ISTITUTO ITALIANO DI TECNOLOGIA
CENTER FOR GENOMIC SCIENCE

Campus IFOM—IEO

Computational challenges in the analysis of high-throughput (epi)genomics sequencing data

Mattia Pelizzola - Center for Genomic Science of IIT@SEMM

Outline of the presentation

- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

Outline of the presentation

- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

Sequencing platforms

	454 Ti RocheT	Illumina HiSeqTM 2000	ABI 5500 (SOLiD)
Amplification	Emulsion PCR	Bridge PCR	Emulsion PCR
Sequencing reaction	Pyrosequencing	Reversible terminators	Ligation-based sequencing
Paired ends/sep	Yes/3kb	Yes/200 bp	Yes/3 kb
Read length	400 bp	100 bp	75 bp
Advantages	Short run times. Longer reads improve mapping in repetitive regions. Ability to detect large structural variations	The most popular platform	Good base call accuracy. Good multiplexing capability
Disadvantages	High reagent cost. Higher error rates in repeat sequences		

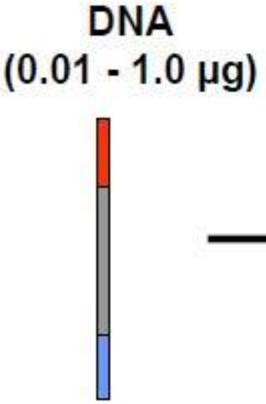
Illumina sequencing

	GAIIx - V4 kits, v1.6 Pipeline	GAIIx - 95Gb Configuration	HiSeq2000
Average Clusters/ GAIIx tile	300,000	387,000	265,000
Data Rate (Gb/day)	5	7	31
Read Length	100bp	150bp	100bp
Error Rate	1.50%	1.40%	0.48%
Yield per run (Gb)	**51**	**97.8**	**248**

Illumina sequencing

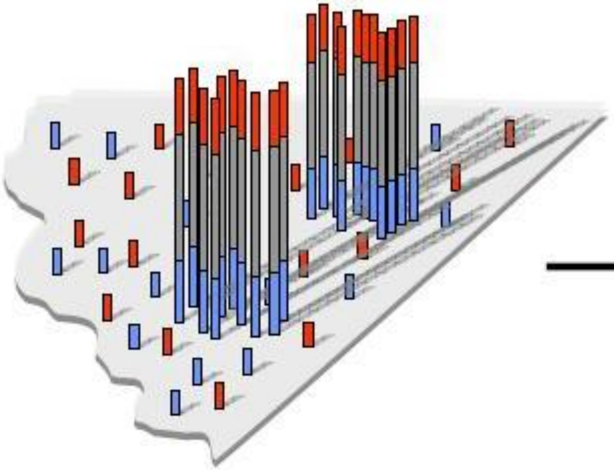
Single base extension with incorporation of fluorescently labeled nucleotides

Library preparation

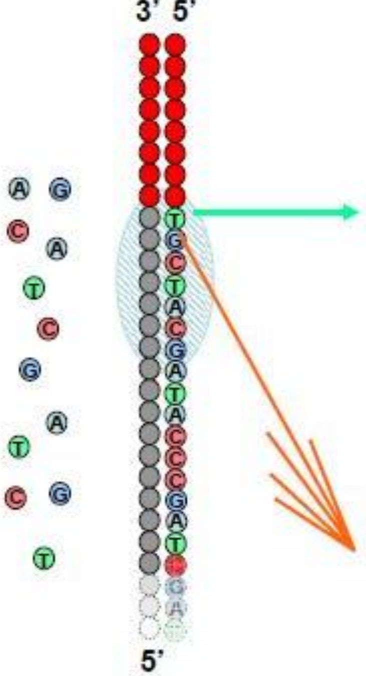


DNA fragmentation and adapter ligation

Automated cluster generation



Attachment to the flow cell
Cluster generation by bridge amplification of DNA fragments



Sequencing

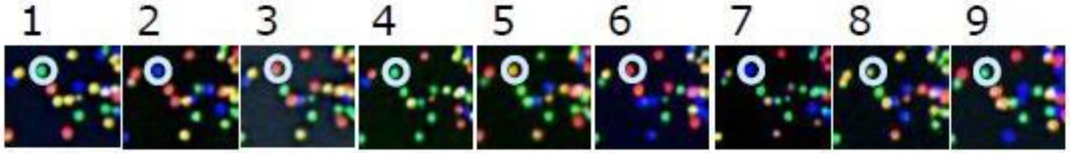
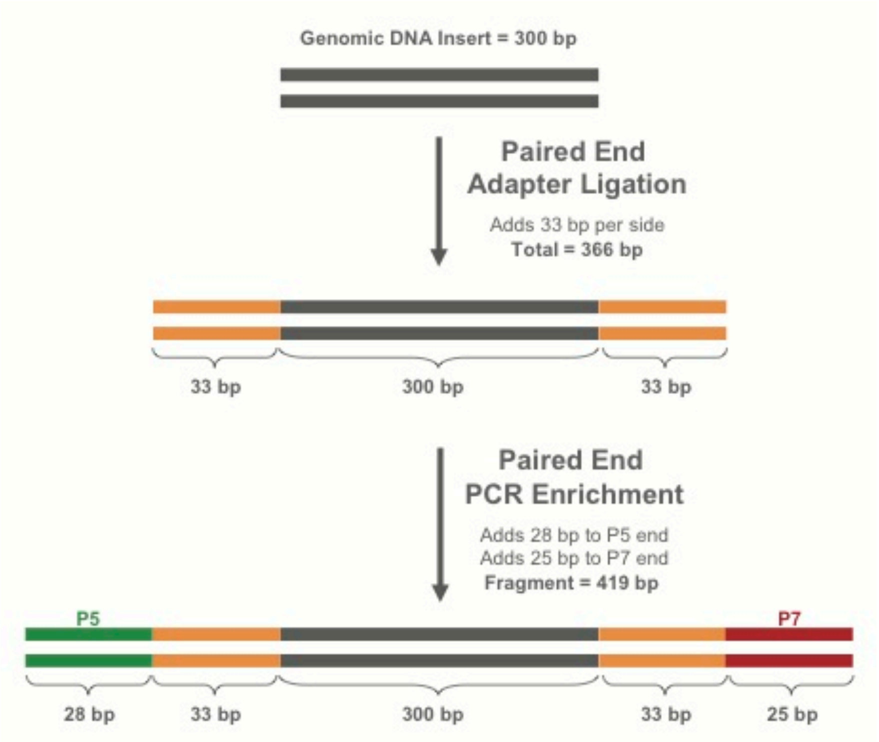


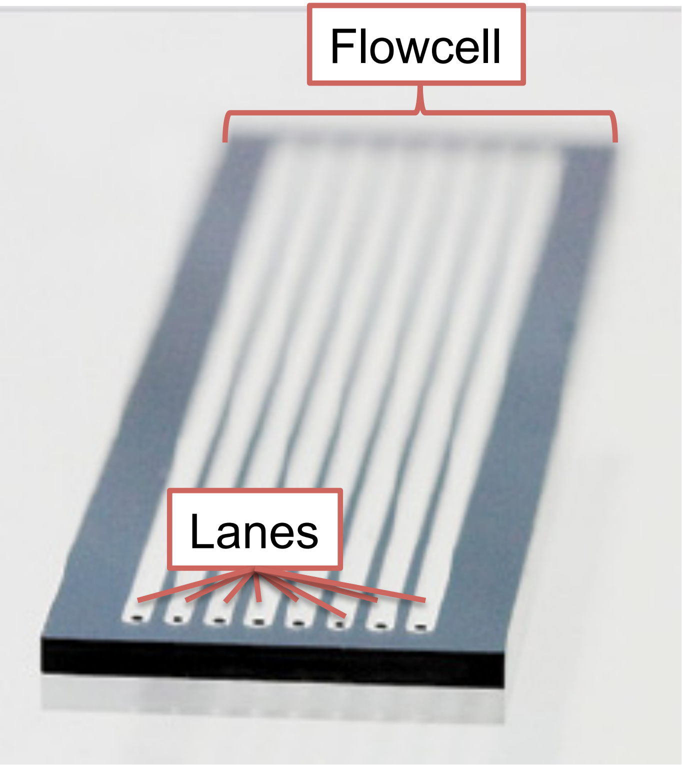
Image acquisition

Base calling

Illumina terminology: libraries, lanes and flow cells



Each reaction produces a unique **library** of DNA fragments for sequencing.

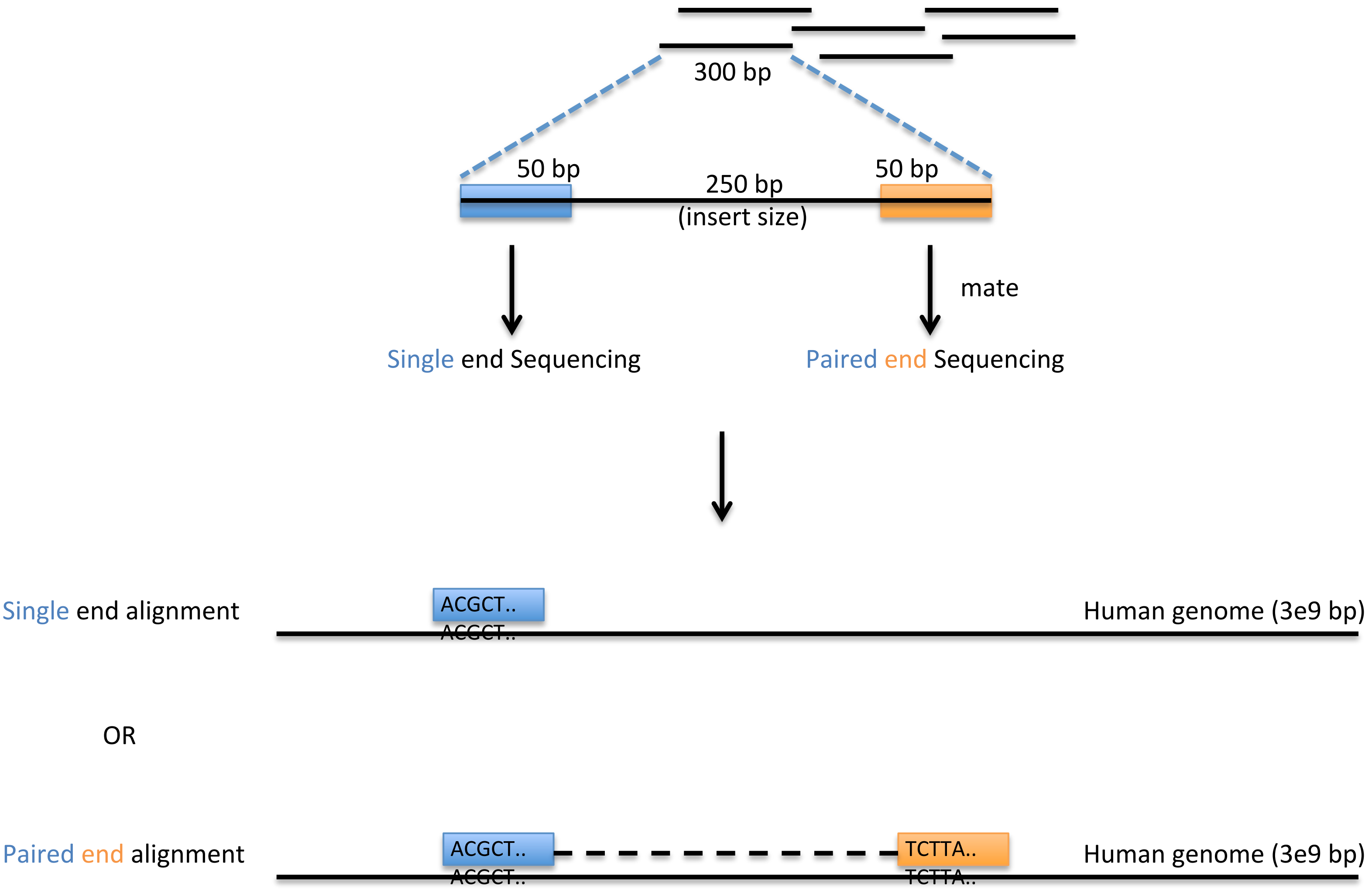


Each NGS machine processes a single **flowcell** containing several independent **lanes** during a single sequencing run

Outline of the presentation

- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

Single end vs paired end sequencing



Adapted from Park P, Nature Review Genetics 2009

Computational workflows in NGS datasets

1. Raw data analysis= image processing and base calling (reads)
2. Storing reads in FASTQ files
3. Quality controls
4. Reads filtering
5. Alignment to the reference genome
6. Storing aligned reads (alignments) in SAM/BAM files
7. Manipulating SAM/BAM files
8. Playing with alignments and genomic intervals

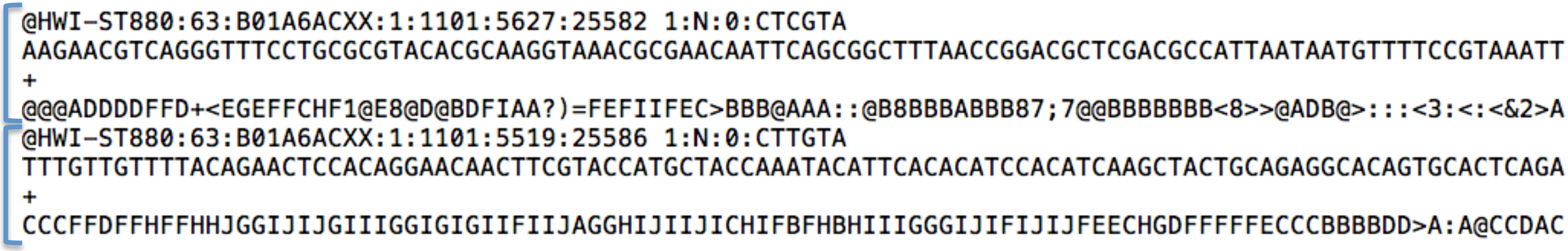
Outline of the presentation

- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

Sequencing reads bases and quality calls: FASTQ file format

FASTQ format is a text-based format for storing a biological sequence and its corresponding quality scores. It has become the standard for storing the output of high throughput sequencing instruments.

EXAMPLE:



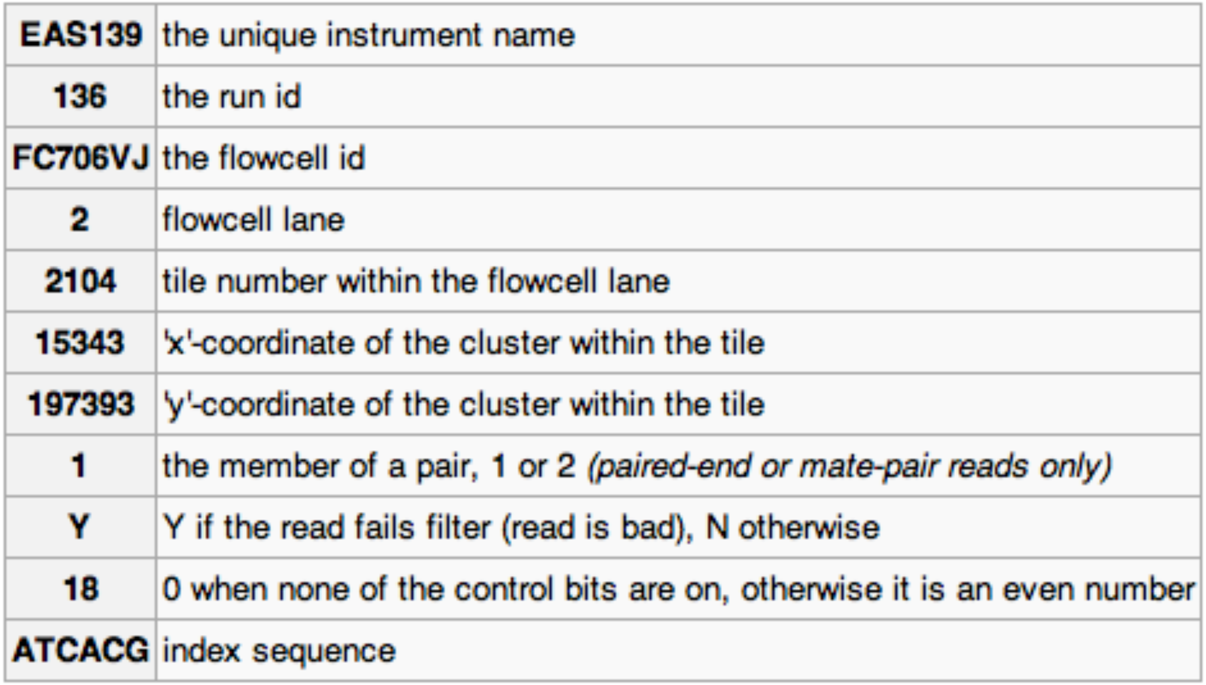
```
@HWI-ST880:63:B01A6ACXX:1:1101:5627:25582 1:N:0:CTCGTA
AAGAACGTCAGGGTTTCCTGCGCGTACACGCAAGGTAAACGCGAACAATTCAGCGGCTTTAACCGGACGCTCGACGCCATTAATAATGTTTTCCGTAAATT
+
@@@ADDDDFFD+<EGEFFCHF1@E8@D@BDFIAA?)=FEFIIFEC>BBB@AAA::@B8BBBABBB87;7@@BBBBBBB<8>>@ADB@>:::<3:<:<&2>A
@HWI-ST880:63:B01A6ACXX:1:1101:5519:25586 1:N:0:CTTGTA
TTTGTTGTTTTACAGAACTCCACAGGAACAACTTCGTACCATGCTACCAAATACATTCACACATCCACATCAAGCTACTGCAGAGGCACAGTGCACTCAGA
+
CCCFFDFFHFFHHJGGIJIJGIIIGGIGIGIIFIIJAGGHIJIIJICHIFBFHBHIIIGGGIJIFIJIJFEECHGDFFFFFECCCBBBBDD>A:A@CCDAC
```

1. begins with a '@' character and is followed by a sequence identifier
2. the raw sequence letters.
3. begins with a '+' character and is *optionally* followed by the same sequence identifier
4. encodes the quality values for the sequence in and must contain the same number of symbols as letters in the sequence.

