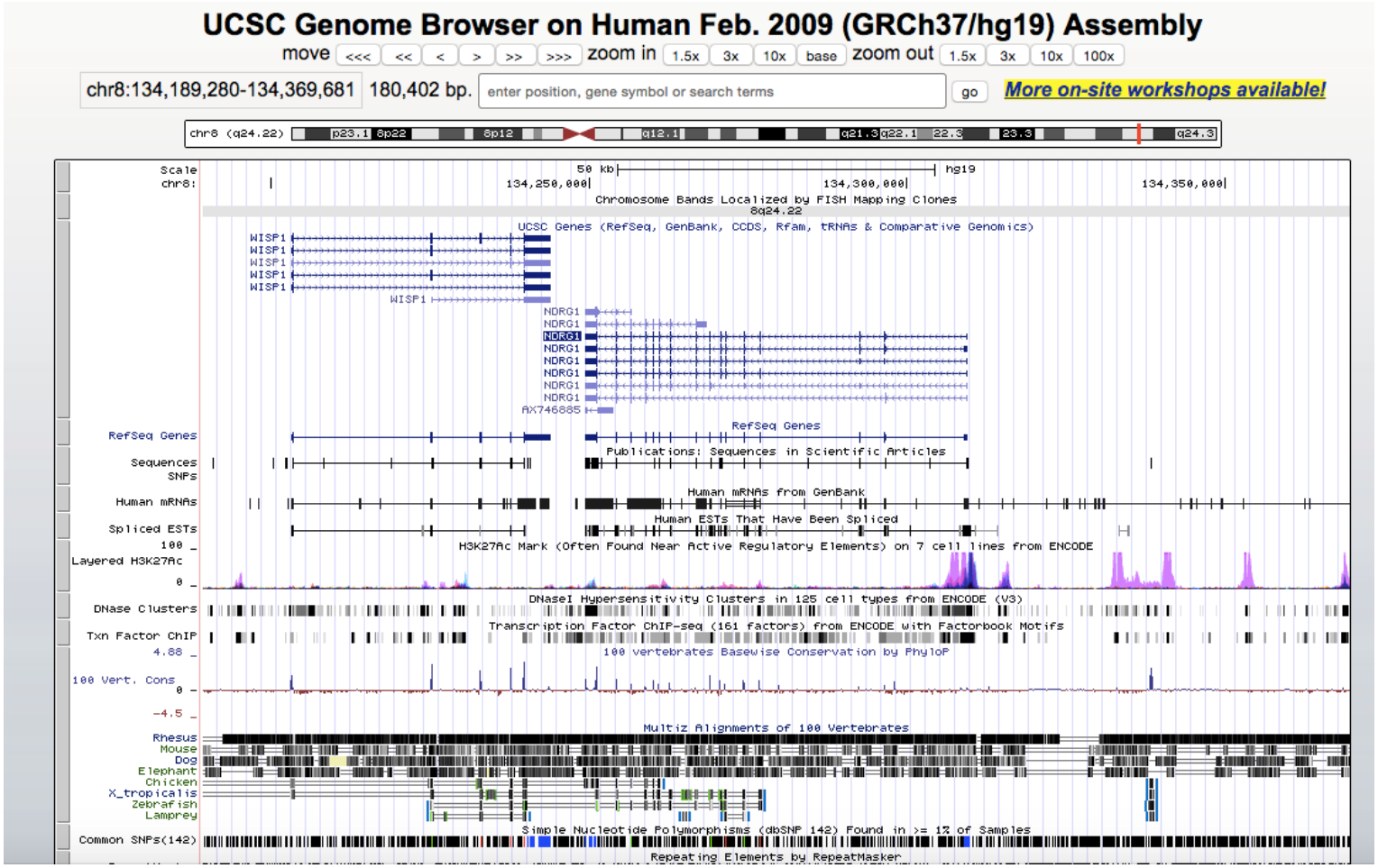
see run_macs2.sh

macs2 callpeak -t GSM409307_UCSD_H3K4me1.bam -c GSM605335_UCSD_Input.bam --format=BAM --broad --gsize=hs --cutoff-analysis --qvalue=0.05 --outdir=macs2_out --name H3K4me1_example

Output is a form of BED file, see:

https://github.com/taoliu/MACS

Visualizing data in genomic context



Displaying your own tracks in the UCSC browser

- http://genome.ucsc.edu/goldenPath/help/customTrack.html

BED file format

BED format provides a flexible way to define the data lines that are displayed in an annotation track. BED lines have three required fields and nine additional optional fields. The number of fields per line must be consistent throughout any single set of data in an annotation track. The order of the optional fields is binding: lower-numbered fields must always be populated if higher-numbered fields are used.

