

# **Introduction to ChIP-Seq data analyses**

# Outline

- Introduction to ChIP-seq experiment.
  - Biological motivation.
  - Experimental procedure.
- Method and software for ChIP-seq peak calling.
  - Protein binding ChIP-seq.
  - Histone modifications.
- Higher order ChIP-seq data analysis.
  - Overlaps of peaks.
  - Differential binding.
  - Correlate with other data such as RNA-seq.

# **Introduction to ChIP-seq experiment**

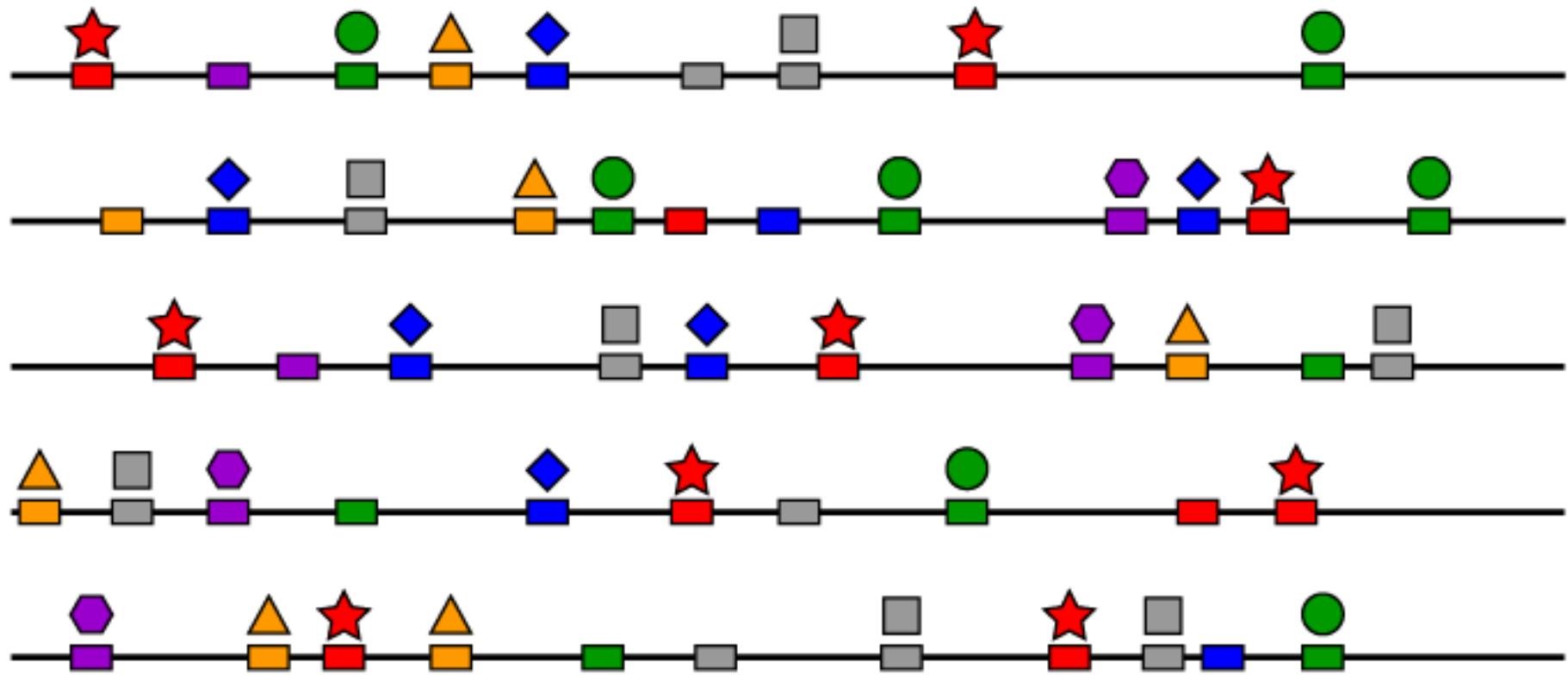
# ChIP-seq: Chromatin ImmunoPrecipitation + sequencing

- Scientific motivation: measure specific biological modifications along the genome:
  - Detect binding sites of DNA-binding proteins (transcription factors, pol2, etc.) .
  - quantify strengths of chromatin modifications (e.g., histone modifications).

# Experimental procedures

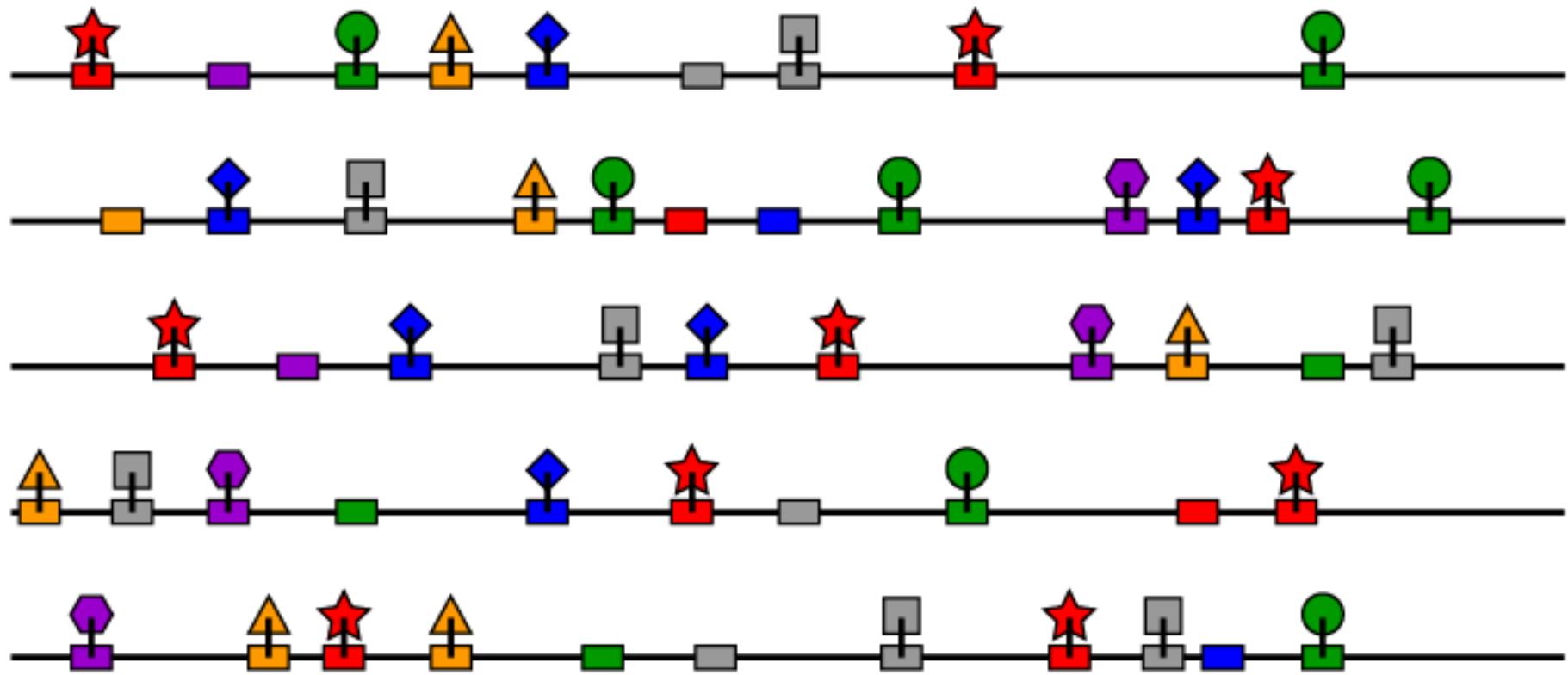
1. Crosslink: fix proteins on Isolate genomic DNA.
2. Sonication: cut DNA in small pieces of ~200bp.
3. IP: use antibody to capture DNA segments with specific proteins.
4. Reverse crosslink: remove protein from DNA.
5. Sequence the DNA segments.