FASTQ files: sequence IDs

```
@EAS139:136:FC706VJ:2:5:1000:12850 1:Y:18:ATCACG
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
BBBBCCCC?<A?BC?7@@???????DBBA@@@@A@@
```

Sequences ID @EAS139:136:FC706VJ:2:5:1000:12850 1:Y:18:ATCACG means:

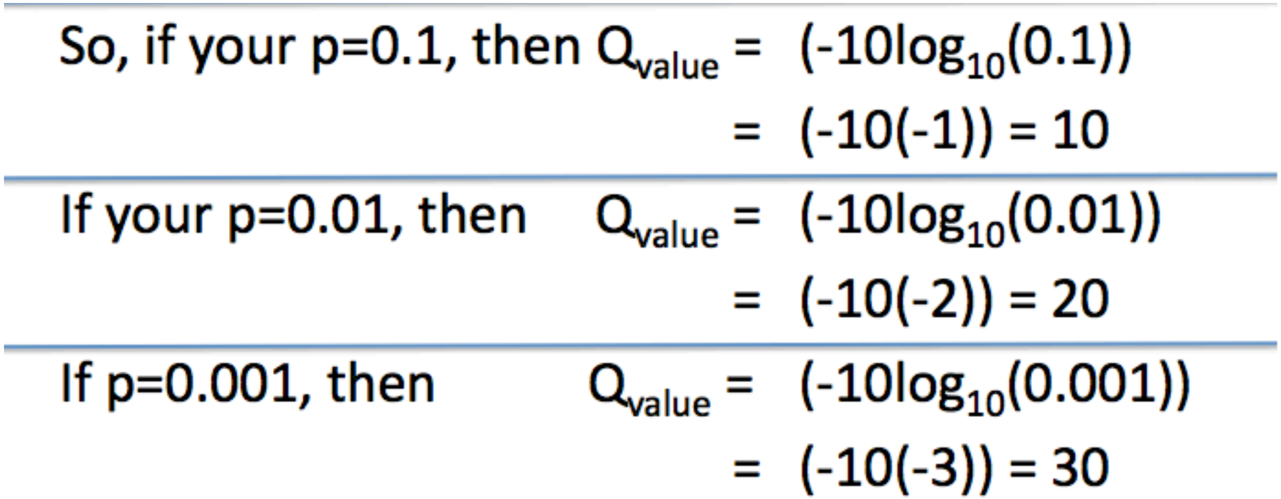


EAS139	the unique instrument name
136	the run id
FC706VJ	the flowcell id
2	flowcell lane
2104	tile number within the flowcell lane
15343	'x'-coordinate of the cluster within the tile
197393	'y'-coordinate of the cluster within the tile
1	the member of a pair, 1 or 2 *(paired-end or mate-pair reads only)*
Y	Y if the read fails filter (read is bad), N otherwise
18	0 when none of the control bits are on, otherwise it is an even number
ATCACG	index sequence

FASTQ files: quality scores

A FASTQ file contains quality information.
Phred quality scores *Q are defined as a property which is logarithmically related to the base-calling error probabilities P*
Where P is the probability that the corresponding base call is incorrect.

$$Q = -10\ \log_{10} P$$



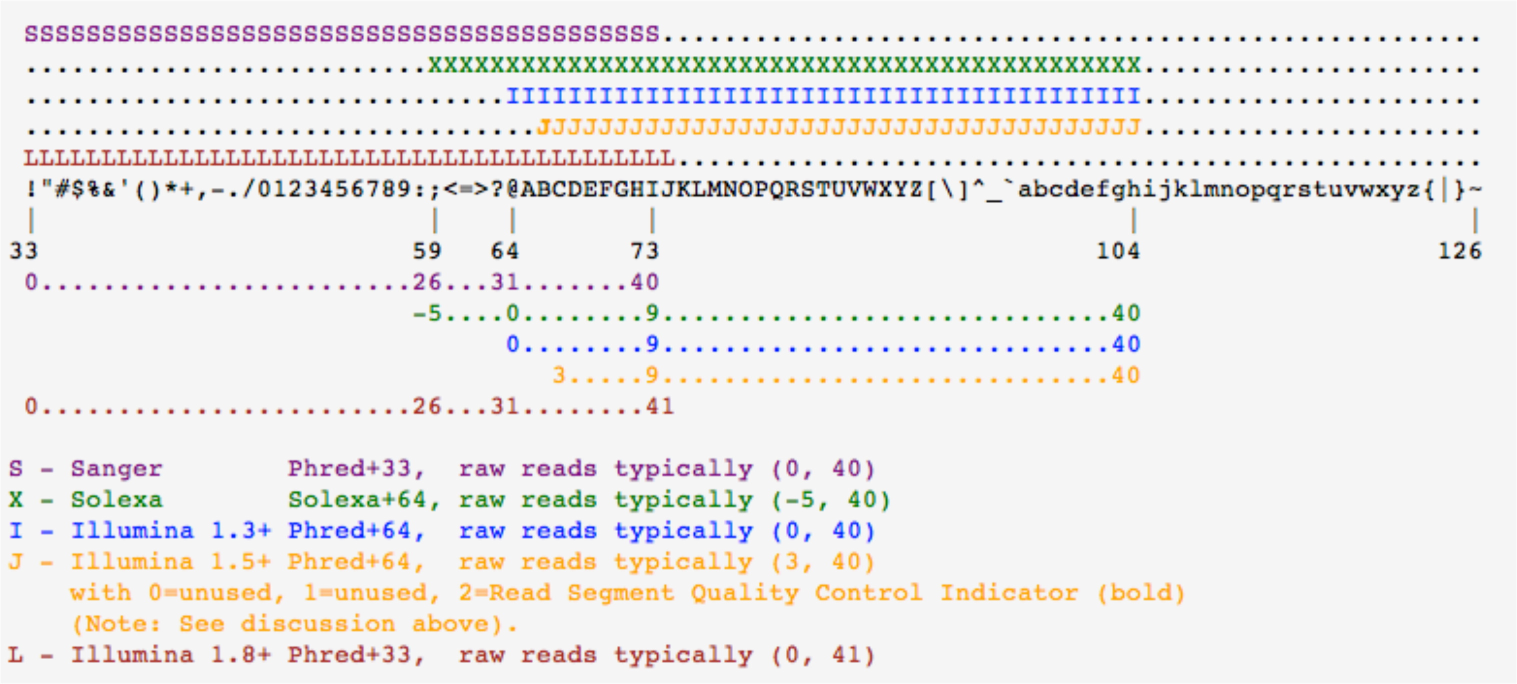
So, if your p=0.1, then $Q_{value} = (-10\log_{10}(0.1))$
$= (-10(-1)) = 10$

If your p=0.01, then $Q_{value} = (-10\log_{10}(0.01))$
$= (-10(-2)) = 20$

If p=0.001, then $Q_{value} = (-10\log_{10}(0.001))$
$= (-10(-3)) = 30$

FASTQ files: quality scores

Phred quality scores *Q are* represented with a single bit in ASCII format.
ASCII stands for American Standard Code for Information Interchange.
ASCII code is the numerical representation of a character such as 'a' or '@'
The first 32 symbols in ASCII are control characters, so we start at 33.



```
 SSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSSS.....................................................
 ..........................XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX......................
 ...............................IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII......................
 .................................JJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJJ......................
 LLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLLL....................................................
 !"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
 |                         |    |        |                              |                     |
33                        59   64       73                            104                   126
 0........................26...31.......40
                          -5....0........9.............................40
                                0........9.............................40
                                   3.....9.............................40
 0........................26...31........41

S - Sanger        Phred+33,  raw reads typically (0, 40)
X - Solexa        Solexa+64, raw reads typically (-5, 40)
I - Illumina 1.3+ Phred+64,  raw reads typically (0, 40)
J - Illumina 1.5+ Phred+64,  raw reads typically (3, 40)
    with 0=unused, 1=unused, 2=Read Segment Quality Control Indicator (bold)
    (Note: See discussion above).
L - Illumina 1.8+ Phred+33,  raw reads typically (0, 41)
```

If your ASCII character is ‘B’ and they are in Sanger format, then 66-33=33, so
$33= (-10\log_{10}p)$
$-3.3=\log_{10}p$
$10^{-3.3}$=p, so p= 0.0005 or 0.05% chance of an incorrect base.

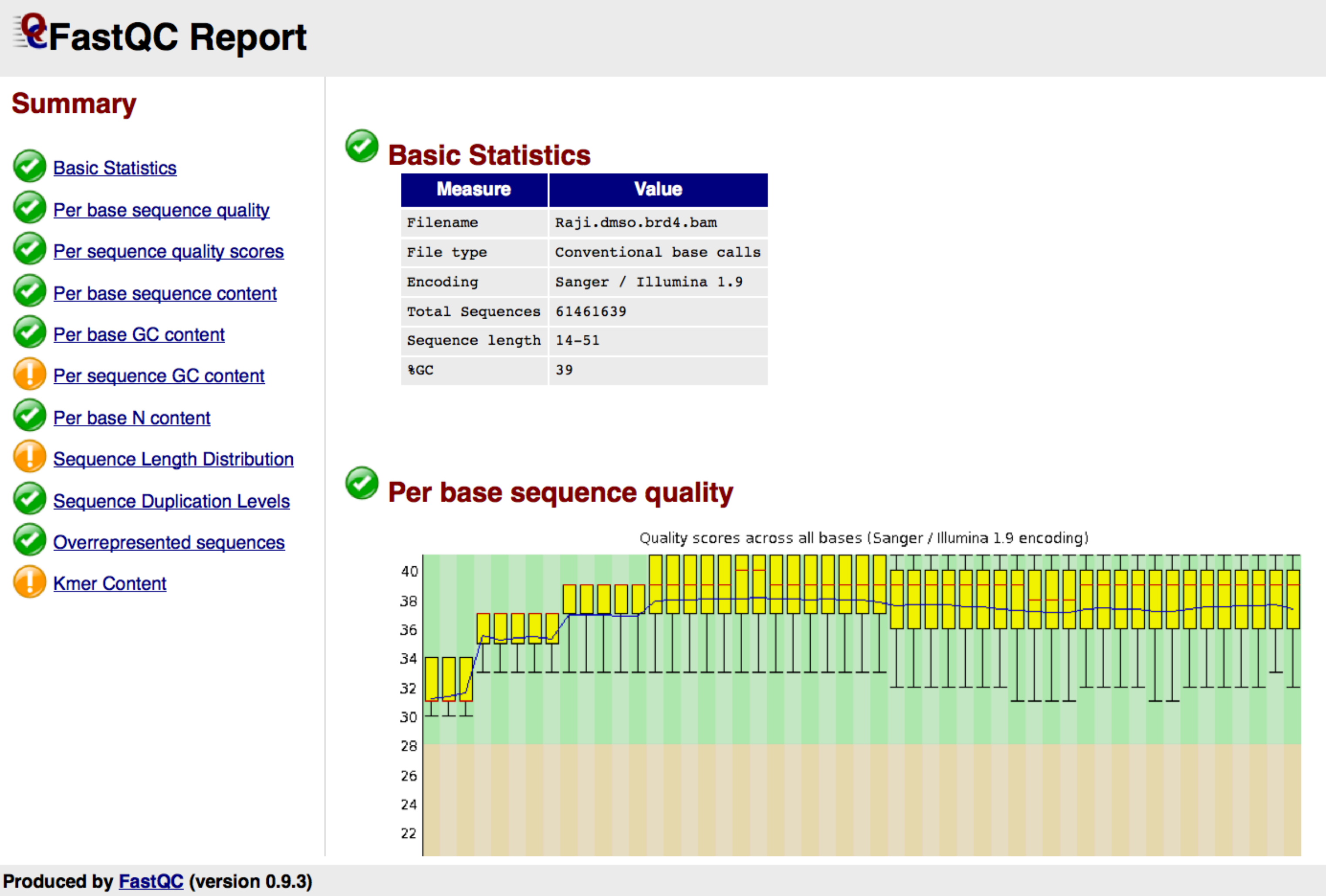
Filtering sequencing reads based on quality scores

- Before alignment there is sometimes the need to preprocess/manipulate the FASTA/FASTQ files to produce better mapping results. It is important to do quality control checks to understand whether your data has any problems of which you should be aware before doing any further analysis

- FastQC: quality control checks on raw sequence data coming from high throughput sequencing pipelines (http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/).

- The FASTX-Toolkit tools perform some of these preprocessing tasks (http://hannonlab.cshl.edu/fastx_toolkit/).
  Two of many useful tools are:
  - FASTQ Quality Filter → Filters sequences based on quality
  - FASTQ Quality Trimmer → Trims (cuts) sequences based on quality

Quality checks: FastqQC reports



FastQC Report

Summary

- ✓ Basic Statistics
- ✓ Per base sequence quality
- ✓ Per sequence quality scores
- ✓ Per base sequence content
- ✓ Per base GC content
- ! Per sequence GC content
- ✓ Per base N content
- ! Sequence Length Distribution
- ✓ Sequence Duplication Levels
- ✓ Overrepresented sequences
- ! Kmer Content

✓ Basic Statistics

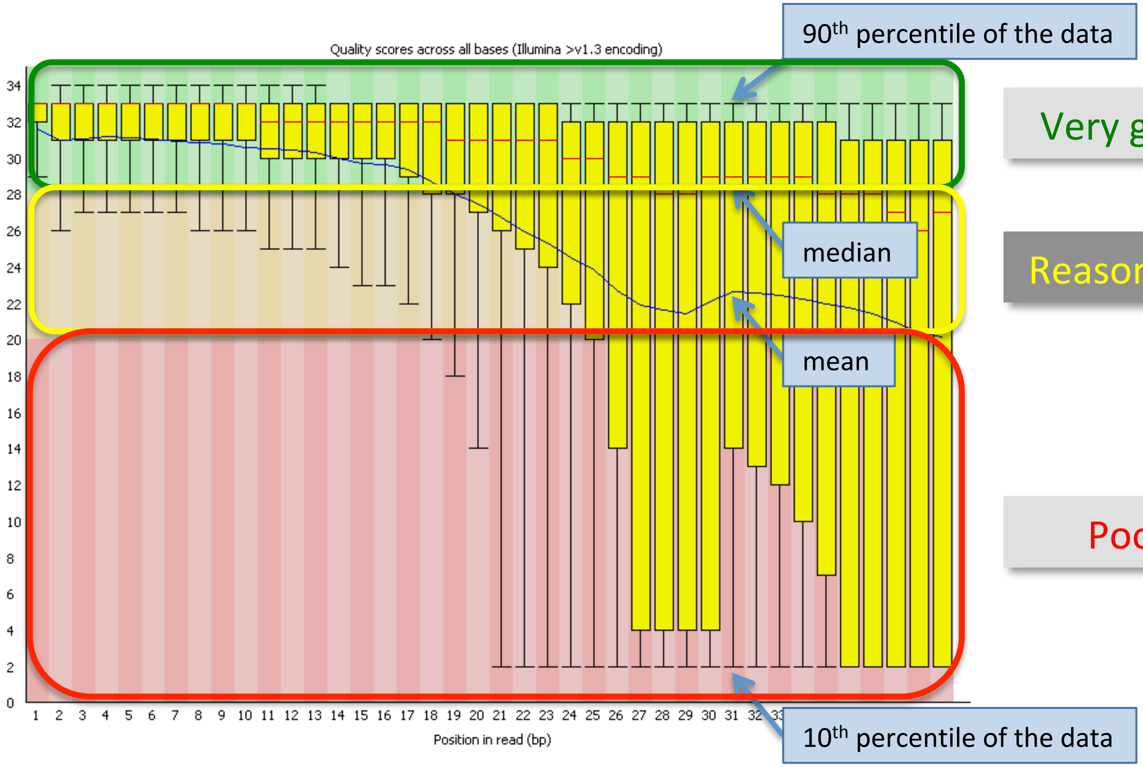
Measure	Value
Filename	Raji.dmso.brd4.bam
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	61461639
Sequence length	14-51
%GC	39

✓ Per base sequence quality

Quality scores across all bases (Sanger / Illumina 1.9 encoding)
40
38
36
34
32
30
28
26
24
22

Produced by FastQC (version 0.9.3)

Quality checks: FastqQC reports



90th percentile of the data

median

mean

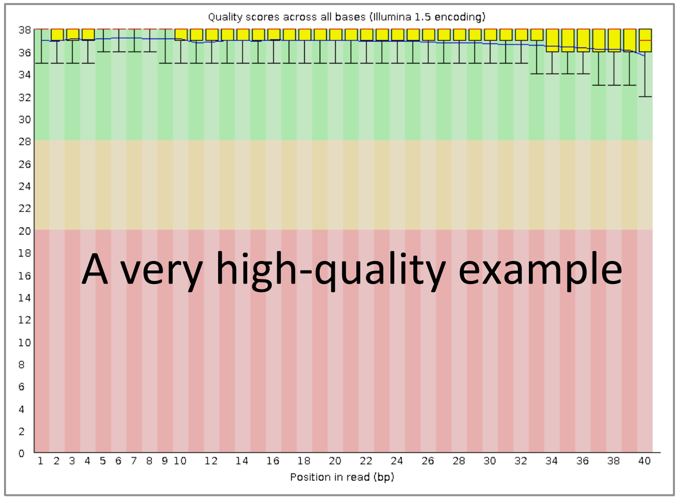
10th percentile of the data

Very good quality

Reasonable quality

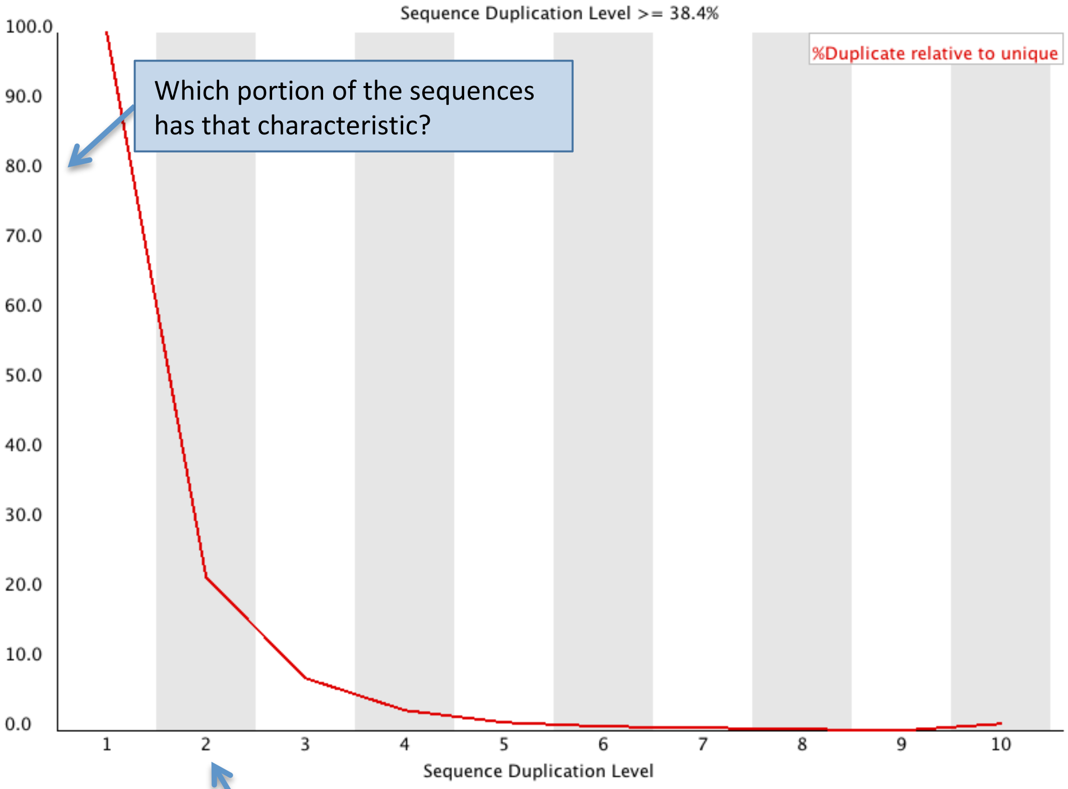
Poor quality

ACCTGGGATCAAACATTCAGGACATATAGCACAATAGGAC



A very high-quality example

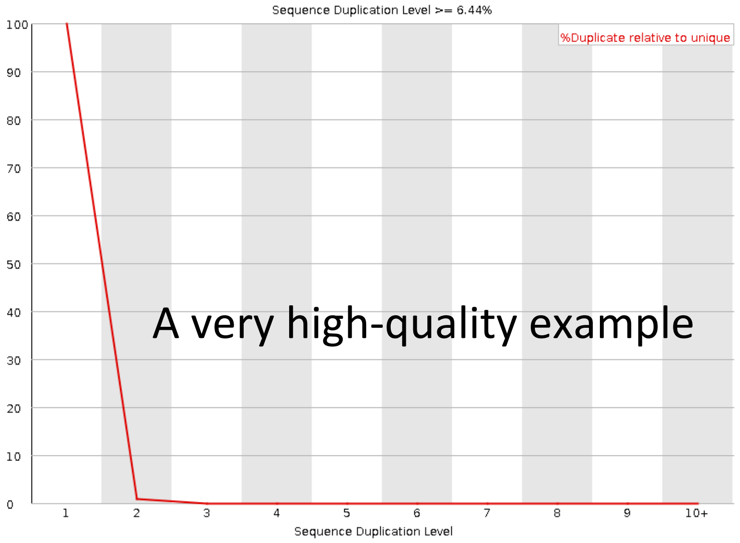
Quality checks: FastqQC reports



Which portion of the sequences has that characteristic?

How many times is the sequence repeated?