The first three required BED fields are:

1. **chrom** - The name of the chromosome (e.g. chr3, chrY, chr2_random) or scaffold (e.g. scaffold10671).
2. **chromStart** - The starting position of the feature in the chromosome or scaffold. The first base in a chromosome is numbered 0.
3. **chromEnd** - The ending position of the feature in the chromosome or scaffold. The *chromEnd* base is not included in the display of the feature. For example, the first 100 bases of a chromosome are defined as *chromStart=0, chromEnd=100*, and span the bases numbered 0-99.

The 9 additional optional BED fields are:

4. **name** - Defines the name of the BED line. This label is displayed to the left of the BED line in the Genome Browser window when the track is open to full display mode or directly to the left of the item in pack mode.
5. **score** - A score between 0 and 1000. If the track line *useScore* attribute is set to 1 for this annotation data set, the *score* value will determine the level of gray in which this feature is displayed (higher numbers = darker gray). This table shows the Genome Browser's translation of BED score values into shades of gray:

shade									
score in range	≤ 166	167-277	278-388	389-499	500-611	612-722	723-833	834-944	≥ 945

6. **strand** - Defines the strand - either '+' or '-'.
7. **thickStart** - The starting position at which the feature is drawn thickly (for example, the start codon in gene displays). When there is no thick part, thickStart and thickEnd are usually set to the chromStart position.
8. **thickEnd** - The ending position at which the feature is drawn thickly (for example, the stop codon in gene displays).
9. **itemRgb** - An RGB value of the form R,G,B (e.g. 255,0,0). If the track line *itemRgb* attribute is set to "On", this RBG value will determine the display color of the data contained in this BED line. NOTE: It is recommended that a simple color scheme (eight colors or less) be used with this attribute to avoid overwhelming the color resources of the Genome Browser and your Internet browser.
10. **blockCount** - The number of blocks (exons) in the BED line.
11. **blockSizes** - A comma-separated list of the block sizes. The number of items in this list should correspond to *blockCount*.
12. **blockStarts** - A comma-separated list of block starts. All of the *blockStart* positions should be calculated relative to *chromStart*. The number of items in this list should correspond to *blockCount*.

In BED files with block definitions, the first *blockStart* value must be 0, so that the first block begins at *chromStart*. Similarly, the final *blockStart* position plus the final *blockSize* value must equal *chromEnd*. Blocks may not overlap.

BED file format example

```
track name=pairedReads description="Clone Paired Reads" useScore=1
chr22 1000 5000 cloneA 960 + 1000 5000 0 2 567,488, 0,3512
chr22 2000 6000 cloneB 900 - 2000 6000 0 2 433,399, 0,3601
```

Wiggle file format

UCSC Genome Bioinformatics

Home - Genomes - Blat - Tables - Gene Sorter - PCR - Session - FAQ - Help

Wiggle Track Format (WIG)

The bigWig format is the recommended format for almost all graphing track needs (for more information, see the following wiki page). The wiggle (WIG) format is an older format for display of dense, continuous data such as GC percent, probability scores, and transcriptome data. Wiggle data elements must be equally sized. The bedGraph format is also an older format used to display sparse data or data that contains elements of varying size.

For speed and efficiency, wiggle data is compressed and stored internally in 128 unique bins. This compression means that there is a minor loss of precision when data is exported from a wiggle track (*i.e.*, with output format "data points" or "bed format" within the table browser). The bedGraph format should be used if it is important to retain exact data when exporting.

variableStep

variableStep format is designed for data with irregular intervals between data points, and is the more commonly used format. It begins with a declaration line, followed by two columns containing **chromosome positions** and **data values**.

The declaration line begins with the word **variableStep** and is followed by space-separated key-value pairs:

- **chrom** (required) - name of chromosome
- **span** (optional, defaults to 1) - the number of bases that each data value should cover

The span allows data to be compressed as follows:

Without span:

```
variableStep chrom=chr2
300701  12.5
300702  12.5
300703  12.5
300704  12.5
300705  12.5
```

With span:

```
variableStep chrom=chr2 span=5
300701  12.5
```

Both of these examples will display a value of 12.5 at position 300701-300705 on chromosome 2.

Wiggle file format

UCSC Genome Bioinformatics

Home - Genomes - Blat - Tables - Gene Sorter - PCR - Session - FAQ - Help

Wiggle Track Format (WIG)

The bigWig format is the recommended format for almost all graphing track needs (for more information, see the following wiki page). The wiggle (WIG) format is an older format for display of dense, continuous data such as GC percent, probability scores, and transcriptome data. Wiggle data elements must be equally sized. The bedGraph format is also an older format used to display sparse data or data that contains elements of varying size.

