Introduction to Bioinformatics

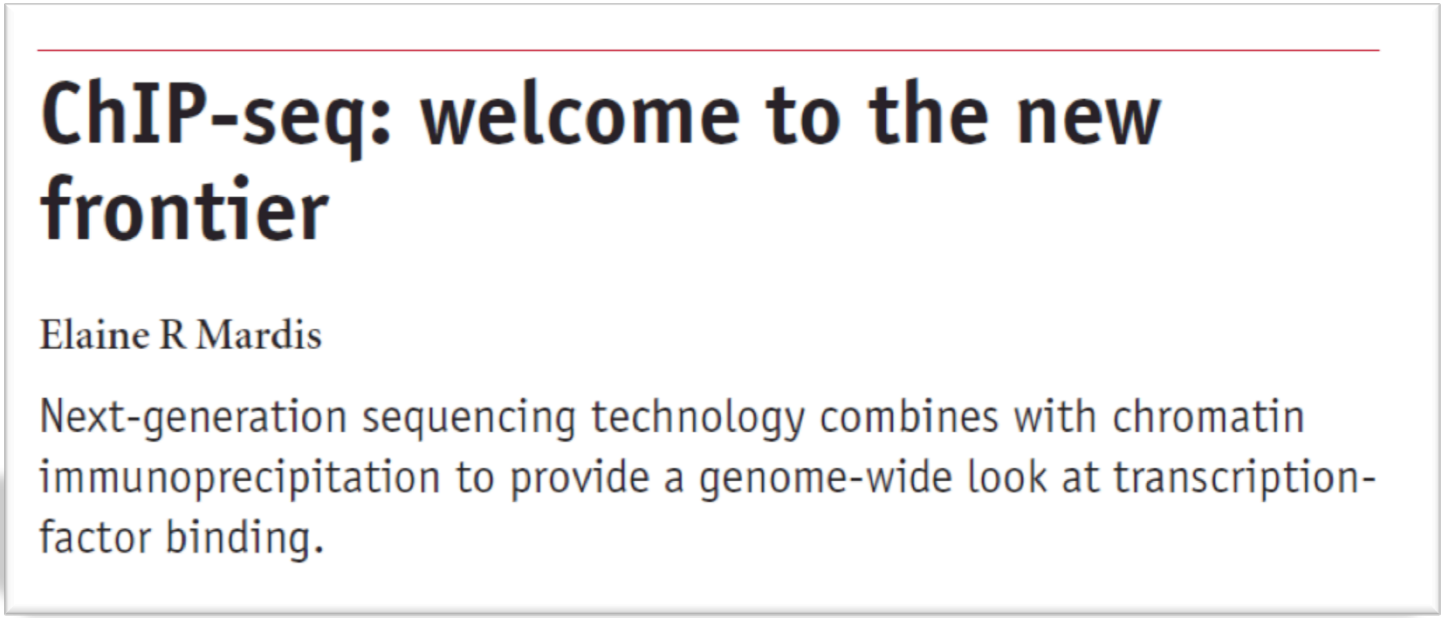
ChIP-data analysis

Lecturer: Jan Baumbach
Teaching assistant(s): Diogo Marinho

Question

"How can we study protein/DNA binding events on a genome-wide scale?"

ChIP-Sequencing



ChIP-seq: welcome to the new frontier

Elaine R Mardis

Next-generation sequencing technology combines with chromatin immunoprecipitation to provide a genome-wide look at transcription-factor binding.

[Nature methods -4, 613-614 (2007)]

- Emerging technology used to study protein/DNA interactions
- Facilitated by next-generation sequencing (NGS) methods
- Combination of ChIP technology with NGS

Quick Introduction to ChIP-Seq

Assumptions: Given a species with known genome. Further given a protein of interest (POI) that binds the DNA.

Questions: Where are its binding sites? What is the motif?

- Chemically fix all protein-DNA binding
- Isolate and fragment DNA with bound proteins
- Cross-link our POI (with bound DNA) to an antibody array
- Remove remaining (unbound) DNA
- Release POI-DNA binding, i.e. release POI-bound DNA for sequencing

Quick Introduction to ChIP-Seq

Assumptions: Given a species with known genome. Further given a protein of interest (POI) that binds the DNA.

Questions: Where are its binding sites? What is the motif?

- Sequence the DNA fragments
- Map DNA fragments to reference genome
- Identify binding site
- Compute sequence motif

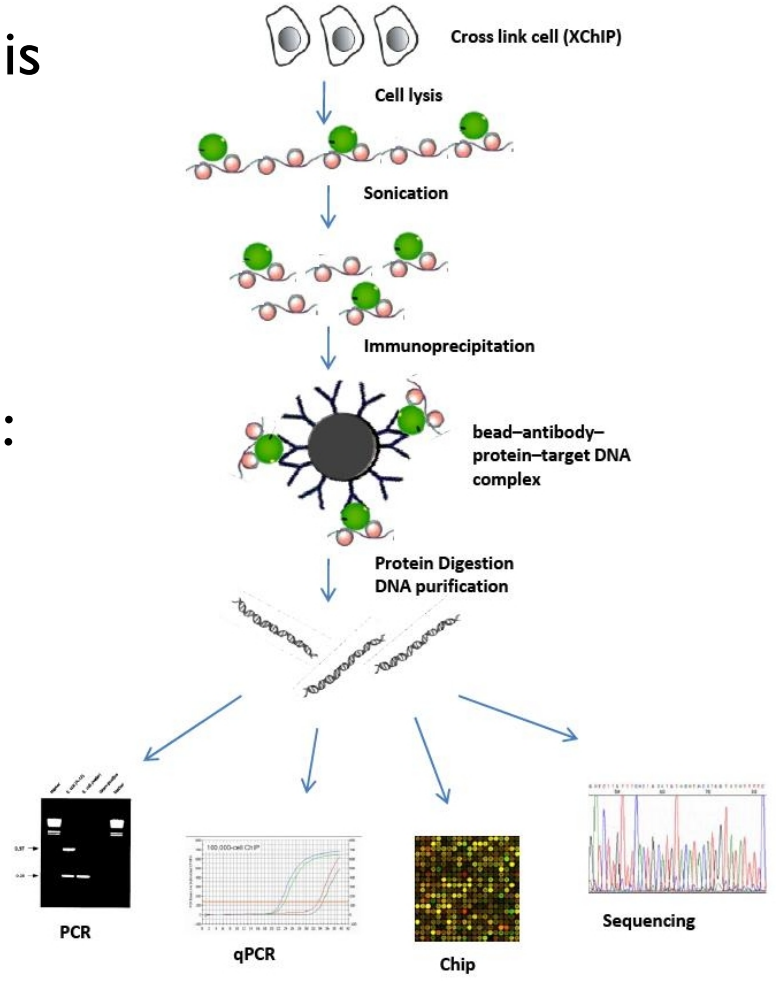
Development

Chromatin immunoprecipitation (ChIP) is well established technology/method (dates back to 1988)

ChIP is part of many research protocols:

- ChIP-chip: micro arrays
- ChIP-PCR: quantitative real-time PCR

ChIP provides a basic building block. With the advances of NGS technology, a new combination appears: **ChIP-seq**



Benefits/Advantages of ChIP-Seq

Table 1 | **Comparison of ChIP–chip and ChIP–seq**

	ChIP–chip	ChIP–seq
Maximum resolution	Array-specific, generally 30–100 bp	Single nucleotide
Coverage	Limited by sequences on the array; repetitive regions are usually masked out	Limited only by alignability of reads to the genome; increases with read length; many repetitive regions can be covered
Cost	US$400–800 per array (1–6 million probes); multiple arrays may be needed for large genomes	Currently US$1,000–2,000 per lane (using the Illumina Genome Analyzer); 6–15 million reads before alignment
Source of platform noise	Cross-hybridization between probes and nonspecific targets	Some GC bias can be present
Experimental design	Single- or double-channel, depending on the platform	Single channel
Cost-effective cases	Profiling of selected regions; when a large fraction of the genome is enriched for the modification or protein of interest (broad binding)	Large genomes; when a small fraction of the genome is enriched for the modification or protein of interest (sharp binding)
Required amount of ChIP DNA	High (a few micrograms)	Low (10–50 ng)
Dynamic range	Lower detection limit; saturation at high signal	Not limited
Amplification	More required	Less required; single-molecule sequencing without amplification is available
Multiplexing	Not possible	Possible

[Peter J Park - ChIP-seq: advantages and challenges of a maturing technology - Nature Review 10:2009]

Benefits/Advantages of ChIP-Seq

Table 1 | **Comparison of ChIP–chip and ChIP–seq**

	ChIP–chip	ChIP–seq
Maximum resolution	Array-specific, generally 30–100 bp	Single nucleotide
Coverage	Limited by sequences on the array; repetitive regions are usually masked out	Limited only by alignability of reads to the genome; increases with read length; many repetitive regions can be covered
Cost	US$400–800 per array (1–6 million probes); multiple arrays may be needed for large genomes	Currently US$1,000–2,000 per lane (using the Illumina Genome Analyzer); 6–15 million reads before alignment
Source of platform noise	Cross-hybridization between probes and nonspecific targets	Some GC bias can be present
Experimental design	Single- or double-channel, depending on the platform	Single channel
Cost-effective cases	Profiling of selected regions; when a large fraction of the genome is enriched for the modification or protein of interest (broad binding)	Large genomes; when a small fraction of the genome is enriched for the modification or protein of interest (sharp binding)
Required amount of ChIP DNA	High (a few micrograms)	Low (10–50 ng)
Dynamic range	Lower detection limit; saturation at high signal	Not limited
Amplification	More required	Less required; single-molecule sequencing without amplification is available
Multiplexing	Not possible	Possible

[Peter J Park - ChIP-seq: advantages and challenges of a maturing technology - Nature Review 10:2009]

When can we use ChIP-seq?

- Genome of species must be known.
- A protein-specific antibody is required for cross-linking.
- Access to next-generation sequencing (NGS) technology (Illumina/Roche454/SOLiD) – It is getting incredibly cheap.

ChIP-Seq Workflow

- Protein interaction with DNA

(e.g. TF or Histone) of interest

- Other DNA-interacting proteins

(we are not interested in)

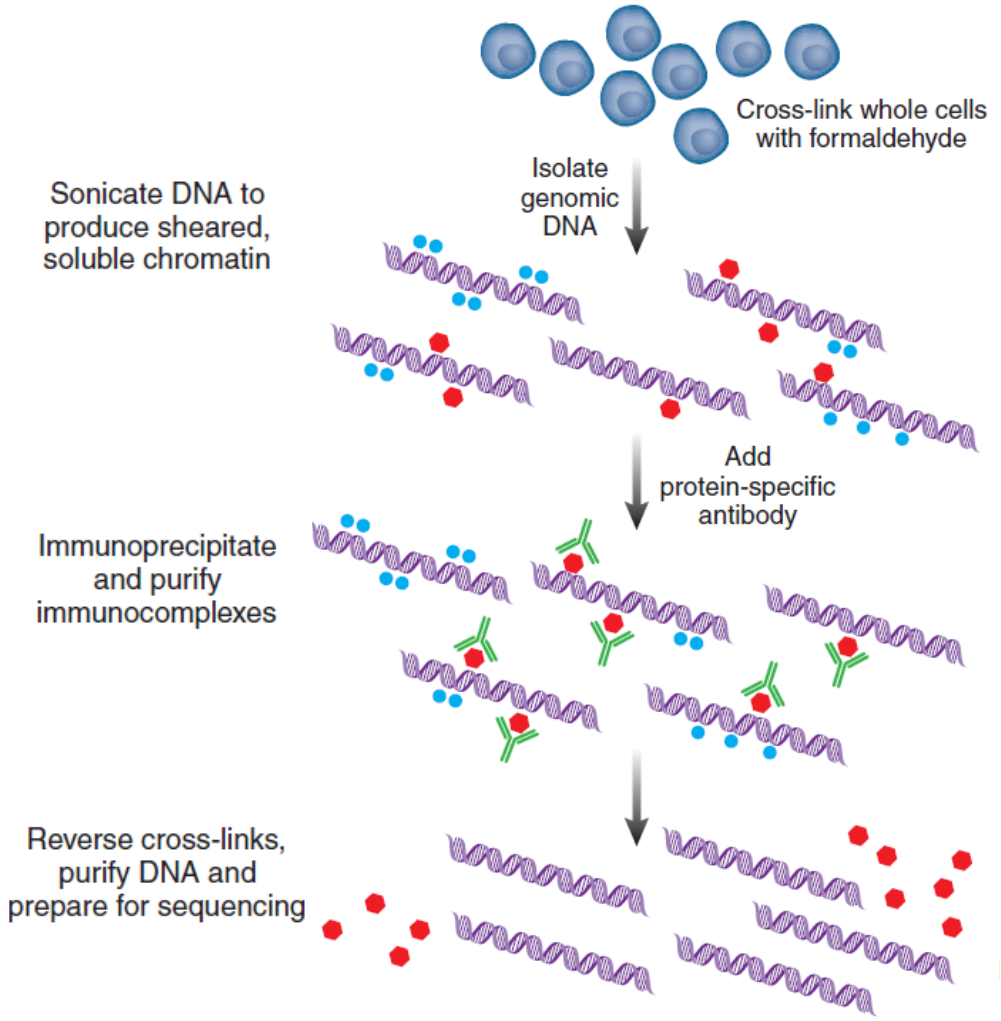
- Protein specific antibody
- Fragmented DNA (should have

