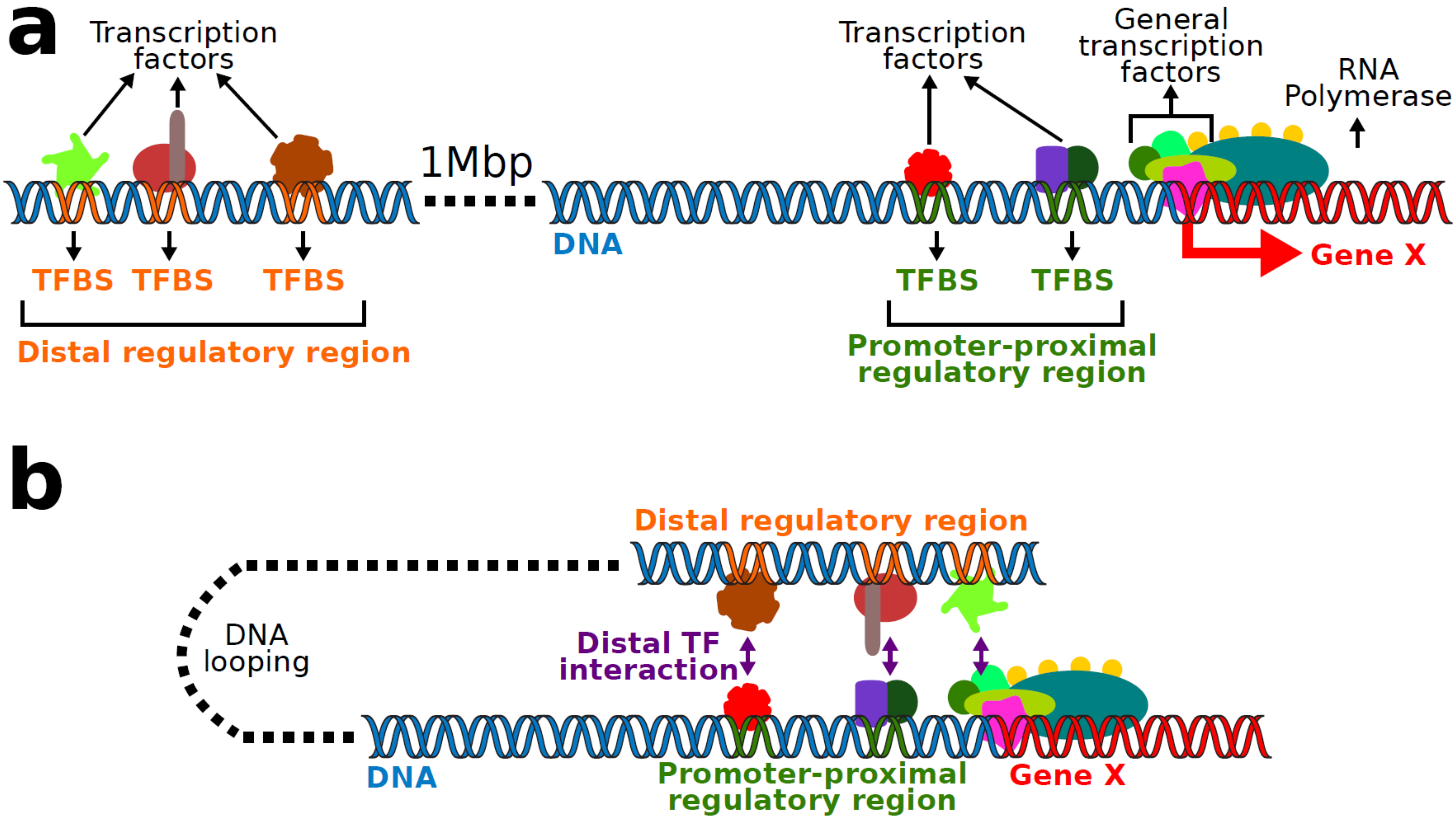
Practical Example: NGS of Regulatory Genomics Data

Ivan Gesteira Costa & Eduardo Gusmao
IZKF Research Group Bioinformatics

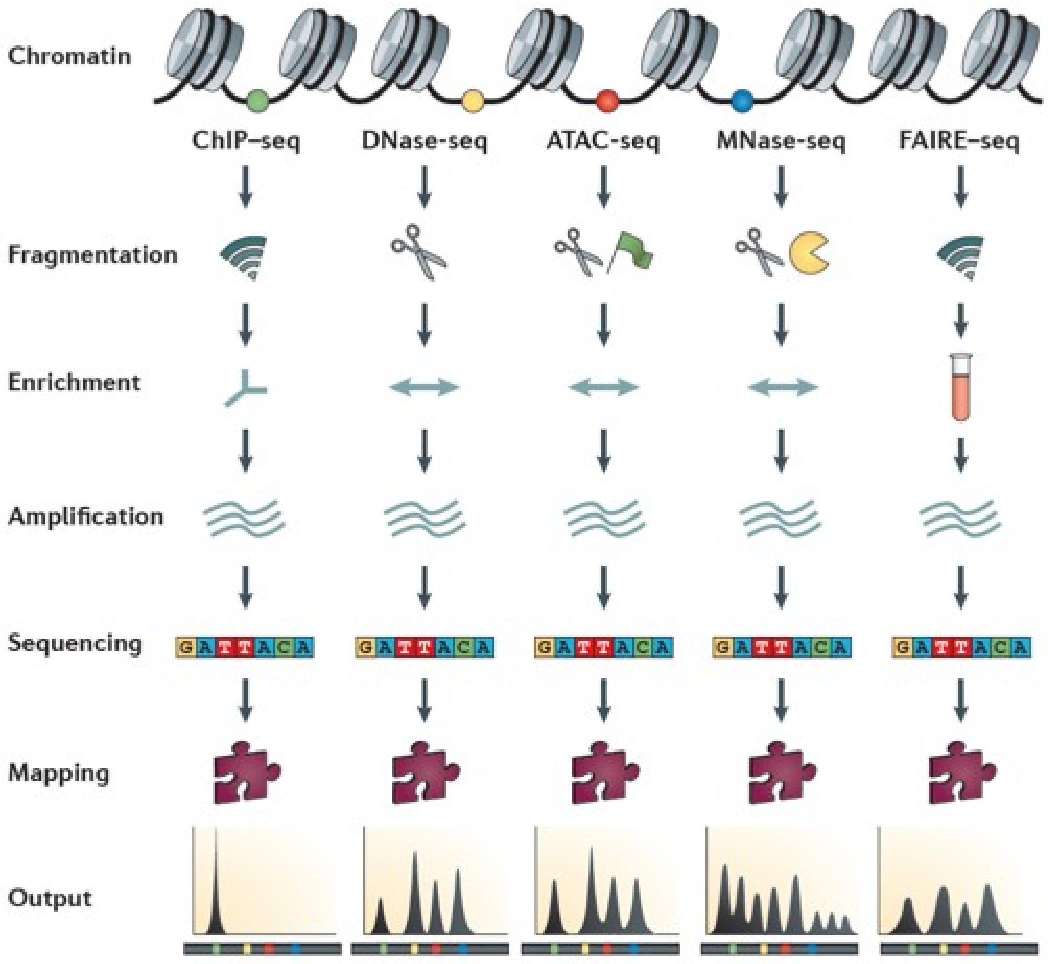
IZKF Interdisziplinäres Zentrum für Klinische Forschung
RWTH AACHEN UNIVERSITY

Review on Next Generation Sequencing (NGS)