For speed and efficiency, wiggle data is compressed and stored internally in 128 unique bins. This compression means that there is a minor loss of precision when data is exported from a wiggle track (*i.e.*, with output format "data points" or "bed format" within the table browser). The bedGraph format should be used if it is important to retain exact data when exporting.

fixedStep

fixedStep format is designed for data with regular intervals between data points, and is the more compact of the two wiggle formats. It begins with a declaration line, followed by a single column of **data values**.

The declaration line begins with the word **fixedStep** and is followed by space-separated key-value pairs:

- **chrom** (required) - name of chromosome
- **start** (required) - start point for the data values
- **step** (required) - distance between data values
- **span** (optional, defaults to 1) - the number of bases that each data value should cover

Without span

```
fixedStep chrom=chr3 start=400601 step=100
11
22
33
```

Displays the values 11, 22, 33 as single-base features, on chromosome 3 at positions 400601, 400701 and 400801 respectively.

With span

```
fixedStep chrom=chr3 start=400601 step=100 span=5
11
22
33
```

Displays the values 11, 22, 33 as 5-base features, on chromosome 3 at positions 400601-400605, 400701-400705 and 400801-400805 respectively.

Lab session: peak annotation

1. gene annotation using bedTools

Log into the cluster.

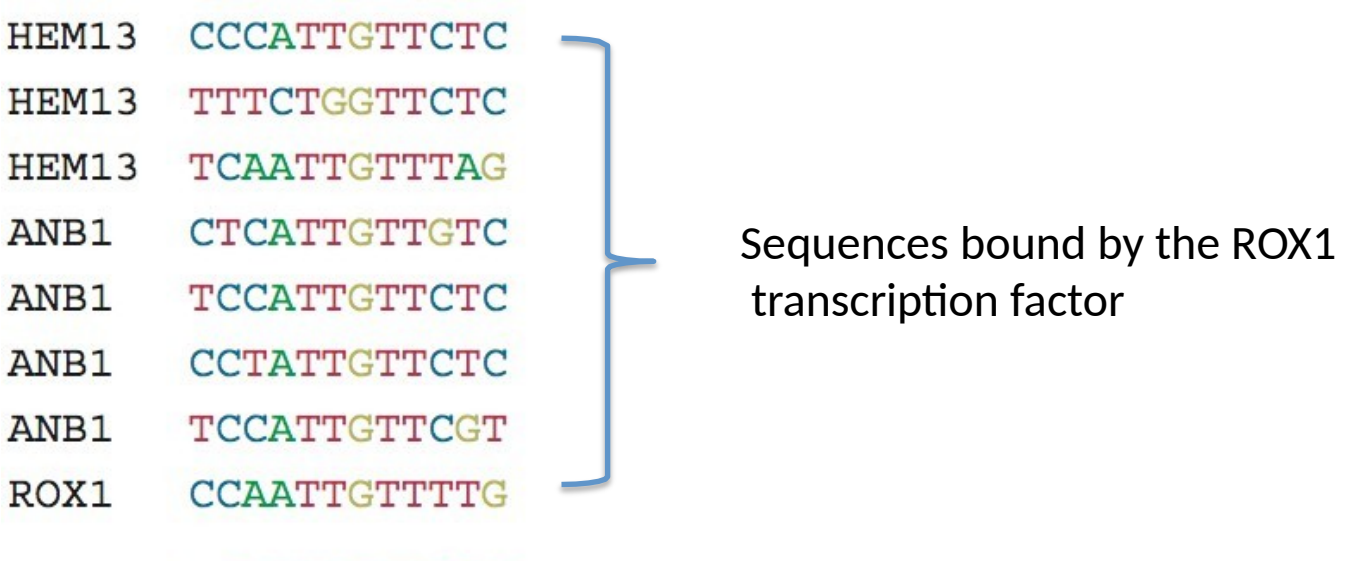
Given the bed output from MACS2, we assign each peak to the gene with the closest TSS

annotate_peaks_bedtools_example.sh

Sequence logo

- Sequence logos are a graphical representation of a sequence motif generated from an amino acid or nucleic acid multiple sequence alignment.
- Developed by Tom Schneider and Mike Stephens.
- A logo consists of stacks of symbols, with one stack for each position in the sequence.
- The overall height of indicates the sequence conservation at that position, while the height of symbols within the stack indicates the relative frequency of each amino or nucleic acid at that position.
- In one common form, the y-axis give the information content (in bits)

How do we construct a DNA binding motif for a transcription factor?