# DNA with proteins



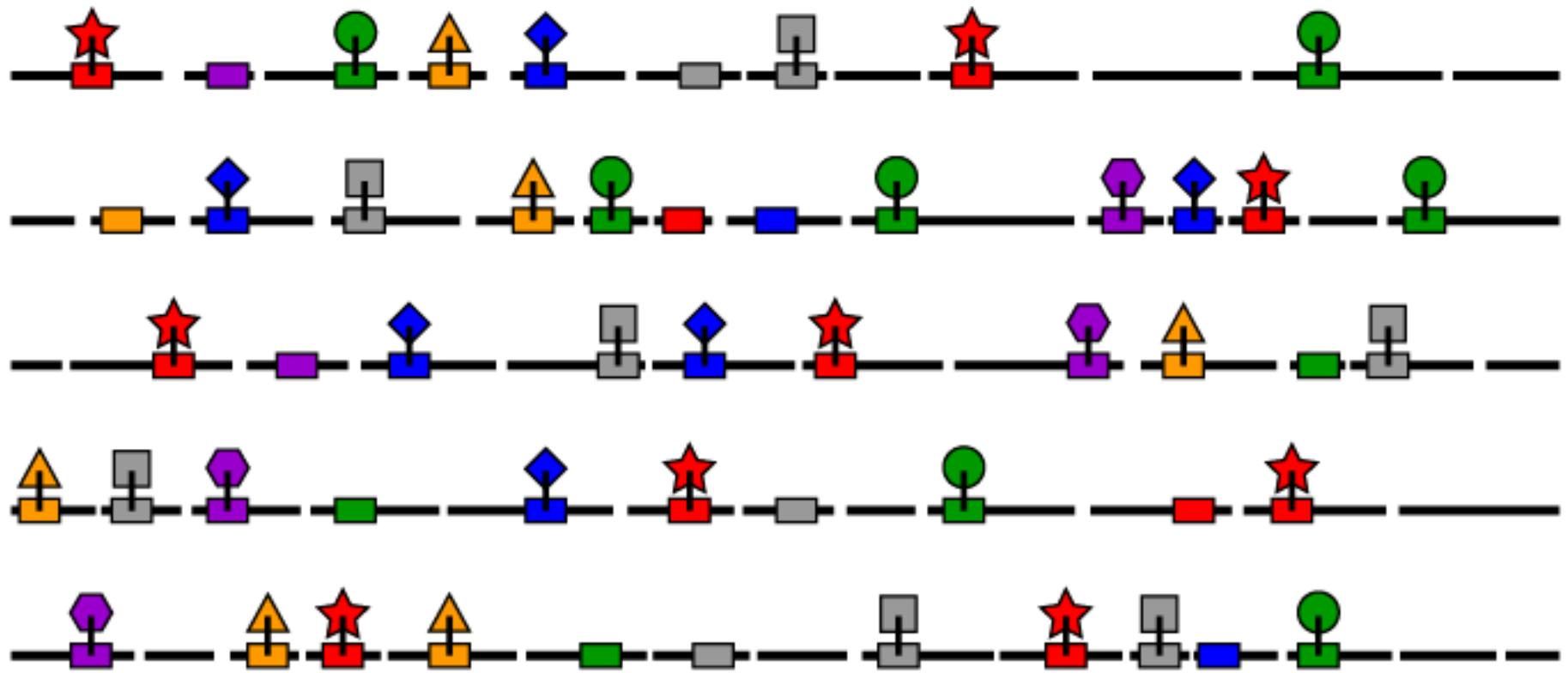
By Richard Bourgon at UC Berkley

# Protein/DNA Crosslinking *in vivo*

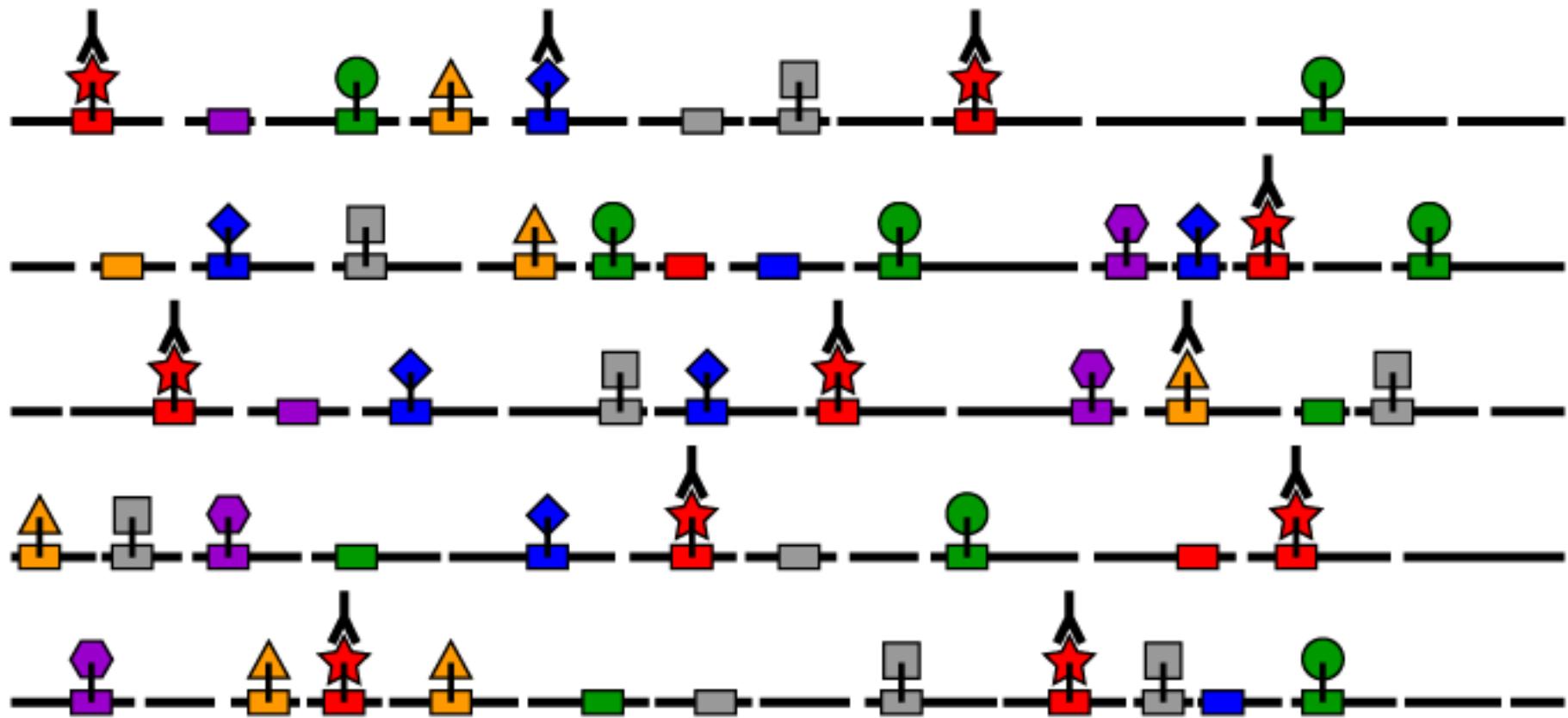


By Richard Bourgon at UC Berkley

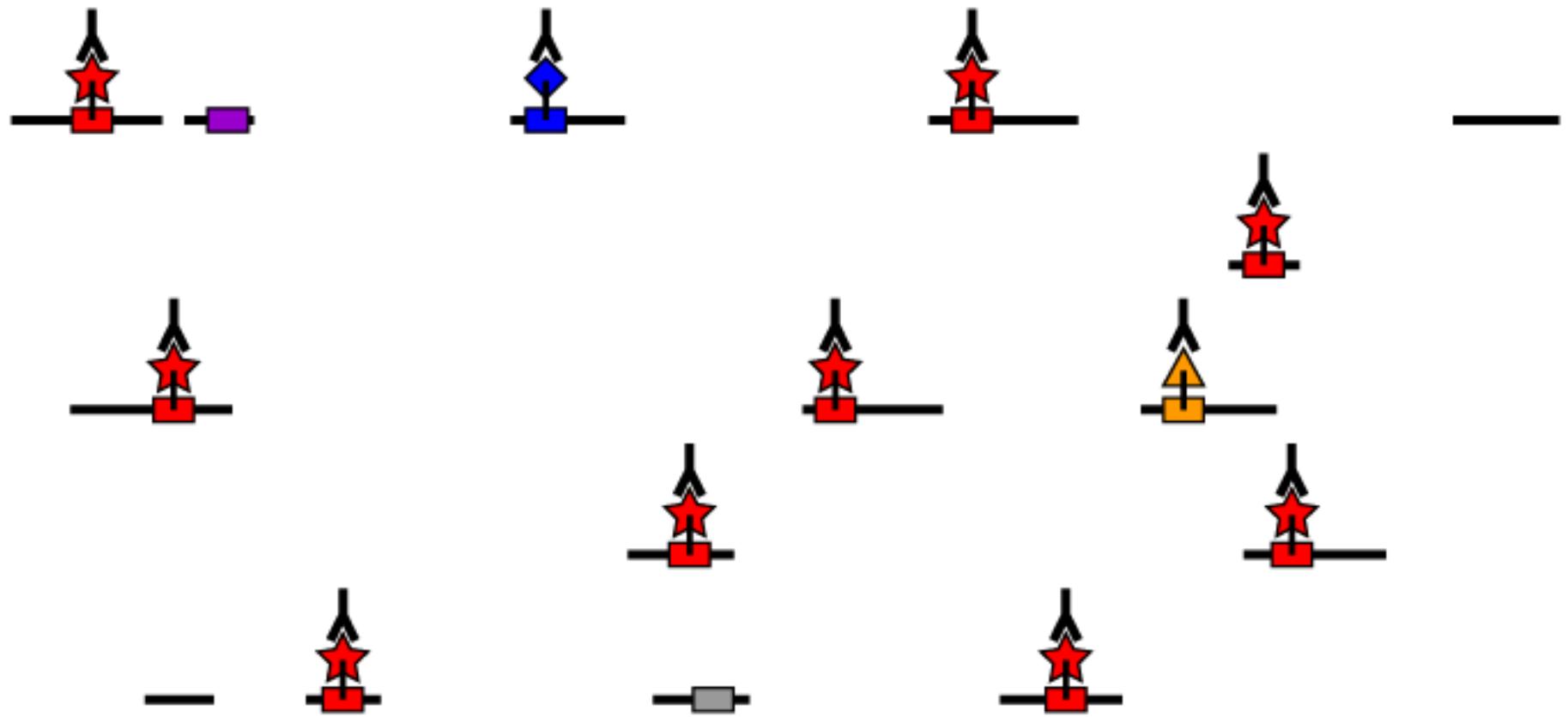
# Sonication (cut DNA into pieces)