uniform length between 200-600bp

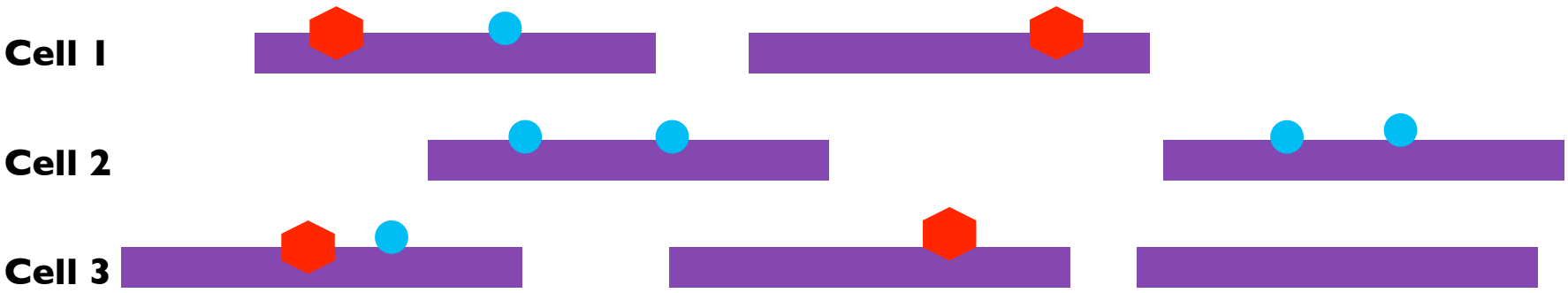
prior to immunoprecipitation)

Part 1: ChIP

Part 2: Sequencing

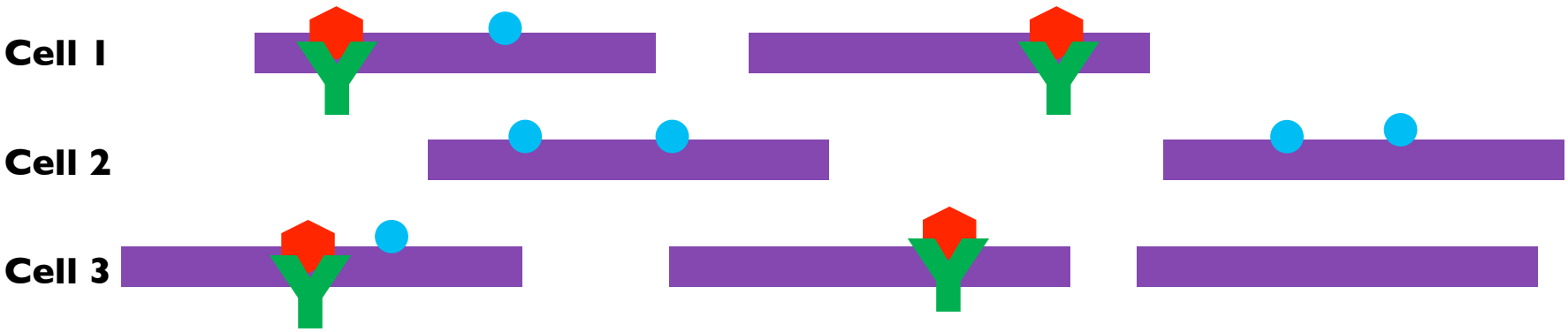


Part 1: ChIP



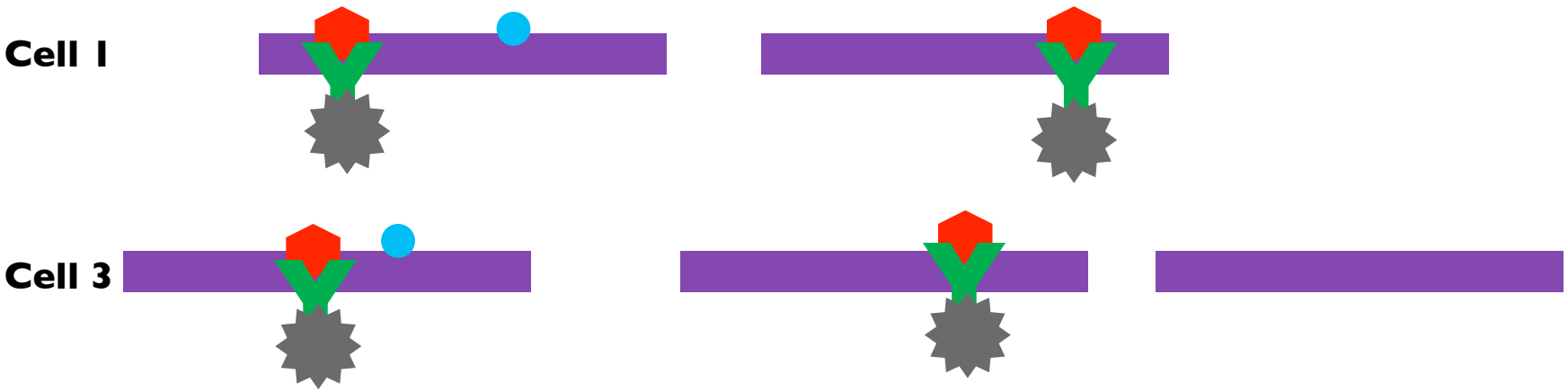
(1) Purify DNA from cross-linked cells. Sheer DNA in order to obtain fragments of size between 200 and 600bp (e.g. by sonication)

Part 1: ChIP



(2) Perform immunoprecipitation (IP): Antibodies bind to protein of interest.

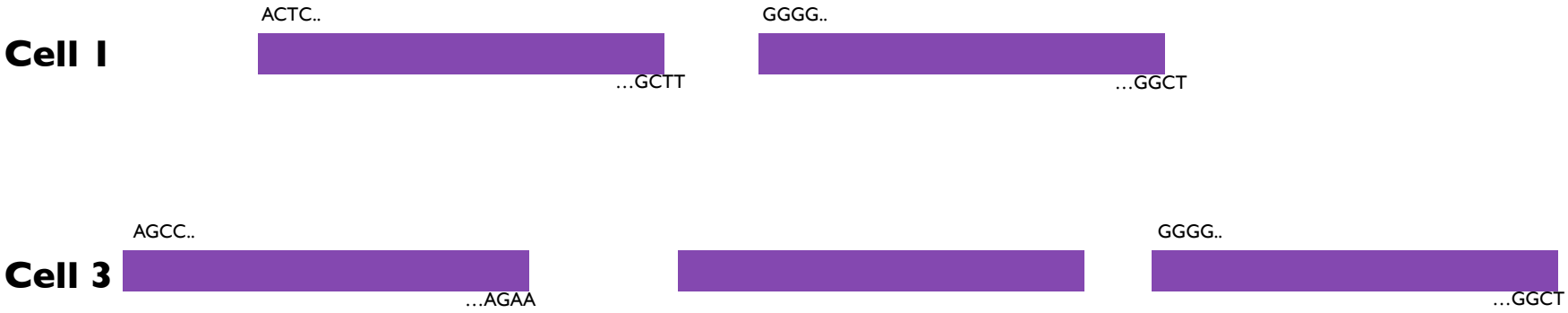
Part 1: ChIP – Purification 1



(3) During IP antibodies get linked to surface (matrix) using e.g. magnetic beads. Physical fixation allows to separate DNA/ protein/antibody complex from the rest.

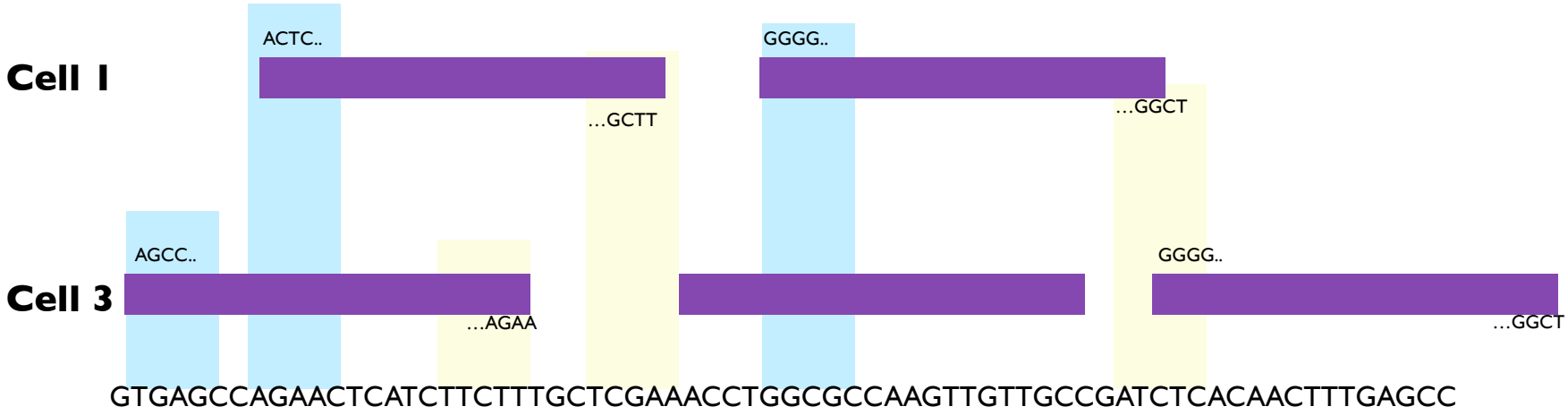
Part 1: ChIP – Purification 2

Part 2: Sequencing



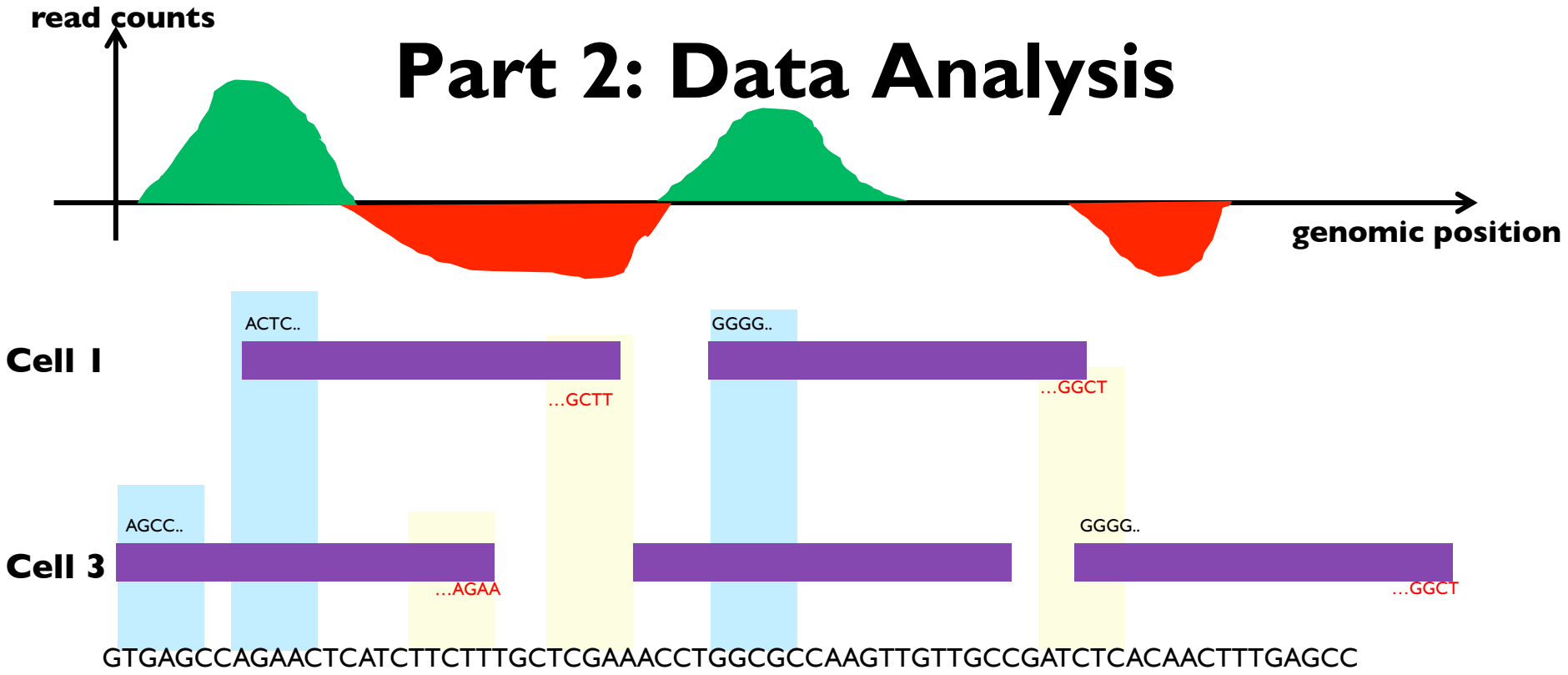
(5) Each fragment is consecutively sequenced using NGS technology. In this example we get **paired-end reads** as we sequence every fragment from both ends.