HEM13 CCCATTGTTCTC
HEM13 TTTCTGGTTCTC
HEM13 TCAATTGTTTAG
ANB1 CTCATTGTTGTC
ANB1 TCCATTGTTCTC
ANB1 CCTATTGTTCTC
ANB1 TCCATTGTTCGT
ROX1 CCAATTGTTTTG
Sequences bound by the ROX1 transcription factor

```
HEM13   CCCATTGTTCTC
HEM13   TTTCTGGTTCTC
HEM13   TCAATTGTTTAG
ANB1    CTCATTGTTGTC
ANB1    TCCATTGTTCTC
ANB1    CCTATTGTTCTC
ANB1    TCCATTGTTCGT
ROX1    CCAATTGTTTTG

        YCHATTGTTCTC
```

```
HEM13   CCCATTGTTCTC
HEM13   TTTCTGGTTCTC
HEM13   TCAATTGTTTAG
ANB1    CTCATTGTTGTC
ANB1    TCCATTGTTCTC
ANB1    CCTATTGTTCTC
ANB1    TCCATTGTTCGT
ROX1    CCAATTGTTTTG

        YCHATTGTTCTC

A 002700000010
C 464100000505
G 000001800112
T 422087088261
```

A matrix of frequencies
“Position frequency matrix” (PFM)

```
HEM13   CCCATTGTTCTC
HEM13   TTTCTGGTTCTC
HEM13   TCAATTGTTTAG
ANB1    CTCATTGTTGTC
ANB1    TCCATTGTTCTC
ANB1    CCTATTGTTCTC
ANB1    TCCATTGTTCGT
ROX1    CCAATTGTTTTG

        YCHATTGTTCTC
```

```
A 002700000010
C 464100000505
G 000001800112
T 422087088261
```

A matrix of frequencies

Visualization of frequencies

```
HEM13   CCCATTGTTCTC
HEM13   TTTCTGGTTCTC
HEM13   TCAATTGTTTAG
ANB1    CTCATTGTTGTC
ANB1    TCCATTGTTCTC
ANB1    CCTATTGTTCTC
ANB1    TCCATTGTTCGT
ROX1    CCAATTGTTTTG

        YCHATTGTTCTC

A 002700000010
C 464100000505
G 000001800112
T 422087088261
```

A matrix of frequencies

Visualization of frequencies

What's the problem with this depiction?

"Information content "
of a position

Frequency of the base (*b*) at this position (*i*)

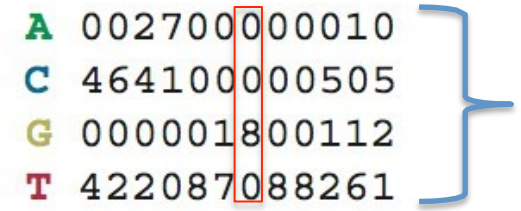
$$I_i = 2 + \sum_b f_{b,i} \log_2 f_{b,i}$$

Add it up for each different base at that position

```
A 002700000010
C 464100000505
G 000001800112
T 422087088261
```

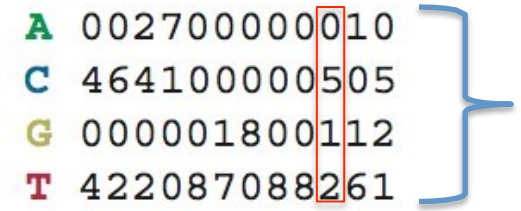
A matrix of frequencies

$$I_7 = 2 + (8/8)*\log_2(8/8)$$



A matrix of frequencies

$$I_7 = 2 + (5/8)*\log_2(5/8) + (1/8)*\log_2(1/8) + (2/8)*\log_2(2/8)$$

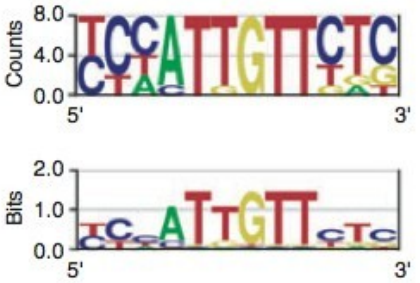


A matrix of frequencies

```
HEM13   CCCATTGTTCTC
HEM13   TTTCTGGTTCTC
HEM13   TCAATTGTTTAG
ANB1    CTCATTGTTGTC
ANB1    TCCATTGTTCTC
ANB1    CCTATTGTTCTC
ANB1    TCCATTGTTCGT
ROX1    CCAATTGTTTTG

        YCHATTGTTCTC

A 002700000010
C 464100000505
G 000001800112
T 422087088261
```



$$I_i = 2 + \sum_b f_{b,i} \log_2 f_{b,i}$$

- Calculate the information content for a position that has two bases that occur at equal frequencies