Computed for the first 200’000 reads to estimate the duplication levels in the whole file



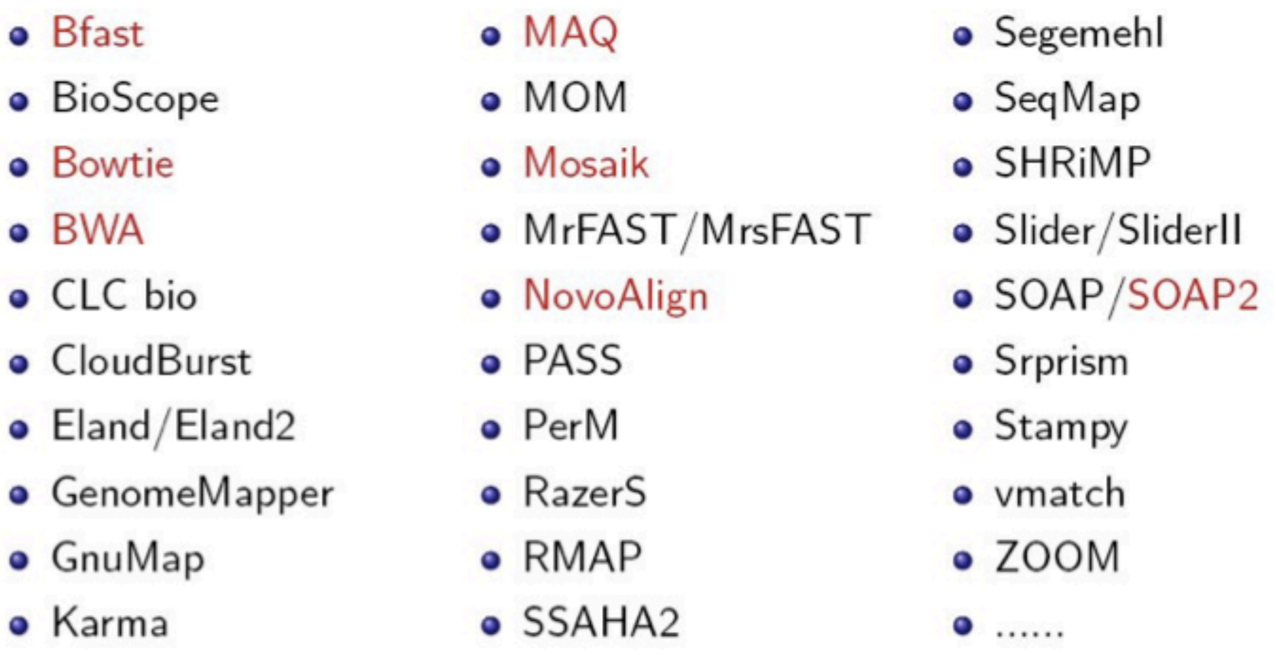
A very high-quality example

Outline of the presentation

- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

Reads alignment

There are many aligners and many short reads aligners:



- Bfast
- BioScope
- Bowtie
- BWA
- CLC bio
- CloudBurst
- Eland/Eland2
- GenomeMapper
- GnuMap
- Karma
- MAQ
- MOM
- Mosaik
- MrFAST/MrsFAST
- NovoAlign
- PASS
- PerM
- RazerS
- RMAP
- SSAHA2
- Segemehl
- SeqMap
- SHRiMP
- Slider/SliderII
- SOAP/SOAP2
- Srprism
- Stampy
- vmatch
- ZOOM
- ......

Li et al, 2010

Reads alignment

- Mapping: quickly identify candidates of hits on the reference genome
- Alignment and report: score the alignment
- Important features:
  - Some software use the base quality score to evaluate alignment, others do not
  - For all the aligners there is a trade off between performance and accuracy
  - Gapped or ungapped alignment
  - Important parameters:
    - Maximum of mismatches
    - Reporting unique hits or multiple hits

Reads alignment: BWA

•It uses Burrows-Wheeler indexing algorithm to speed up alignment time

• Fast and moderate memory usage

• Work for different sequencing platforms, for SE and PE

• Gapped alignment for both SE and PE reads

• Effective pairing to achieve high alignment accuracy; suboptimal hits considered in pairing.

• Non-unique read is placed randomly with a mapping quality 0.

• Reports ambiguous hits

References:

1.Li, H. and Durbin, R., Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25** (14), 1754 (2009).

2.http://bio-bwa.sourceforge.net/bwa.shtml

Reads alignment output : SAM/BAM file formats

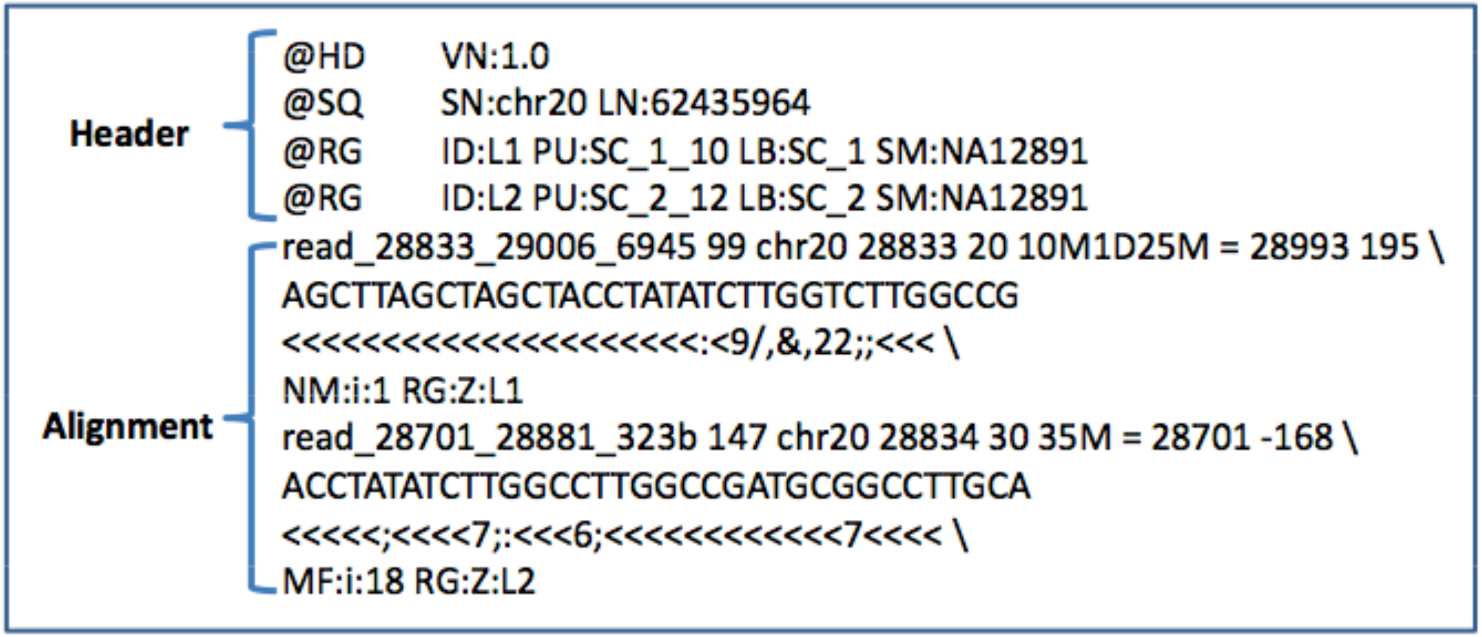
The Sequence Alignment/Map (SAM) format:

- Format for the storage of sequence alignments and their mapping coordinates
- Supports different sequencing platforms
- Flexible in style, compact in size, computationally efficient to access

BAM is the binary version of the SAM format

Samtools is a set of tools for manipulating and controlling SAM/BAM files

Reads alignment output : SAM/BAM file formats


Header
Alignment

```
@HD     VN:1.0
@SQ     SN:chr20 LN:62435964
@RG     ID:L1 PU:SC_1_10 LB:SC_1 SM:NA12891
@RG     ID:L2 PU:SC_2_12 LB:SC_2 SM:NA12891
read_28833_29006_6945 99 chr20 28833 20 10M1D25M = 28993 195 \
AGCTTAGCTAGCTACCTATATCTTGGTCTTGGCCG
<<<<<<<<<<<<<<<<<<<<<:<9/,&,22;;<<< \
NM:i:1 RG:Z:L1
read_28701_28881_323b 147 chr20 28834 30 35M = 28701 -168 \
ACCTATATCTTGGCCTTGGCCGATGCGGCCTTGCA
<<<<<;<<<<7;:<<<6;<<<<<<<<<<<<7<<<< \
MF:i:18 RG:Z:L2
```

Tag Description:

@HD The header line.

@SQ Reference sequence dictionary. The order of @SQ lines defines the alignment sorting order.

LN Reference sequence length. Range: [1,229-1]

@RG Read group. Unordered multiple @RG lines are allowed.

ID Read group identifier. Each @RG line must have a unique ID. The value of ID is used in the RG tags of alignment records. Must be unique among all read groups in header section.

CN Name of sequencing center producing the read.

LB Library.

PU Platform unit

SM Sample. Use pool name where a pool is being sequenced.

Reads alignment output : SAM/BAM file formats

```
@HD    VN:1.0  GO:none SO:coordinate
@SQ    SN:chrM     LN:16571
@SQ    SN:chr1     LN:247249719
@SQ    SN:chr2     LN:242951149
[cut for clarity]
@SQ    SN:chr9     LN:140273252
@SQ    SN:chr10    LN:135374737
@SQ    SN:chr11    LN:134452384
[cut for clarity]
@SQ    SN:chr22    LN:49691432
@SQ    SN:chrX     LN:154913754
@SQ    SN:chrY     LN:57772954
@RG    ID:20FUK.1    PL:illumina    PU:20FUKAAXX100202.1    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.2    PL:illumina    PU:20FUKAAXX100202.2    LB:Solexa-18484 SM:NA12878    CN:BI
@RG    ID:20FUK.3    PL:illumina    PU:20FUKAAXX100202.3    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.4    PL:illumina    PU:20FUKAAXX100202.4    LB:Solexa-18484 SM:NA12878    CN:BI
@RG    ID:20FUK.5    PL:illumina    PU:20FUKAAXX100202.5    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.6    PL:illumina    PU:20FUKAAXX100202.6    LB:Solexa-18484 SM:NA12878    CN:BI
@RG    ID:20FUK.7    PL:illumina    PU:20FUKAAXX100202.7    LB:Solexa-18483 SM:NA12878    CN:BI
@RG    ID:20FUK.8    PL:illumina    PU:20FUKAAXX100202.8    LB:Solexa-18484 SM:NA12878    CN:BI
@PG    ID:BWA  VN:0.5.7       CL:tk
@PG    ID:GATK TableRecalibration      VN:1.0.2864
20FUKAAXX100202:1:1:12730:189900      163    chrM    1     60     101M    =     282    381
         GATCACAGGTCTATCACCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTA…[more bases]
         ?BA@A>BBBBACBBAC@BBCBBCBC@BC@CAC@:BBCBBCACAACBABCBCCAB…[more quals]
         RG:Z:20FUK.1    NM:i:1  SM:i:37 AM:i:37 MD:Z:72G28     MQ:i:60 PG:Z:BWA      UQ:i:33
```

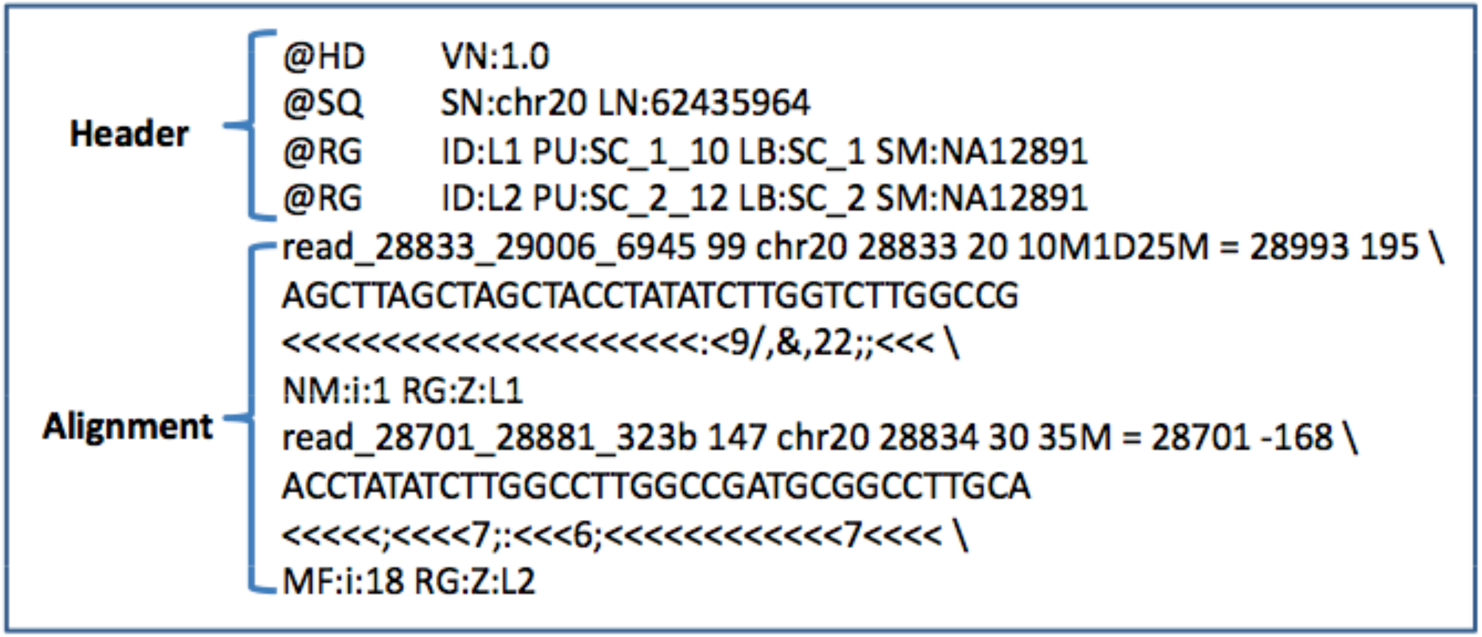
Required: Standard header

Essential: contigs of aligned reference sequence. Should be in karotypic order.

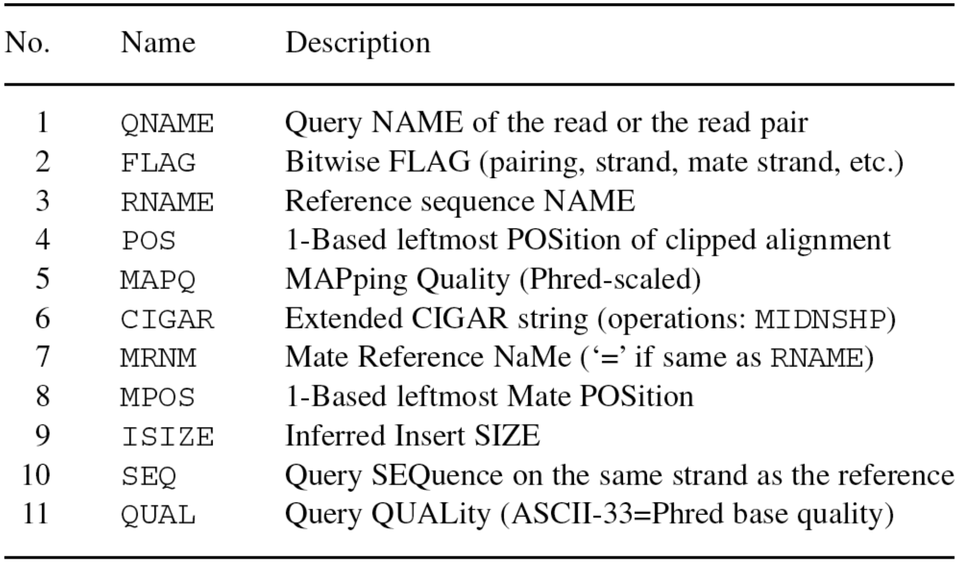
Essential: read groups. Carries platform (PL), library (LB), and sample (SM) information. Each read is associated with a read group

Useful: Data processing tools applied to the reads

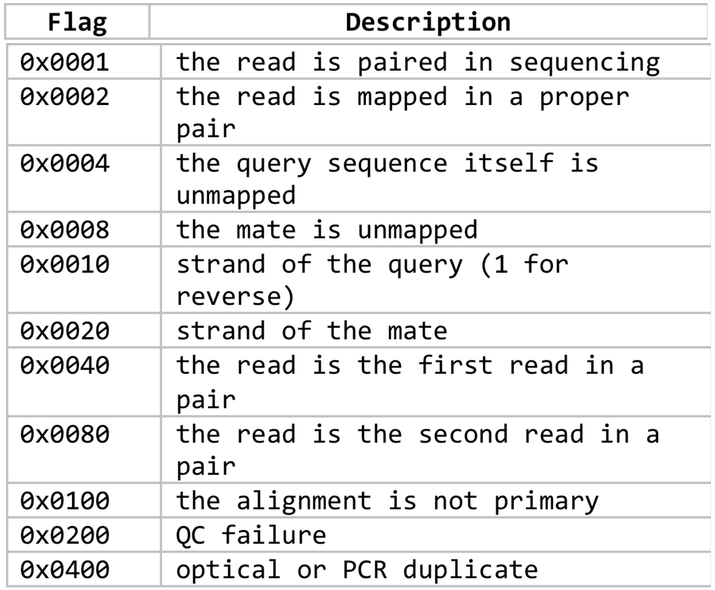
Reads alignment output : SAM/BAM file formats


Header
Alignment

```
@HD     VN:1.0
@SQ     SN:chr20 LN:62435964
@RG     ID:L1 PU:SC_1_10 LB:SC_1 SM:NA12891
@RG     ID:L2 PU:SC_2_12 LB:SC_2 SM:NA12891
read_28833_29006_6945 99 chr20 28833 20 10M1D25M = 28993 195 \
AGCTTAGCTAGCTACCTATATCTTGGTCTTGGCCG
<<<<<<<<<<<<<<<<<<<<<:<9/,&,22;;<<< \
NM:i:1 RG:Z:L1
read_28701_28881_323b 147 chr20 28834 30 35M = 28701 -168 \
ACCTATATCTTGGCCTTGGCCGATGCGGCCTTGCA
<<<<<;<<<<7;:<<<6;<<<<<<<<<<<<7<<<< \
MF:i:18 RG:Z:L2
```

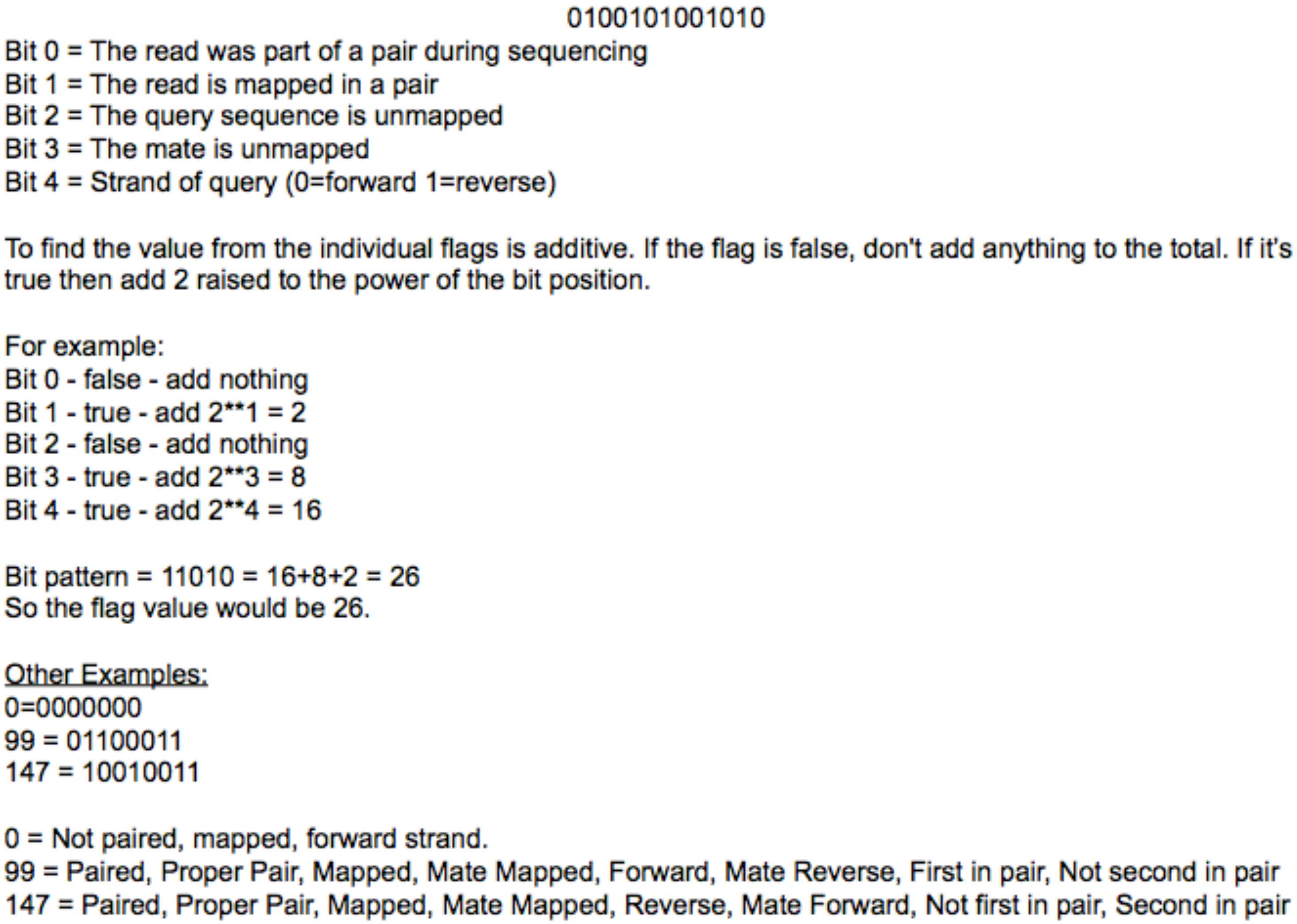


No.	Name	Description
1	QNAME	Query NAME of the read or the read pair
2	FLAG	Bitwise FLAG (pairing, strand, mate strand, etc.)
3	RNAME	Reference sequence NAME
4	POS	1-Based leftmost POSition of clipped alignment
5	MAPQ	MAPping Quality (Phred-scaled)
6	CIGAR	Extended CIGAR string (operations: MIDNSHP)
7	MRNM	Mate Reference NaMe ('=' if same as RNAME)
8	MPOS	1-Based leftmost Mate POSition
9	ISIZE	Inferred Insert SIZE
10	SEQ	Query SEQuence on the same strand as the reference
11	QUAL	Query QUALity (ASCII-33=Phred base quality)



Flag	Description
0x0001	the read is paired in sequencing
0x0002	the read is mapped in a proper pair
0x0004	the query sequence itself is unmapped
0x0008	the mate is unmapped
0x0010	strand of the query (1 for reverse)
0x0020	strand of the mate
0x0040	the read is the first read in a pair
0x0080	the read is the second read in a pair
0x0100	the alignment is not primary
0x0200	QC failure
0x0400	optical or PCR duplicate

Reads alignment output : SAM/BAM file formats



0100101001010

Bit 0 = The read was part of a pair during sequencing
Bit 1 = The read is mapped in a pair
Bit 2 = The query sequence is unmapped
Bit 3 = The mate is unmapped
Bit 4 = Strand of query (0=forward 1=reverse)

To find the value from the individual flags is additive. If the flag is false, don't add anything to the total. If it's true then add 2 raised to the power of the bit position.