Part 2: Short-read Mapping



(6) Every sequenced fragment is mapped to the genome. Non-unique mappings and low quality reads can be discarded.

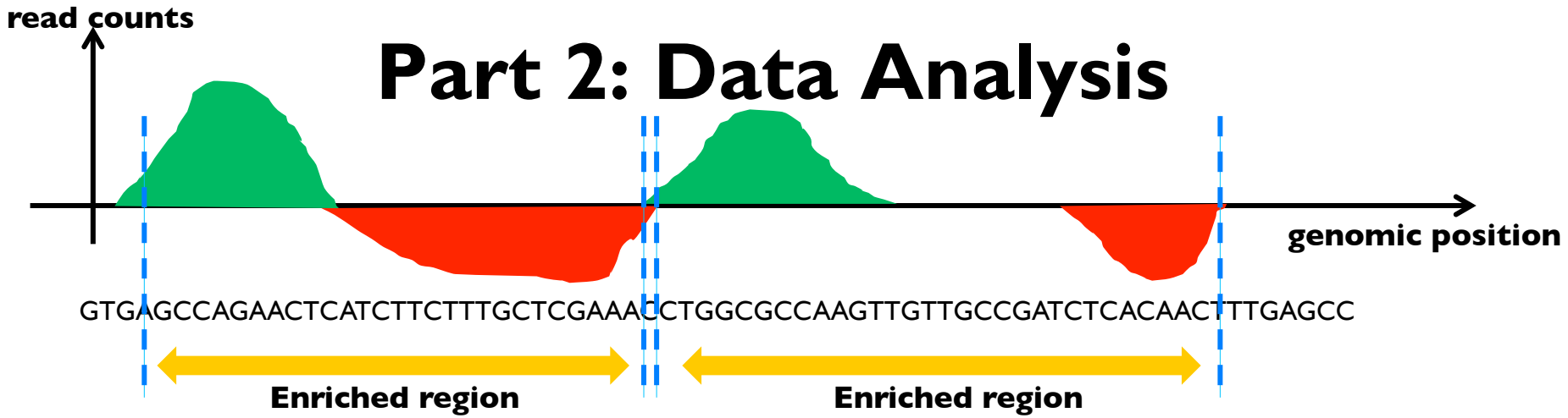
Part 2: Data Analysis



Build histogram (signal map) for forward- and reverse reads for all uniquely mapped reads, i.e.: Count how often each genomic position is covered by a short sequence read.

Assumption: Binding sites should be somewhere between corresponding peaks on the forward- and reverse strand in the histogram.

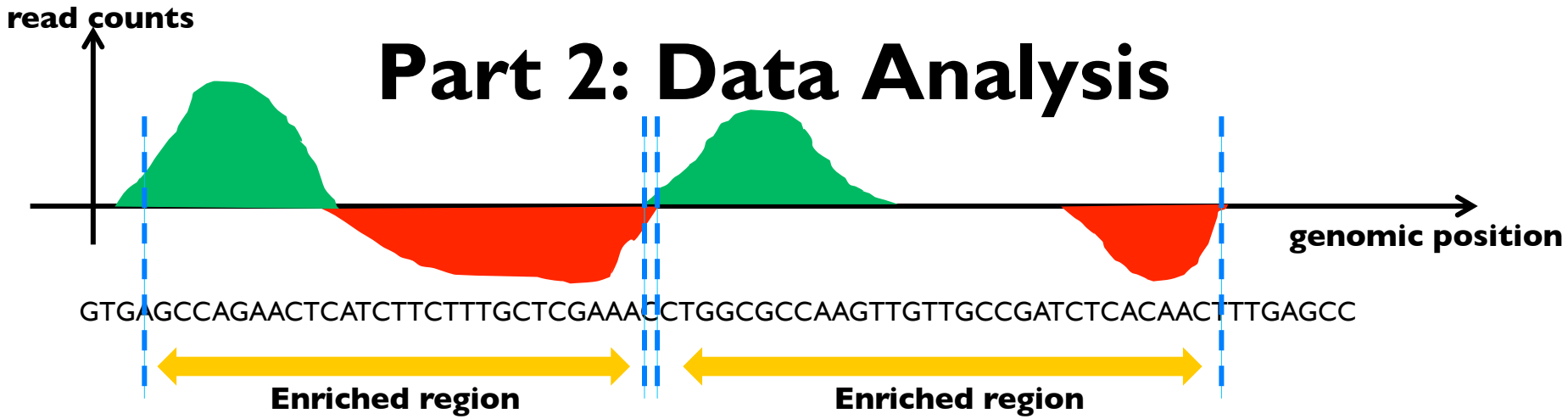
Part 2: Data Analysis



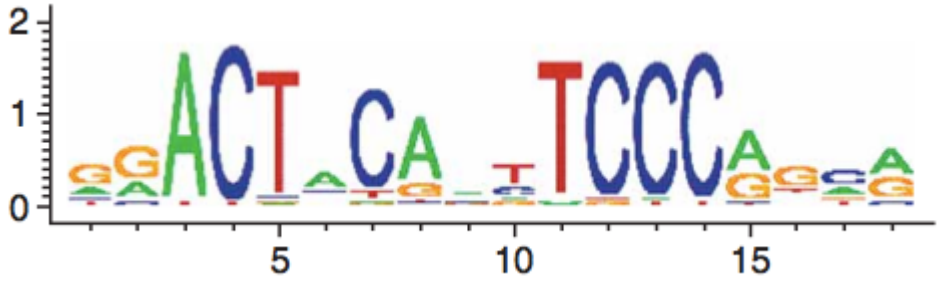
Data shows "bi-modal enrichment pattern": A peak on forward strand is followed by similar peak on the reverse strand.

Shift size between peaks is an experimental parameter (~100-250b)

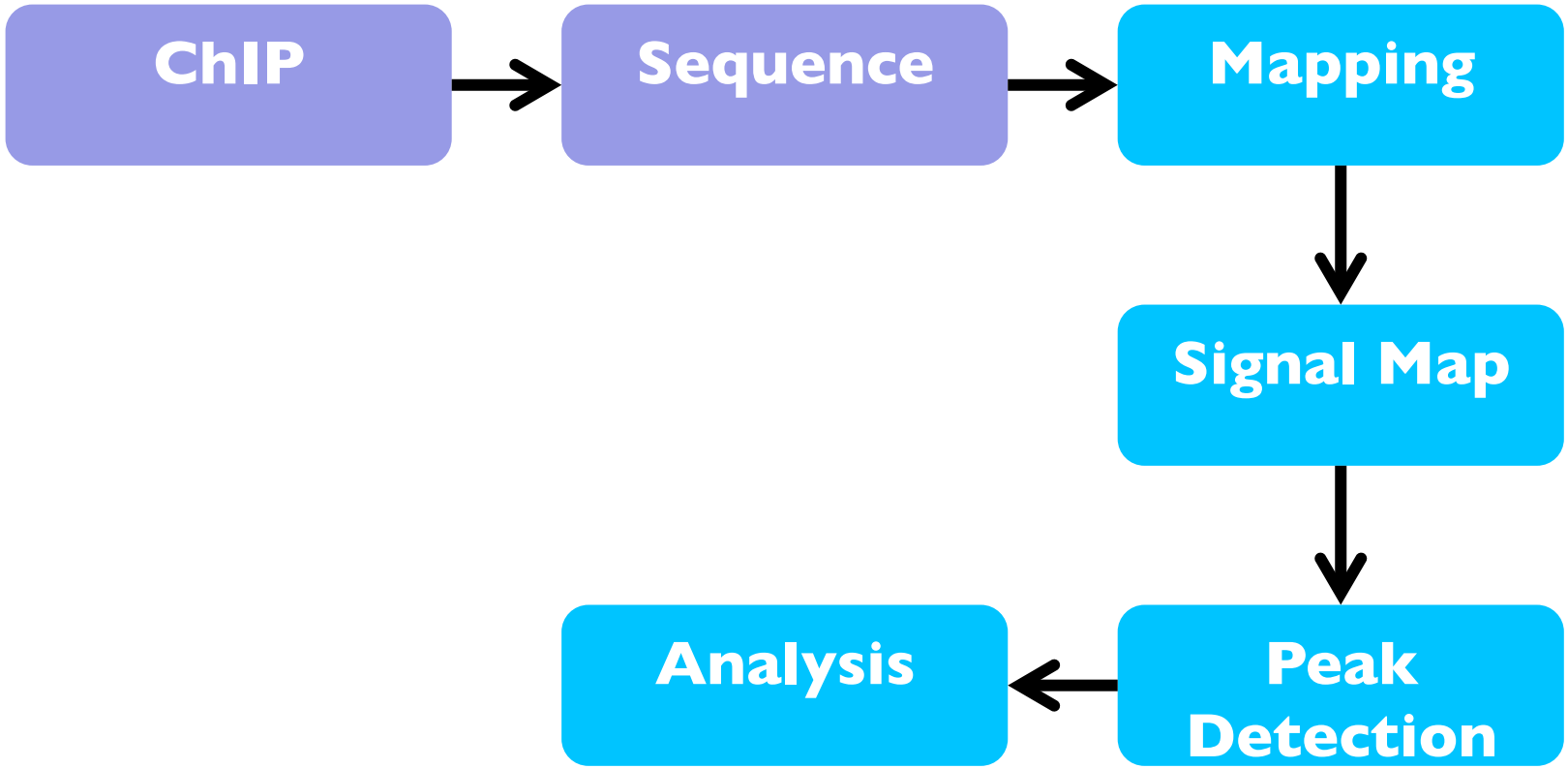
Part 2: Data Analysis



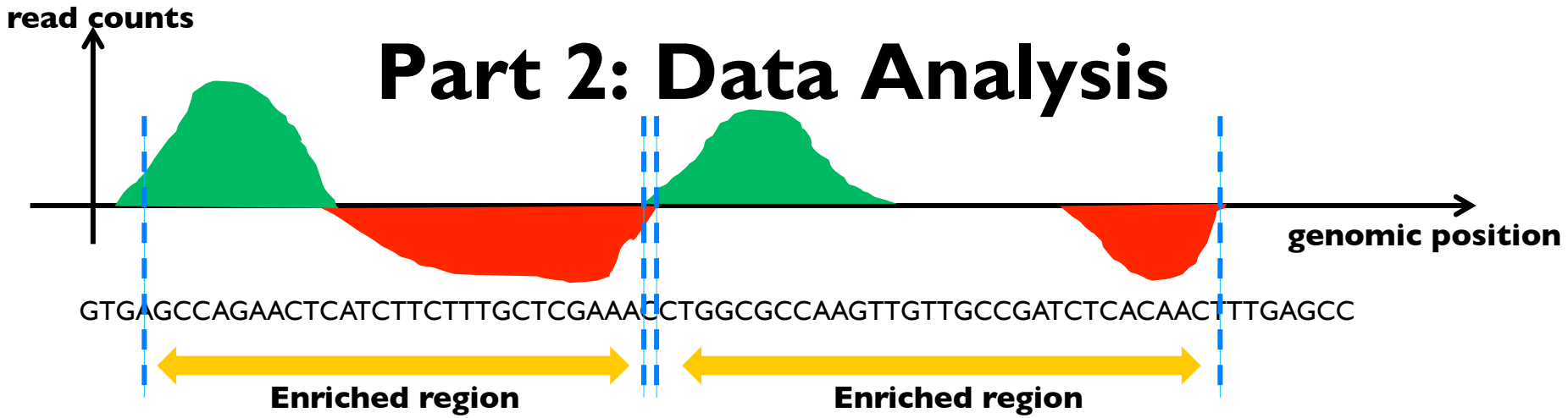
Apply your favorite motif discovery tool (e.g. MEME) and identify conserved motif.



The Pipeline:



Part 2: Data Analysis



That is basically how ChIP-seq works! **But:**

In real-life applications you always perform two experiments

(1) Experiment as explained
(2) Control experiment where you sequence everything, i.e. no immunoprecipitation (ChIP) step

→ You get two signal maps. Do differential analysis on them!

Bioinformatics Challenges

(1) Sequencing:

- Rapid mapping of short-sequence reads (possibly paired end) to a genome. (using >10million reads).

(2) Peak Calling:

- How to identify the peaks?
- Find the exact location of the binding site.

(3) Data visualization

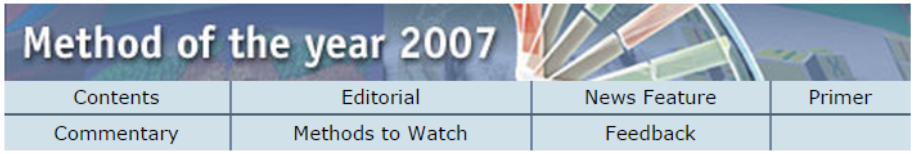
- How to visualize thousands of binding sites?