- Calculate the information for a single position at which all four bases occur with equal frequencies

$$I_i = 2 + \sum_b f_{b,i} \log_2 f_{b,i}$$

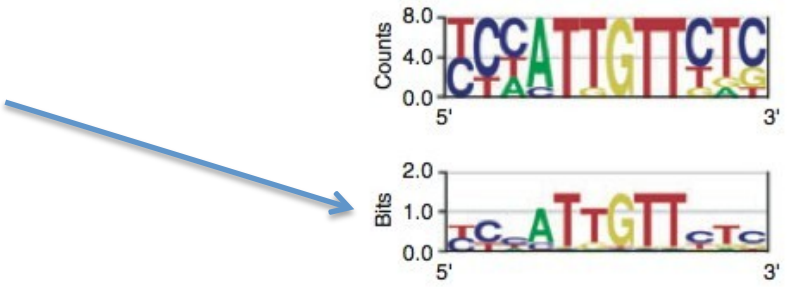
- Calculate the information content for a position that has two bases that occur at equal frequencies
- > 2 + 2*(0.5*log2(0.5))
- [1] 1

- Calculate the information for a single position at which all four bases occur with equal frequencies
- > 2 + 4*(0.25*log2(0.25))
- [1] 0

```
HEM13   CCCATTGTTCTC
HEM13   TTTCTGGTTCTC
HEM13   TCAATTGTTTAG
ANB1    CTCATTGTTGTC
ANB1    TCCATTGTTCTC
ANB1    CCTATTGTTCTC
ANB1    TCCATTGTTCGT
ROX1    CCAATTGTTTTG

        YCHATTGTTCTC

A 002700000010
C 464100000505
G 000001800112
T 422087088261
```

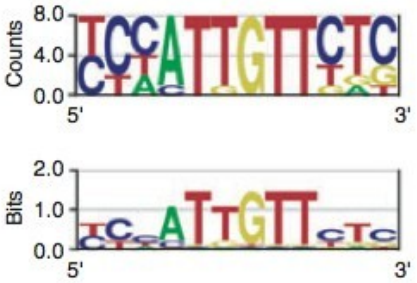


What's the problem with this?

```
HEM13   CCCATTGTTCTC
HEM13   TTTCTGGTTCTC
HEM13   TCAATTGTTTAG
ANB1    CTCATTGTTGTC
ANB1    TCCATTGTTCTC
ANB1    CCTATTGTTCTC
ANB1    TCCATTGTTCGT
ROX1    CCAATTGTTTTG

        YCHATTGTTCTC

A 002700000010
C 464100000505
G 000001800112
T 422087088261
```



Average GC content

E. coli (51%)

Humans (41%)

S. cerevisisae (38%)

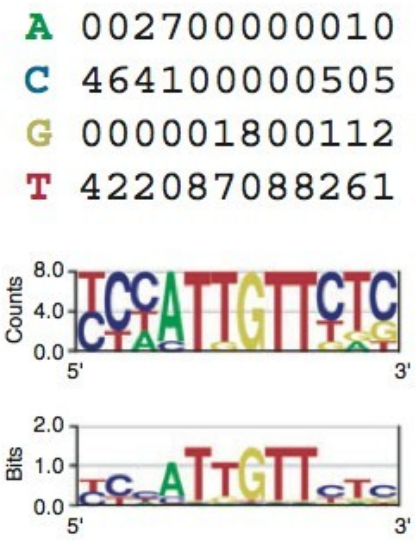
C. elegans (36%)

P. falciparum (19%)

What's the problem with this?

$$I_{seq}(i) = -\sum_{b} f_{b,i} \log_2 \frac{f_{b,i}}{p_b}$$

Frequency of that base (A,C,G,T) in the genome



Average GC content

E. coli (51%)

Humans (41%)

S. cerevisisae (38%)

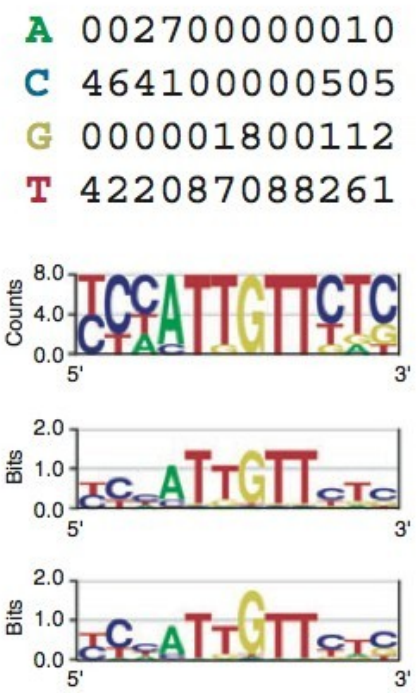
C. elegans (36%)