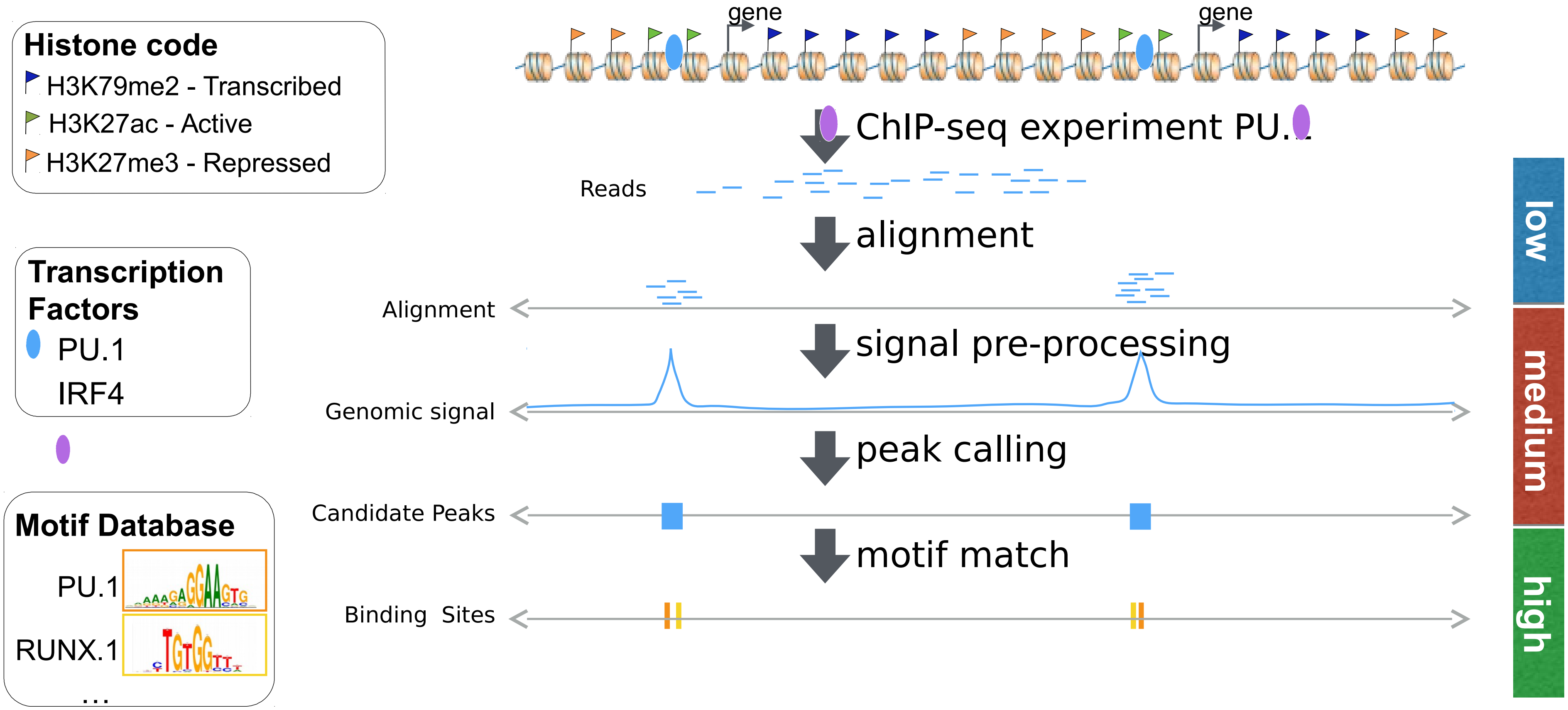
Understanding Gene Regulation



Understanding Gene Regulation



ChIP-seq Experimental Pipeline



Histone code

H3K79me2 - Transcribed

H3K27ac - Active

H3K27me3 - Repressed

Transcription Factors

PU.1

IRF4

Motif Database

PU.1

RUNX.1

...

ChIP-seq experiment PU.1

alignment

signal pre-processing

peak calling

motif match

Input Data – DNA sequences

FASTA File

- Store DNA sequences in a text-based file.

- Mainly used to store large genomic sequences.

- Header (lines that start with '>') + DNA sequence.

- DNA alphabet: A, C, G, T, N.

```
>SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
```

FASTQ File

- Also text-based. Mainly used to store short DNA sequences (reads) from NGS-based experiments.

- **Line 1:** Begins with a '@' character and is followed by a sequence identifier and an optional description (like a FASTA title line).
- **Line 2:** DNA sequence.
- **Line 3:** Begins with a '+' character and is optionally followed by the same sequence identifier (and any description) again.
- **Line 4:** Encodes the quality values for the sequence in Line 2, and must contain the same number of symbols as letters in the sequence.

```
@SEQ_ID
GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
+
!''*((((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

Input Data Download

- Download PU.1 ChIP-seq experiment results – FASTQ file compressed as an SRA file.

Link: ftp://ftp-trace.ncbi.nlm.nih.gov/sra/sra-instant/reads/ByExp/sra/SRX%2FSRX540%2FSRX540701/SRR1283891/SRR1283891.sra

- Download mouse genome version mm9.

Link: http://hgdownload.soe.ucsc.edu/goldenPath/mm9/chromosomes/chr19.fa.gz

Short DNA Sequence Alignment

SRA Toolkit

- Set of tools to modify genomic data and perform file conversions.
- Example: fastq-dump to convert SRA to FASTQ.

- More Information:
  Information: http://www.ncbi.nlm.nih.gov/books/NBK158900/
  Website: http://www.ncbi.nlm.nih.gov/Traces/sra/sra.cgi?view=software

Convert SRA to FASTQ

- SRA is a compressed version of a FASTQ file.

- Use SRA toolkit to convert SRA to FASTQ.

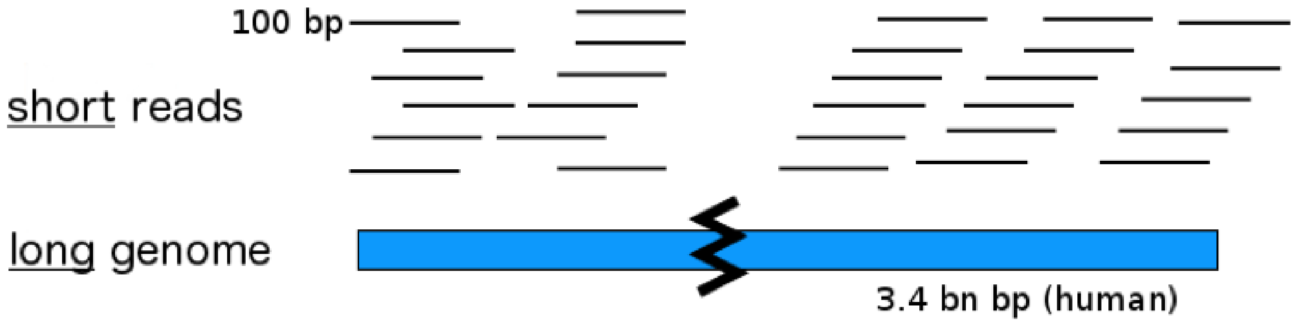
```
fastq-dump SRR1283891.sra
```

Alignment Problem

- A large reference sequence is given (genome)
  - up to billions of base pairs

- Query: short reads (DNA sequences with length < 200 bps)

- Find most probable position of the read in the genome (by inexact string matching)



Burrows-Wheeler Alignment Tool (BWA)

- Align reads to the genome:
  1. Prefix trie and string matching
  2. Burrows–Wheeler transform
  3. Suffix array interval and sequence alignment
  4. Exact matching: backward search
  5. Inexact matching: bounded traversal/backtracking

- More Information:

  Paper: http://bioinformatics.oxfordjournals.org/content/25/14/1754.long
  Website: http://bio-bwa.sourceforge.net/

Perform Alignment

- BWA Pipeline:

1. **index:** creates genome's index for fast look-up in transformation.

```
bwa index chr19.fa
```

2. **aln:** perform BWA alignment.

```
bwa aln chr19.fa SRR1283891.fastq > align.sai
```

3. **samse:** convert output to SAM file.