→ We only focus on (1) and (2)

Next Generation Sequencing

- A single NGS reads up-to 55billion bp/day
- Run over 10 days gives = 600 Gigabases
- Total data: 1.1 Tb data (fastq)

- Operated by single person



Nature Methods - **5**, 11 - 14 (2008)
doi:10.1038/nmeth1154

The year of sequencing

Next Generation Sequencing

NGS	Sequencing technology	Average fragment read length	Throughput
Solexa (Illumina)	Synthesis with reversible terminators	30–100b	55Gb / hour
454 (Roche)	Pyrosequencing	400–1000b	1million 400b reads / 10hours
SOLiD (ABS)	Capillary Electrophoresis	60–75b (paired-end)	20–30Gb / day

File Format

NGS output usually contains the data fields:

someID, seq-read, quality score, (direction)

Downside: There exist as many file formats & subtle variations as sequencing machines and manufactures. see: http://emboss.sourceforge.net/docs/themes/SequenceFormats.html

- Most widely used: FASTQ files (with variation for Solexa/Illumina)

→ Better master your favorite scripting language, bioperl/biopython, GNU tools for fast conversion

→ NGS related question: ask/search www.seqanswers.com/

Sample: FASTQ file

```
@HWUSI-EAS729_615HF:1:1:1025:3361#0/1
NAGCACAAAGGGAATGATTTTACGGTAGGCGTAGGAAAAGTGGCCCCATTTTAAAAACACGTCTCAACGACTCCTAGTGCTCTAATAACCGGGACAAGGA
+HWUSI-EAS729_615HF:1:1:1025:3361#0/1
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWUSI-EAS729_615HF:1:1:1025:11153#0/1
NGCGGAGGCGTATGCGGAAGCATCGGAAGCAGCACGCAAGGCTTCATCCTCAAGGGTTTCAATGTTTTCTGACGGCGCCCACACGATATTGGCTAGGGCT
+HWUSI-EAS729_615HF:1:1:1025:11153#0/1
BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWUSI-EAS729_615HF:1:1:1025:7786#0/1
NTGGTGAGCTCCGAATCCTCGAAGCCATTGGTTGCGATAACTGCTACTTTTCGGTTGCTTAGTTTTGCCATTGTTAATCCTCCTTGCTTGAGACTTTTCG
+HWUSI-EAS729_615HF:1:1:1025:7786#0/1
BJIIJHHIIL```W`WWWWWVVTUWW```O`WSWWWVQWVOVWWWWW`````T``````````T``JJKGKTTTRQT`U``````````````L````U``
@HWUSI-EAS729_615HF:1:1:1026:4614#0/1
NACAATCGCTGCCGCTGCCTCTAAGAATGCATCATTTTCCGAGGGCAGCCCAATGGTGGTTCGCAAGAACCCAGAAATTCCTACATCGCGGATGAGCACA
+HWUSI-EAS729_615HF:1:1:1026:4614#0/1
BJJGHGJMGGWYWYV``````````````V`WUWYY`VV``WQWKT``````````BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
@HWUSI-EAS729_615HF:1:1:1026:2845#0/1
NATTTTCGGTATCCCAGATGACGATGCCAAACATGCCGCGAAGATGATTGACTACATCCTTGCCCCAGTGGTGATAGCCCACGGCAATTGGTTCGCCGTC
+HWUSI-EAS729_615HF:1:1:1026:2845#0/1
BHHHFIIFFF`````T``T````````````````U```U`BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

Example of fastq file format. Each entry consist of four consecutive lines:

(1) Some ID (2) Sequence (3) Same ID (4) Phred quality scores per base (encoded as char per base)

Read Mapping

The next step is to align the short (paired-end) reads against the whole genome. Task known as **mapping.**

Data consists of millions of reads → usual approaches like dynamic programming will fail!

A mapping algorithm has to deal with:

- Mismatches induced by mutations and sequencing errors
- (Memory)efficient data-structures, hashing, compression, parallelization

Tons of tools are available:

http://seqanswers.com/wiki/Special:BrowseData/Bioinformatics%20application?Bioinformatics_method=Mapping

Mapping Software: Bowtie

We will have a closer look at the BOWTIE software

Genome Biol. 2009;10(3):R25. doi: 10.1186/gb-2009-10-3-r25. Epub 2009 Mar 4.

Ultrafast and memory-efficient alignment of short DNA sequences to the human genome.

Langmead B[1], Trapnell C, Pop M, Salzberg SL.

[1]Center for Bioinformatics and Computational Biology, Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA. langmead@cs.umd.edu

http://bowtie-bio.sourceforge.net/index.shtml

http://genomebiology.com/content/10/3/R25

- Small memory footprint: less than 3Gb while mapping to the human genome.
- Aligns up-to 30 million 35bp reads per hour.

Mapping Software: Bowtie

Bowtie uses the Burrows-Wheeler transform (BWT) to reduce the memory footprint of the genome index

- BWT is a lossless text compression algorithm

[Burrows M and Wheeler D (1994), A block sorting lossless data compression algorithm, Technical Report 124, Digital quipment Corporation]

BWT(T):

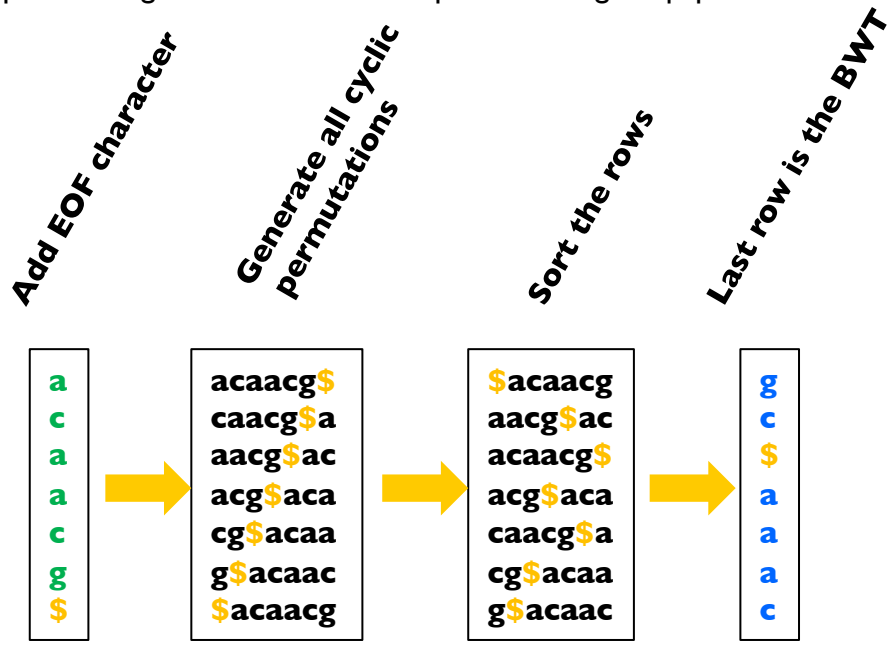
(1) Add additional EOF character ($)

(2) Generate all cyclic permutations of the input string T

(3) Sort the columns in lexicographic order

(4) The last row is the transform BWT(T)

- BWT(T) easy to compress (run-length coding)