P. falciparum (19%)

Bits of information can be converted into a probability or an expected frequency in the genome

$$I_{seq}(i) = -\sum_{b} f_{b,i} \log_2 \frac{f_{b,i}}{p_b}$$

Frequency of that base (A,C,G,T) in the genome



Average GC content

E. coli (51%)

Humans (41%)

S. cerevisisae (38%)

C. elegans (36%)

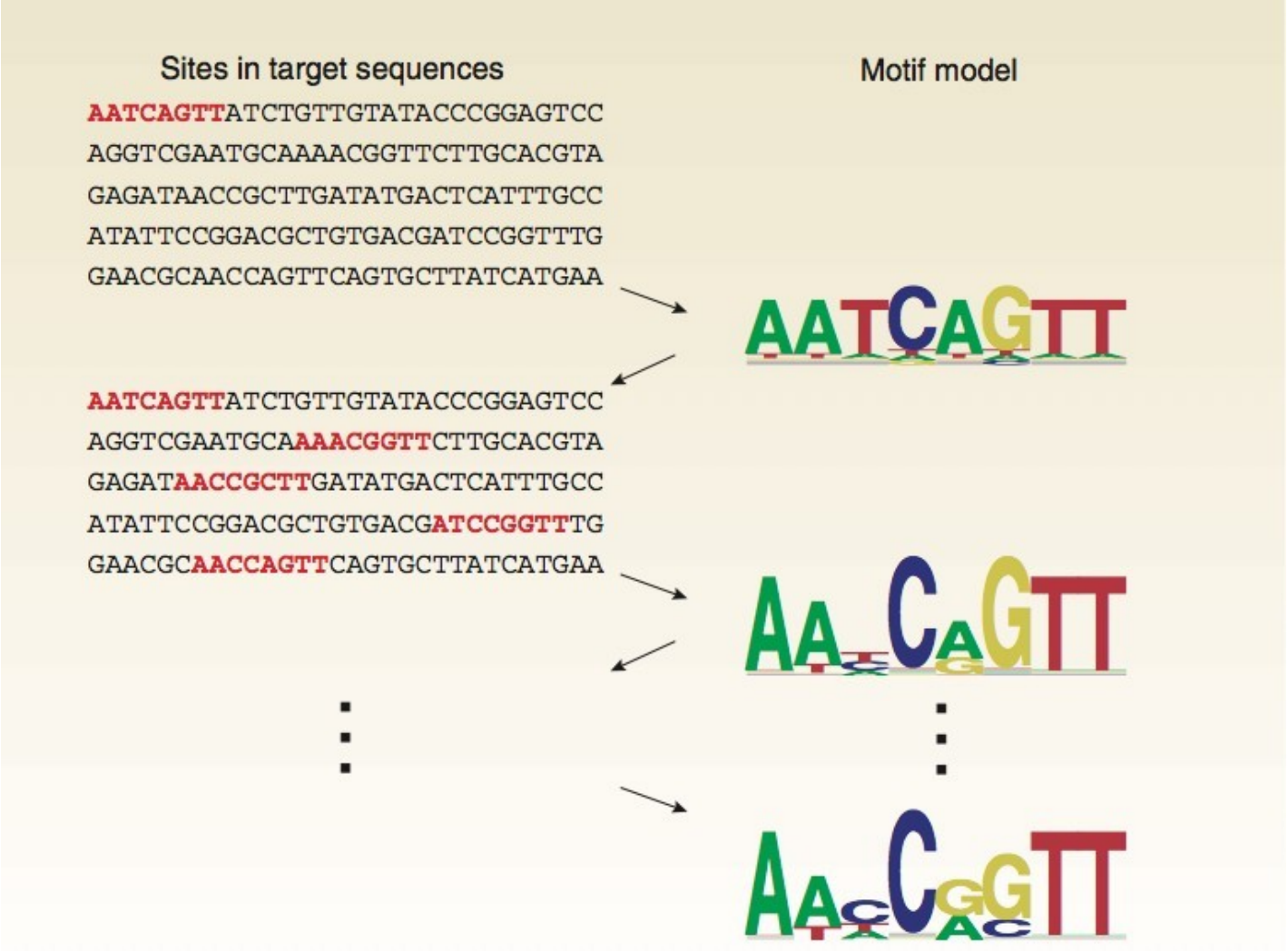
P. falciparum (19%)

Expected freq in genome = 1 in $2^{11.27}$

= 1 in 2,469 bases

How are enriched motifs found?