# Capture using TF-specific Antibody

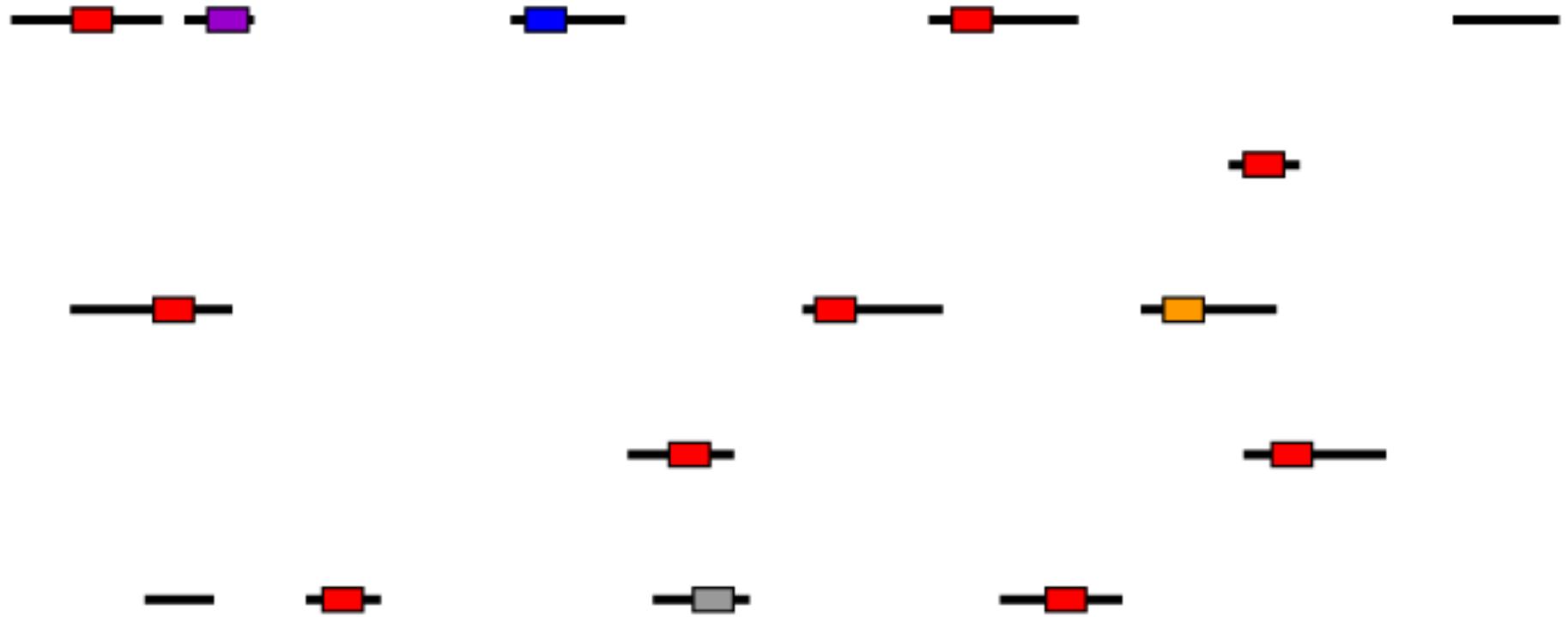


# Immunoprecipitation (IP)



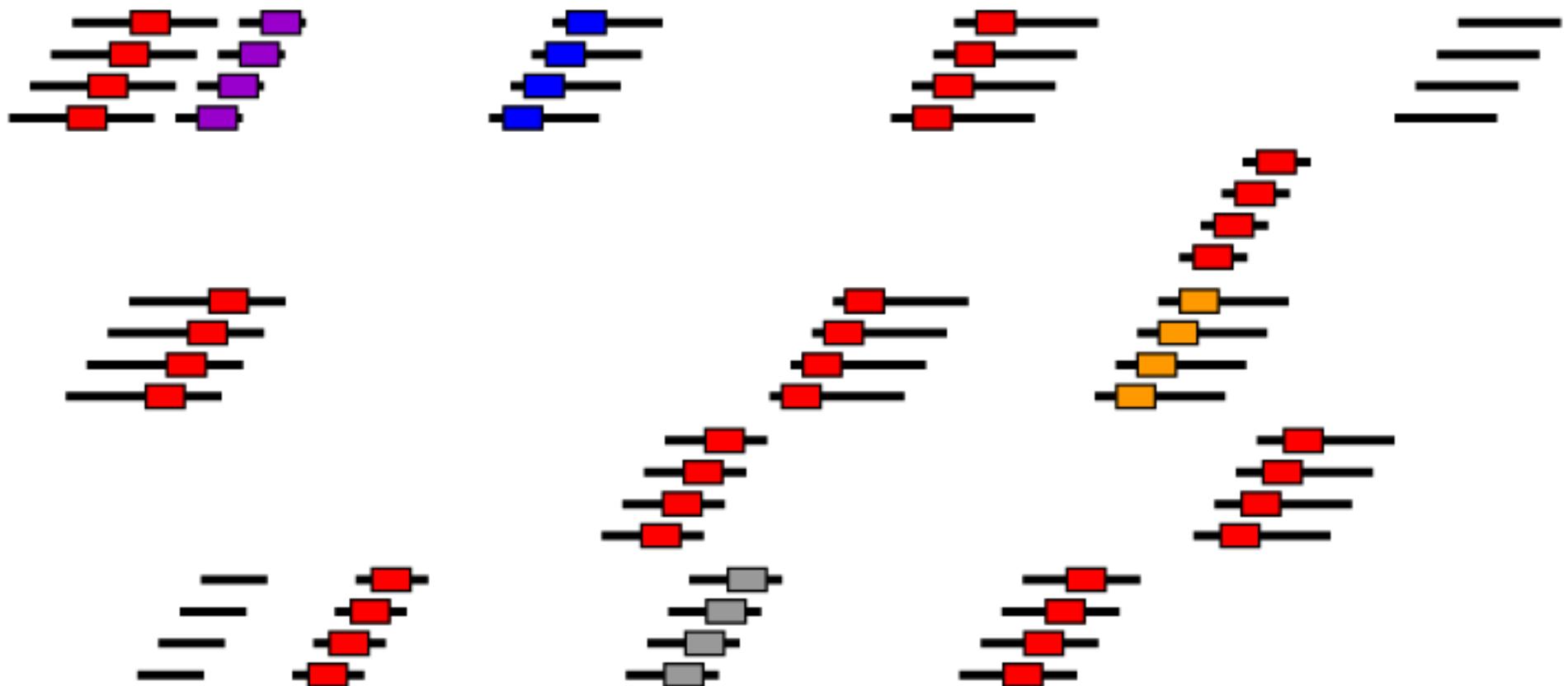
By Richard Bourgon at UC Berkley

# Reverse Crosslink and DNA Purification



By Richard Bourgon at UC Berkley

# Amplification (PCR)



By Richard Bourgon at UC Berkley

# Other similar sequencing technologies

- “Captured sequencing” – enrich and then sequence selected genomic regions.
- Similar technologies:
  - MeDIP-seq: measure methylated DNA.
  - DNase-seq: detect DNase I hypersensitive sites.
  - FAIRE-seq: detect open chromatin sites.
  - Hi-C: study 3D structure of chromatin conformation.
  - GRO-seq: map the position, amount and orientation of transcriptionally engaged RNA polymerases.
  - Ribo-seq: detect ribosome occupancy on mRNA. Captured mRNA-seq.
  - MeRIP-seq: measure RNA methylation. Captured mRNA-seq.
- Analysis techniques are more or less similar.

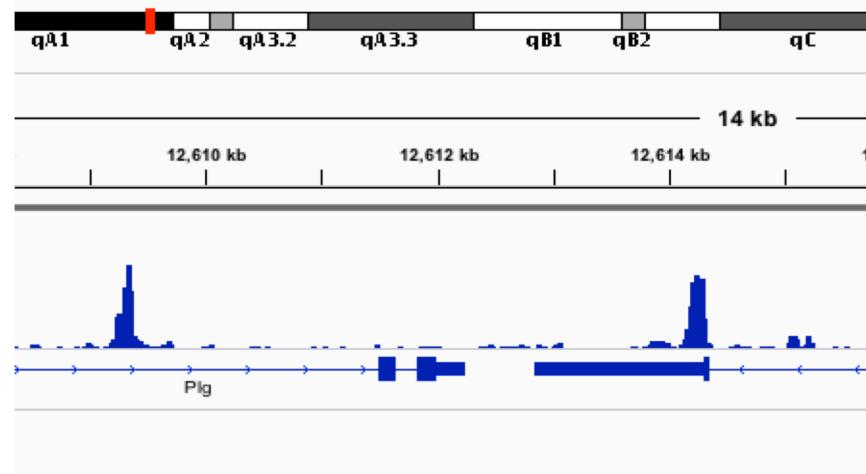
# **Methods and software for ChIP-seq peak/block calling**

# Data from ChIP-seq

- Raw data: sequence reads.
- After alignments: genome coordinates (chromosome/position) of all reads.
- Often, aligned reads are summarized into “counts” in equal sized bins genome-wide:
  1. segment genome into small bins of equal sizes (50bps).
  2. Count number of reads started at each bin.

# ChIP-seq “peak” detection

- When plot the read counts against genome coordinates, the binding sites show a tall and pointy peak. So “peaks” are used to refer to protein binding or histone modification sites.



- Peak detection is the most fundamental problem in ChIP-seq data analysis.