For example:
Bit 0 - false - add nothing
Bit 1 - true - add 2**1 = 2
Bit 2 - false - add nothing
Bit 3 - true - add 2**3 = 8
Bit 4 - true - add 2**4 = 16

Bit pattern = 11010 = 16+8+2 = 26
So the flag value would be 26.

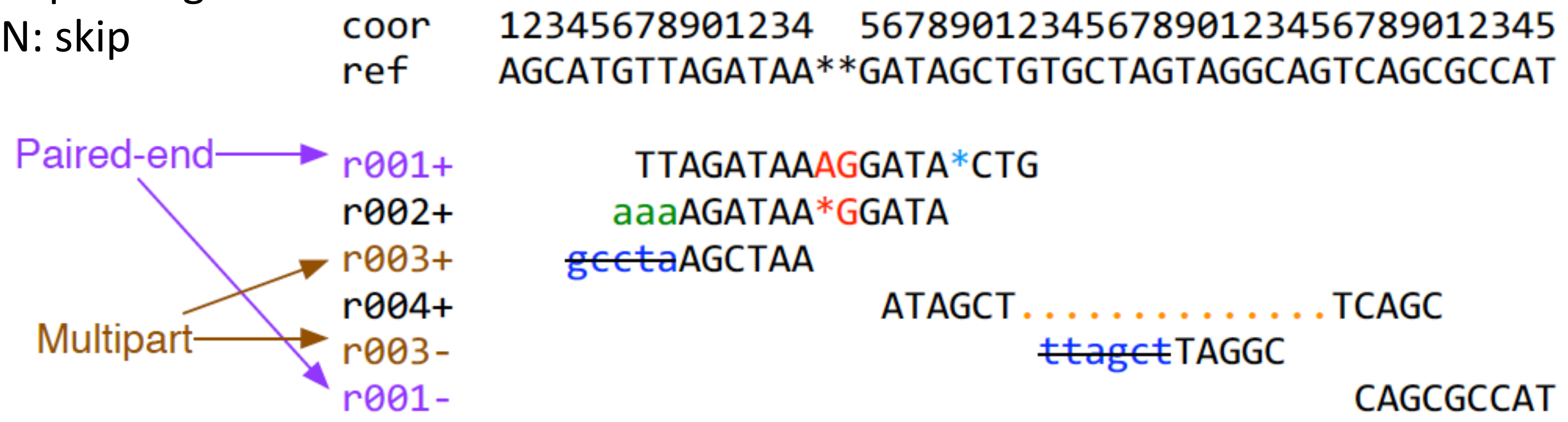
Other Examples:
0=0000000
99 = 01100011
147 = 10010011

0 = Not paired, mapped, forward strand.
99 = Paired, Proper Pair, Mapped, Mate Mapped, Forward, Mate Reverse, First in pair, Not second in pair
147 = Paired, Proper Pair, Mapped, Mate Mapped, Reverse, Mate Forward, Not first in pair, Second in pair

Reads alignment output : SAM/BAM file formats

CIGAR string

- M: match/mismatch
- I: insertion
- D: deletion
- S: softclip
- H: hardclip
- P: padding
- N: skip



Paired-end

Multipart

```
coor     12345678901234  5678901234567890123456789012345
ref      AGCATGTTAGATAA**GATAGCTGTGCTAGTAGGCAGTCAGCGCCAT

r001+          TTAGATAAAGGATA*CTG
r002+         aaaAGATAA*GGATA
r003+       ~~gccta~~AGCTAA
r004+                    ATAGCT..............TCAGC
r003-                           ~~ttagct~~TAGGC
r001-                                         CAGCGCCAT
```

```
                @SQ SN:ref LN:45
Ins & padding   r001 163 ref  7 30 8M2I4M1D3M = 37  39 TTAGATAAAGGATACTA *
Soft clipping   r002   0 ref  9 30 3S6M1P1I4M *  0   0 AAAAGATAAGGATA    *
                r003   0 ref  9 30 5H6M       *  0   0 AGCTAA    *    NM:i:1
Splicing        r004   0 ref 16 30 6M14N5M    *  0   0 ATAGCTTCAGC       *
Hard clipping   r003  16 ref 29 30 6H5M       *  0   0 TAGGC     *    NM:i:0
                r001  83 ref 37 30 9M         =  7 -39 CAGCGCCAT         *
```

Outline of the presentation

- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

Manipulating Reads alignments: SAMtools

- Library and software package that manipulate BAM/SAM files
- SAM→BAM conversion (samtools view)

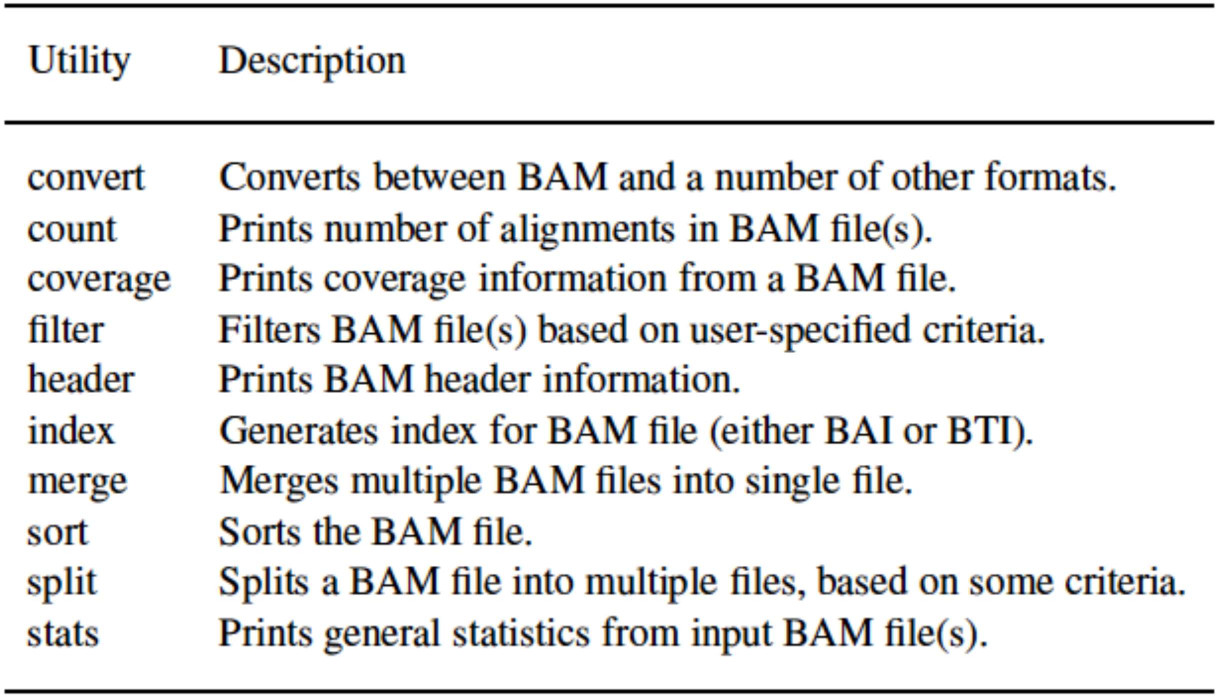
- Samtools view –f INT file.bam
  Only output alignments with all bits in INT present in the FLAG field.

- Samtools view –F INT file.bam
  Skip alignments with bits present in INT [0]

- Creates sorted and indexed BAM files from SAM files (samtools sort /samtools index)
- Removing PCR duplicates (samtools rmdup)
- Merging alignments (samtools merge)
- Visualization of alignments from BAM files
- **SNP calling and short indel detection**

References:
1. http://samtools.sourceforge.net/
2.http://samtools.sourceforge.net/samtools.shtml

Manipulating Reads alignments: BamTools



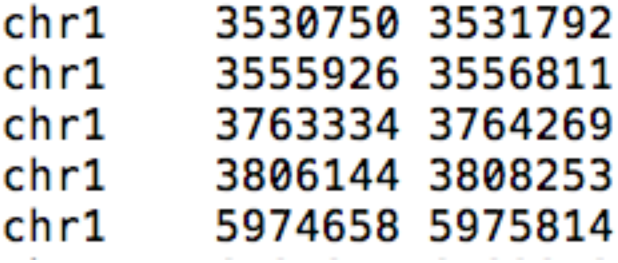
Utility	Description
convert	Converts between BAM and a number of other formats.
count	Prints number of alignments in BAM file(s).
coverage	Prints coverage information from a BAM file.
filter	Filters BAM file(s) based on user-specified criteria.
header	Prints BAM header information.
index	Generates index for BAM file (either BAI or BTI).
merge	Merges multiple BAM files into single file.
sort	Sorts the BAM file.
split	Splits a BAM file into multiple files, based on some criteria.
stats	Prints general statistics from input BAM file(s).

Freely available at
http://github.org/pezmaster31/bamtools

Manipulating Reads alignments and genomic intervals: BEDTools

The BEDTools utilities allow one to address common genomics tasks such as finding feature overlaps and computing coverage. The utilities are largely based on four widely-used file formats: BED, GFF/GTF, VCF and SAM/BAM.

BED format



```
chr1     3530750 3531792
chr1     3555926 3556811
chr1     3763334 3764269
chr1     3806144 3808253
chr1     5974658 5975814
```

- **slopBed** Adjusts each BED entry by a requested number of base pairs.
- **shuffleBed** Randomly permutes the locations of a BED file among a genome.
- **intersectBed (BAM)** Returns overlaps between two BED/GFF/VCF files.
- **genomeCoverageBed (BAM)** Creates a "per base" report of genome coverage.
- **subtractBed** Removes the portion of an interval that is overlapped by another feature.
- **mergeBed** Merges overlapping features into a single feature.

References:

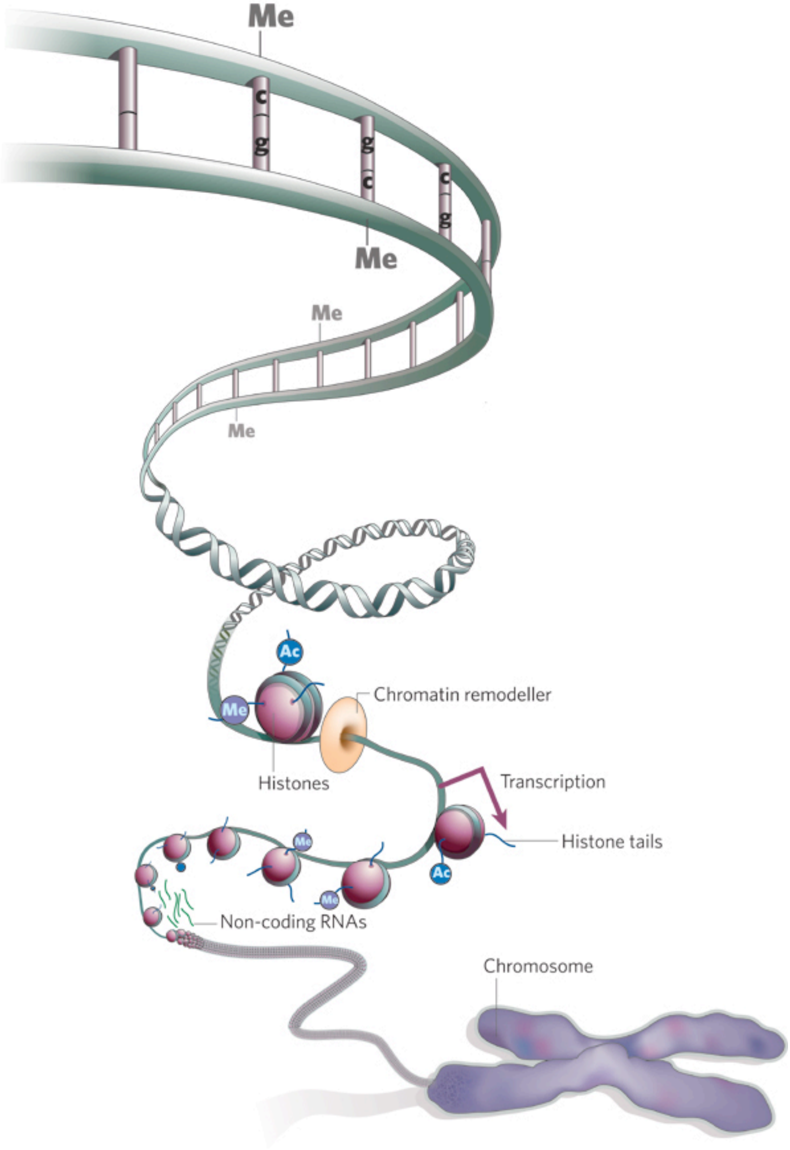
1 Quinlan AR and Hall IM, 2010. BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics. 26, 6, pp. 841–842.

2 http://code.google.com/p/bedtools/downloads/detail?name=BEDTools-User-Manual.v4.pdf

Outline of the presentation

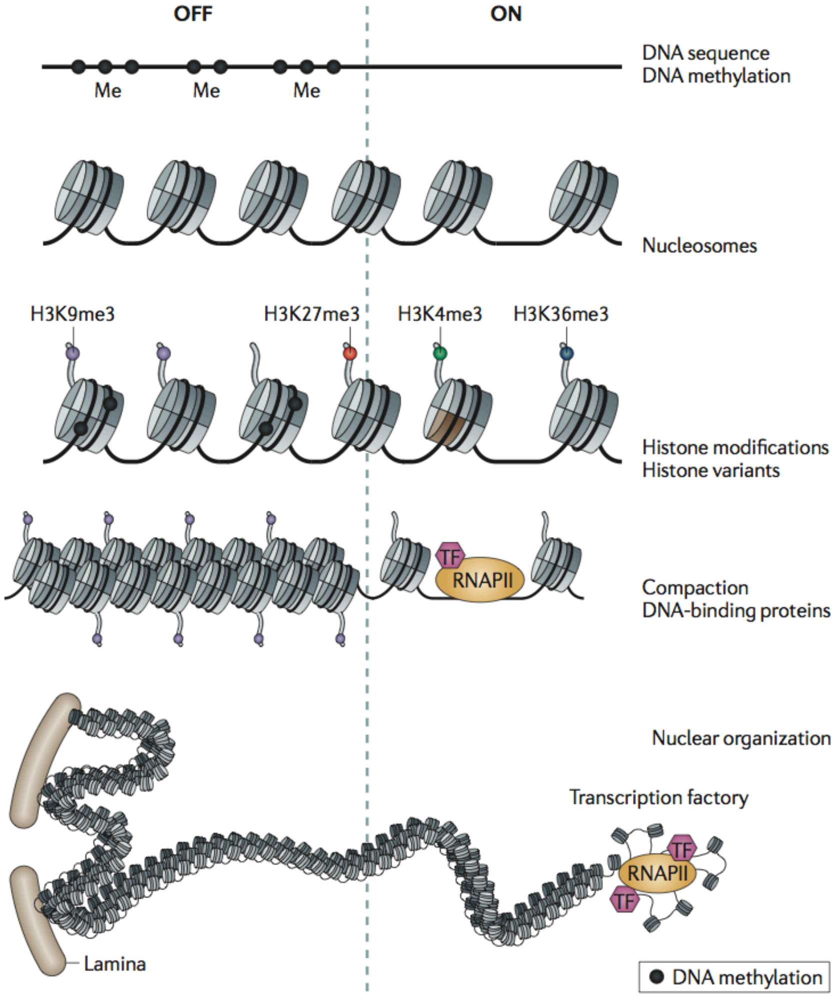
- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

Layers of chromatin organization



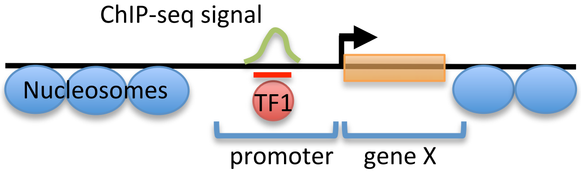
Nature 454, 711-715 (Aug 2008)

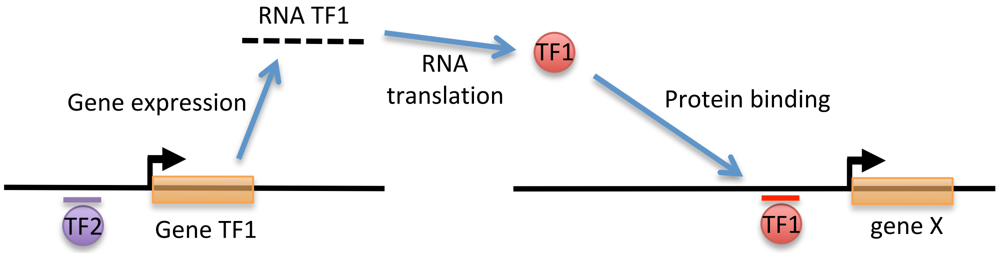
Layers of chromatin organization



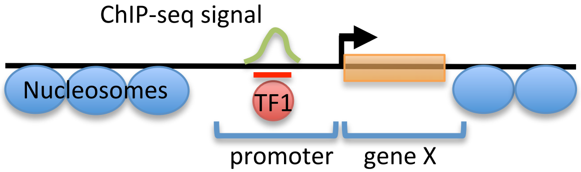
Zhou VW *et al*, Nature Review Genetics 2010

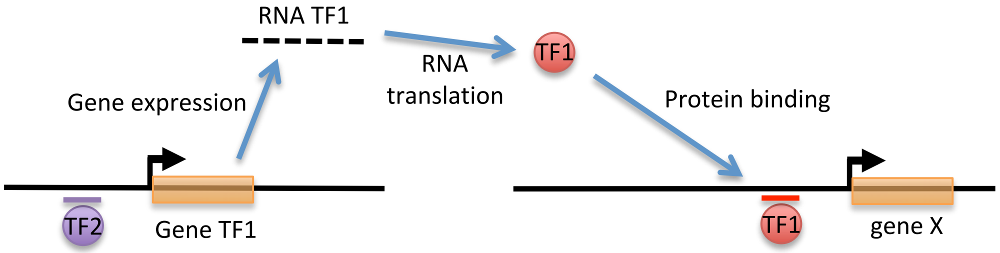
Transcription factors (TFs)



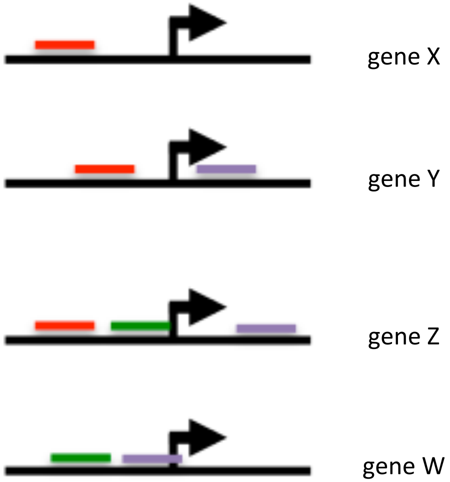


Transcription factors (TFs)