Deterministic optimization



Expectation-Maximization optimization

- e.g. MEME
- https://meme-suite.org/meme/ D

1. performs a single iteration for each k-mer in target sequence (i.e. partially enumerative)
2. Selects the best motif from this set
3. Tests each n-mer in the sequence to calculate probability that it was generated by the motif
4. Updates the motif based on those probabilities (using Expectation-Maximization)
5. Continues process until convergence

Lab session: Chip-seq motif discovery

Using peaks for de novo discovery of enriched motifs

(i) Load motif_example.R into RStudio. This shows how to plot your own motifs.

(ii) on prince cluster

Follow comments in *meme_motif_analysis.sh* to submit parallel job to cluster queueing system.

Lab session: ChIP-seq motif discovery-
analysis of results

MEME-ChIP calls a pipeline of analyses, including:

1. discover novel DNA-binding motifs (with MEME and DREME),
2. determine which motifs are most centrally enriched,
3. analyze them for similarity to known binding motifs, and
4. automatically group significant motifs by similarity,
5. create a GFF file for viewing each motif's predicted sites in a genome browser.

Differential chIP-seq

MACS2 can give a differential chIP-seq analysis between two conditions

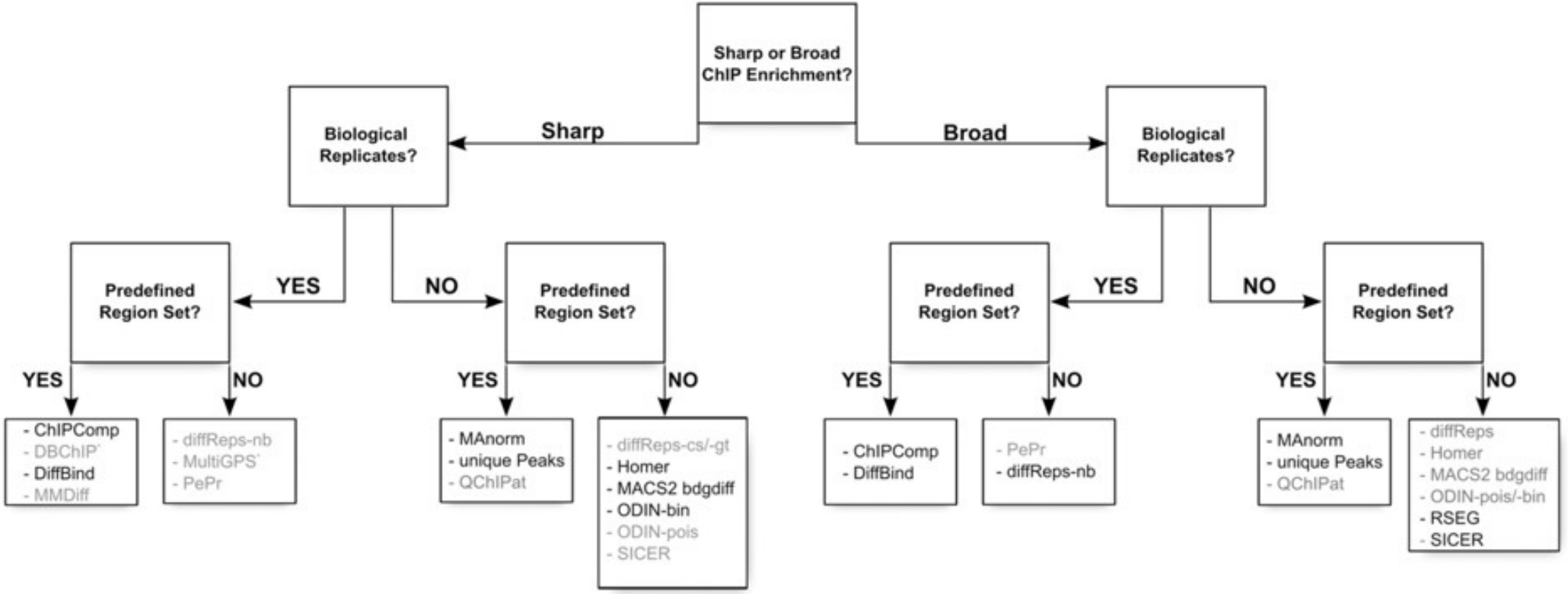
- Returns peaks differing between the two conditions.
- Uses the "callpeak" subcommand: **`macs2 callpeak`**
- For details see:

  https://github.com/taoliu/MACS/wiki/Call-differential-binding-events

Differential chIP-seq

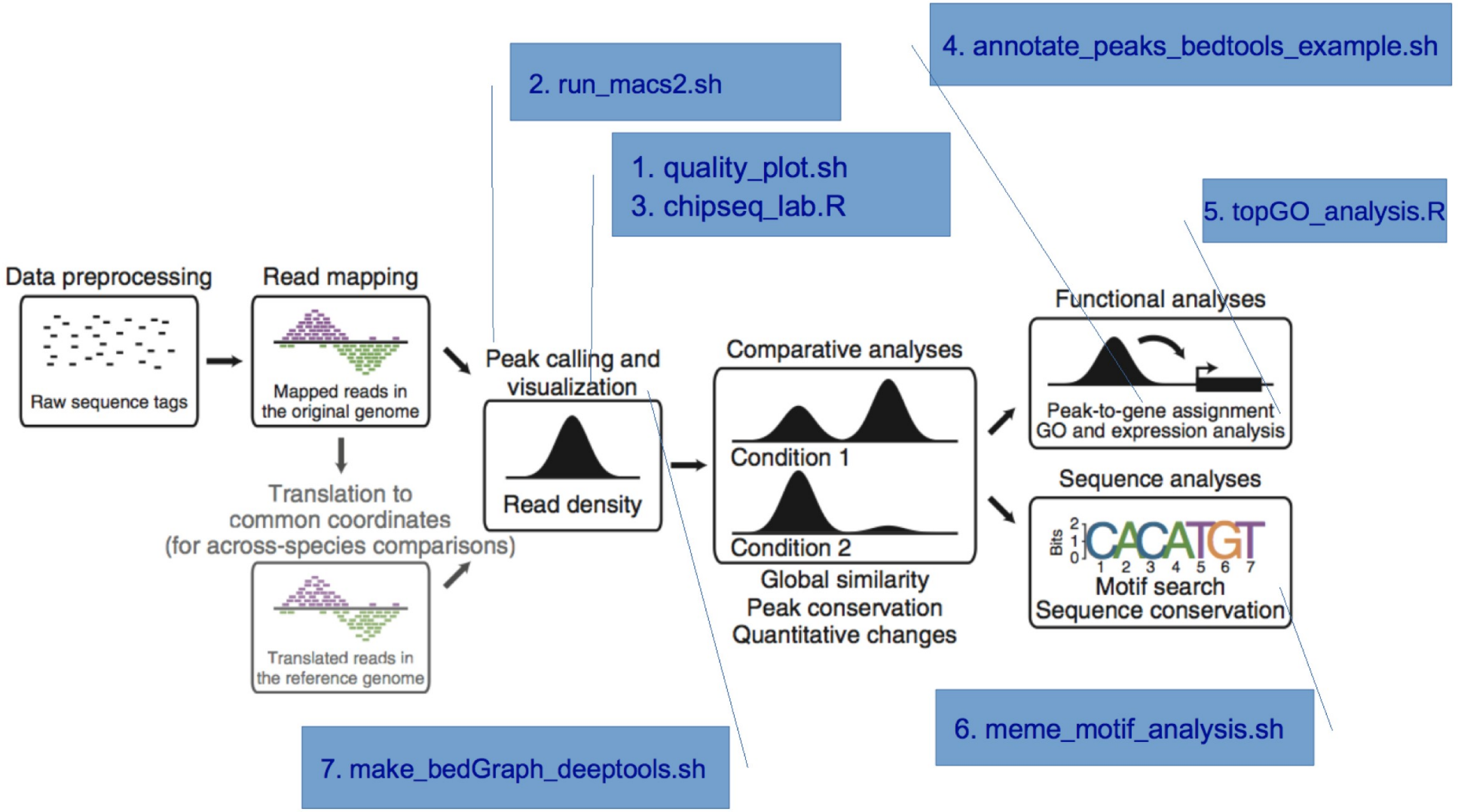
Other peak-calling programs can better use replicate information.

- See S. Steinhauser et al. "A comprehensive comparison of tools for differential chIP-seq". Brief Bioinform. 2016;17(6):953-966.



ChIP-seq pipeline- summary

(see chipseq_pipeline.sh)



*A. Bardet, Q. He, J. Zeitlinger & A. Stark. "A computational pipeline for comparative ChIP-seq analyses. Nature Protocols 7, 45–61 (2012)

End of ChIP-seq pipeline

ATAC-seq: based on transposon insertion