```
bwa samse chr19.fa align.sai SRR1283891.fastq > align.sam
```

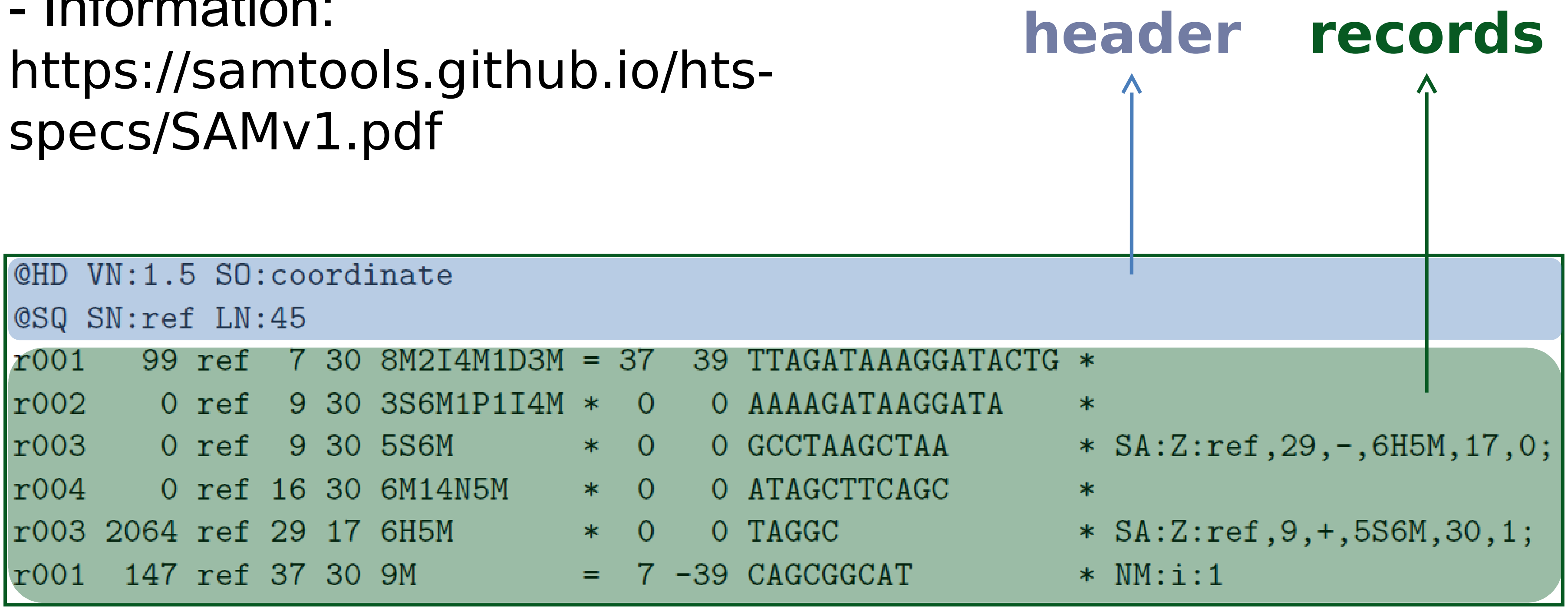
SAM File

- Sequence Alignment/Map format.

- Text-based tab-delimited file.

- Header + records (aligned reads)

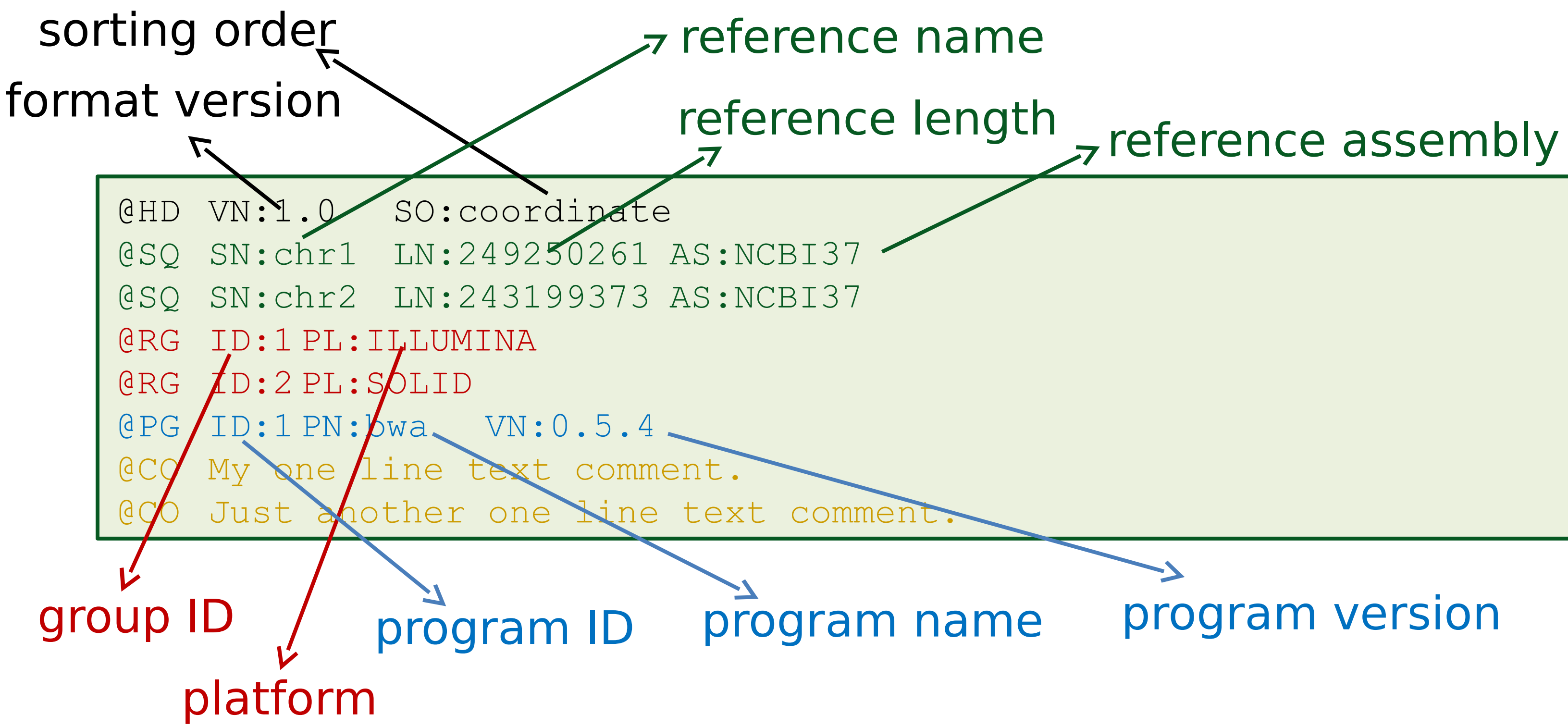
- Information: https://samtools.github.io/hts-specs/SAMv1.pdf



```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

SAM Header

- @HD – Header line.
- @SQ – Reference genome information.
- @RG – Read group information.
- @PG – Program (software) information.
- @CO – Commentary line.



sorting order

format version

reference name

reference length

reference assembly

```
@HD VN:1.0   SO:coordinate
@SQ SN:chr1  LN:249250261 AS:NCBI37
@SQ SN:chr2  LN:243199373 AS:NCBI37
@RG ID:1 PL:ILLUMINA
@RG ID:2 PL:SOLID
@PG ID:1 PN:bwa  VN:0.5.4
@CO My one line text comment.
@CO Just another one line text comment.
```