# Simple ideas for peak detection

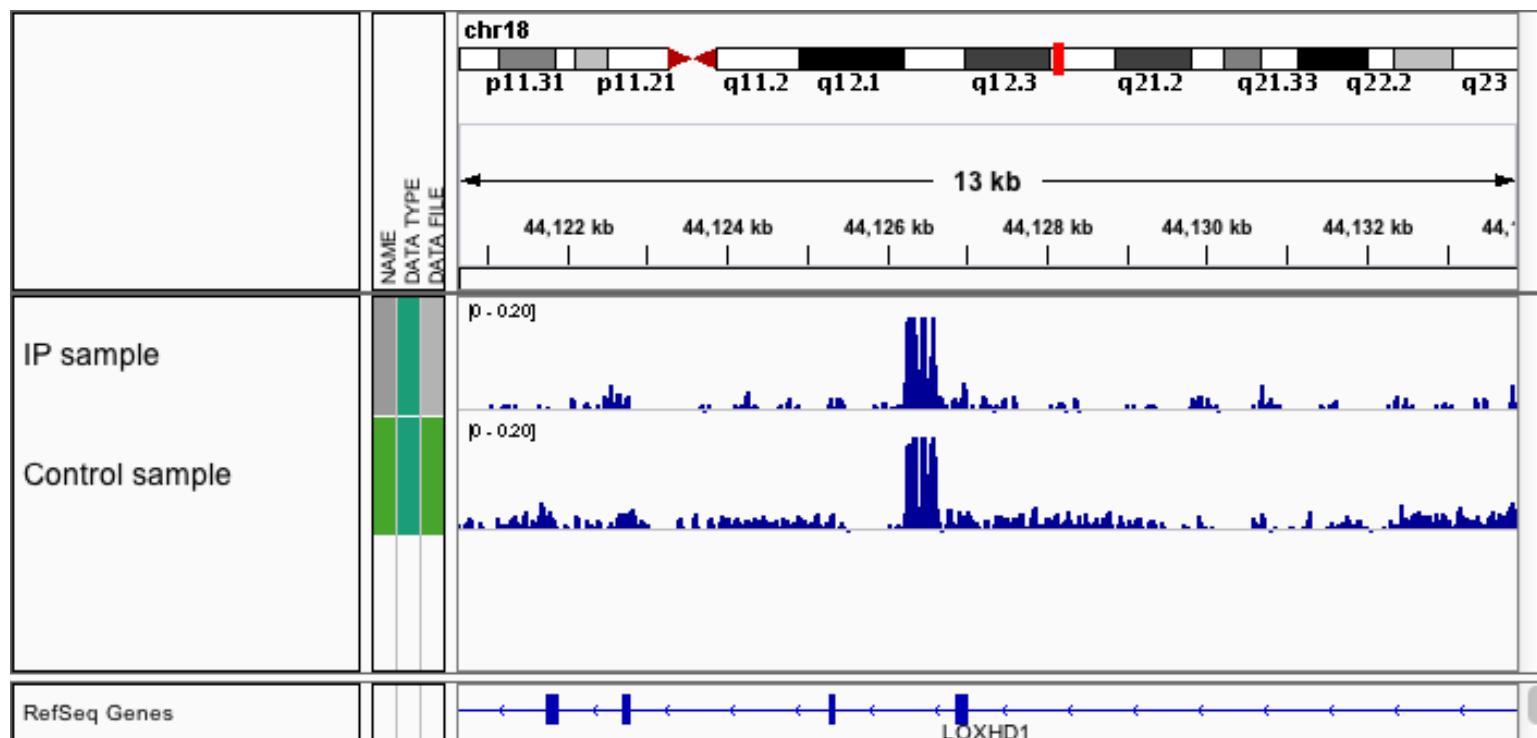
- Regions with reads clustered are likely to be peaks.
- Counts from neighboring windows need to be combined to make inference (so that it's more robust).
- To combine counts:
  - Smoothing based: moving average (MACS, CisGenome), HMM-based (Hpeak).
  - Model clustering of reads starting position (PICS, GPS).
- Moreover, some special characteristics of the data can be incorporated to improve the peak calling performance.

# **Before peak detection: what do we know about ChIP-seq?**

- Artifacts need to be considered.
  - DNA sequence: can affect amplification process or sequencing process
  - Chromatin structure (e.g., open chromatin region or not): may affect the DNA sonication process.
  - A control sample is necessary to correct artifacts.
- Reads clustered around binding sites to form two distinct peaks on different strands.
- Alignment issue: mappability.

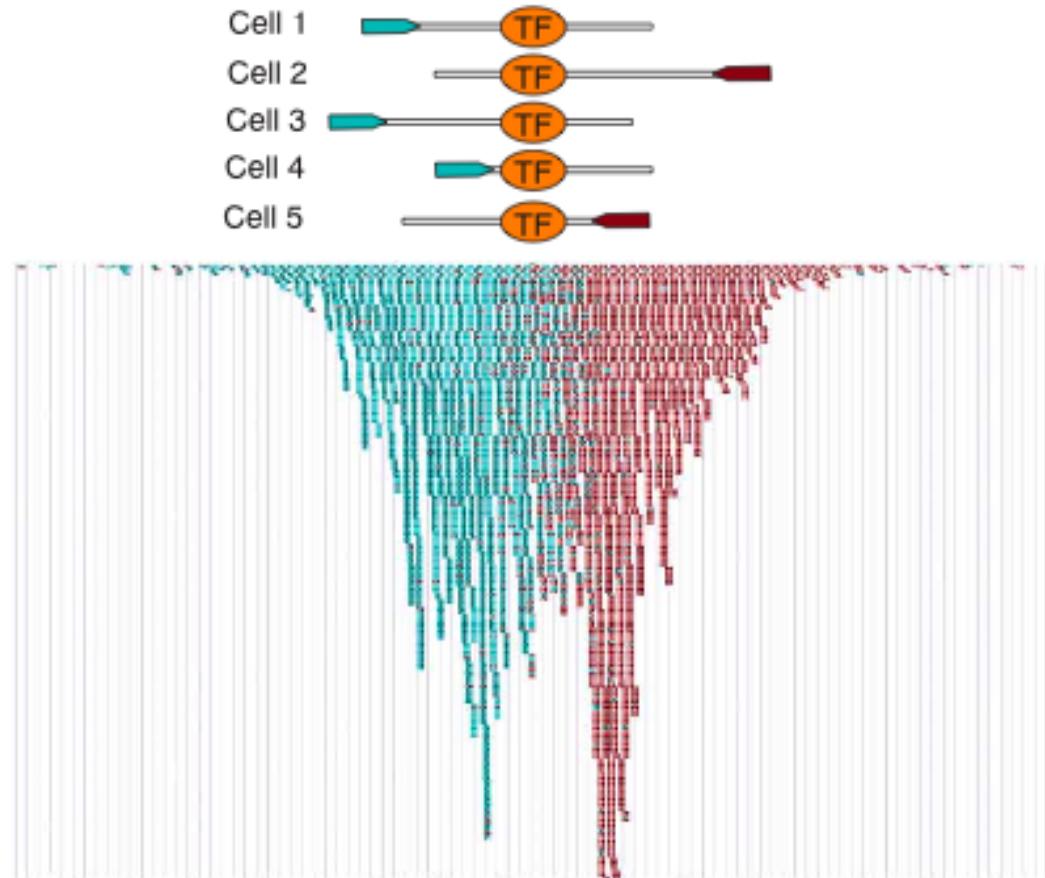
# Control sample is important

- A control sample is necessary for correcting many artifacts: DNA sequence dependent artifacts, chromatin structure, repetitive regions, etc.



# Reads aligned to different strands

- Number of Reads aligned to different strands form two distinct peaks around the true binding sites.
- This information can be used to help peak detection.



# Mappability

- For each basepair position in the genome, whether a sequence read starting from this position can be uniquely mapped to a genome location.
- Regions with low mappability (highly repetitive) cannot have high counts, thus affect the ability to detect peaks.

**Table 1 Genome mappability fraction**

Organism	Genome size (Mb)	Nonrepetitive sequence		Mappable sequence	
		Size (Mb)	Percentage	Size (Mb)	Percentage
<i>Caenorhabditis elegans</i>	100.28	87.01	86.8%	93.26	93.0%
<i>Drosophila melanogaster</i>	168.74	117.45	69.6%	121.40	71.9%
<i>Mus musculus</i>	2,654.91	1,438.61	54.2%	2,150.57	81.0%
<i>Homo sapiens</i>	3,080.44	1,462.69	47.5%	2,451.96	79.6%

# Peak detection software

- MACS
- Cisgenome
- QuEST
- Hpeak
- PICS
- GPS
- PeakSeq
- MOSAiCS
- ...

# MACS (Model-based Analysis of ChIP-Seq)

## Zhang et al. 2008, *GB*

- Estimate shift size of reads  $d$  from the distance of two modes from + and – strands.
- Shift all reads toward 3' end by  $d/2$ .
- Use a dynamic Possion model to scan genome and score peaks. Counts in a window are assumed to following Poisson distribution with rate:  $\lambda_{\text{local}} = \max(\lambda_{\text{BG}}, [\lambda_{1k},] \lambda_{5k}, \lambda_{10k})$ 
  - The dynamic rate capture the local fluctuation of counts.
- FDR estimates from sample swapping: flip the IP and control samples and call peaks. Number of peaks detected under each p-value cutoff will be used as null and used to compute FDR.

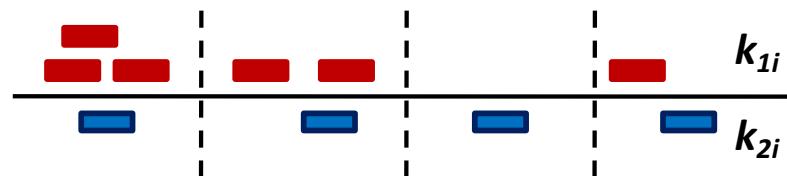
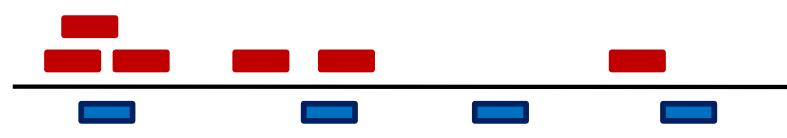
# Using MACS

- Written in python
- Newer versions are MACS2 and MACS3:
  - [https://hbctraining.github.io/Intro-to-ChIPseq/lessons/05\\_peak\\_calling\\_macs.html](https://hbctraining.github.io/Intro-to-ChIPseq/lessons/05_peak_calling_macs.html)
  - <https://github.com/macs3-project/MACS>
- Syntax:

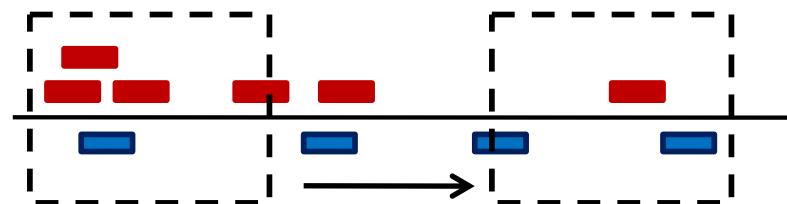
```
macs2 callpeak -t ChIP.bam -c Control.bam \
-f BAM -g hs -n output
```

# Cisgenome (Ji et al. 2008, NBT)

- Implemented with Windows GUI.
- Use a Binomial model to score peaks.