- Inverse exists: Construct input from last column



Mapping Software: Inverse BWT

The BWT maintains the following invariants:

(1) Last-first (LF) mapping: The ith occurrence of any character in the BWT (last row in matrix) corresponds to the ith occurrence of the same character in the first row of the matrix (which is simply the sorted input string) The function LFc(i) maps the ith occurrence of character c in the BWT to the corresponding position of the character c in the first row.

(2) Any character in position i of the first row is the successor of the corresponding character at position i from the last row (the BWT) in the original text. (E.g. the first character in BWT is the last character in the original text - since it is followed by "$" in the first row)

To recreate T from BWT(T), start with *i* = 0 and T = BWT [0] and repeatedly apply rule 2:

$$T = BWT\,[LF(i)] + T\,; i = LF(i)$$

Where LF(*i*) maps row *i* to the row whose first character corresponds to *i*'s last character according to the LF mapping:

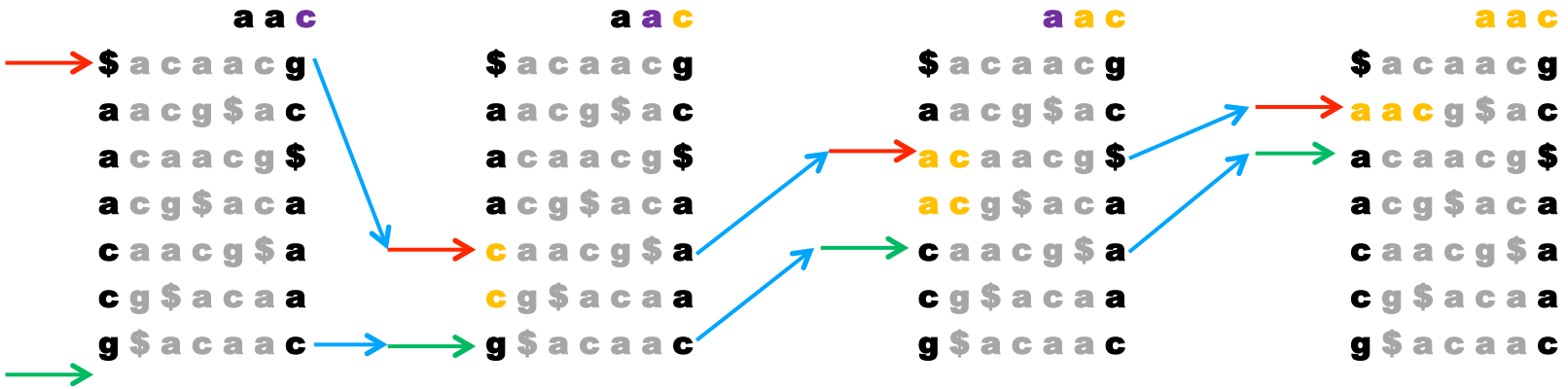
Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc



In progressive rounds, top/bot delimit rows beginning with progressively longer suffixes of Q

Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc

In progressive rounds, top/bot delimit rows beginning with progressively longer suffixes of Q

Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc

Find c in left side. Mark first and last occurrence of c with top and bot .

Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc

If c is preceded by an a, there must be an a in range (!) at the right side (because of the cycle permutation).

Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc

Find the first and last occurrence of a in range (!) on the right side.

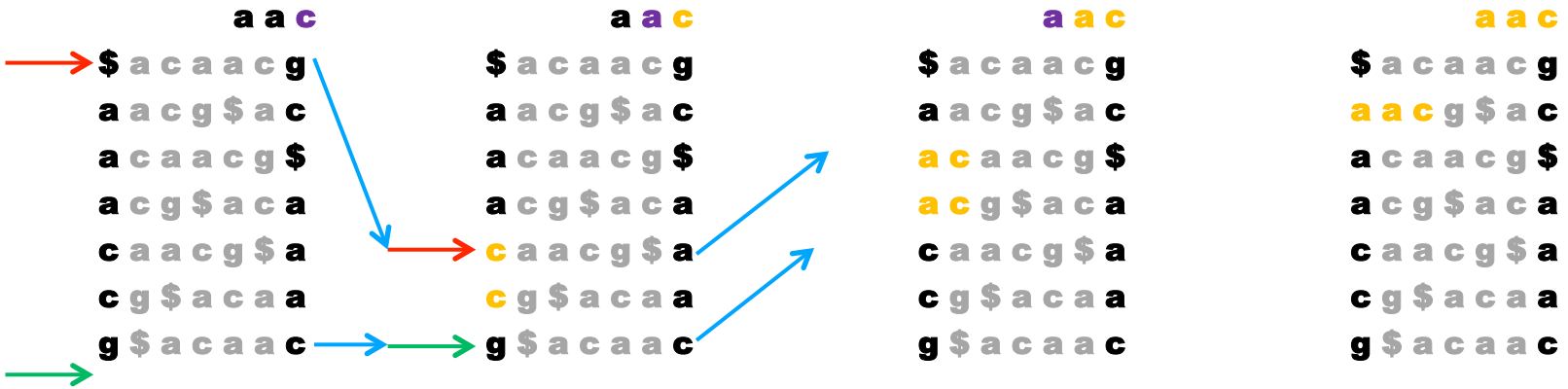
Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc



Find the first and last occurrence of a in range (!) on the right side. It is the 2nd and the 3rd occurrence of an a on the right side.

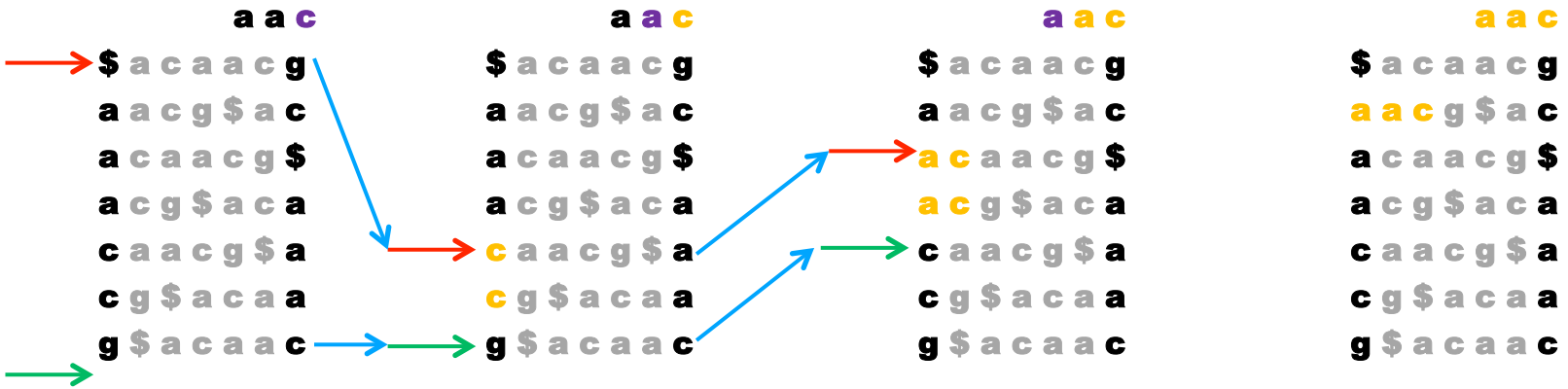
Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc



Mark 2nd and 3rd occurrence of a on the left side with top and bot.

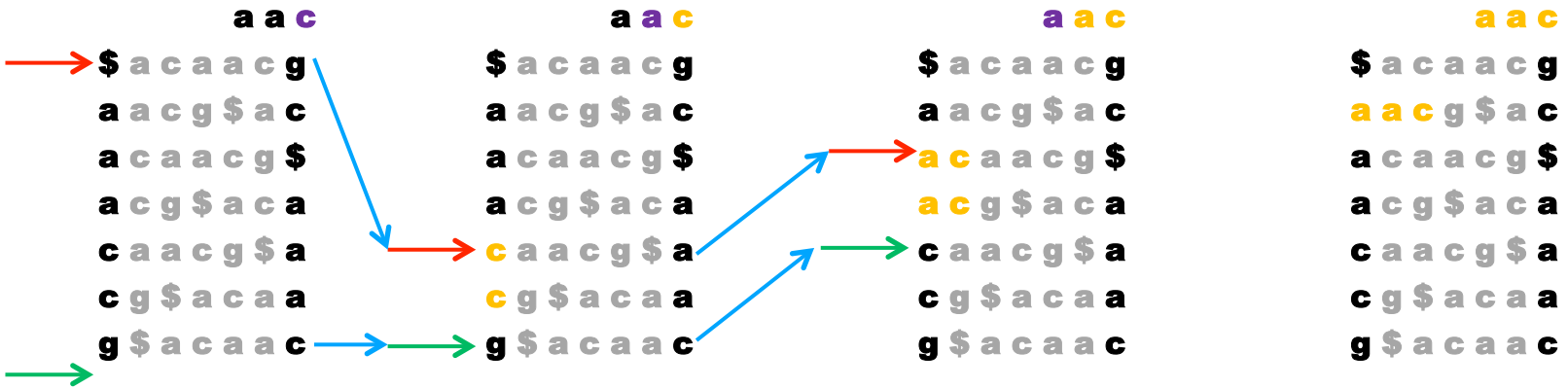
Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc



To be preceded by another a, there must be more a's in range (!) on the right side. Find their first and last occurrences. Here: It is the first occurrence of an a at the right side.

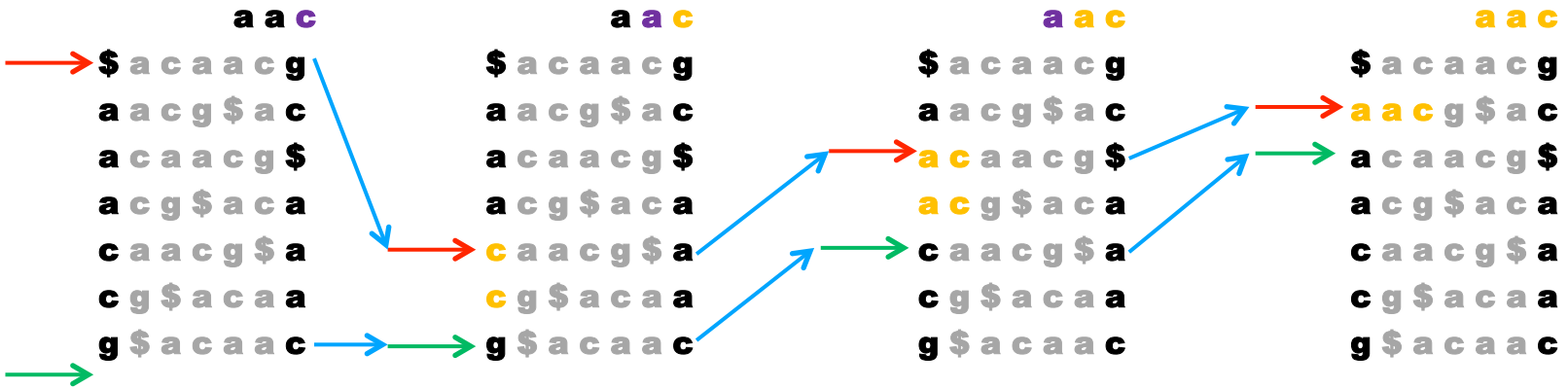
Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc



Mark the 1st occurrence of a on the left side with top and bot.

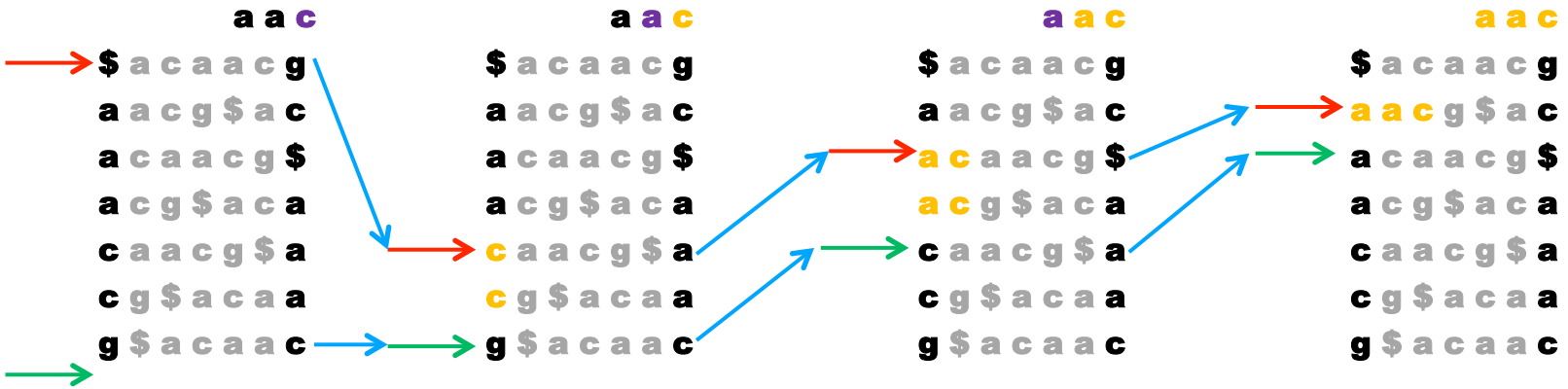
Mapping Software: Bowtie

Exact Matching

To match a query string Q in text T using BWT(T), repeatedly apply rule 3:

top = LF(top, qc): bot = LF(bot, qc)

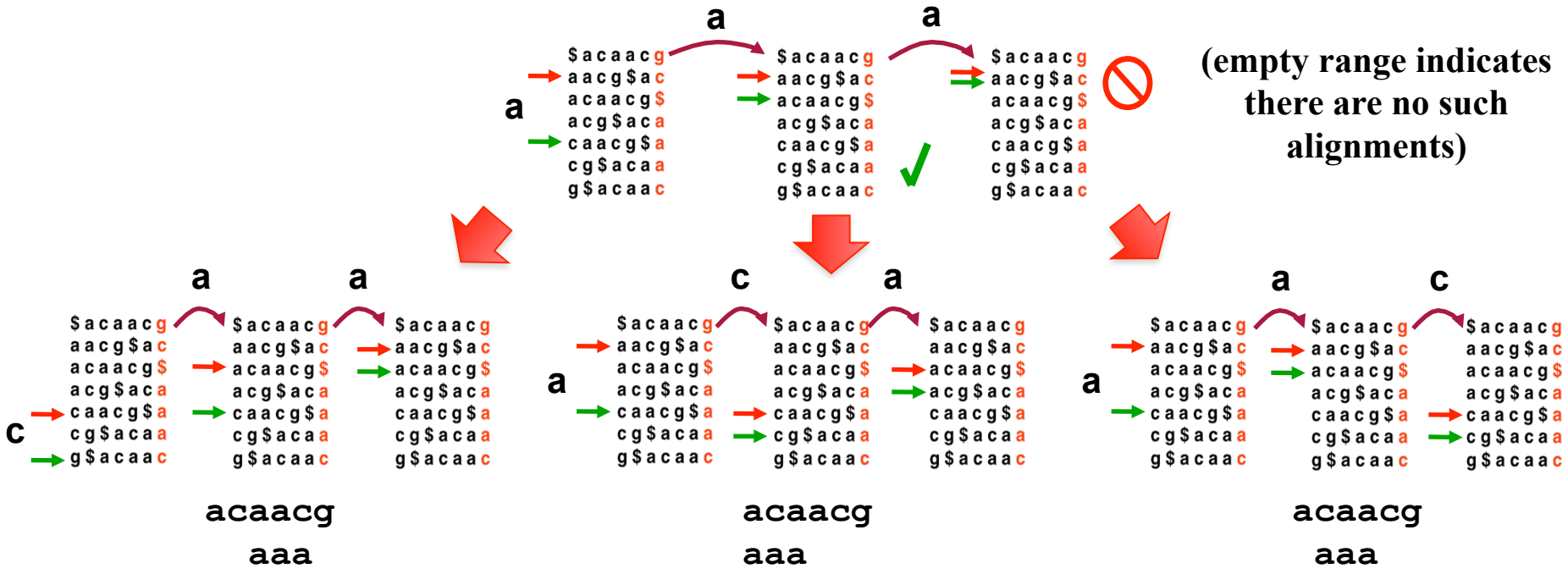
where qc is the next character in Q (right-to-left) and LF(*i*, qc) maps row *i* to the row whose first character corresponds to *i*'s last character as if it were qc



As all letters where matched and top > bot, we see a match. When top == bot, we have got a mismatch.

Inexact matching with Bowtie

Quality-aware, greedy, depth-first backtracking



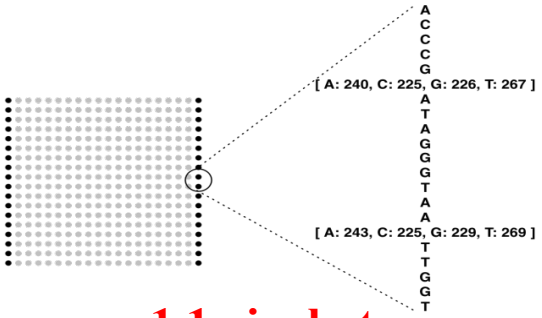
Above: Backtracking scenarios leading to 3 distinct 1-mismatch alignments for “aaa”

Mapping Software: Bowtie

- Memory efficient
- Parallelism (no need to split the index just copy it to n machines)
- Compares well with Maq and SOAP
- BWT has been adapted by many competing methods e.g. SOAP2

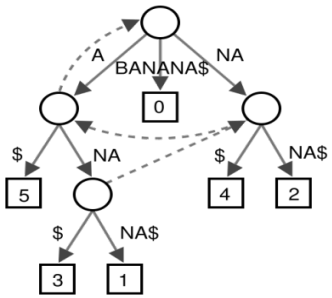
	Wall clock time (s)	Reads mapped per hour (millions)	Peak virtual memory footprint (megabytes)	Bowtie speedup	Reads aligned (%)
Bowtie –v 2 (server)	15m:41	33.8	1,149	351x	67.4
SOAP (server)	91h:47m:46	0.10	13,619		67.3
Bowtie (PC)	17m:57	29.5	1,353	59.8x	71.9
Maq (PC)	17h:53m:07	0.49	804		74.7
Bowtie (server)	18m:26	28.8	1,353	107x	71.9
Maq (server)	32h:58m:39	0.27	804		74.7

Bowtie Index



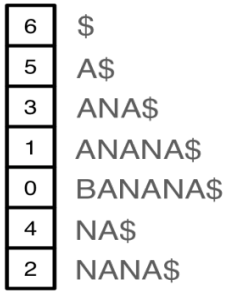
1.1 gigabytes
(2.2 incl. mirror index)

Suffix Tree



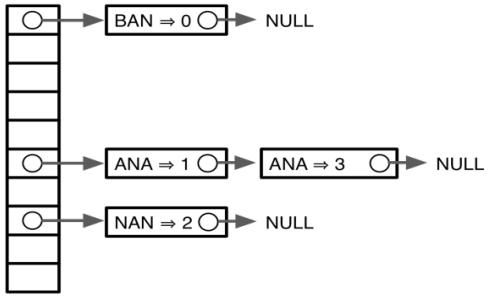
>35 gigabytes

Suffix Array



>12 gigabytes

k-mer Hash Tables



>12 gigabytes

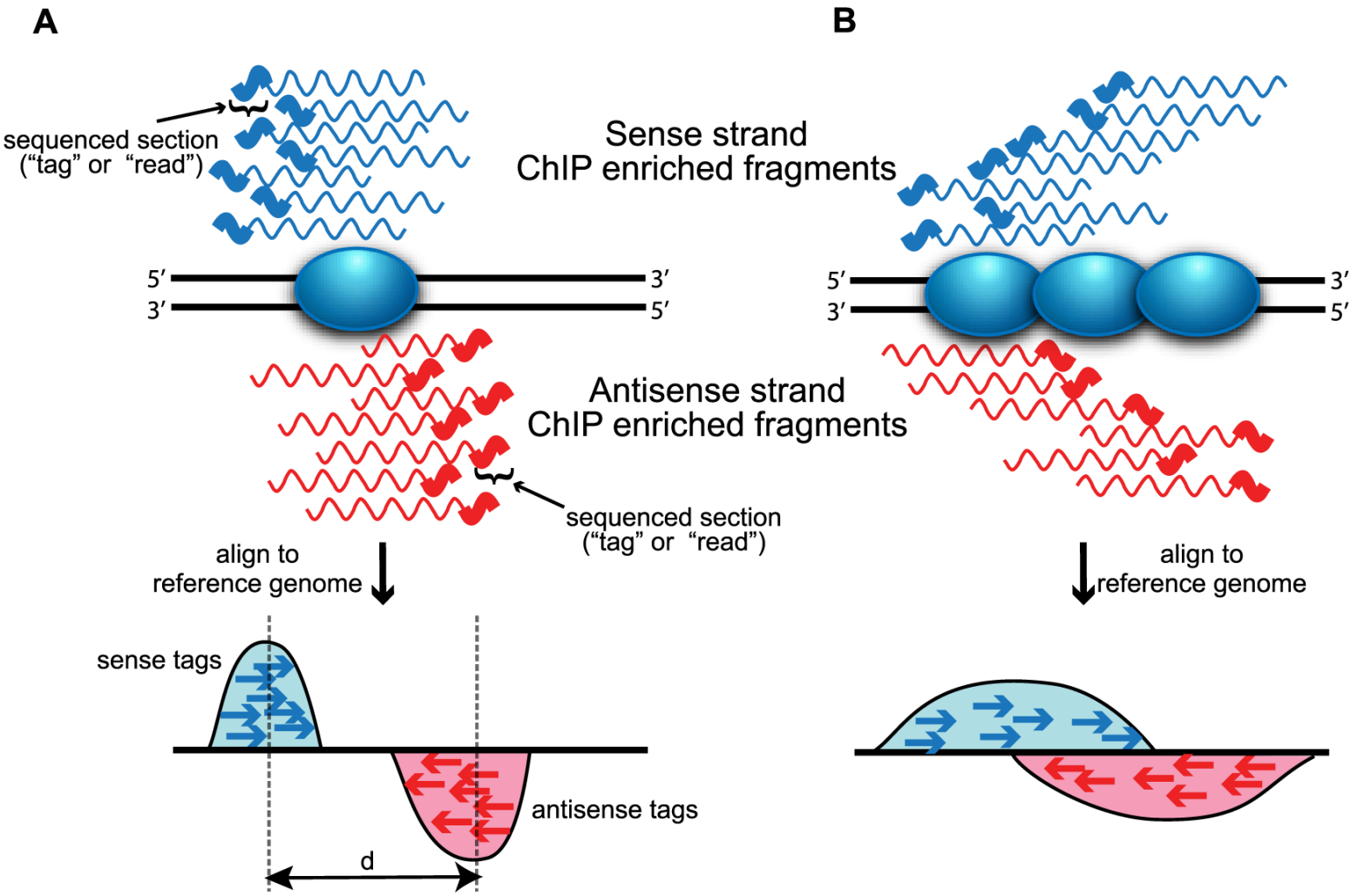
Read Mapping – File Format