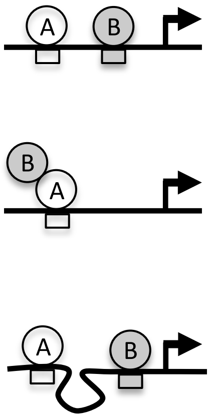




Combinatorial control



Interaction modes



Histone modifications

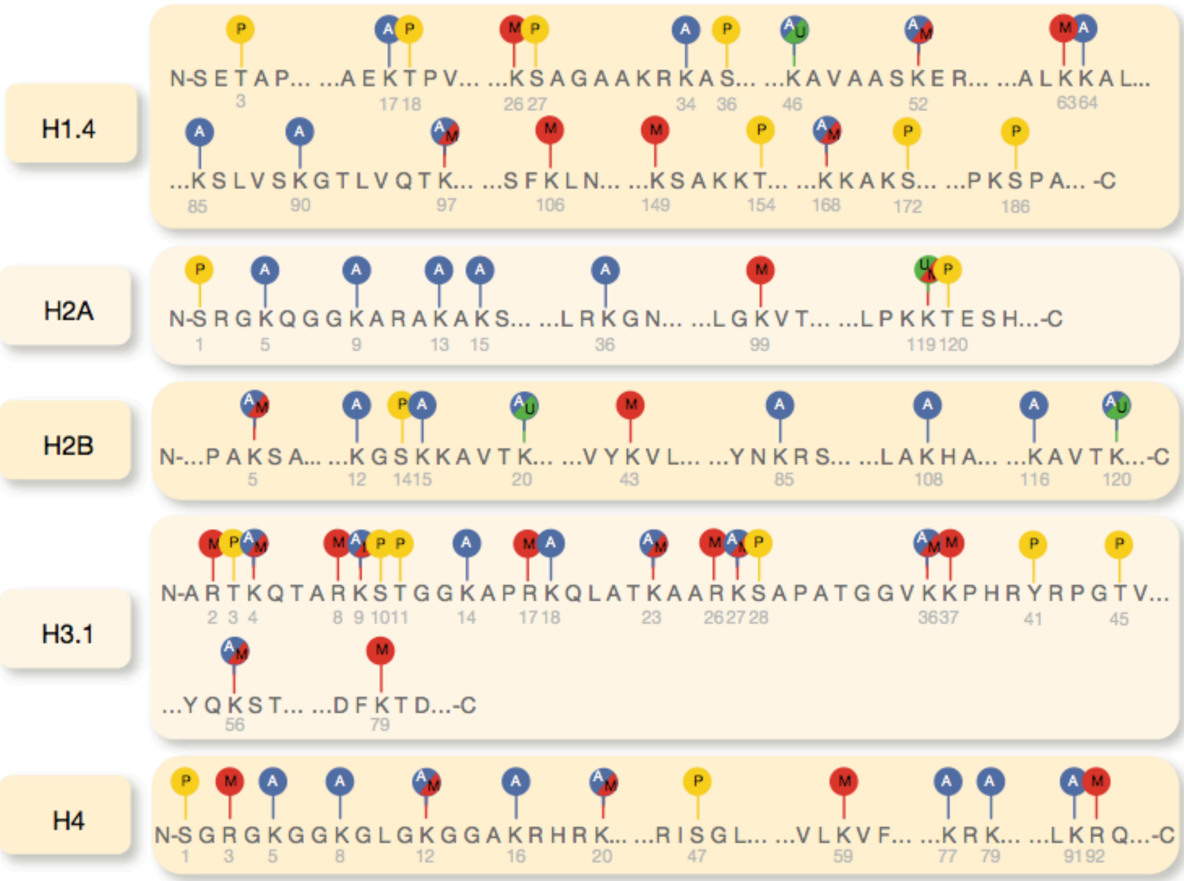
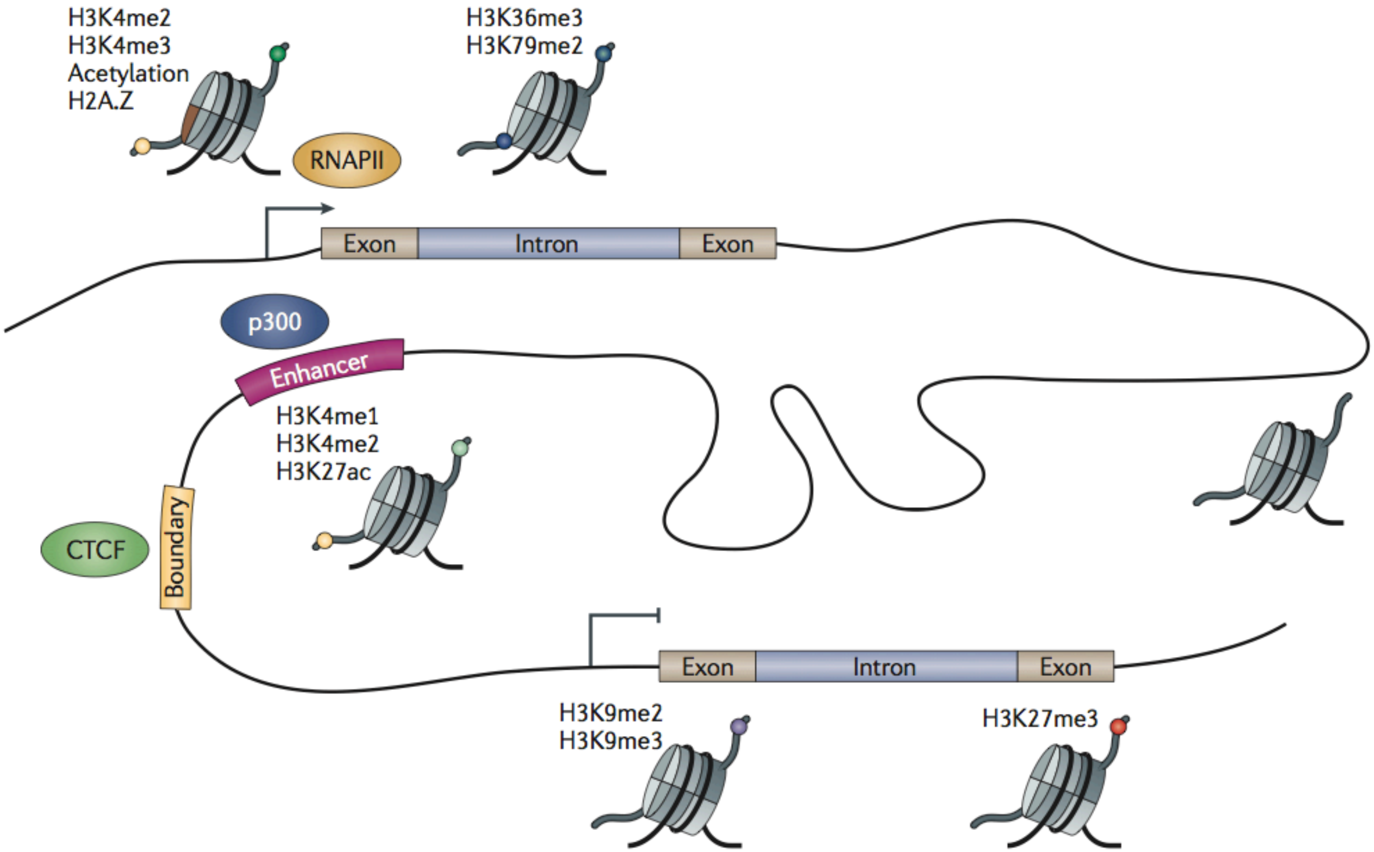

H1.4
N-S E T A P... ...A E K T P V... ...K S A G A A K R K A S... ...K A V A A S K E R... ...A L K K A L...
3 17 18 26 27 34 36 46 52 63 64
...K S L V S K G T L V Q T K... ...S F K L N... ...K S A K K T... ...K K A K S... ...P K S P A... -C
85 90 97 106 149 154 168 172 186
H2A
N-S R G K Q G G K A R A K A K S... ...L R K G N... ...L G K V T... ...L P K K T E S H...-C
1 5 9 13 15 36 99 119 120
H2B
N-...P A K S A... ...K G S K K A V T K... ...V Y K V L... ...Y N K R S... ...L A K H A... ...K A V T K...-C
5 12 14 15 20 43 85 108 116 120
H3.1
N-A R T K Q T A R K S T G G K A P R K Q L A T K A A R K S A P A T G G V K K P H R Y R P G T V...
2 3 4 8 9 10 11 14 17 18 23 26 27 28 36 37 41 45
...Y Q K S T... ...D F K T D...-C
56 79
H4
N-S G R G K G G K G L G K G G A K R H R K... ...R I S G L... ...V L K V F... ...K R K... ...L K R Q...-C
1 3 5 8 12 16 20 47 59 77 79 91 92

Figure 3 Histone modifications. All histones are subject to post-transcriptional modifications, which mainly occur in histone tails. The main post-transcriptional modifications are depicted in this figure: acetylation (blue), methylation (red), phosphorylation (yellow) and ubiquitination (green). The number in gray under each amino acid represents its position in the sequence.

Portela A and Esteller M, Nature Biotechnology 2010

Histone modifications



Zhou VW *et al*, Nature Review Genetics 2010

Histone modifications may serve as ‘dials’ or ‘switches’ for cell type specificity

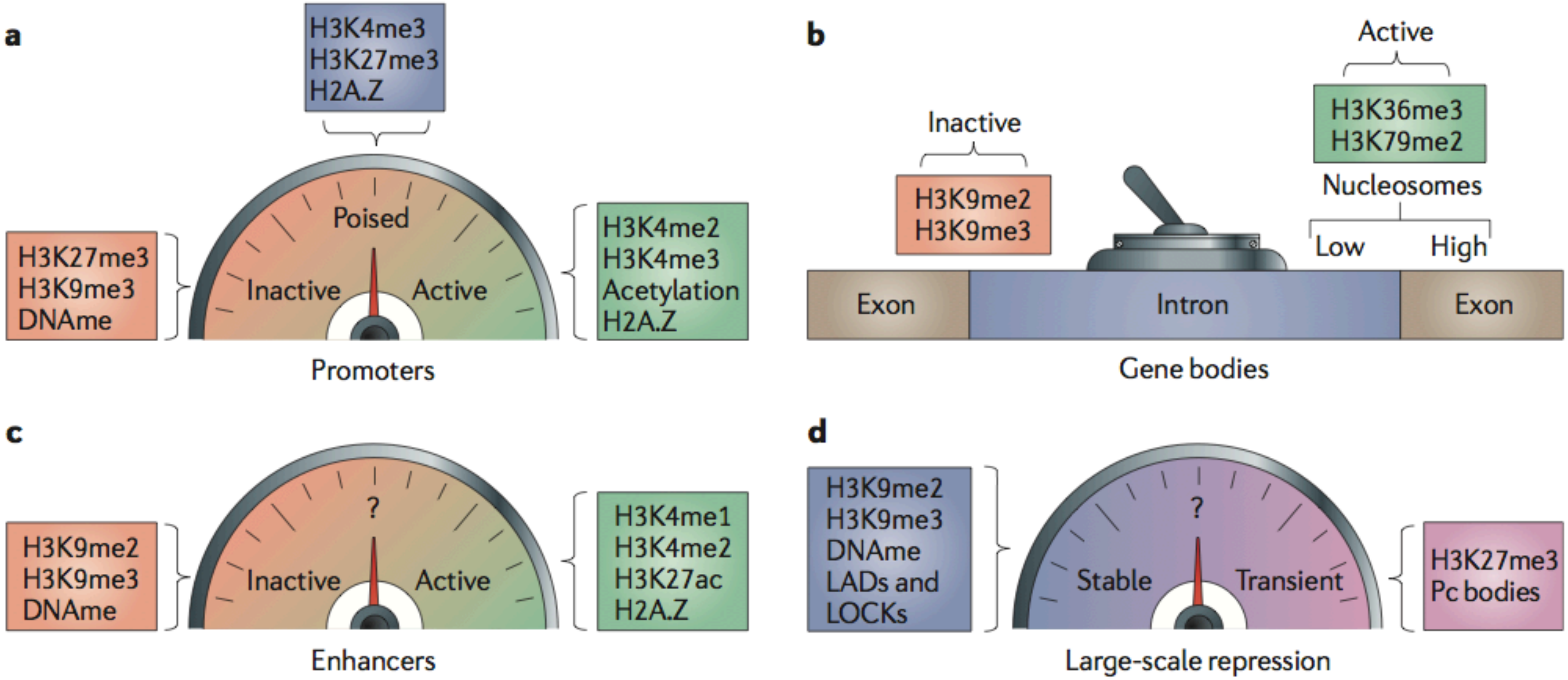


Figure 4 | **‘Dashboard’ of histone modifications for fine-tuning genomic elements.** In addition to enabling annotation, histone modifications may serve as ‘dials’ or ‘switches’ for cell type specificity. **a** | At promoters, they can contribute to fine-tuning of expression levels — from active to poised to inactive — and perhaps even intermediate levels. **b** | At gene bodies, they discriminate between active and inactive conformations. In addition, exons in active genes have higher nucleosome occupancy and thus more histone H3 lysine 36 trimethylation (H3K36me3) and H3K79me2-modified histones than introns. **c** | At distal sites, histone marks correlate with levels of enhancer activity. **d** | On a global scale, they may confer repression of varying stabilities and be associated with different genomic features. For example, lamina-associated domains (LADs) in the case of stable repression and Polycomb (Pc) bodies in the case of context-specific repression. DNAme, DNA methylation; LOCK, large organized chromatin K modification.

Zhou VW *et al*, Nature Review Genetics 2010

Epigenetic modifications in human diseases

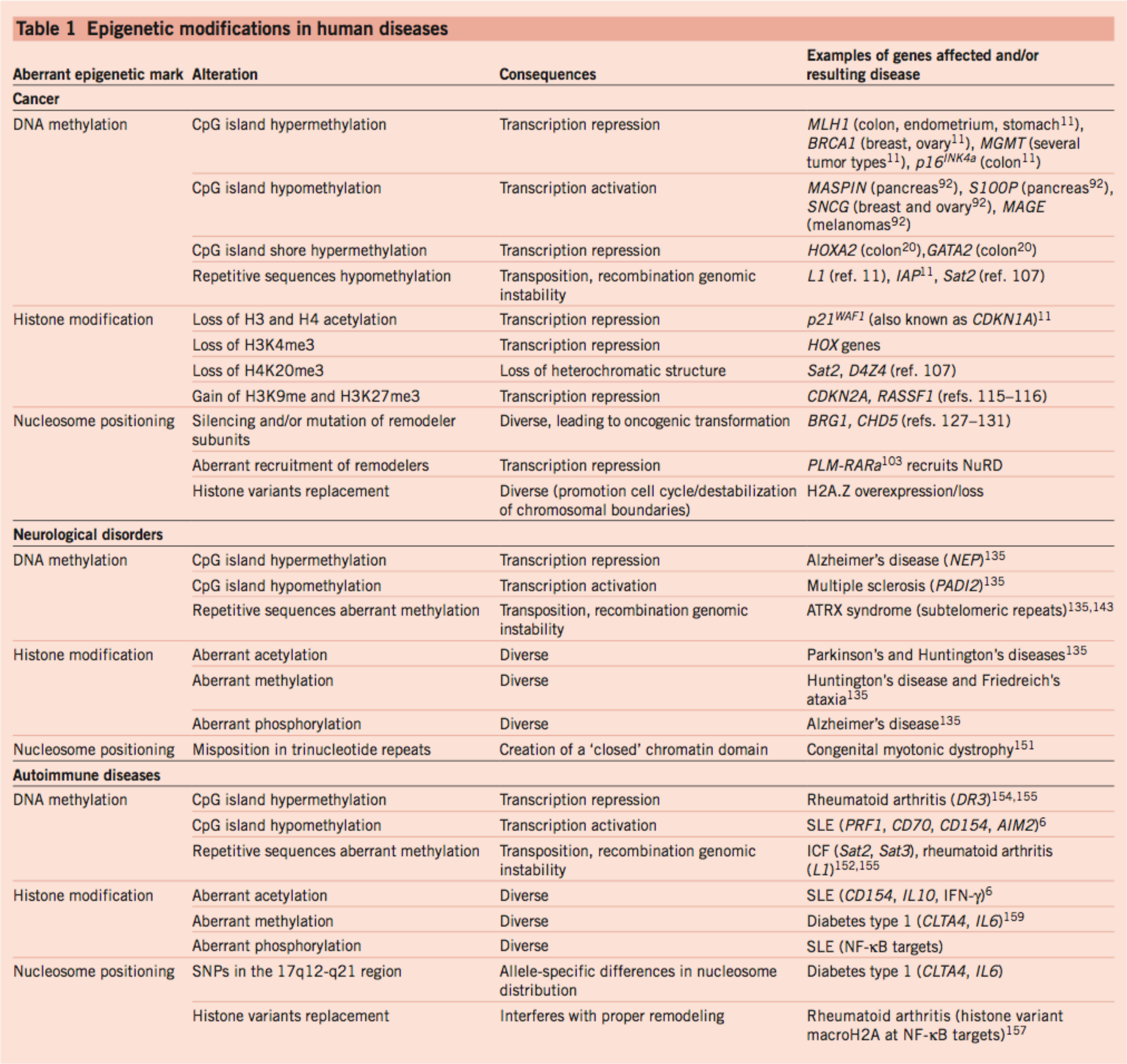
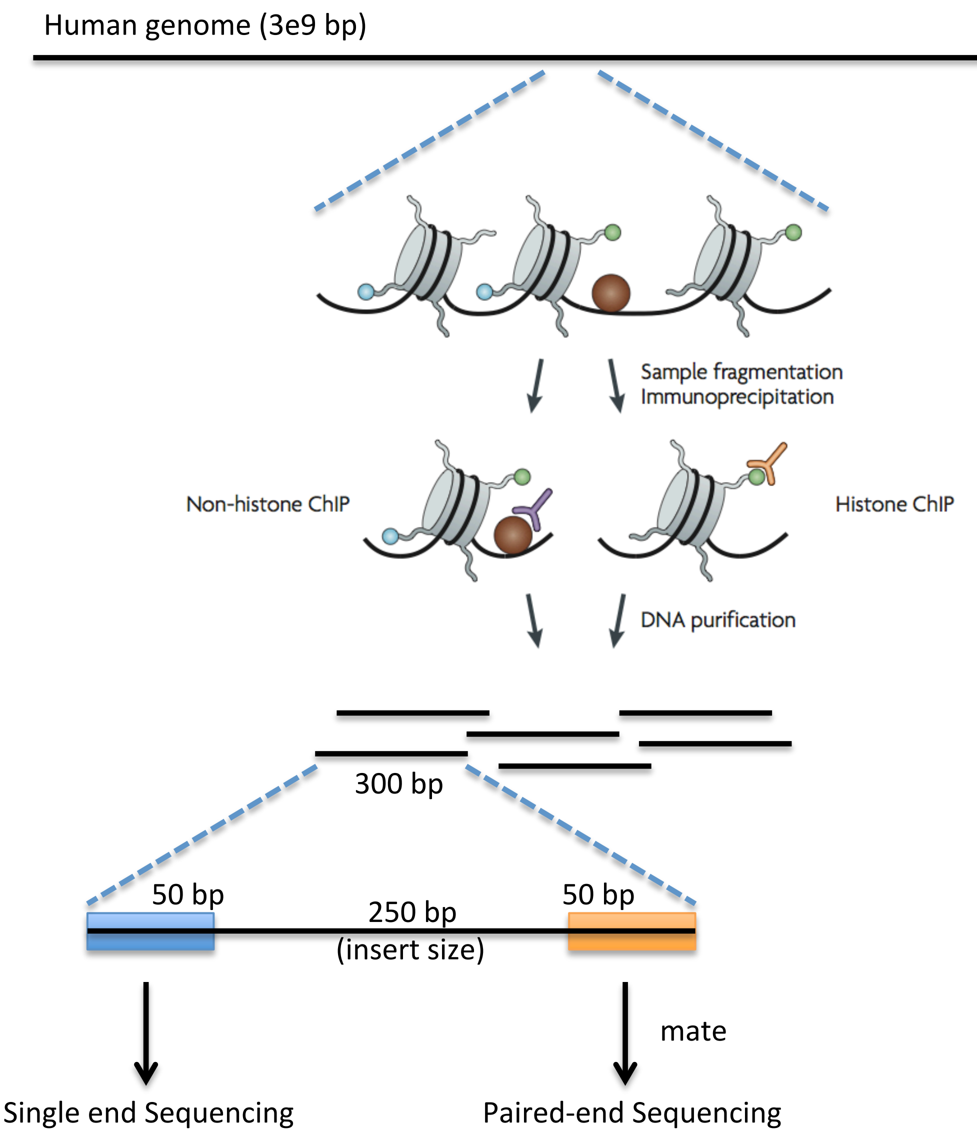