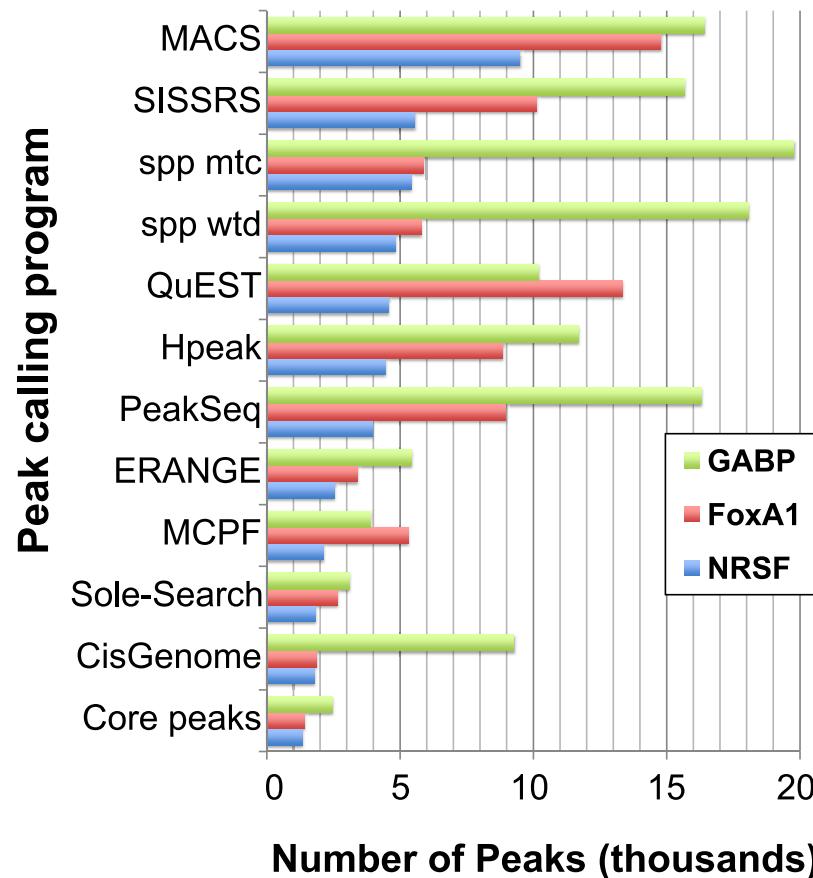


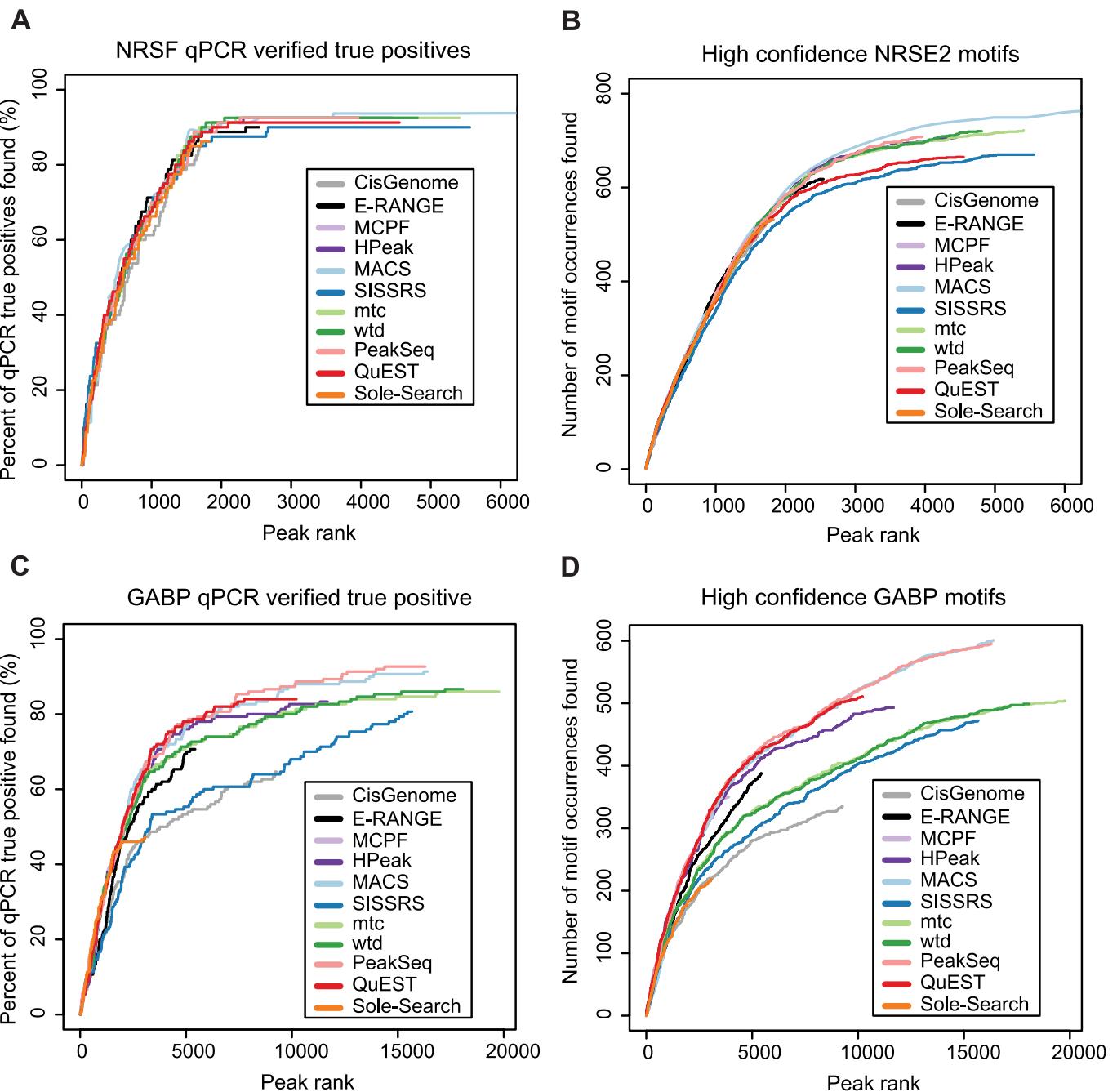
$$n_i = k_{1i} + k_{2i}$$
$$k_{1i} / n_i \sim \text{Binom}(n_i, p_0)$$



# Comparing peak calling algorithms

- Wilbanks et al. (2010) *PloS One*
- Laajala et al. (2009) *BMC Genomics*





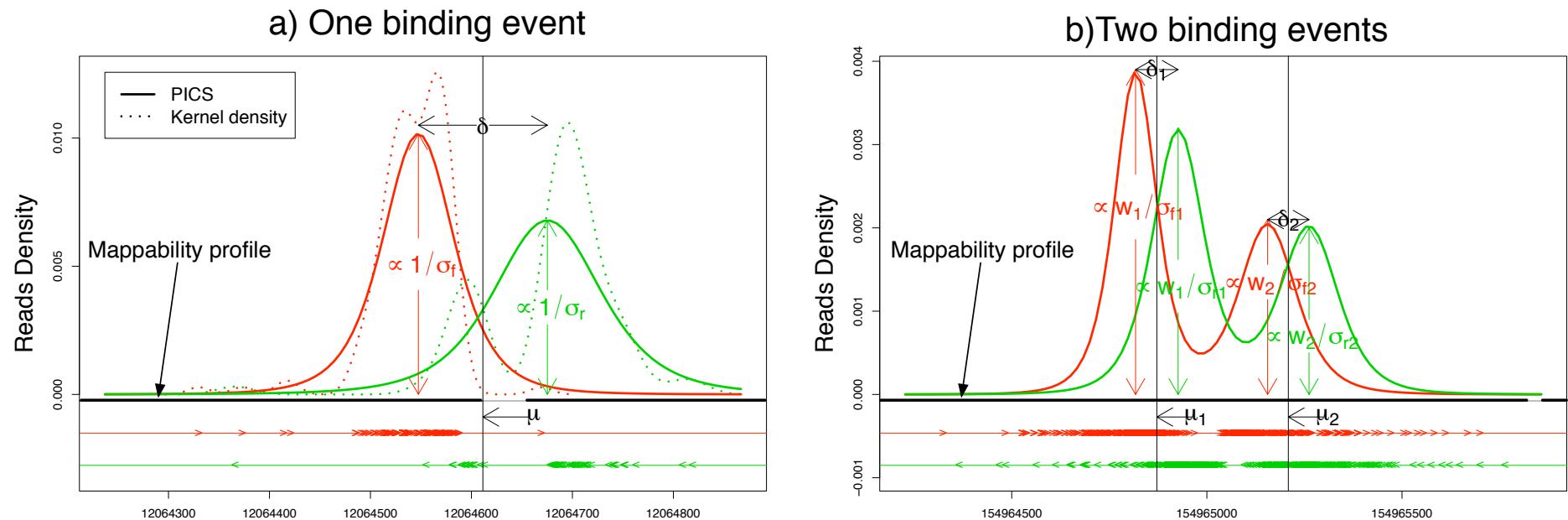
# **Another type of approach: modeling the read locations**

- Regions with more reads clustered tend to be binding sites.
- This is similar to using binned read counts.
- Reads mapped to forward/reverse strands are considered separately.
- Peak shapes can be incorporated.

# PICS: Probabilistic Inference for ChIP-seq (Zhang *et al.* 2010 *Biometrics*)

- Use shifted t-distributions to model peak shape.
- Can deal with the clustering of multiple peaks in a small region.
- A two step approach:
  - Roughly locate the candidate regions.
  - Fit the model at each candidate region and assign a score.
- EM algorithm for estimating parameters.
- Computationally very intensive.