group ID

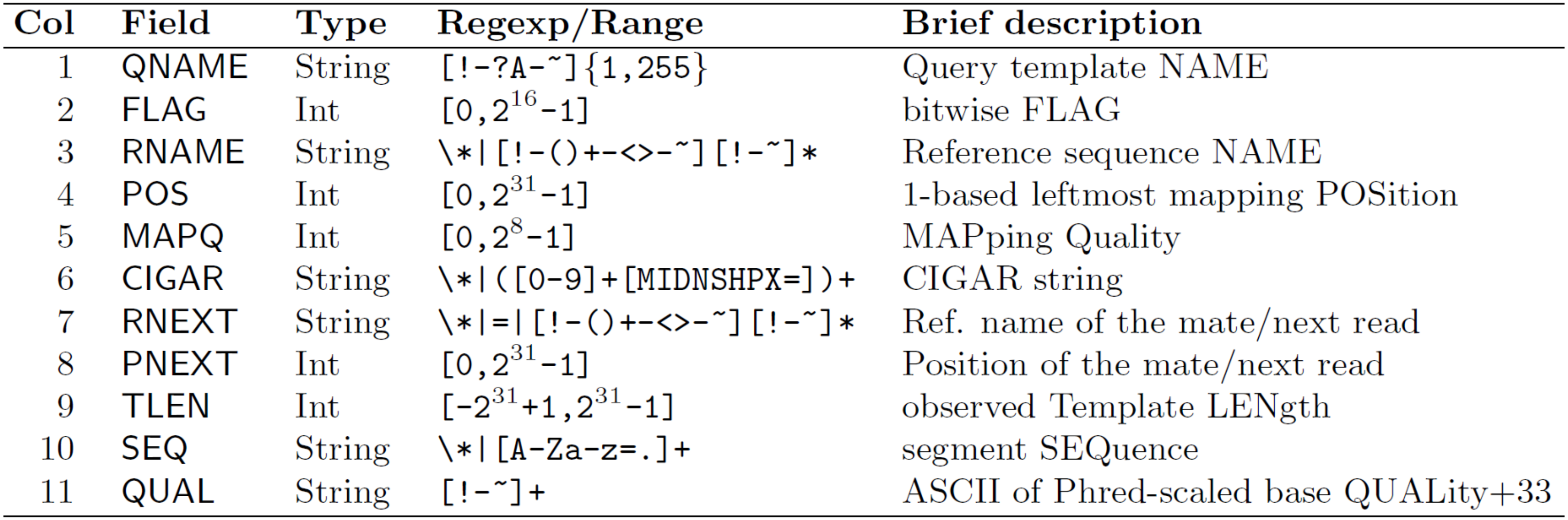
program ID

program name

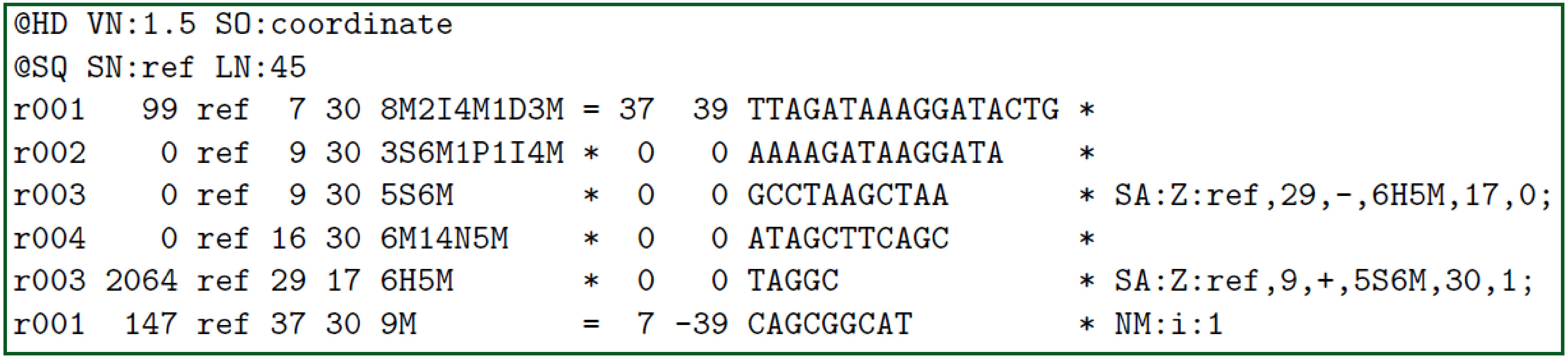
program version

platform

SAM Records



Col	Field	Type	Regexp/Range	Brief description
1	QNAME	String	`[!-?A-~]{1,255}`	Query template NAME
2	FLAG	Int	$[0,2^{16}-1]$	bitwise FLAG
3	RNAME	String	`\*|[!-()+-<>-~][!-~]*`	Reference sequence NAME
4	POS	Int	$[0,2^{31}-1]$	1-based leftmost mapping POSition
5	MAPQ	Int	$[0,2^{8}-1]$	MAPping Quality
6	CIGAR	String	`\*|([0-9]+[MIDNSHPX=])+`	CIGAR string
7	RNEXT	String	`\*|=|[!-()+-<>-~][!-~]*`	Ref. name of the mate/next read
8	PNEXT	Int	$[0,2^{31}-1]$	Position of the mate/next read
9	TLEN	Int	$[-2^{31}+1,2^{31}-1]$	observed Template LENgth
10	SEQ	String	`\*|[A-Za-z=.]+`	segment SEQuence
11	QUAL	String	`[!-~]+`	ASCII of Phred-scaled base QUALity+33



```
@HD VN:1.5 SO:coordinate
@SQ SN:ref LN:45
r001   99 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTG *
r002    0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
r003    0 ref  9 30 5S6M       *  0   0 GCCTAAGCTAA       * SA:Z:ref,29,-,6H5M,17,0;
r004    0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
r003 2064 ref 29 17 6H5M       *  0   0 TAGGC             * SA:Z:ref,9,+,5S6M,30,1;
r001  147 ref 37 30 9M         =  7 -39 CAGCGGCAT         * NM:i:1
```

BAM File

- Binary Alignment/Map format – compressed version of SAM.

- Compression: BGZF block compression.

- Efficient random access: UCSC bin/chunk scheme.

- BAI index files.

- More Information:

  http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2723002/
  http://www.ncbi.nlm.nih.gov/pmc/articles/PMC186604/

Samtools

- Provides various utilities for manipulating alignments in the SAM format.

- Tools useful for quality check and bias correction.

- More Information:

  Paper: http://www.ncbi.nlm.nih.gov/pubmed/19505943
  Website: http://samtools.sourceforge.net/

Convert SAM to BAM

- Using samtools:

1. **view:** shows binary format.

```
samtools view -bS align.sam > align.bam
```

2. **sort:** sorts alignment by coordinates.

```
samtools sort align.bam align.sorted
```

3. **index:** creates alignment's index fast random access.

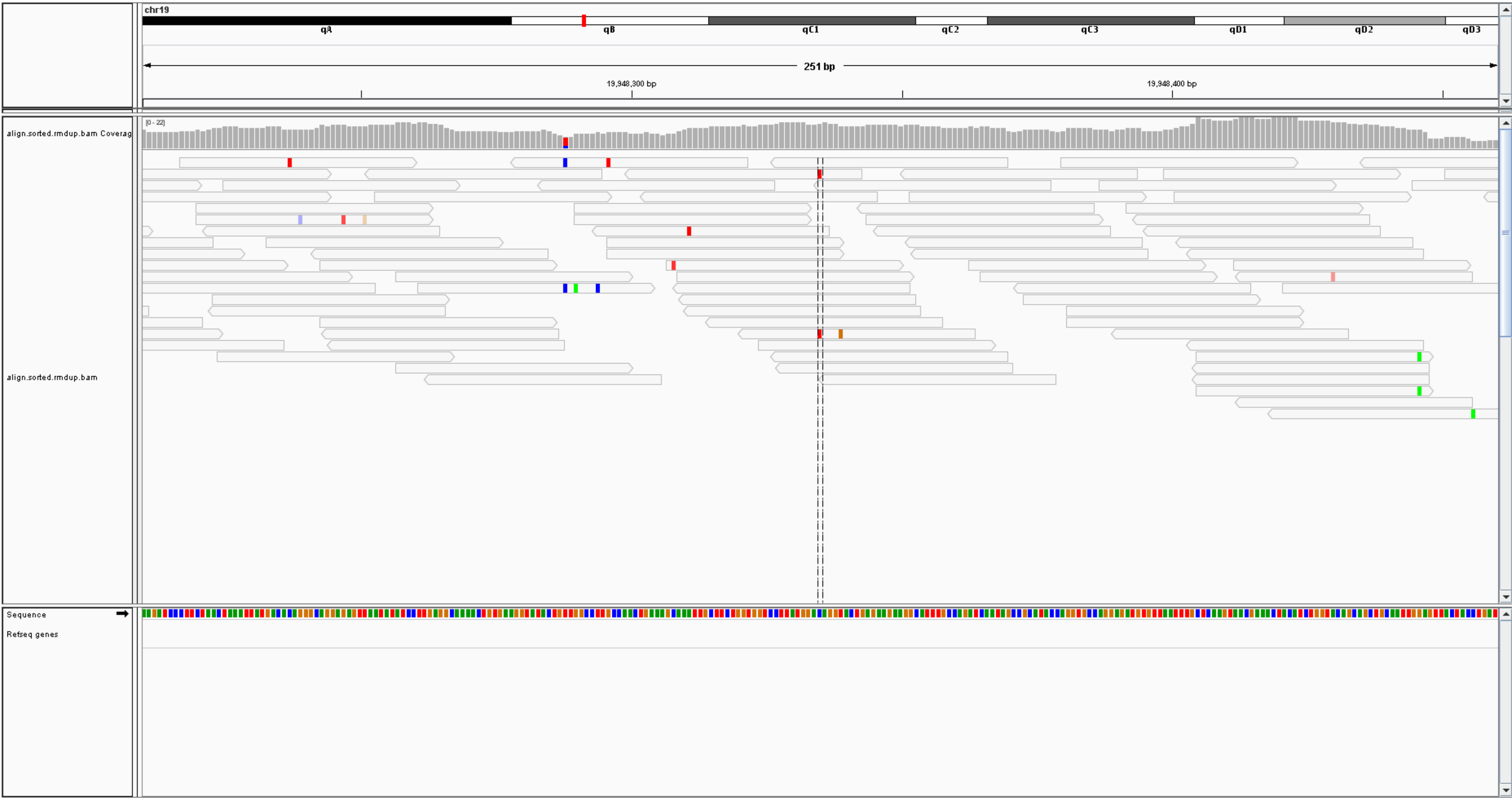
```
samtools index align.sorted.bam
```

Example of quality check

- Removing duplicate reads:

```
samtools rmdup align.sorted.bam align.sorted.rmdup.bam
samtools index align.sorted.rmdup.bam
```

Example of quality check

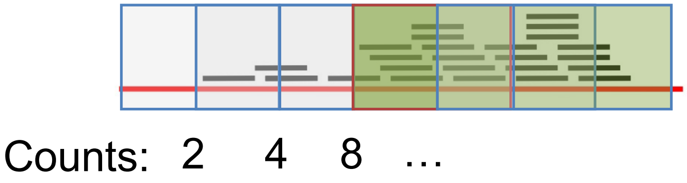


Peak Calling

Peak Calling

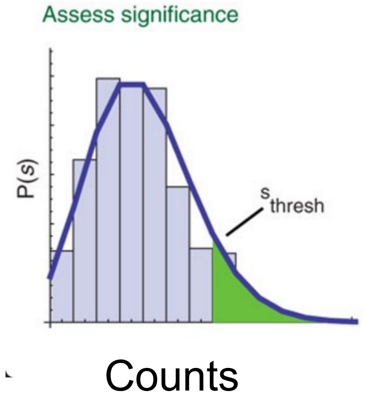
Problem definition: Find genomic regions (of arbitrary size) with more aligned reads than expected by chance.