Table 1 Epigenetic modifications in human diseases

Aberrant epigenetic mark	Alteration	Consequences	Examples of genes affected and/or resulting disease
Cancer			
DNA methylation	CpG island hypermethylation	Transcription repression	*MLH1* (colon, endometrium, stomach[11]), *BRCA1* (breast, ovary[11]), *MGMT* (several tumor types[11]), *p16*^{INK4a} (colon[11])
	CpG island hypomethylation	Transcription activation	*MASPIN* (pancreas[92]), *S100P* (pancreas[92]), *SNCG* (breast and ovary[92]), *MAGE* (melanomas[92])
	CpG island shore hypermethylation	Transcription repression	*HOXA2* (colon[20]),*GATA2* (colon[20])
	Repetitive sequences hypomethylation	Transposition, recombination genomic instability	*L1* (ref. 11), *IAP*[11], *Sat2* (ref. 107)
Histone modification	Loss of H3 and H4 acetylation	Transcription repression	*p21*^{WAF1} (also known as *CDKN1A*)[11]
	Loss of H3K4me3	Transcription repression	*HOX* genes
	Loss of H4K20me3	Loss of heterochromatic structure	*Sat2*, *D4Z4* (ref. 107)
	Gain of H3K9me and H3K27me3	Transcription repression	*CDKN2A, RASSF1* (refs. 115–116)
Nucleosome positioning	Silencing and/or mutation of remodeler subunits	Diverse, leading to oncogenic transformation	*BRG1, CHD5* (refs. 127–131)
	Aberrant recruitment of remodelers	Transcription repression	*PLM-RARa*[103] recruits NuRD
	Histone variants replacement	Diverse (promotion cell cycle/destabilization of chromosomal boundaries)	H2A.Z overexpression/loss
Neurological disorders			
DNA methylation	CpG island hypermethylation	Transcription repression	Alzheimer's disease (*NEP*)[135]
	CpG island hypomethylation	Transcription activation	Multiple sclerosis (*PADI2*)[135]
	Repetitive sequences aberrant methylation	Transposition, recombination genomic instability	ATRX syndrome (subtelomeric repeats)[135,143]
Histone modification	Aberrant acetylation	Diverse	Parkinson's and Huntington's diseases[135]
	Aberrant methylation	Diverse	Huntington's disease and Friedreich's ataxia[135]
	Aberrant phosphorylation	Diverse	Alzheimer's disease[135]
Nucleosome positioning	Misposition in trinucleotide repeats	Creation of a 'closed' chromatin domain	Congenital myotonic dystrophy[151]
Autoimmune diseases			
DNA methylation	CpG island hypermethylation	Transcription repression	Rheumatoid arthritis (*DR3*)[154,155]
	CpG island hypomethylation	Transcription activation	SLE (*PRF1, CD70, CD154, AIM2*)[6]
	Repetitive sequences aberrant methylation	Transposition, recombination genomic instability	ICF (*Sat2, Sat3*), rheumatoid arthritis (*L1*)[152,155]
Histone modification	Aberrant acetylation	Diverse	SLE (*CD154, IL10*, IFN-γ)[6]
	Aberrant methylation	Diverse	Diabetes type 1 (*CLTA4, IL6*)[159]
	Aberrant phosphorylation	Diverse	SLE (NF-κB targets)
Nucleosome positioning	SNPs in the 17q12-q21 region	Allele-specific differences in nucleosome distribution	Diabetes type 1 (*CLTA4, IL6*)
	Histone variants replacement	Interferes with proper remodeling	Rheumatoid arthritis (histone variant macroH2A at NF-κB targets)[157]

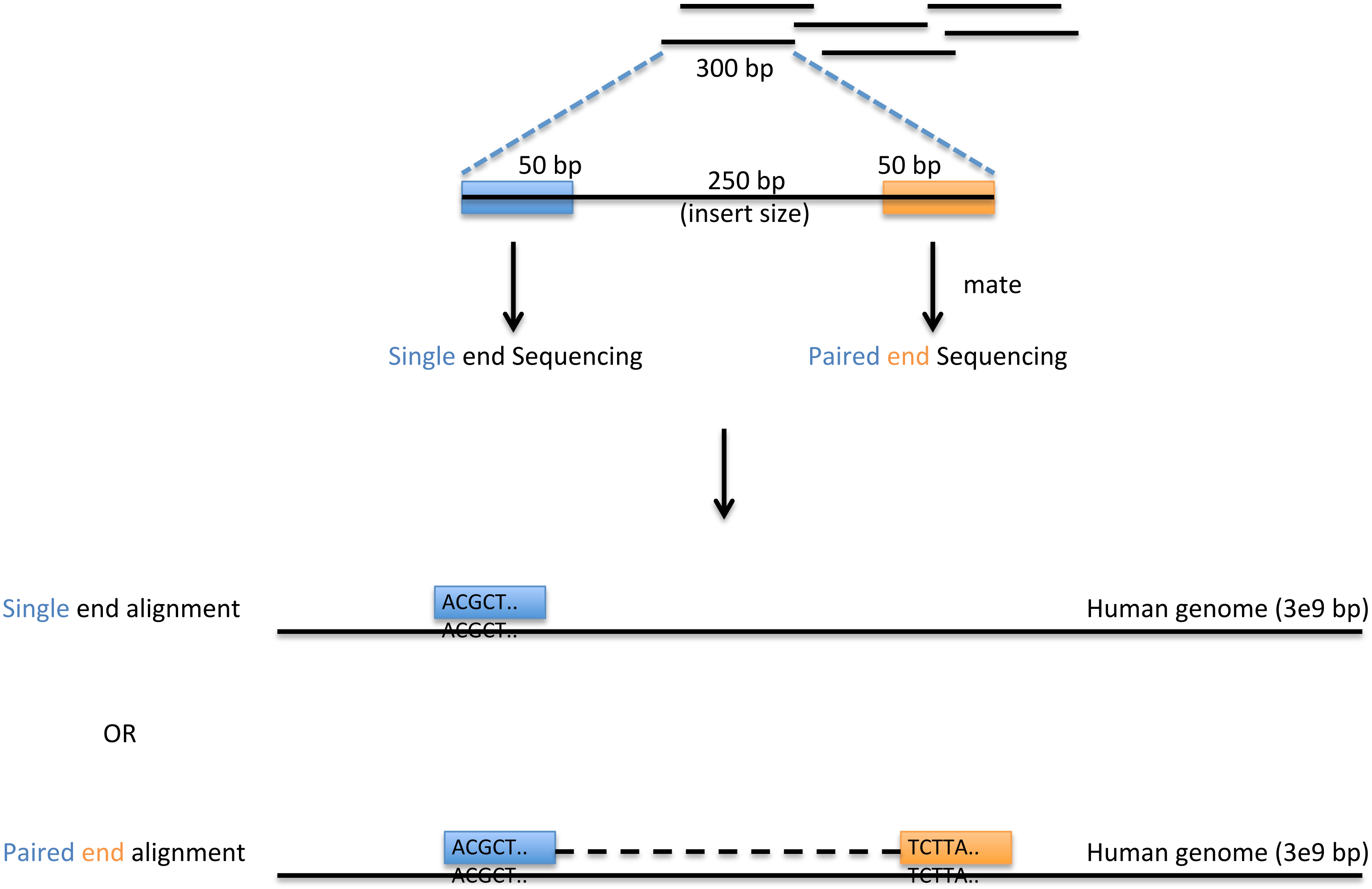
Portela A and Esteller M, Nature Biotechnology 2010

Identifying **TFs** or **Histone modifications** through ChIP-seq experiments



Adapted from Park P, Nature Review Genetics 2009

Identifying **TFs** or **Histone modifications** through ChIP-seq experiments



300 bp

50 bp

250 bp
(insert size)

50 bp

mate

Single end Sequencing

Paired end Sequencing

Single end alignment

ACGCT..

ACGCT..

Human genome (3e9 bp)

OR

Paired end alignment

ACGCT..

ACGCT..

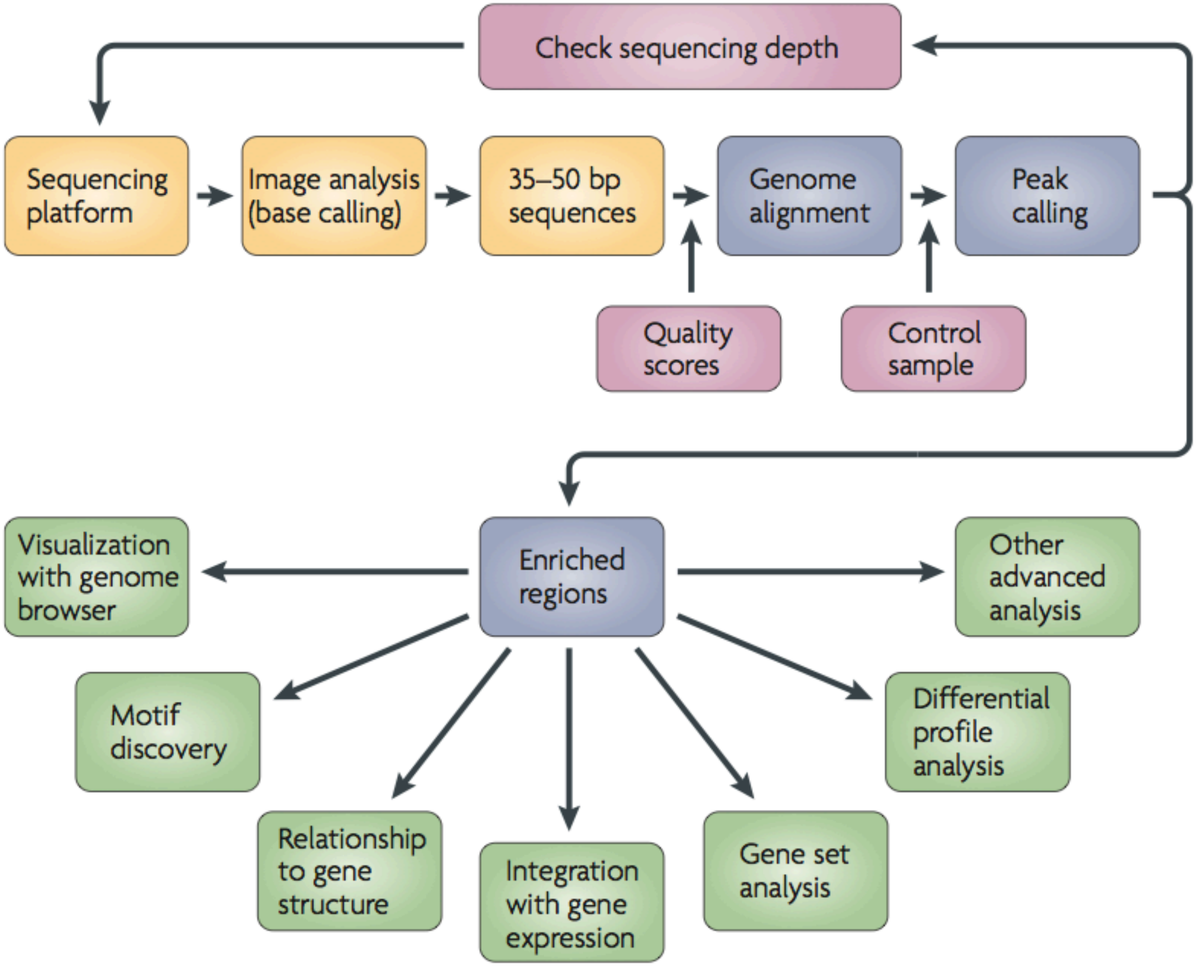
TCTTA..

TCTTA..

Human genome (3e9 bp)

Adapted from Park P, Nature Review Genetics 2009

ChIP-seq analysis workflow

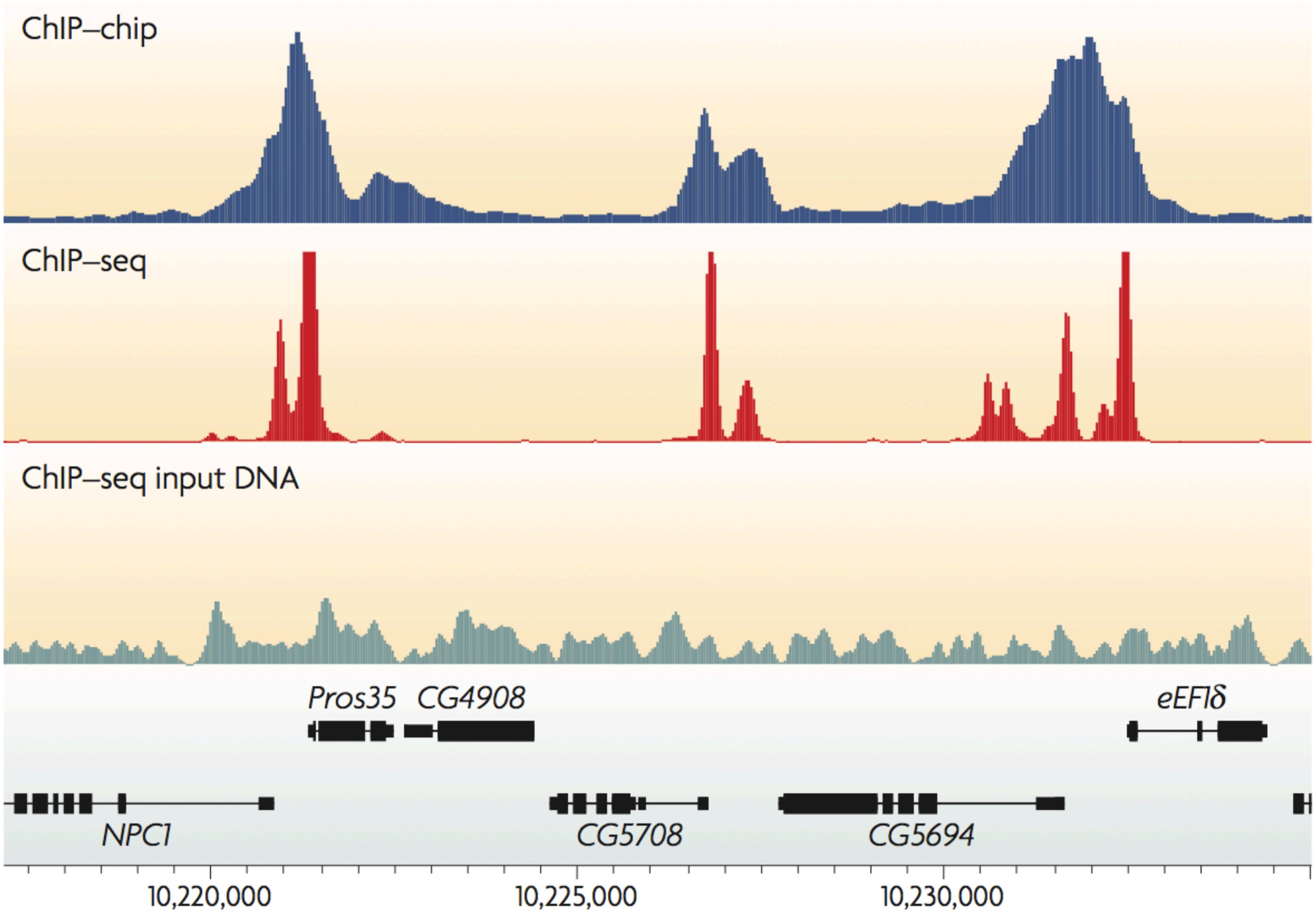


Park P, Nature Review Genetics 2009

Outline of the presentation

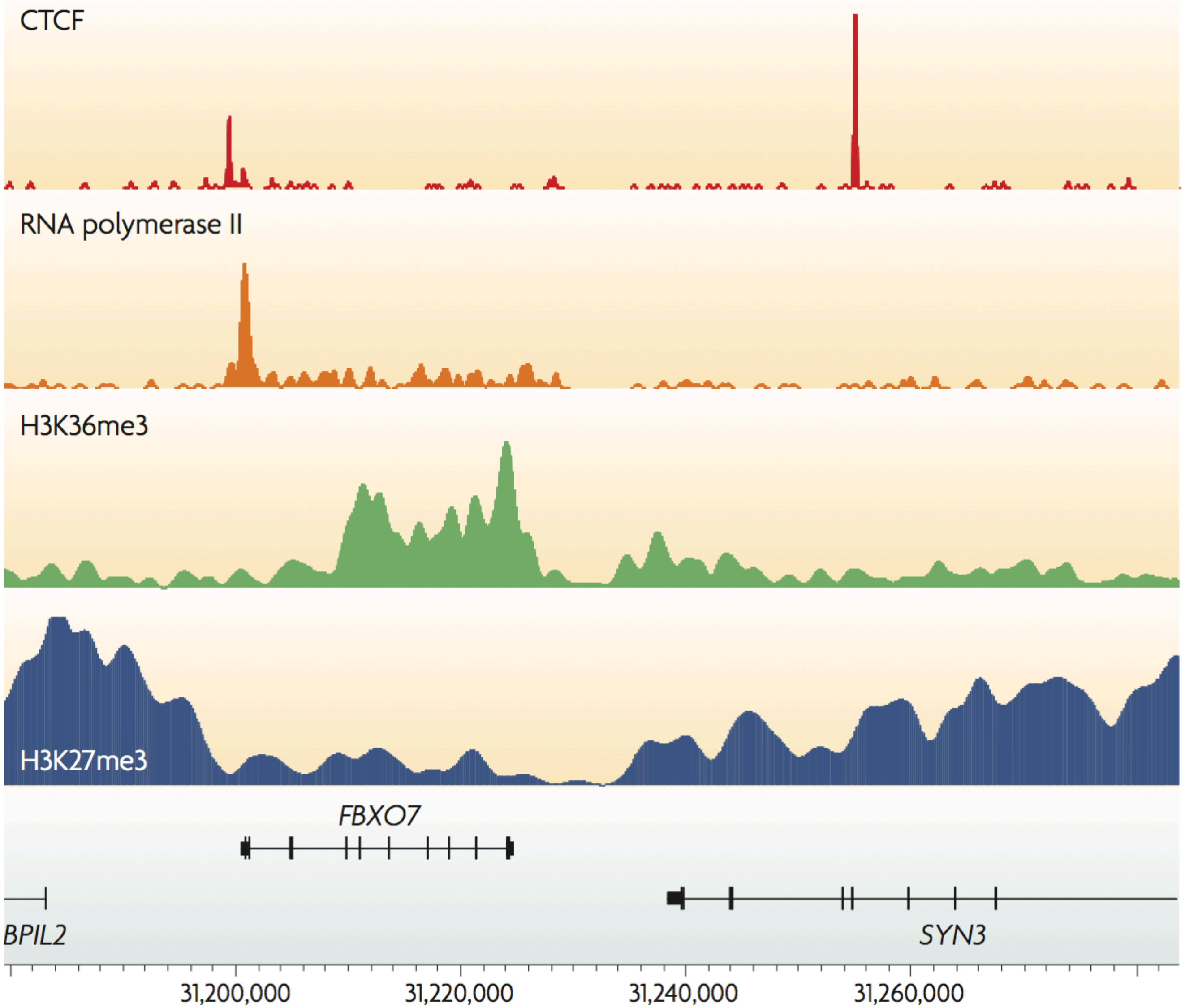
- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