# PICS



$$f_i \sim \sum_{k=1}^K w_k t_4(\mu_{fk}, \sigma_{fk}^2) \stackrel{d}{=} g_f(f_i | \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\delta}, \boldsymbol{\sigma}_f)$$

$$r_j \sim \sum_{k=1}^K w_k t_4(\mu_{rk}, \sigma_{rk}^2) \stackrel{d}{=} g_r(r_j | \boldsymbol{w}, \boldsymbol{\mu}, \boldsymbol{\delta}, \boldsymbol{\sigma}_r)$$

# GPS

**Guo *et al.* 2010, *Bioinformatics***

- The general idea is very similar to PICS.
- Use non-parametric distribution to model the peak shape.
- Estimation of peak shape and peak detection are iterated until convergence.

# Use GPS

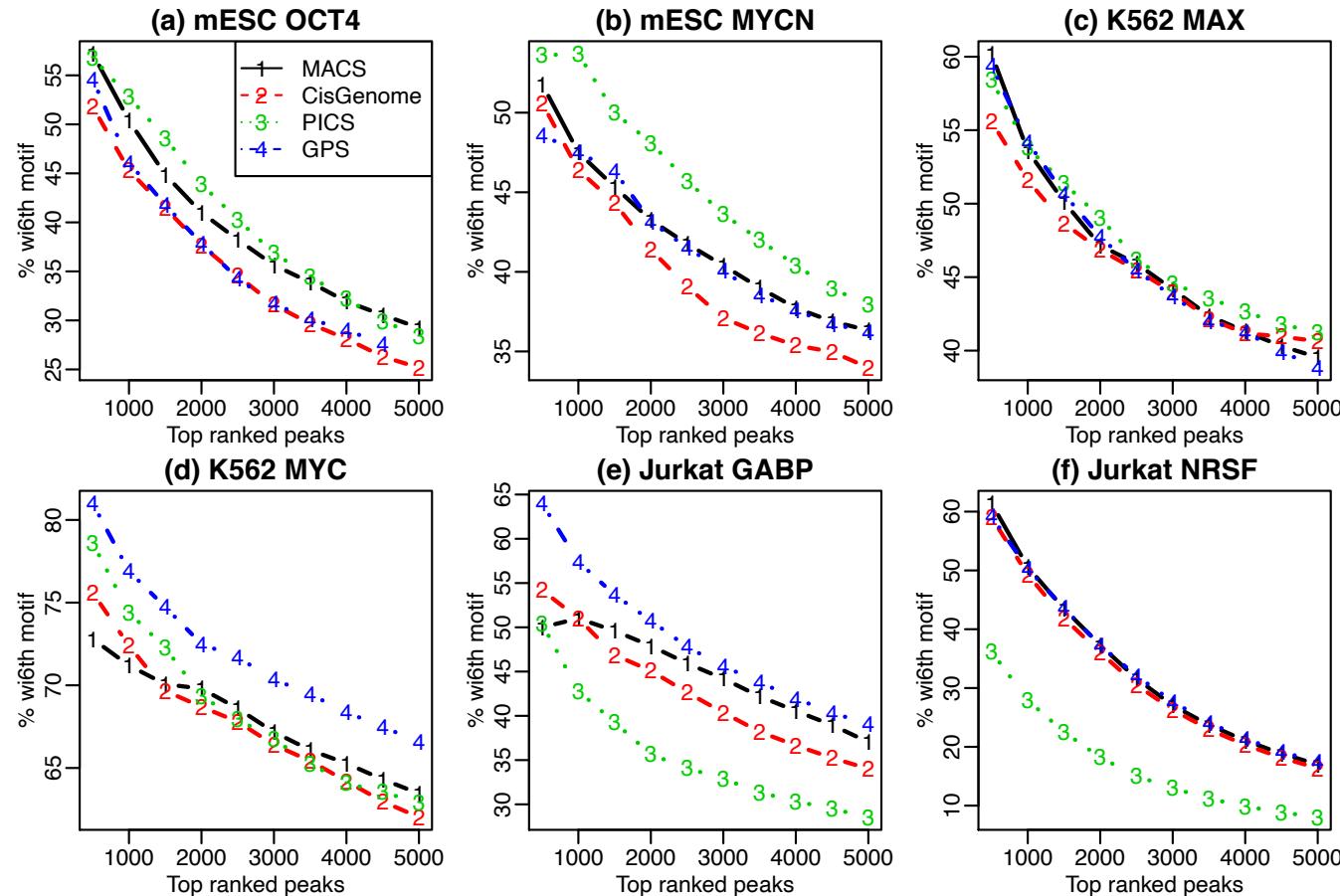
- Run following command:

```
java -Xmx1G -jar gps.jar --g mm8.info --d  
Read_Distribution_default.txt --expt  
IP.bed --ctrl control.bed --f BED --out  
result
```

- It's much slower than MACS or CisGenome.  
So we won't do it in the lab.

# A little more comparison

- I found that using peak shapes helps. GPS tend to perform better. PICS seems not stable.



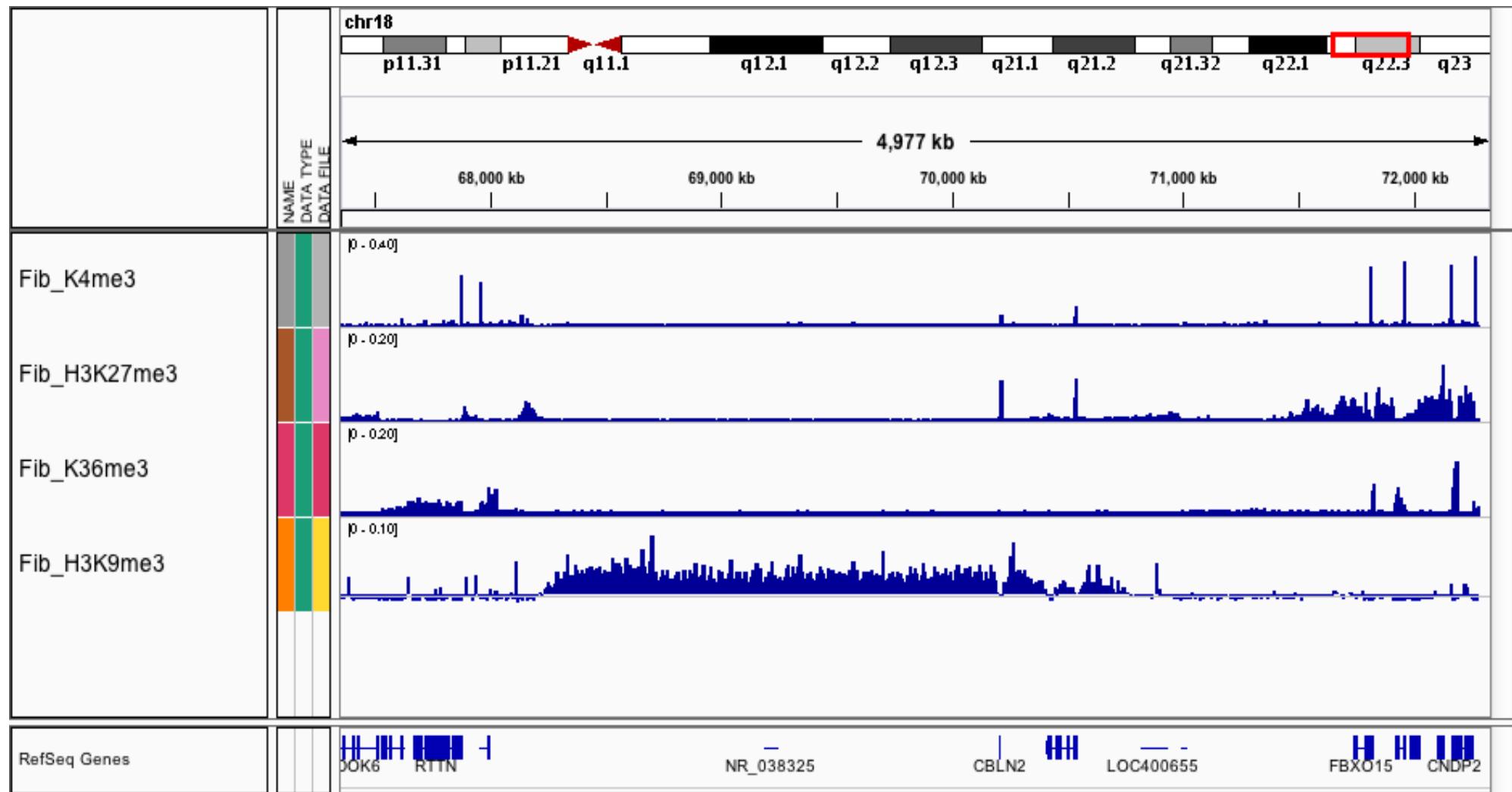
# Bioconductor packages for protein binding ChIP-seq

- There are several packages: chipseq, ChIPseqR, BayesPeak, PICS, etc., but not very popular.
- Most people use command line driven software like MACS or CisGenome GUI.

# ChIP-seq for histone modification

- Histone modifications have various patterns.
  - Some are similar to protein binding data, e.g., with tall, sharp peaks: H3K4.
  - Some have wide (mega-bp) “blocks”: H3k9.
  - Some are variable, with both peaks and blocks: H3k27me3, H3k36me3.

# Histone modification ChIP-seq data



# **Peak/block calling from histone ChIP-seq**

- Use the software developed for TF data:
  - Works fine for some data (K4, K27, K36).
  - Not ideal for K9: it tends to separate a long block into smaller pieces.
- Many existing methods, mostly based on smoothing, HMM or wavelet.

# Complications in histone peak/block calling

- Smoothing-based method:
  - Long block requires bigger smoothing span, which hurts boundary detection.
  - Data with mixed peak/block (K27me3, K36me3) requires varied span: adaptive fitting is computationally infeasible.
- HMM based method:
  - Tend to over fit. Sometimes need to manually specify transition matrix.