Aligned Reads



Example of a simple peak caller :

1. Use a fix window to scan through the genome and obtain a distribution of counts per bin.
2. Define a statistical test to evaluate if the number of reads in higher than expected by change.



Peak Calling

Problems:

- Which window size/offset to use?
-
- Distinct proteins have distinct peak sizes.

- Proper quantification of read counts require several further steps:
  - Fragment size estimation.
  - CG bias correction.
  - Mappability.

MACS Peak Caller

- Model-based Analysis for ChIP-seq.

- Two important steps:

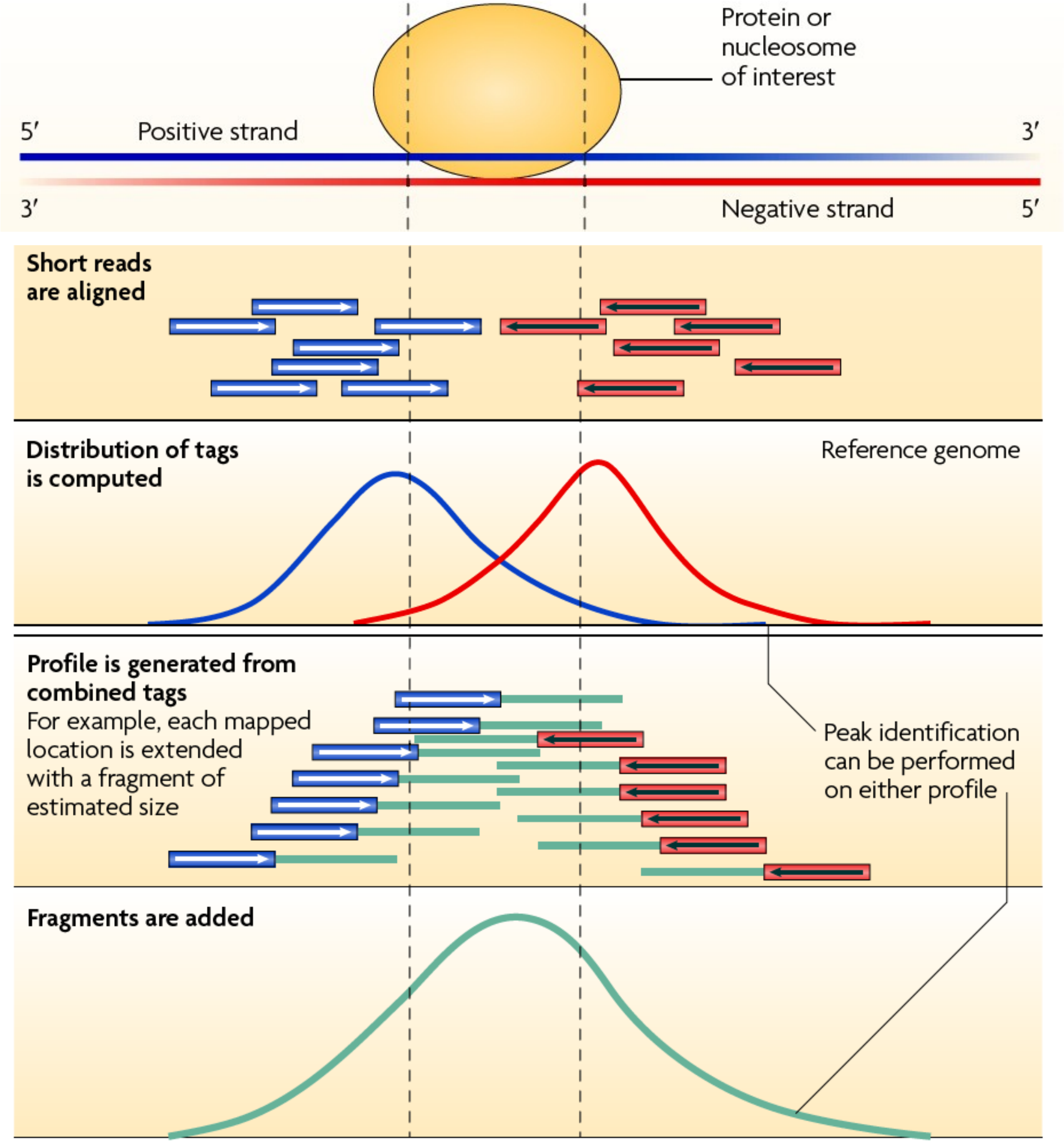
1. MACS empirically models the shift size of ChIP-seq reads, and uses it to improve the spatial resolution of inferred TF binding sites.

2. MACS estimates a dynamic background reads distribution to effectively capture local biases in the genome, allowing for more robust identifications.

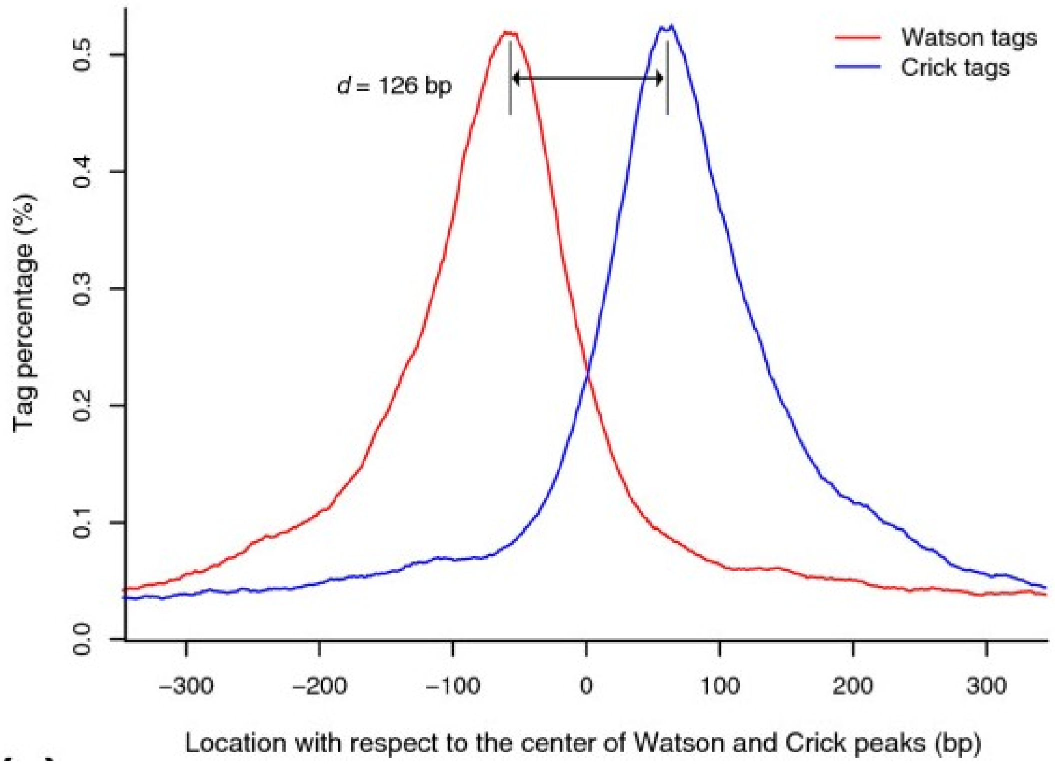
- More Information:

Paper: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2592715/
Website: http://liulab.dfci.harvard.edu/MACS/