```
R2D2_0091_7_1_1060_5044    2690948            2690984   >> TGGTGGCACCAAGCTTTTCCAGGCCGCTTTGATGCAT
        TGGTGGCACCAAGCTTTTCCAGGCCGCTTTGATGCAT        0
R2D2_0091_7_1_1061_14822   1889387   1889423   >> CGTCAACGGCTGCGCATTACCTTGGAGAATCTTGTCT
        CGTCAACGGCTGCGCATTACCTTGGAGAATCTTGTCT        0
R2D2_0091_7_1_1062_8962    724115    724151    >> GCTCGGCGGCGACAACATCC
        GCTCGGCGGCGACAACATCCCATTGATCTGGACCGCA                0
R2D2_0091_7_1_1062_20915   2678319   2678355   >> TCCACACCACTTCCACTAACC
        TCCACACCACTTCCACTAACCACGCAGCACATGCCCA        0
R2D2_0091_7_1_1062_20915   3122206   3122242   >> TCCACACCACTTCCACTAACC
        TCCACACCACTTCCACTAACCACGCAGCACATGCCCA        0
R2D2_0091_7_1_1063_6167    2644261   2644297   >> GACAGGAACTGCAGACCAGG
        GACAGGAACTGCAGACCAGGGGTGAAAAGCTCCGCTC        0
R2D2_0091_7_1_1064_20970   1733342   1733378   >> AGCCGATGAGAGCGGTCACG
        AGCCGATGAGAGCGGTCACGATACGTGCTTGGAGGCG        0
R2D2_0091_7_1_1064_15533   141762    141798    >> GGAGAAAACTGCCCACGGAT
        GGAGAAAACTGCCCACGGATTGGAGGCGTAAGCGGTG        0
R2D2_0091_7_1_1064_18341   2562733   2562769   >> CCTTGACCTCTTCCACGACGG
        CCTTGACCTCTTCCACGACGGAGGTATCAACAACGGT        0
R2D2_0091_7_1_1065_18624   1787746   1787782   >> GTTAGCGCCGTTCTCACCATC
        GTTAGCGCCGTTCTCACCATCCTTGCCATTTTCACCA        0
```

Again, there are several file formats for mapping and they usually contain the following entries in some order/coding:

- ID related to read file (fastq, etc…)
- start position in the genome
- end position in the genome
- forward (>>/+) or reverse (<</-) strand
- the sequence read
- the mapped sequence from the genome
- number of mismatches

Peak Detection



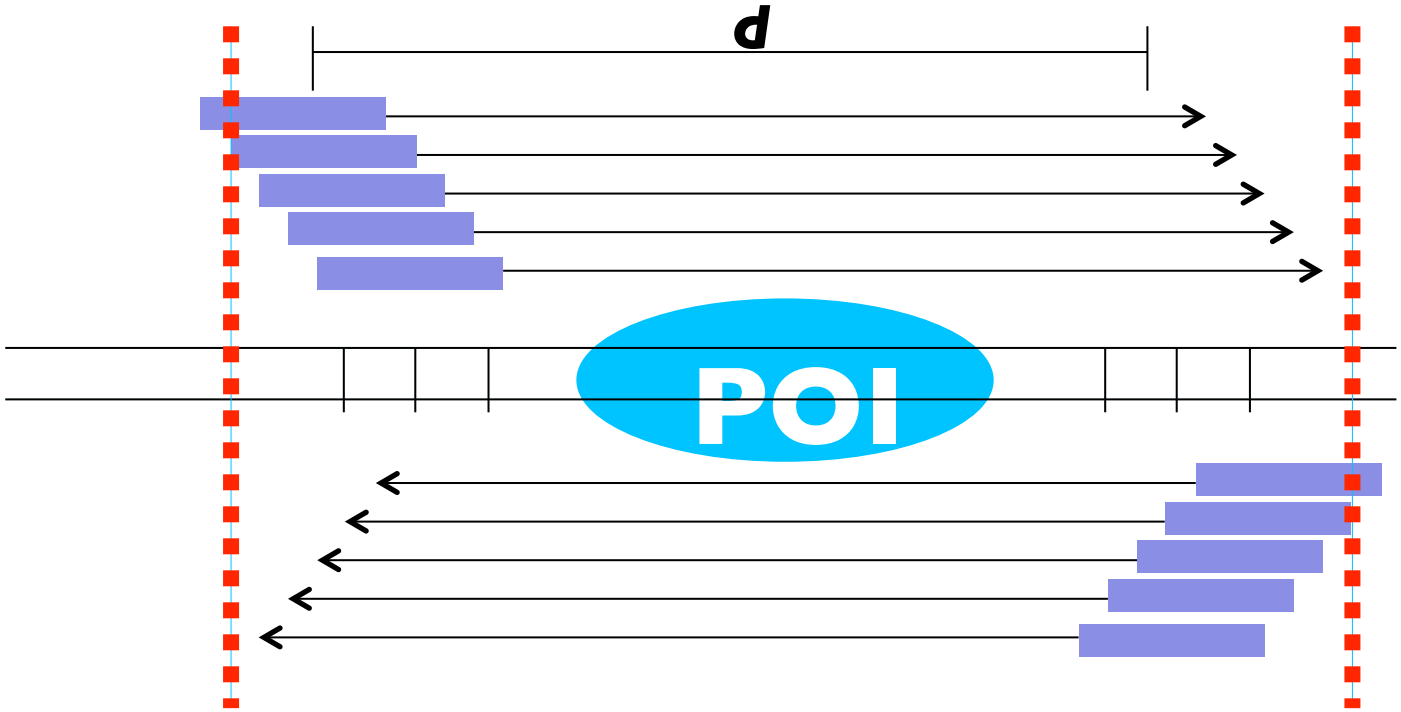
ChIP-Seq fragment length

Why does the separation between peaks (d) correspond to the average sequenced fragment length?

Ans: Library preparation

- Most Illumina protocols require that DNA is fragmented to less than 800 nt.
- Ideally, fragments have uniform size
- Sonication uses ultrasound waves in solution to shear DNA.
- Ultrasound waves pass through the sample, expanding and contracting liquid, creating "bubbles" in a process called cavitation.

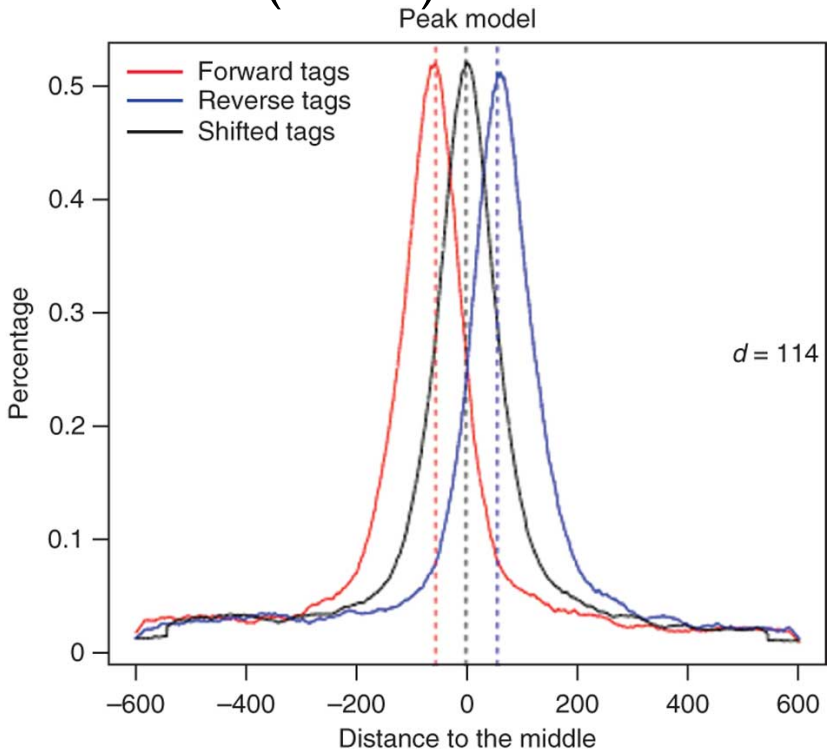
ChIP-Seq fragment length



Blue boxes: 36bps sequences obtained from sequencer.
The entire fragment is longer, with the exact size depending on the experimental fragmentation protocol. On average, the protein of interest (POI) is located in the middle of the fragment, so that the average distance between reads corresponds to the average fragment length

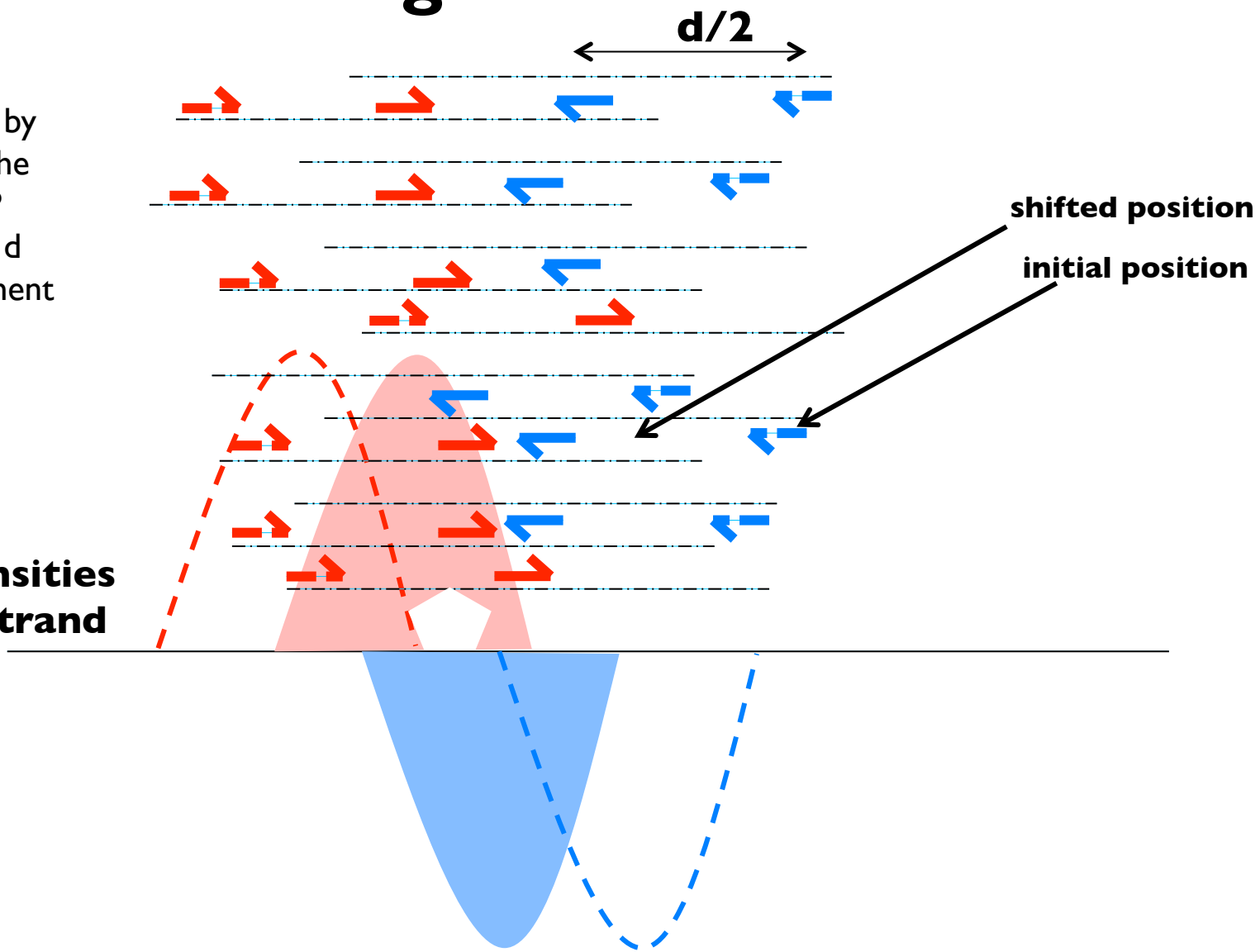
Tag Shift

- Positive/negative strand read peaks ***do not represent the true location of the binding site***
- Reads can be **shifted** by *d*/2 where *d* is the band size (MACS)
  → increased resolution
- Tags are shifted/extended towards the 3' direction to better represent the precise protein-DNA interaction site