Data resolution and ChIP vs Input



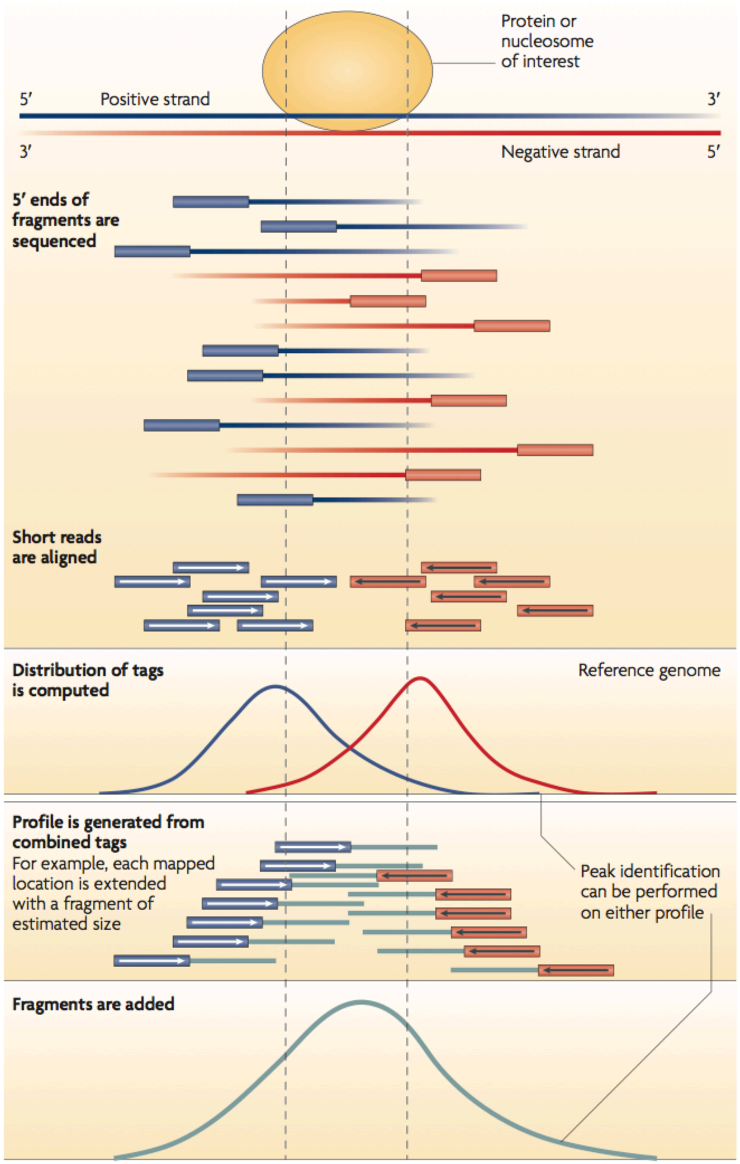
Park P, Nature Review Genetics 2009

Broad and sharp ChIP-seq signals



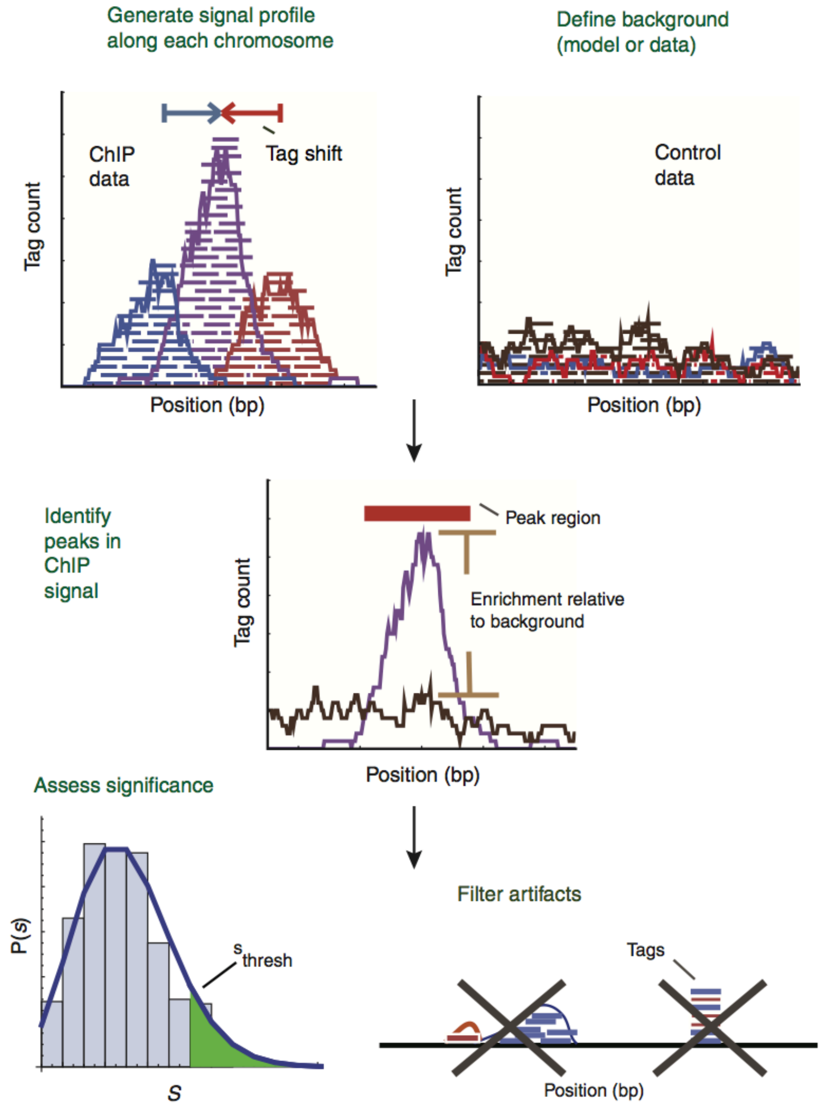
Park P, Nature Review Genetics 2009

ChIP-seq peaks finding


Protein or nucleosome of interest
5′
Positive strand
3′
3′
Negative strand
5′
5′ ends of fragments are sequenced
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

Park P, Nature Review Genetics 2009

ChIP-seq peaks finding

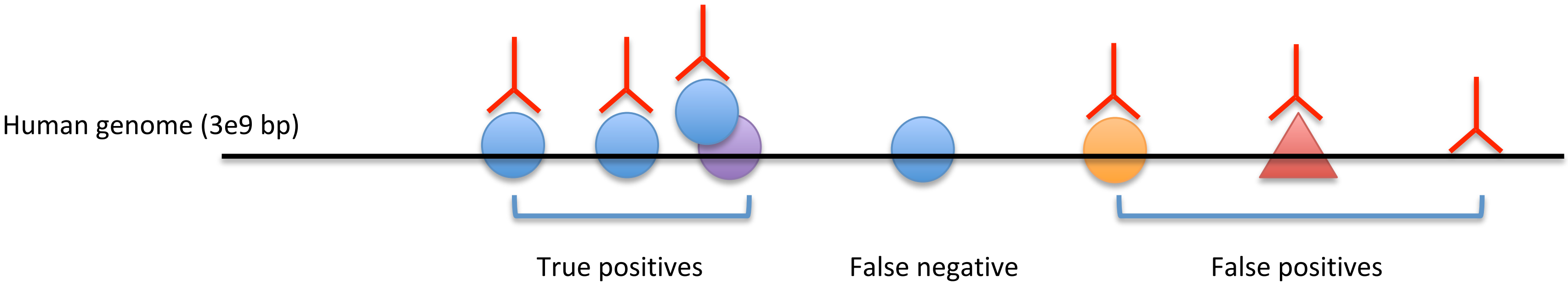


Pepke S and Mortazavi A, Nature Methods 2009

Outline of the presentation

- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

False positives and negatives



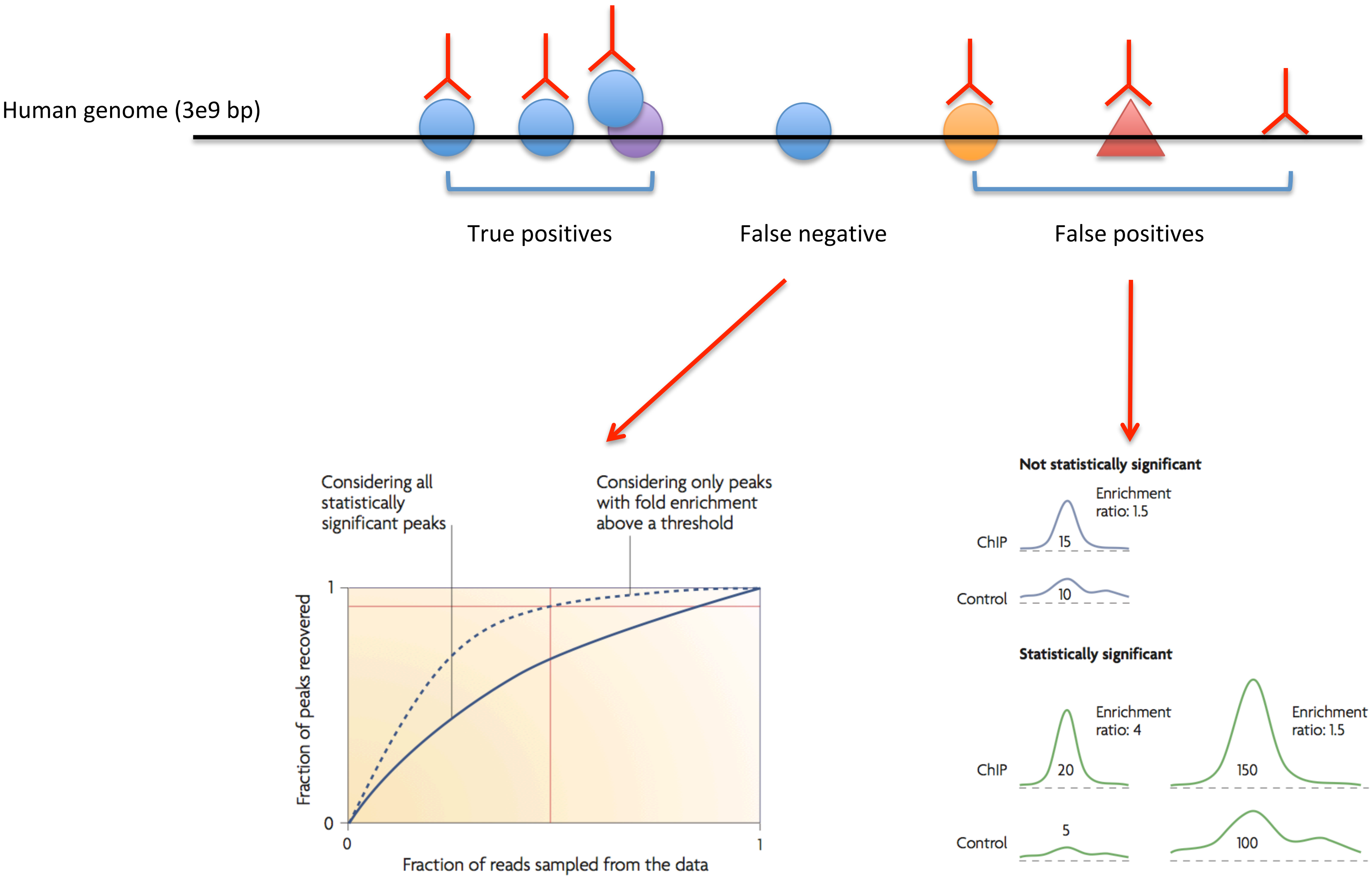
Human genome (3e9 bp)

True positives

False negative

False positives

False positives and negatives


Human genome (3e9 bp)
True positives
False negative
False positives
Considering all statistically significant peaks
Considering only peaks with fold enrichment above a threshold
1
0
Fraction of peaks recovered
0
1
Fraction of reads sampled from the data
Not statistically significant
Enrichment ratio: 1.5
ChIP
15
Control
10
Statistically significant
Enrichment ratio: 4
ChIP
20
Control
5
Enrichment ratio: 1.5
150
100

ChIP-seq peaks callers

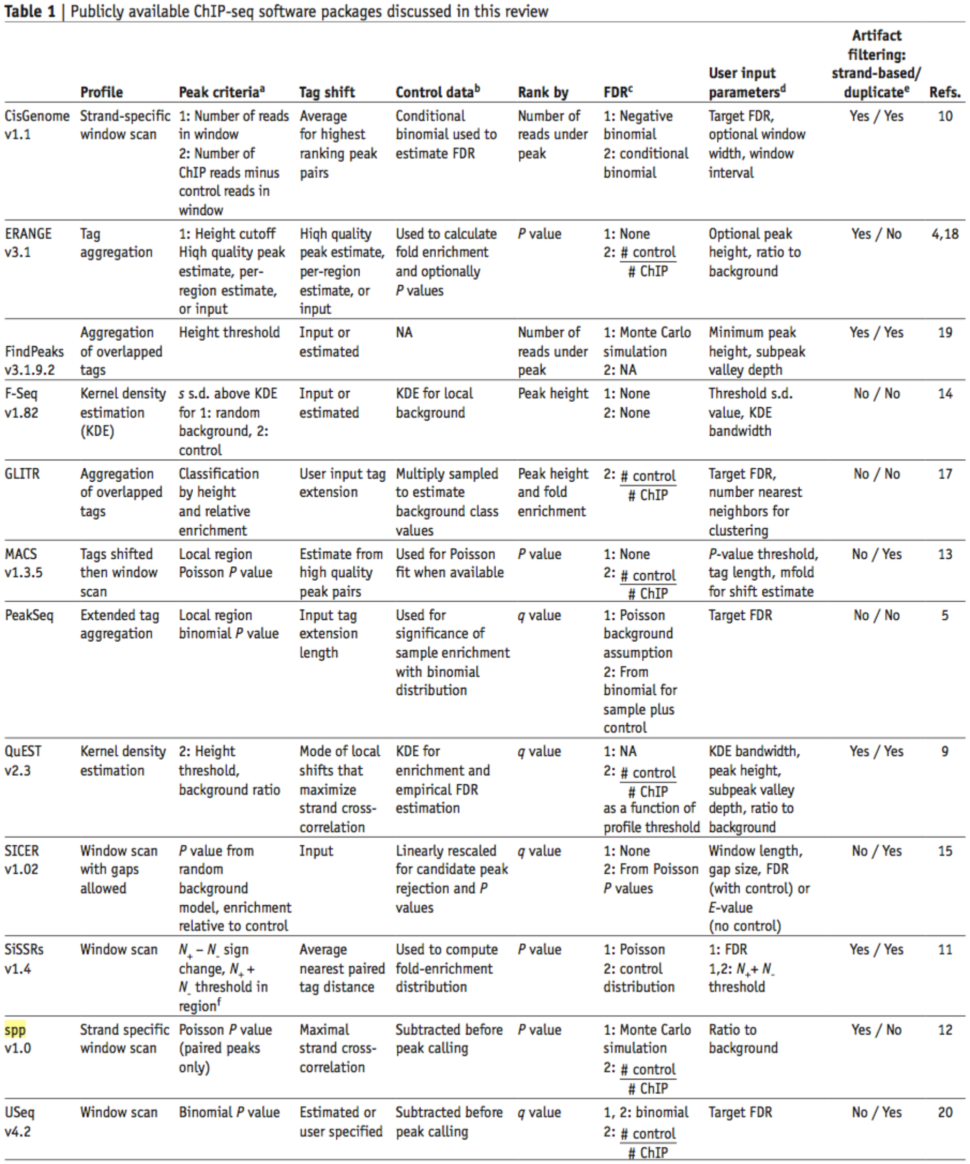


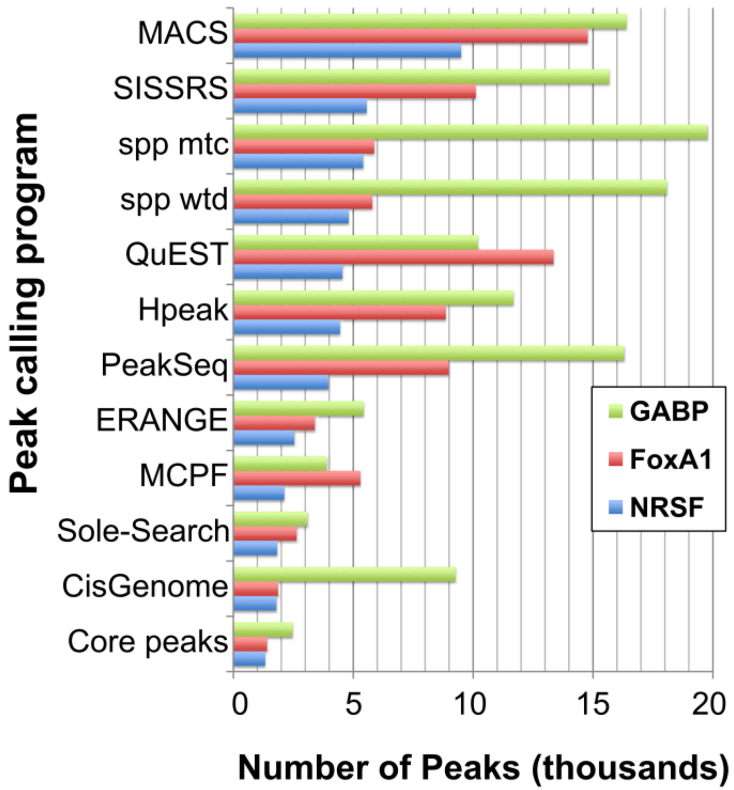
Table 1 | Publicly available ChIP-seq software packages discussed in this review

	Profile	Peak criteria[a]	Tag shift	Control data[b]	Rank by	FDR[c]	User input parameters[d]	Artifact filtering: strand-based/ duplicate[e]	Refs.
CisGenome v1.1	Strand-specific window scan	1: Number of reads in window 2: Number of ChIP reads minus control reads in window	Average for highest ranking peak pairs	Conditional binomial used to estimate FDR	Number of reads under peak	1: Negative binomial 2: conditional binomial	Target FDR, optional window width, window interval	Yes / Yes	10
ERANGE v3.1	Tag aggregation	1: Height cutoff High quality peak estimate, per-region estimate, or input	High quality peak estimate, per-region estimate, or input	Used to calculate fold enrichment and optionally *P* values	*P* value	1: None 2: $\frac{\#\ \text{control}}{\#\ \text{ChIP}}$	Optional peak height, ratio to background	Yes / No	4,18
FindPeaks v3.1.9.2	Aggregation of overlapped tags	Height threshold	Input or estimated	NA	Number of reads under peak	1: Monte Carlo simulation 2: NA	Minimum peak height, subpeak valley depth	Yes / Yes	19
F-Seq v1.82	Kernel density estimation (KDE)	*s* s.d. above KDE for 1: random background, 2: control	Input or estimated	KDE for local background	Peak height	1: None 2: None	Threshold s.d. value, KDE bandwidth	No / No	14
GLITR	Aggregation of overlapped tags	Classification by height and relative enrichment	User input tag extension	Multiply sampled to estimate background class values	Peak height and fold enrichment	2: $\frac{\#\ \text{control}}{\#\ \text{ChIP}}$	Target FDR, number nearest neighbors for clustering	No / No	17
MACS v1.3.5	Tags shifted then window scan	Local region Poisson *P* value	Estimate from high quality peak pairs	Used for Poisson fit when available	*P* value	1: None 2: $\frac{\#\ \text{control}}{\#\ \text{ChIP}}$	*P*-value threshold, tag length, mfold for shift estimate	No / Yes	13
PeakSeq	Extended tag aggregation	Local region binomial *P* value	Input tag extension length	Used for significance of sample enrichment with binomial distribution	*q* value	1: Poisson background assumption 2: From binomial for sample plus control	Target FDR	No / No	5
QuEST v2.3	Kernel density estimation	2: Height threshold, background ratio	Mode of local shifts that maximize strand cross-correlation	KDE for enrichment and empirical FDR estimation	*q* value	1: NA 2: $\frac{\#\ \text{control}}{\#\ \text{ChIP}}$ as a function of profile threshold	KDE bandwidth, peak height, subpeak valley depth, ratio to background	Yes / Yes	9
SICER v1.02	Window scan with gaps allowed	*P* value from random background model, enrichment relative to control	Input	Linearly rescaled for candidate peak rejection and *P* values	*q* value	1: None 2: From Poisson *P* values	Window length, gap size, FDR (with control) or *E*-value (no control)	No / Yes	15
SiSSRs v1.4	Window scan	$N_+ - N_-$ sign change, $N_+ + N_-$ threshold in region[f]	Average nearest paired tag distance	Used to compute fold-enrichment distribution	*P* value	1: Poisson 2: control distribution	1: FDR 1,2: $N_+ + N_-$ threshold	Yes / Yes	11
spp v1.0	Strand specific window scan	Poisson *P* value (paired peaks only)	Maximal strand cross-correlation	Subtracted before peak calling	*P* value	1: Monte Carlo simulation 2: $\frac{\#\ \text{control}}{\#\ \text{ChIP}}$	Ratio to background	Yes / No	12
USeq v4.2	Window scan	Binomial *P* value	Estimated or user specified	Subtracted before peak calling	*q* value	1, 2: binomial 2: $\frac{\#\ \text{control}}{\#\ \text{ChIP}}$	Target FDR	No / Yes	20