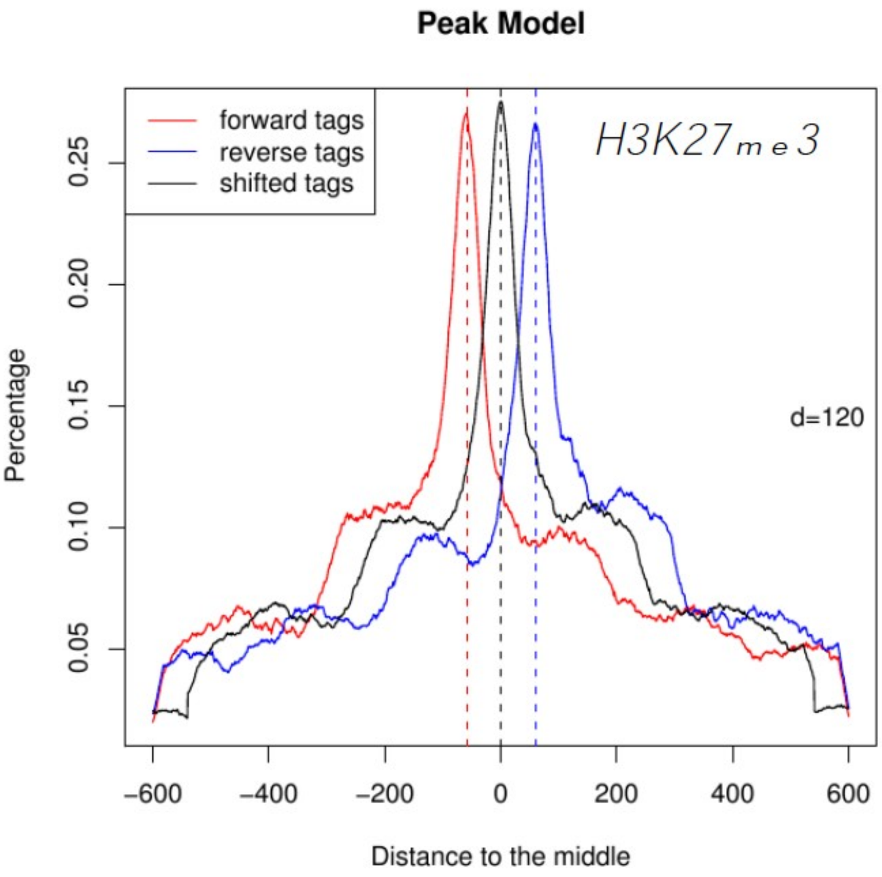
MACS Peak Caller

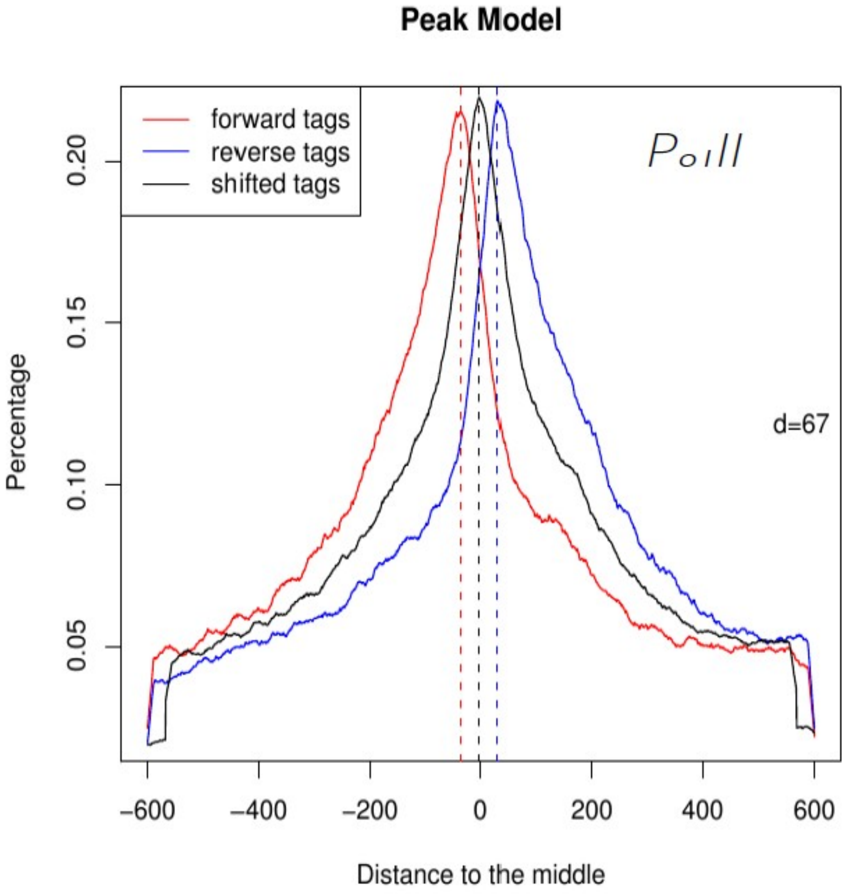

Protein or nucleosome of interest
5′
Positive strand
3′
3′
Negative strand
5′
Short reads are aligned
Distribution of tags is computed
Reference genome
Profile is generated from combined tags
For example, each mapped location is extended with a fragment of estimated size
Peak identification can be performed on either profile
Fragments are added

MACS Peak Caller



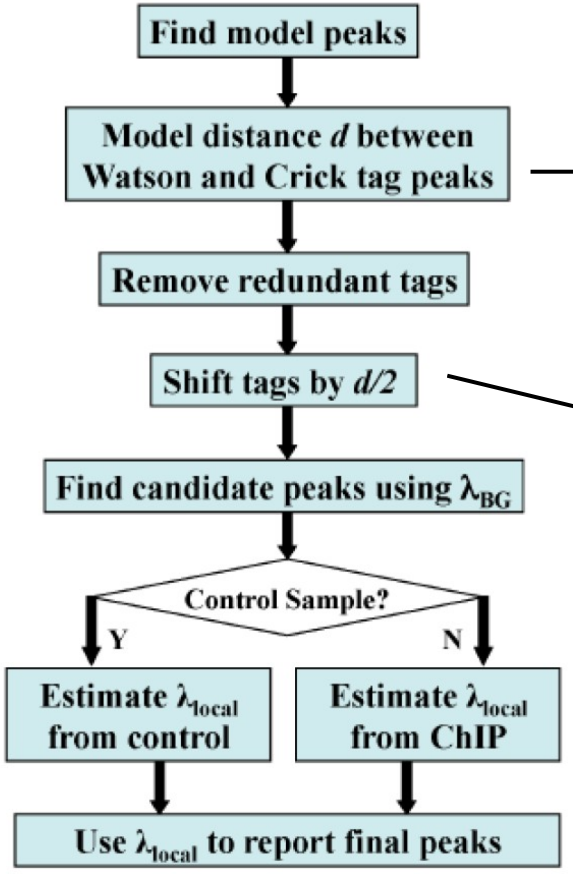
MACS Peak Caller

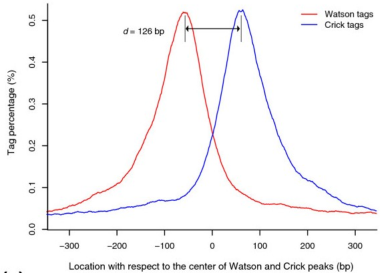


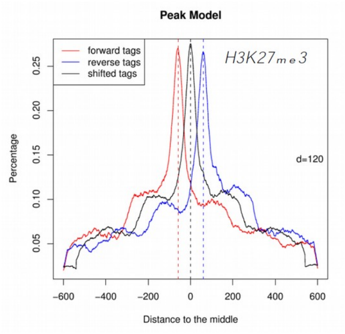


MACS Peak Caller

- Model the reads using a Poisson distribution
- Advantage: only one parameter (λ) which models both mean and variance.
- Peaks are defined given a p-value on the Poisson model


Find model peaks
Model distance d between Watson and Crick tag peaks
Remove redundant tags
Shift tags by $d/2$
Find candidate peaks using λ_{BG}
Control Sample?
Y
N
Estimate λ_{local} from control
Estimate λ_{local} from ChIP
Use λ_{local} to report final peaks

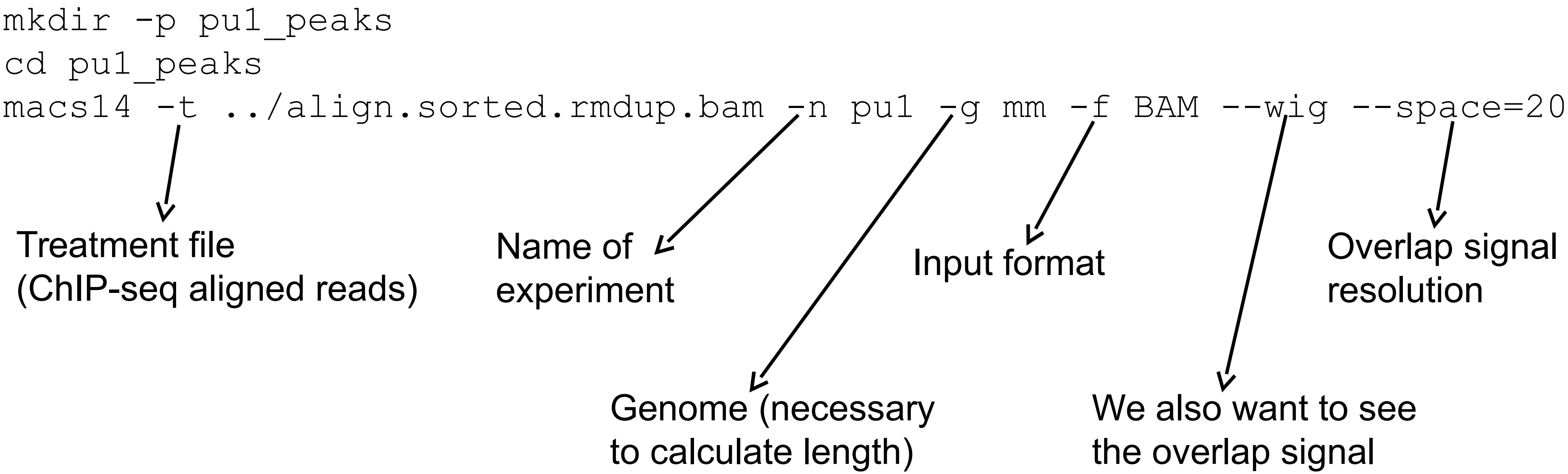




$$\lambda_{local} = \max(\lambda_{BG}, [\lambda_{1k},] \lambda_{5k}, \lambda_{10k})$$