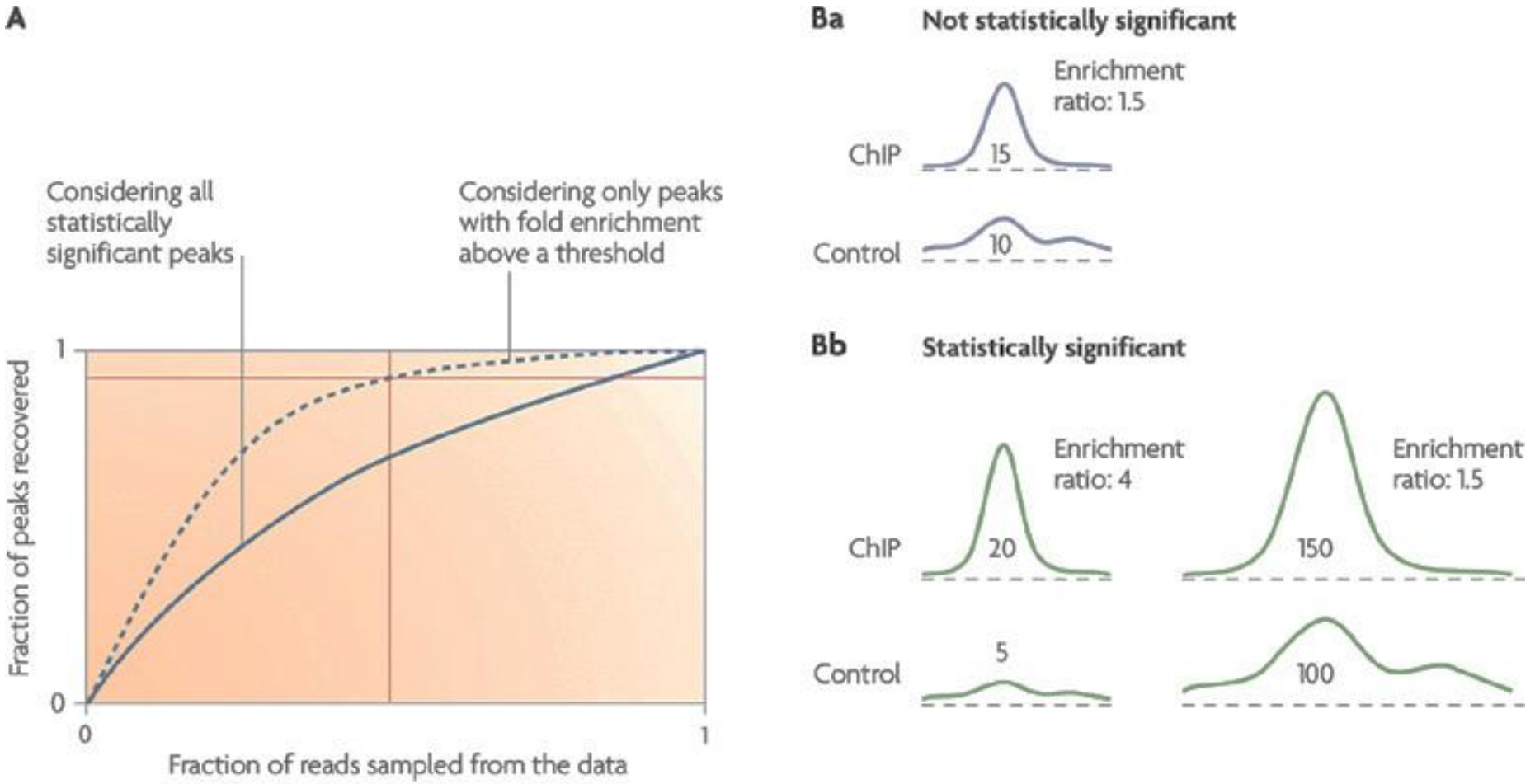


Tag Shift

Each tag is shifted by d/2 (i.e towards the middle of the IP fragment) where d represent the fragment length

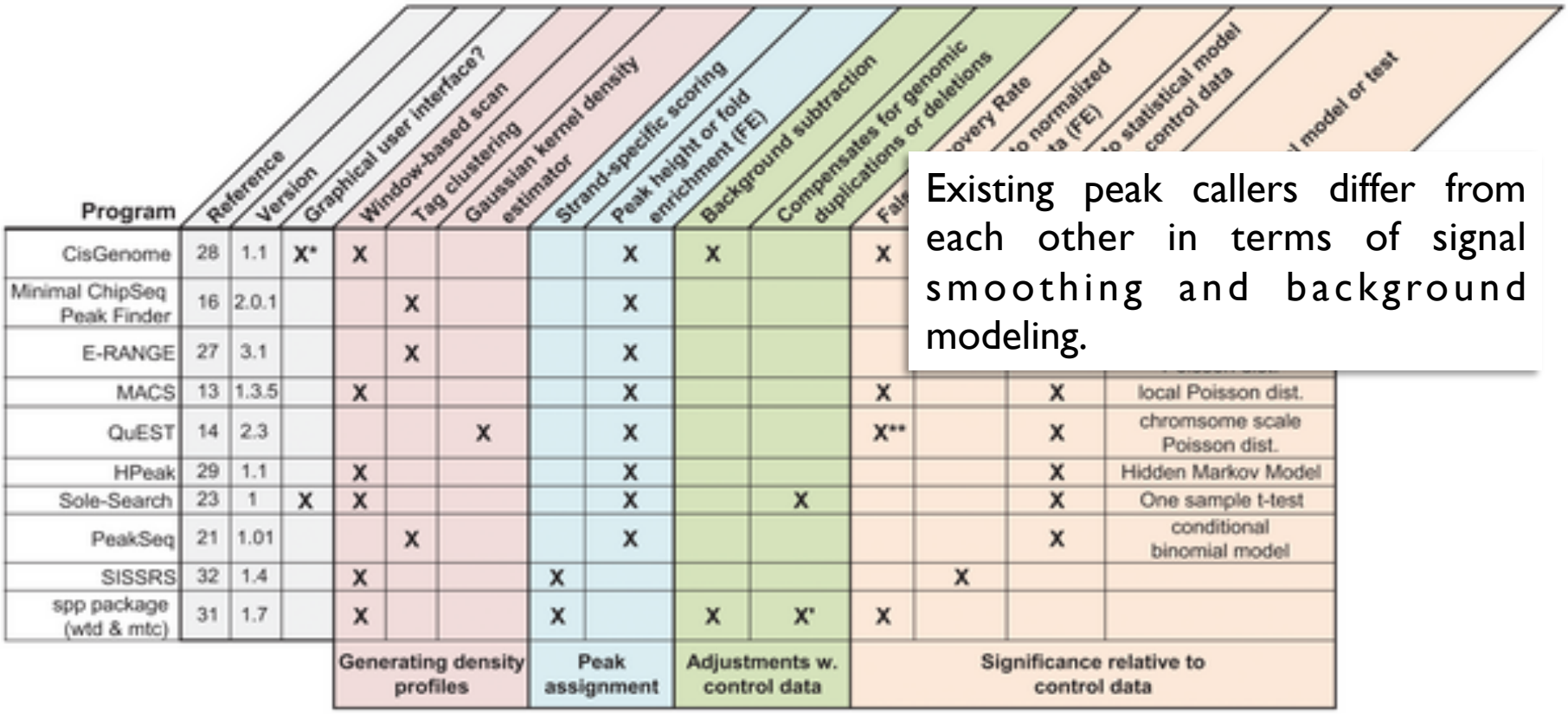


Depth of Sequencing



Nature Reviews | Genetics

Peak Detection Algorithms



Program	Reference	Version	Graphical user interface?	Window-based scan	Tag clustering	Gaussian kernel density estimator	Strand-specific scoring	Peak height or fold enrichment (FE)	Background subtraction	Compensates for genomic duplications or deletions	False	...ecovery Rate	...o normalized ...ta (FE)	...o statistical model ...control data	...l model or test
CisGenome	28	1.1	X*	X				X	X		X				
Minimal ChipSeq Peak Finder	16	2.0.1			X			X							
E-RANGE	27	3.1			X			X							
MACS	13	1.3.5		X				X			X		X	local Poisson dist.	
QuEST	14	2.3				X		X			X**		X	chromsome scale Poisson dist.	
HPeak	29	1.1		X				X					X	Hidden Markov Model	
Sole-Search	23	1	X	X				X		X			X	One sample t-test	
PeakSeq	21	1.01			X			X					X	conditional binomial model	
SISSRS	32	1.4		X			X					X			
spp package (wtd & mtc)	31	1.7		X			X		X	X'	X				
				Generating density profiles			Peak assignment		Adjustments w. control data		Significance relative to control data				

Existing peak callers differ from each other in terms of signal smoothing and background modeling.

X* = Windows-only GUI or cross-platform command line interface

X** = optional if sufficient data is available to split control data

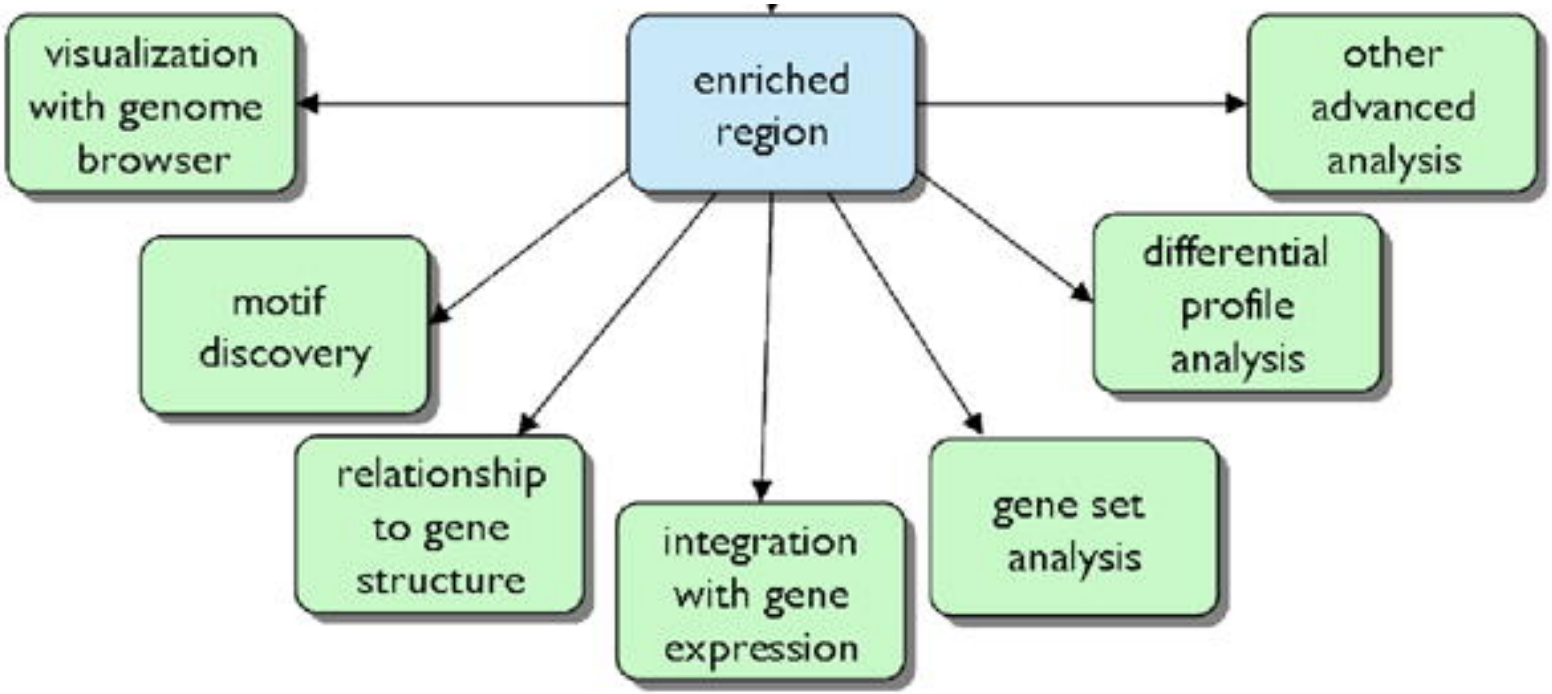
X' = method exludes putative duplicated regions, no treatment of deletions

Wilbanks EG, Facciotti MT (2010) Evaluation of Algorithm Performance in ChIP-Seq Peak Detection. PLoS ONE 5(7): e11471. doi:10.1371/journal.pone.0011471

http://www.plosone.org/article/info:doi/10.1371/journal.pone.0011471

Downstream Analysis

The enriched regions allow for further data analysis to answer the respective research question.



References

[Slides - Peter N. Robinson - ChIP-Seq - Peak Calling]

[Slides - J.van Helden et al ChIP-Seq analysis]

[Yong Zhang et al - Model-based Analysis of ChIP-Seq - Genome Biology 2008, 9:R137]

[Peter J Park - ChIP-seq: advantages and challenges of a maturing technology - Nature Review 10:2009]

[Elaine R Mardis - ChIP-seq: welcome to a new frontier - Nature Methods - 4, 613-614 (2007)]

[Wilbanks EG et al - Evaluation of Algorithm Performance in ChIP-Seq Peak Detection. PLoS ONE 5(7): e11471. (2010)]

[Ben Langmead et al - Ultrafast and memory-efficient alignment of short DNA sequences to the human genome - Genome Biology 2009, 10:R25]

[Burrows M and Wheeler D (1994), A block sorting lossless data compression algorithm, Technical Report 124, Digital Equipment Corporation]

[Ferragina P et al - Opportunistic data structures with applications - Proceedings of the 41st Annual Symposium on Foundations of Computer Science. IEEE Computer Society (2000)]