# Available methods/software for histone data peak calling

- MACS2
- BCP (Bayesian change point caller)
- SICER
- RSEG
- UW Hotspot
- BroadPeak
- mosaicsHMM
- WaveSeq
- ZINBA
- ARHMM
- ...

# MACS2

- Has an option for broad peak calling, which uses post hoc approach to combine nearby peaks.
- Syntax:

```
macs2 callpeak -t ChIP.bam -c Control.bam \
--broad -g hs --broad-cutoff 0.1
```

# RSEG

- By Andrew Smith at USC:  
<http://smithlabresearch.org/software/rseg/>
- Use negative binomial distribution to model the bin counts, NBDiff distribution for differences between IP and control.
- HMM (3-state for TF data, 2-state for epigenomic domains) for genome segmentation. Use permutation to calculate p-values and determine boundaries.

# Use RSEG

- Inputs are bed files.
- First determine “deadzone” (low or unmappable regions). Deadzones for different species can be obtained from their website.

```
deadzone -s fa -k 32 -o deadzones-mm9-k32.bed mm9
```

- Then call blocks:

```
rseg-diff -c mouse-mm9-size.bed -o output.bed -i  
20 -v -mode 2 -d deadzone-mm9-k32.bed IP.bed  
control.bed
```

# SICER

Zang *et al.* 2009, Bioinformatics

- Algorithm:
  - Cut genome into non-overlapping windows and compute a score for each window based on a Poisson model.
  - Identify “islands” by thresholding the scores.
  - Compute a score for each island. This is the tricky part.

# Use SICER

- The software is written in python.
- Inputs are bed files for IP and control.
- Good computational performance.
- Results are sometimes sensitive to the parameters.
- A typical command is like:

```
SICER.sh . h3k27me3.bed control.bed .\  
hg19 2 200 150 0.74 600 0.01
```

# ARHMM

## Rashid *et al.* (2014) JASA

- Use ARHMM (auto-regressive HMM) to model the binned read counts.
  - The AR part has smoothing effects which overcomes the problem of HMM that it tends to generate smaller blocks.
- Has capability to include more covariates, and do model selection.
  - Consider IP counts are response, covariates can be control counts, GC content, mappability, TF bindings, etc.
- According to my limited experience, the results seem to be desirable.
- An R package is available at <https://code.google.com/p/hmmcov/>, but not in very good shape.

# Summary for ChIP-seq peak/block calling

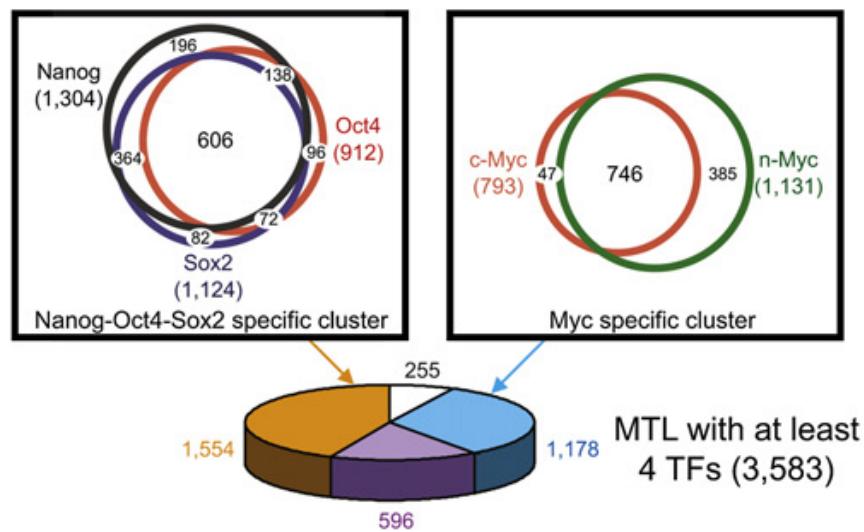
- Detect regions with reads enriched.
- Control sample is important.
- Incorporate some special characteristics of the data improves results.
- Calling blocks (long peaks) is harder.
- Many software available.

# After peak/block calling

- Compare results among different samples:
  - Presence/absence of peaks.
  - Differential binding.
  - Combinatory patterns.
- Compare results with other type of data:
  - Correlate TF binding with gene expressions from RNA-seq.

# Comparison of multiple ChIP-seq

- It's important to understand the co-occurrence patterns of different TF bindings and/or histone modifications.
- Post hoc methods: look at overlaps of peaks and represent by Venn Diagram.
  - This can be done using different tools. We'll practice using Bioconductor packages in the lab.



# Differential binding (DB)

- This is different from the overlapping analysis, because it considers quantitative changes.
- Straightforward methods:
  - Call peaks from individual dataset.
  - Union the called peaks to form candidate regions.
  - Treat the candidate regions as genes, then use RNA-seq method to test. Or model the differences of normalized counts from two conditions

# Issues to consider in DB analysis

- How to use control data:
  - Need to model the IP-control relationship.
  - Simply subtracting control might not be ideal.
- Normalization between experiments:
  - Signal to noise ratios (SNRs) are different due to technical and biological artifacts.
- Biological variations and experimental design (same as in RNA-seq).

# Existing method/software for DB analysis

- ChIPDiff (Xu et al. 2008, *Bioinformatics*): HMM on differences of normalized IP counts between two groups.
- DIME (Taslim *et al.* 2009, 2011, *Bioinformatics*): finite mixture model on differences of normalized IP counts.
- MAnorm (Shao *et al.* 2012, *Genome Biology*): normalization based on MA plot of counts from two groups, then use normalized “M” values to rank differential peaks.
- ChIPnorm (Nair *et al.* 2012, *PLoS One*): quantile normalization for each data. *Ad hoc* method for detecting differential peak.
- DBChIP (Liang *et al.* 2012 *Bioinformatics*) and DiffBind: Bioconductor packages, based on RNA-seq method.
- ChIPComp (Chen *et al.* 2015 *Bioinformatics*): Based on linear model framework, works for general design.

# Combine ChIP- and RNA-seq

- It is of great interest to study how the gene expressions are controlled by protein bindings and epigenetic modifications.
- Easy approach:
  - Look at the correlation of promoter TF binding (from ChIP-seq), and gene expression (from RNA-seq).
- More advanced approaches:
  - Build a model to predict gene expression (from RNA-seq) from protein binding and epigenetic data (from ChIP-seq).
  - Build a network for all ChIP- and RNA-seq data.

# Predict expression from TF binding

## Ouyang et al. (2009) PNAS

- Goal: to build a model to predict gene expressions using 12 TF binding datasets.
- Data: mouse ESC TF data from a cell paper by a Singapore group.
- Method: regression based.
- A similar paper using histone modification to predict gene expression is Karlic et al. (2010) PNAS.

# Procedures in Ouyang *et al.*

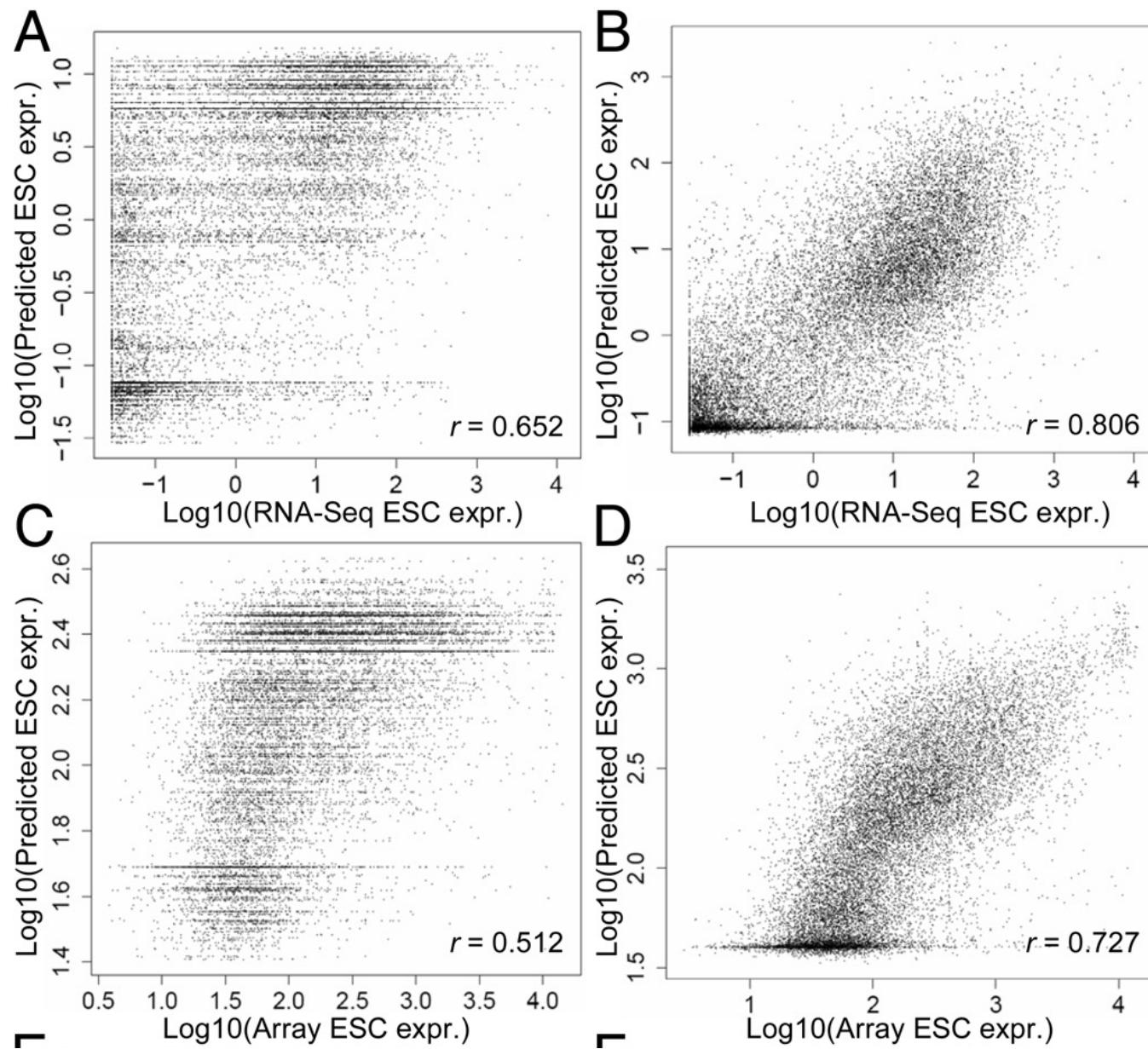
- Read counts are first summarized into gene level.
- Association strength between TF  $j$  and gene  $i$  is:

$$a_{ij} = \sum_k g_k e^{-d_k/d_0},$$

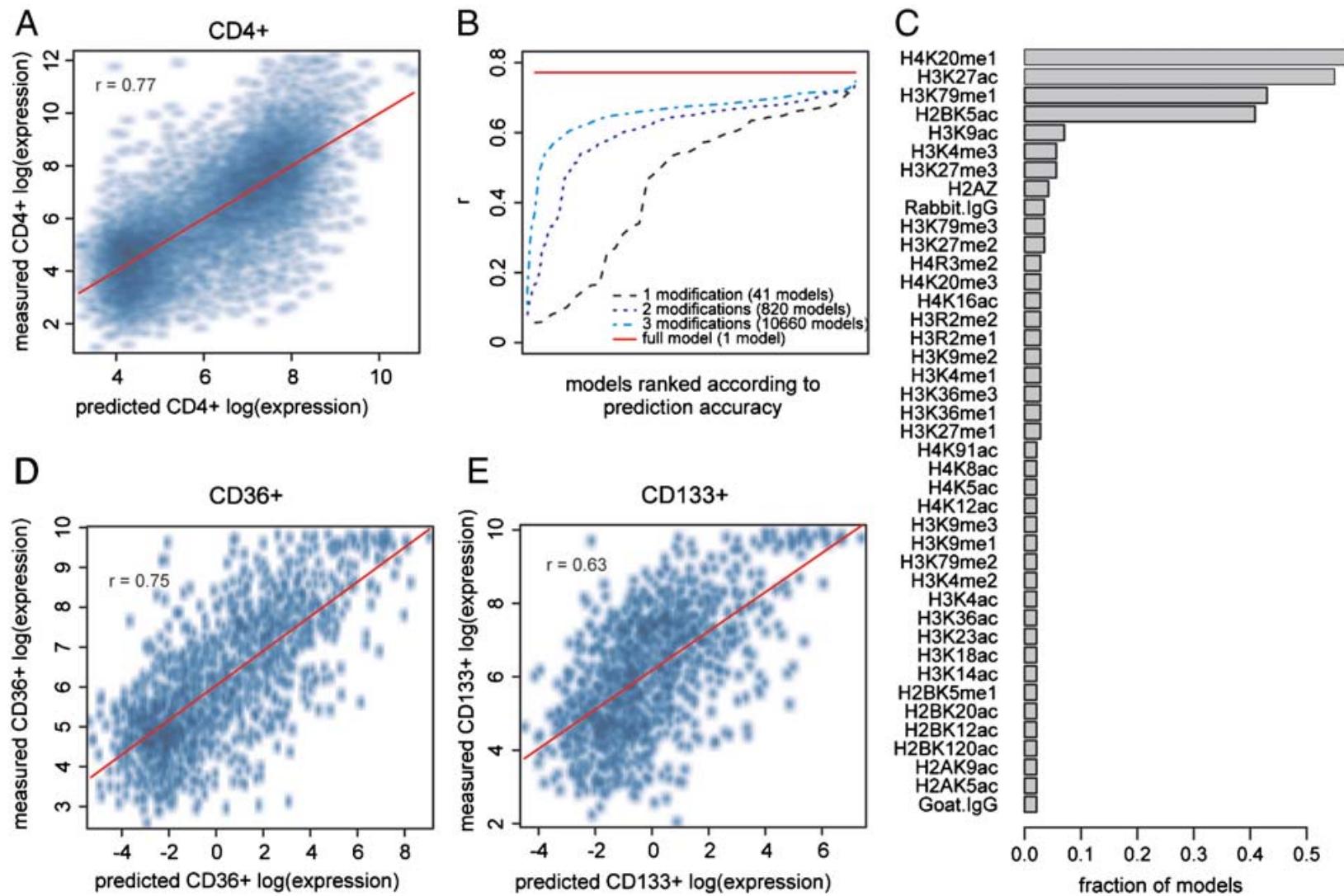
where  $g_k$  is the intensity (number of reads aligned to the coordinate) of the  $k$ th binding peak of the TF  $j$ ,  $d_k$  is the distance (number of nucleotides) between the TSS of gene  $i$  and the  $k$ th binding peak in the reference genome, and  $d_0$  is a constant. In theory, the summation is over all binding peaks of a given TF.

- Result  $a_{ij}$  is a matrix of  $n_{\text{genes}}$  by  $n_{\text{TF}}$ .
- PCA on  $a_{ij}$  to avoid having one TF dominating.
- log-linear model:  $\log Y_i = \mu + \sum_{j=1}^M \beta_j X_{ij} + \varepsilon_i,$

# Prediction results from TF binding



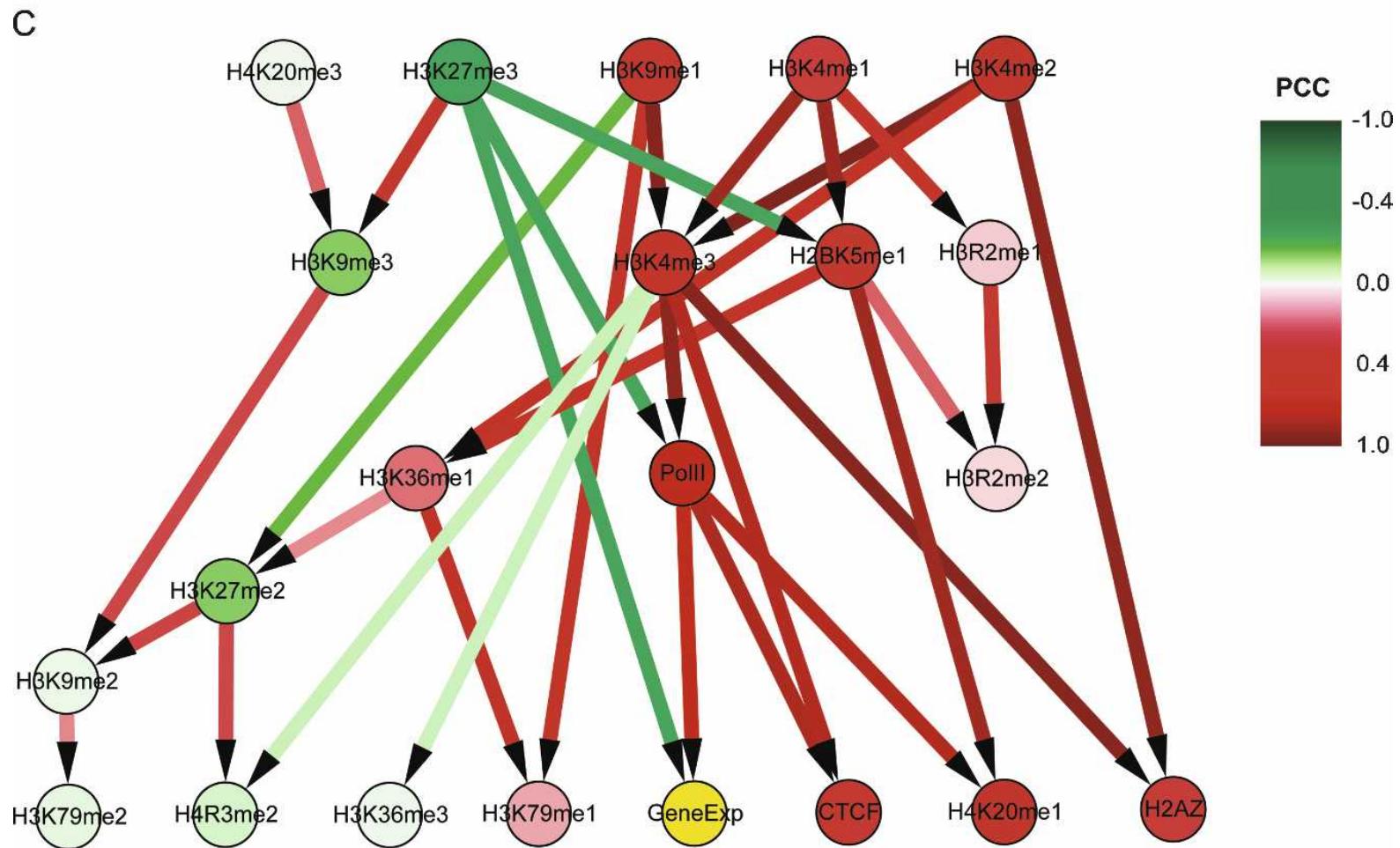
# Prediction results from histone modification



# Network based analysis of multiple ChIP-seq

- Yu et al. (2008) *Genome Research*.
- Data used: human CD4+ T-cell chip-seq for 23 histones and TF binding (from Keji Zhao's Cell paper). Read counts are summarized into TSS +/- 1kb region.
- Method:
  - Bayesian network on discretized counts using WinMine. A randomization procedure is implemented to select the robust edges.

# Result from BN



# Review

- ChIP-seq detects TFBS or measure histone modifications along the genome.
- Peak (short and long) detection is the major goal of data analysis.
- Number of aligned reads are input data. Data in neighboring regions need to be combined to call peaks.
- Many similar technologies, and the method are more or less the same.