Calling Peaks

- Calling peaks using MACS:



```
mkdir -p pu1_peaks
cd pu1_peaks
macs14 -t ../align.sorted.rmdup.bam -n pu1 -g mm -f BAM --wig --space=20
```

Treatment file
(ChIP-seq aligned reads)

Name of
experiment

Input format

Overlap signal
resolution

Genome (necessary
to calculate length)

We also want to see
the overlap signal

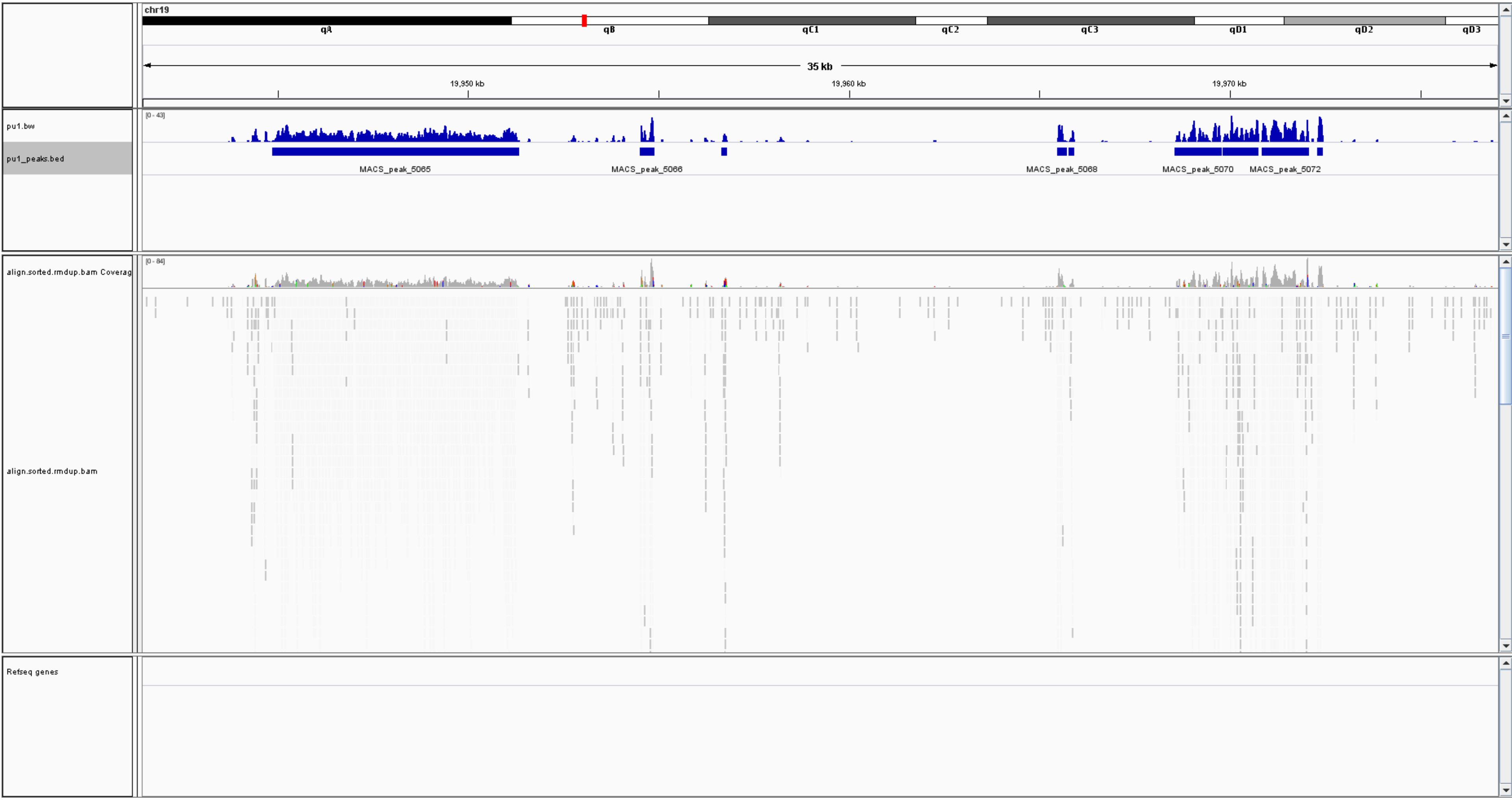
Visualization: The ChIP-seq signal

- MACS generates the signal as a wiggle (wig) file.

- Converting signal in wig format to compressed bigwig (bw) format to visualize it:

```
cd pu1_MACS_wiggle/treat/
gunzip pu1_treat_afterfiting_chr19.wig.gz
fetchChromSizes mm9 > mm9.chrom.sizes
wigToBigWig pu1_treat_afterfiting_chr19.wig mm9.chrom.sizes pu1.bw
```

Visualization: PU.1 Peaks and Signal



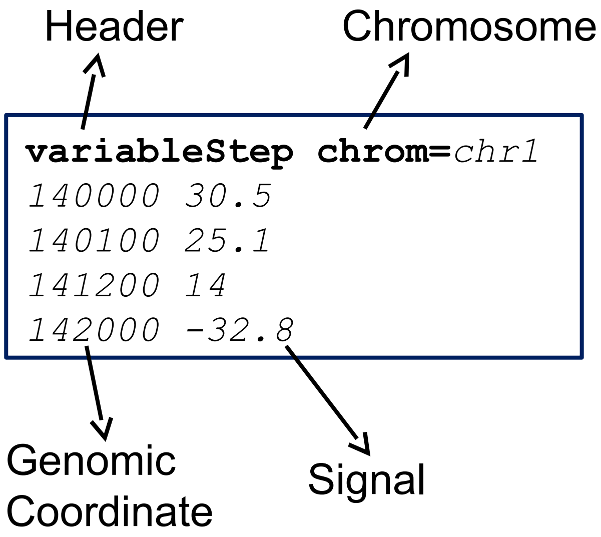
BED File: Storing genomic regions

- WIG: Text-based tab-delimited file to store genomic signals.
- Fields:
  1. **chrom**: The name of the chromosome.
  2. **chromStart**: The starting position of the coordinate (start = 0).
  3. **chromEnd**: The ending position of the coordinate (outside interval).
  4. **name**: Label of the coordinate.
  5. **score**: A score between 0 and 1000.
  6. **strand**: Either ‘+’ or ‘-’.
- Example

```
chr1   140000   140100   read1   160   +
chr1   140200   140300   read2   200   -
chr1   140400   140500   read3   250   +
chr1   141000   141100   read4   400   -
```

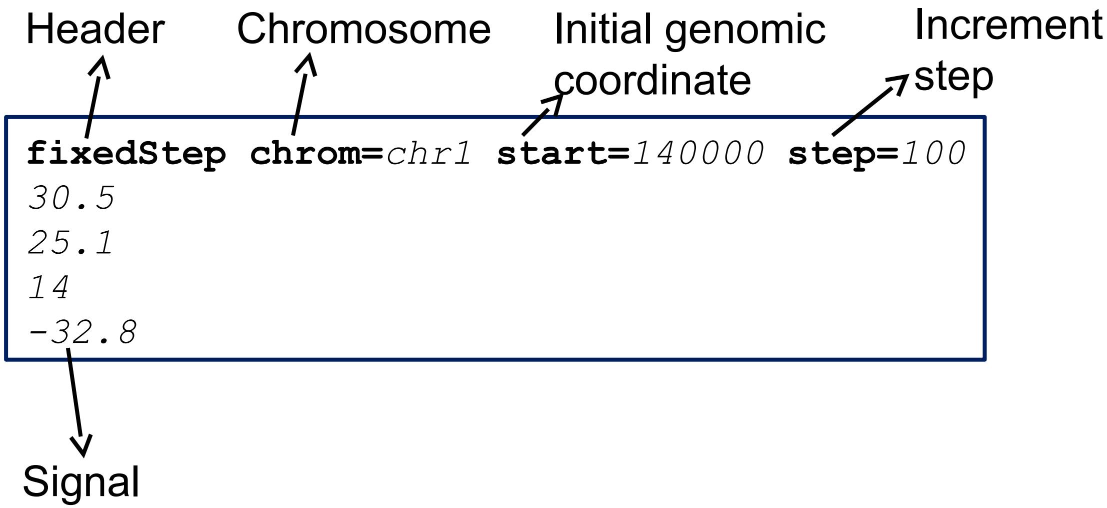
WIG & BIGWIG Files: Storing genomic signal