Pepke S and Mortazavi A, Nature Methods 2009

Comparing peak callers

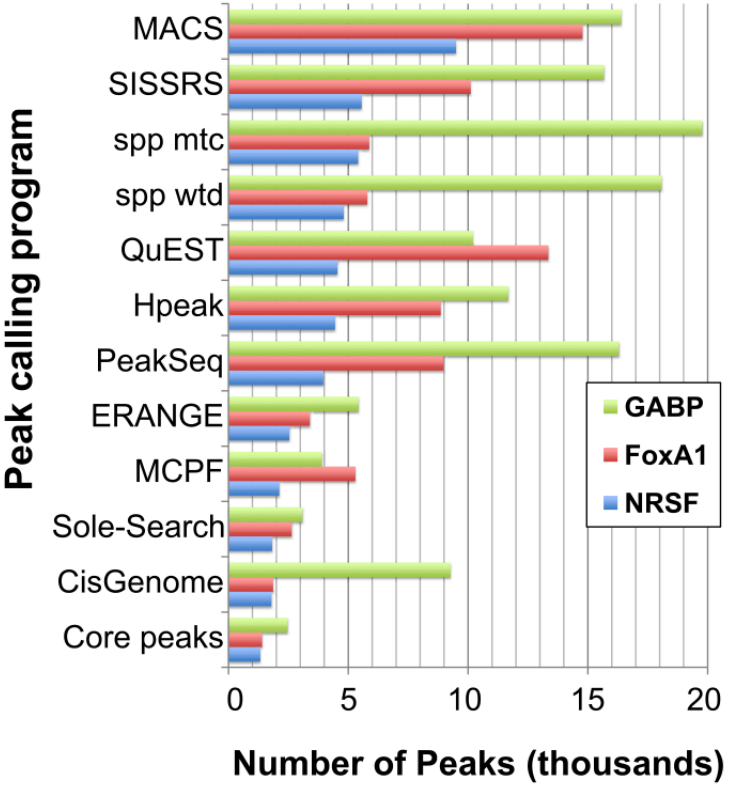
Number of peaks



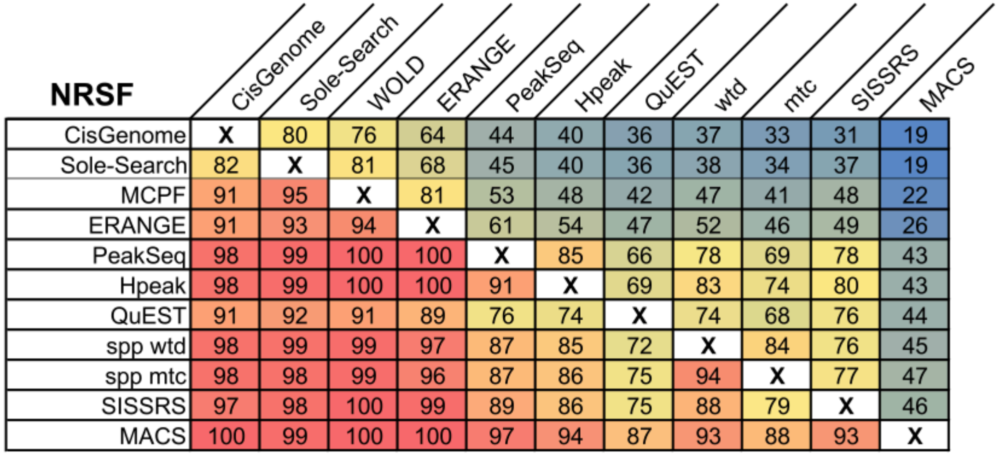
Wilbanks *et al*, Plos ONE 2010

Comparing peak callers

Number of peaks



Peaks overlap



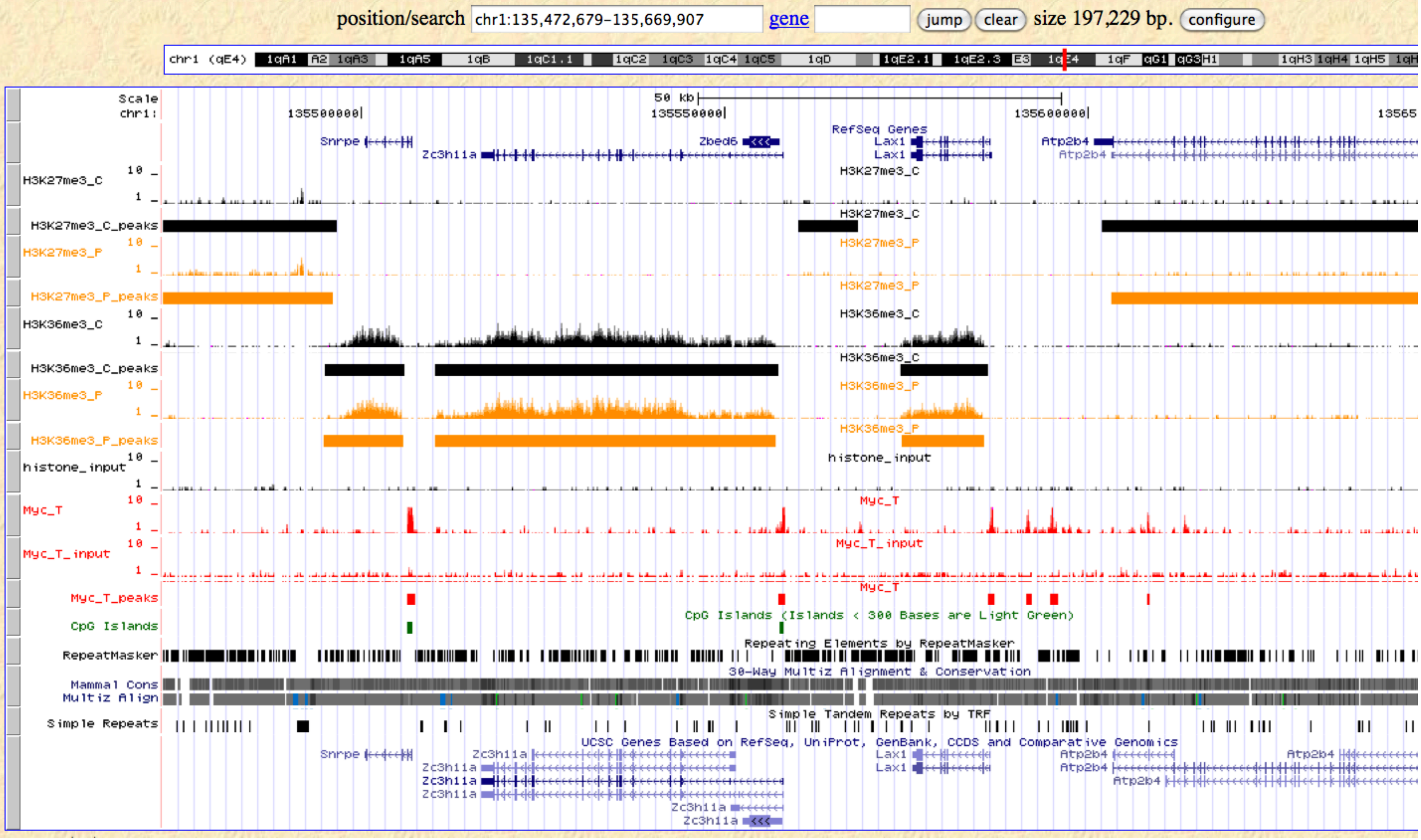
NRSF	CisGenome	Sole-Search	WOLD	ERANGE	PeakSeq	Hpeak	QuEST	wtd	mtc	SISSRS	MACS
CisGenome	X	80	76	64	44	40	36	37	33	31	19
Sole-Search	82	X	81	68	45	40	36	38	34	37	19
MCPF	91	95	X	81	53	48	42	47	41	48	22
ERANGE	91	93	94	X	61	54	47	52	46	49	26
PeakSeq	98	99	100	100	X	85	66	78	69	78	43
Hpeak	98	99	100	100	91	X	69	83	74	80	43
QuEST	91	92	91	89	76	74	X	74	68	76	44
spp wtd	98	99	99	97	87	85	72	X	84	76	45
spp mtc	98	98	99	96	87	86	75	94	X	77	47
SISSRS	97	98	100	99	89	86	75	88	79	X	46
MACS	100	99	100	100	97	94	87	93	88	93	X

Wilbanks *et al*, Plos ONE 2010

Outline of the presentation

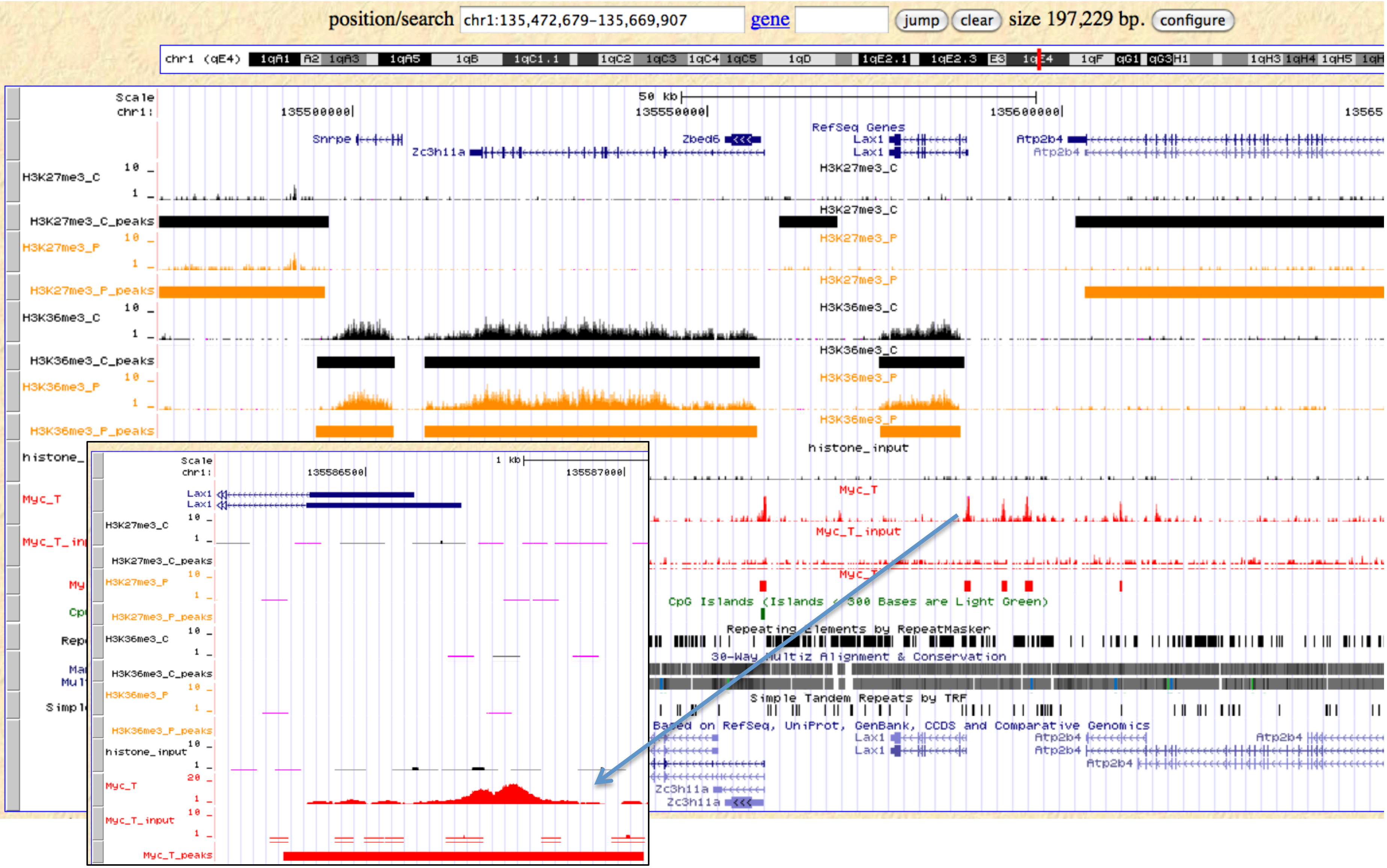
- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

Broad and sharp ChIP-seq signals



position/search chr1:135,472,679-135,669,907 gene jump clear size 197,229 bp. configure

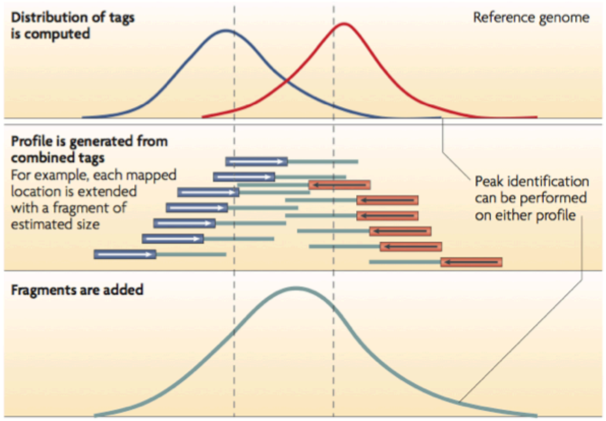
Broad and sharp ChIP-seq signals

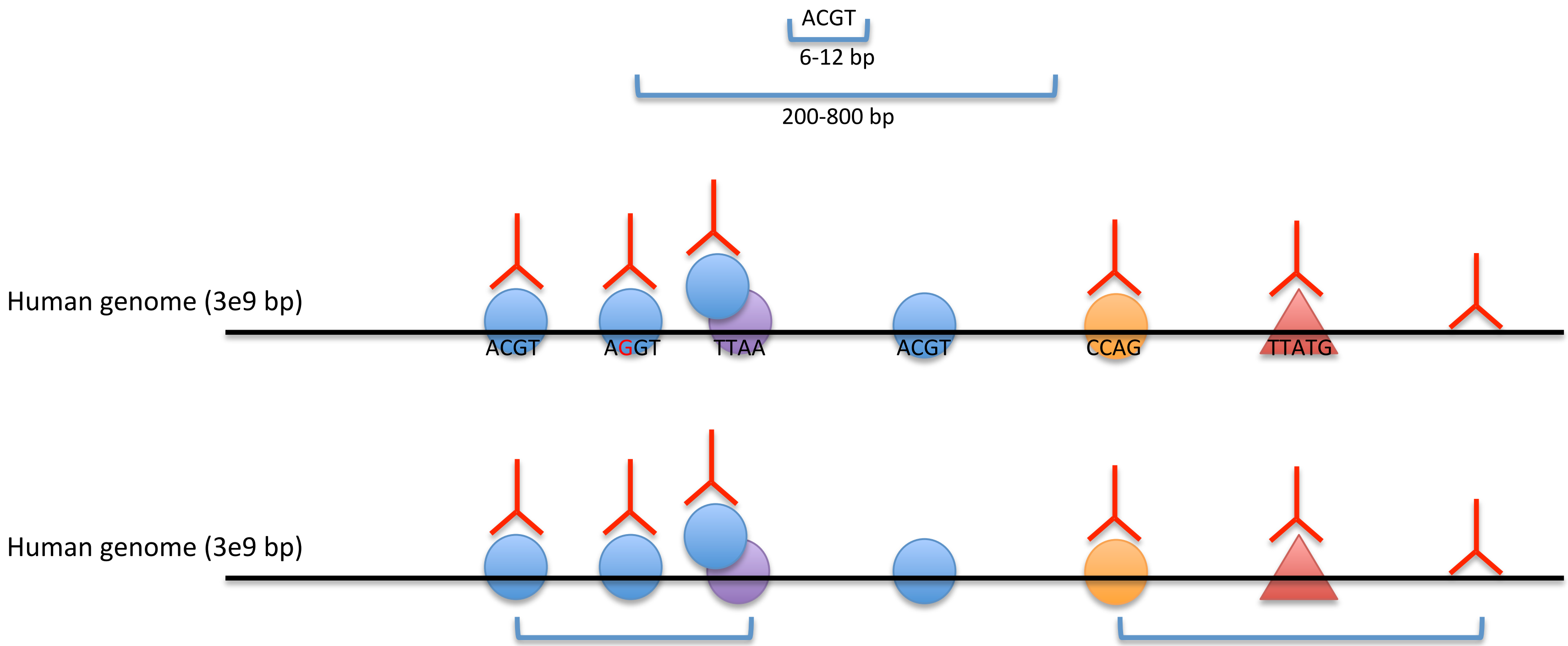


Outline of the presentation

- NGS technology
- NGS Computational workflows and data types
  - Sequencing reads: FASTQ
  - Reads alignments: SAM/BAM
  - Manipulating alignments and genomic intervals
- ChIP-seq background
- Peak calling
- Evaluating the quality of the results
- Visualizing the results on the genome browser
- Motif discovery

TF matching a specific DNA motif



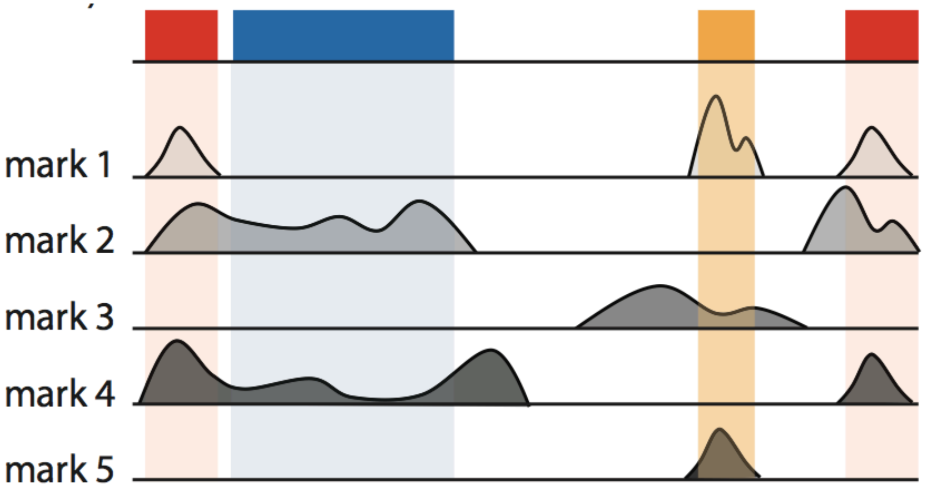


True positives

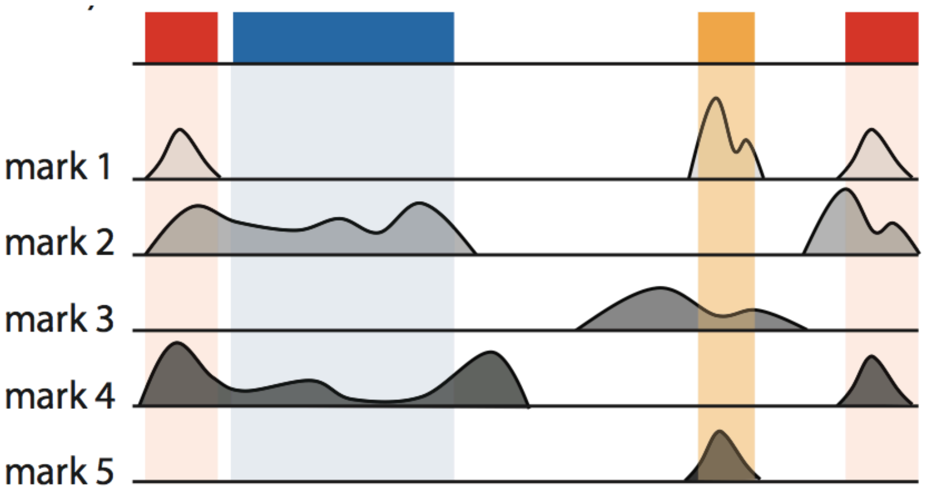
False negative

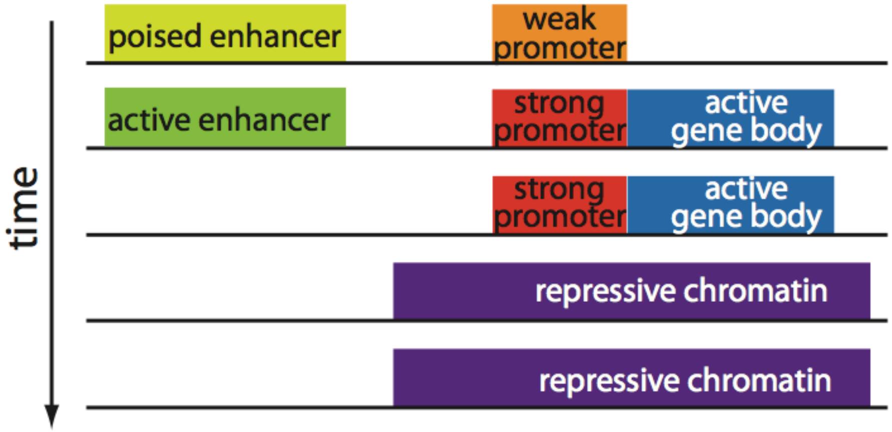
False positives

Identification of chromatin states



Identification of chromatin states


mark 1
mark 2
mark 3
mark 4
mark 5


poised enhancer
weak promoter
active enhancer
strong promoter
active gene body
strong promoter
active gene body
repressive chromatin
repressive chromatin
time