- WIG: Text-based tab-delimited file to store genomic signals.
- Variable Step



```
variableStep chrom=chr1
140000 30.5
140100 25.1
141200 14
142000 -32.8
```

- Fixed Step



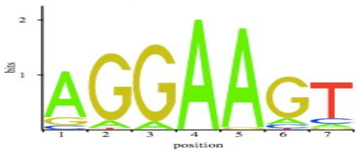
```
fixedStep chrom=chr1 start=140000 step=100
30.5
25.1
14
-32.8
```

- BIGWIG: Binary compression of WIG file
- Similar to SAM/BAM

DNA Motif Analysis

Motif Search

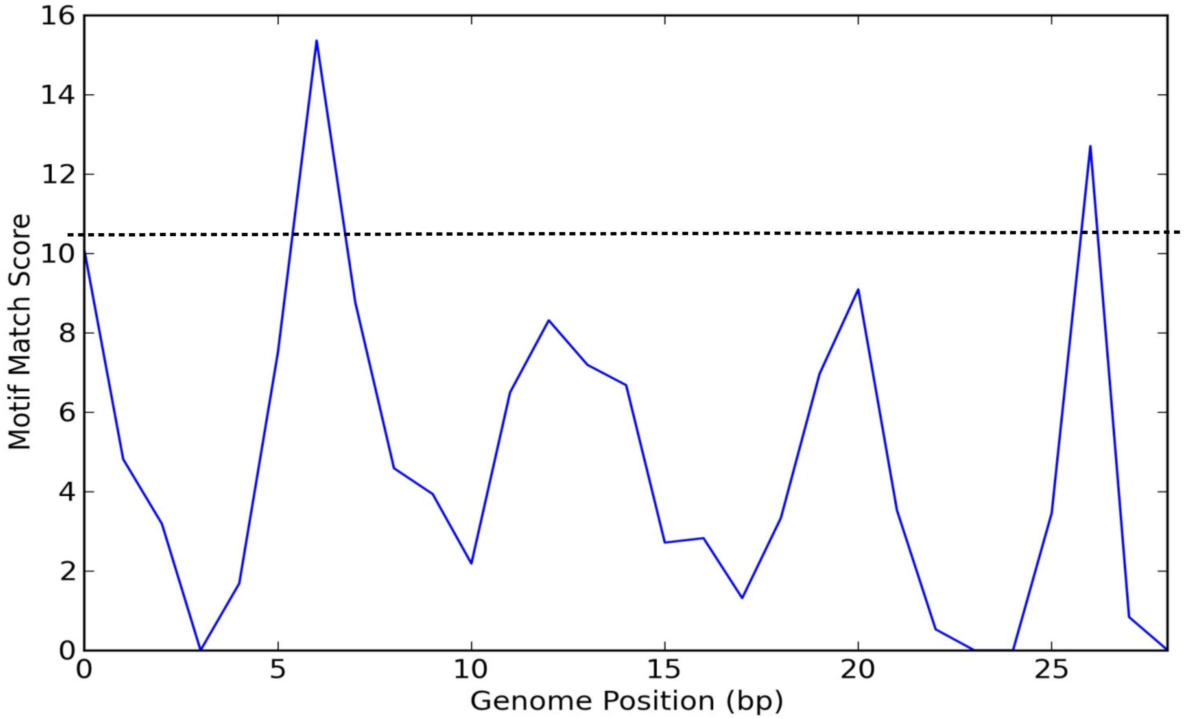
PU.1 PWM



Genome TATCTTTGGAAGTGAAACTACTATCCTGAAAGTCGAA

Score 10.06 3.19 …

4.81



Statistical Test

FDR = 1 x 10^{-4}

FDR- False Discovery Rate

Motif Analysis

- Fetching a subset of high-quality peaks

```
awk -v threshold="1000" '$5 > threshold' pu1_peaks.bed > pu1_peaks_1000.bed
```

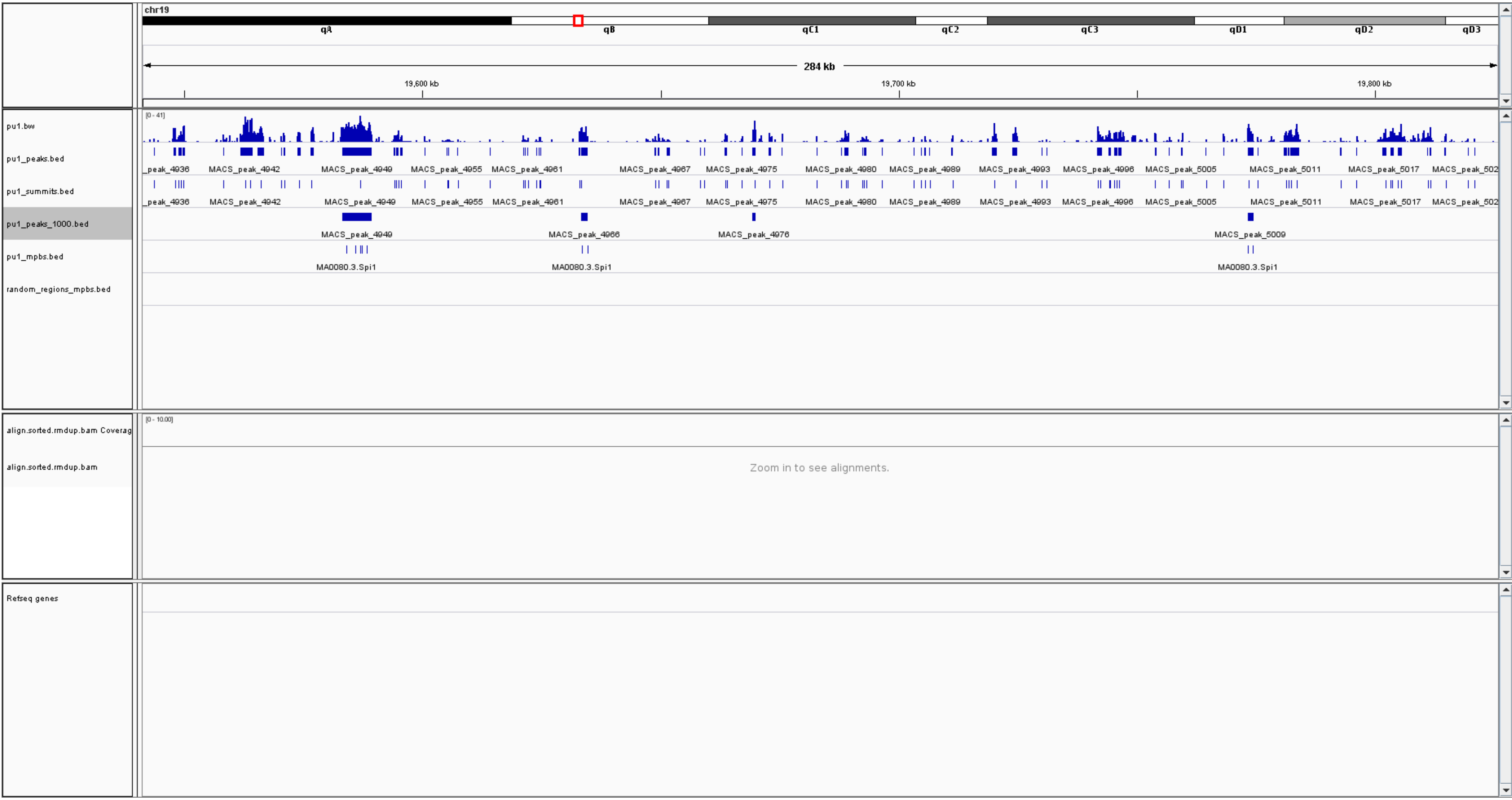
- Fix to search for PU.1 motifs only

```
cd ~/rgtdata
cp -r ./motifs/jaspar_vertebrates ~/rgtdata/motifs/pu1
cp ./motifs/jaspar_vertebrates.fpr ~/rgtdata/motifs/pu1.fpr
find ./motifs/pu1 ! -name 'MA0080.3.Spi1.pwm' -type f -exec rm -f {} +
python setupGenomicData.py --mm9
```

- Searching PU.1 binding sites within PU.1 peaks using RGT

```
rgt-motifanalysis --matching ./motifmatch.txt
```

Motif Analysis



Zoom in to see alignments.