[Zhengdong D. Zhang et.al. - Modeling ChIP Sequencing In Silico with Applications - PLoS Computational Biol 4(8): e1000158 (2008)]

[Ruiqiang Li - SOAP2: an improved ultrafast tool for short read alignment - Bioinformatics (2009) 25 (15): 1966-1967]

[Yoav Benjamini - Controlling the false discovery rate: a practical and powerful approach to multiple testing - Journal of the Royal Statistical Society, 57 (1): 289–300 (1995)]

[Feng J et al – Identifying ChIP-seq enrichment using MACS – Nature Protocols 7, 1728-1740 (2012)]

Thank you!

Peak Detection: MACS

Model-based analysis of ChIP-seq (MACS) is a computational algorithm that identifies genome-wide locations of transcription/chromatin factor binding or histone modification from ChIP-seq data.

MACS consists of four steps:

- removing redundant reads,
- adjusting read position,
- calculating peak enrichment and
- estimating the empirical false discovery rate (FDR).

MACS: Estimation of fragment size

Given a sonication size *(bandwidth*) and a high-confidence fold-enrichment *(mfold)*, MACS slides windows of length 2 *X* bandwidth across the genome to find regions with tags more than mfold enriched relative to a random tag genome distribution

- slide a window of size BANDWIDTH
- retain top regions with MFOLD enrichment of treatment vs. input
- plot average +/- strand read densities → estimate d

MACS: Estimation of fragment size

Algorithm 1 Estimate Fragment Size

1: Slide a window of 2 X bandwidth* across genome

2: Identify regions of moderate enrichment (mfold: 10-30 fold)

3: **for each** peak *i* of 1000 randomly chosen enriched regions

do

4: separate reads into + and - strand

5: calculate mode of + and - summit

6: $d_i \leftarrow |mode_+ - mode_-|$

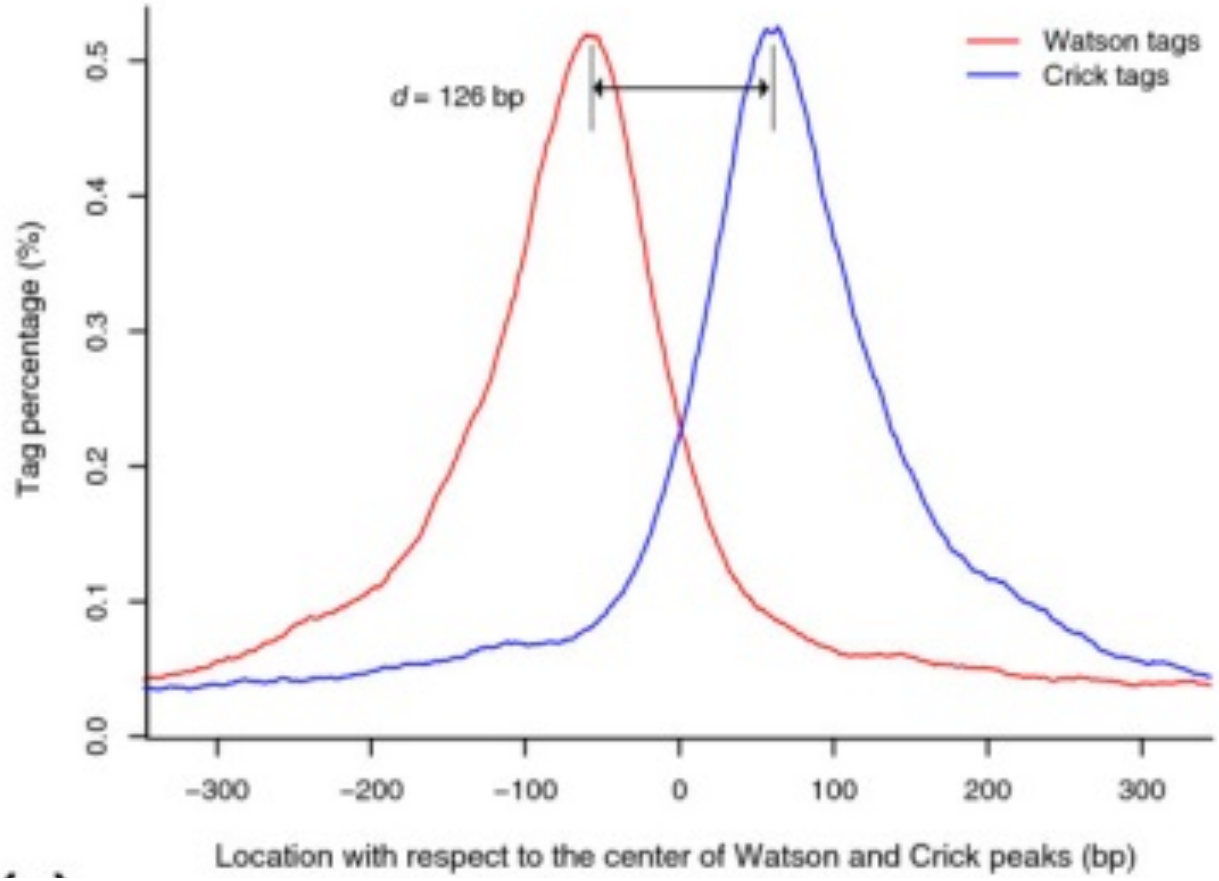
7: **end for**

8: $d \leftarrow average_i(d_i)$

Thus, the distance between bimodal summits is assumed to be the the estimated DNA fragment size d

* Roughly twice the size of the sheared chromatin across the genome

MACS: Estimation of fragment size



MACS: Estimation of fragment size

Once d has been estimated, all reads are shifted by d/2 to their 3' end, i.e., towards the center of the overall peak.

- A statistical test is then used to determine significant peaks
- A dynamic $\lambda\downarrow local$ is defined to capture local biases in the genome.

MACS: background bias

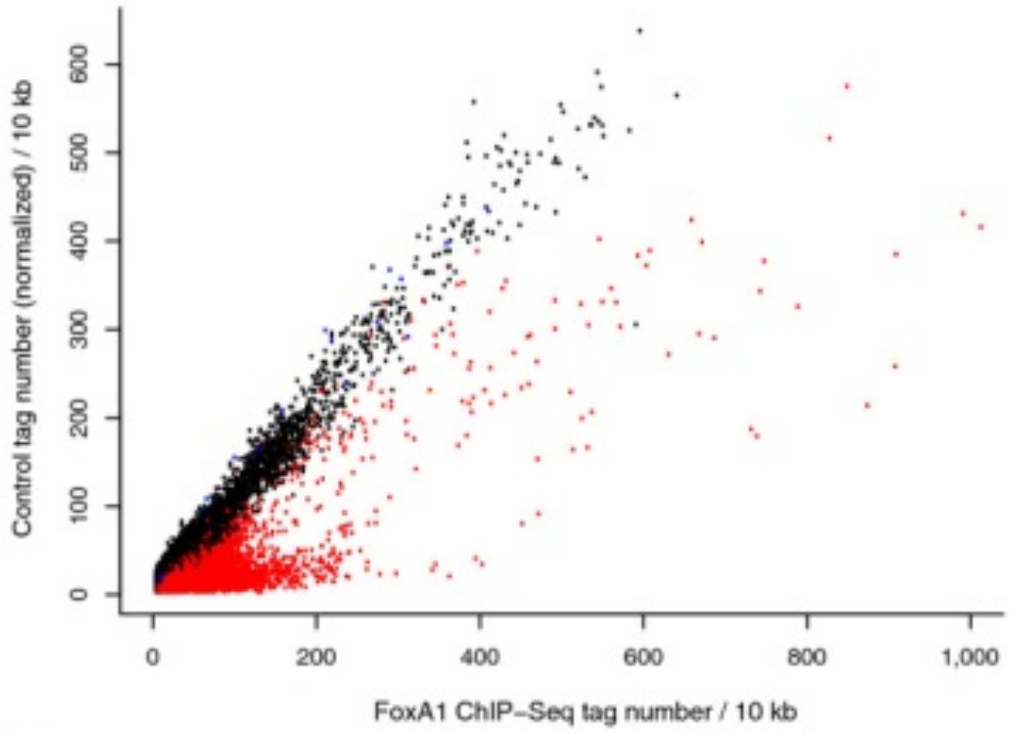
Similar to the situation with read-depth analysis in genome sequencing, local characteristics of the genome can lead to a bias in the number of reads being mapped.

- chromatin state (e.g. euchromatin fragments easier than silenced chromatin)
- GC content
- Therefore, ChIP-Seq experiments often include a control sample, consisting of the input material of the ChIP processed with an unspecific immunoprecipitation with "generic" (i.e., mixed) IgG

MACS: background bias

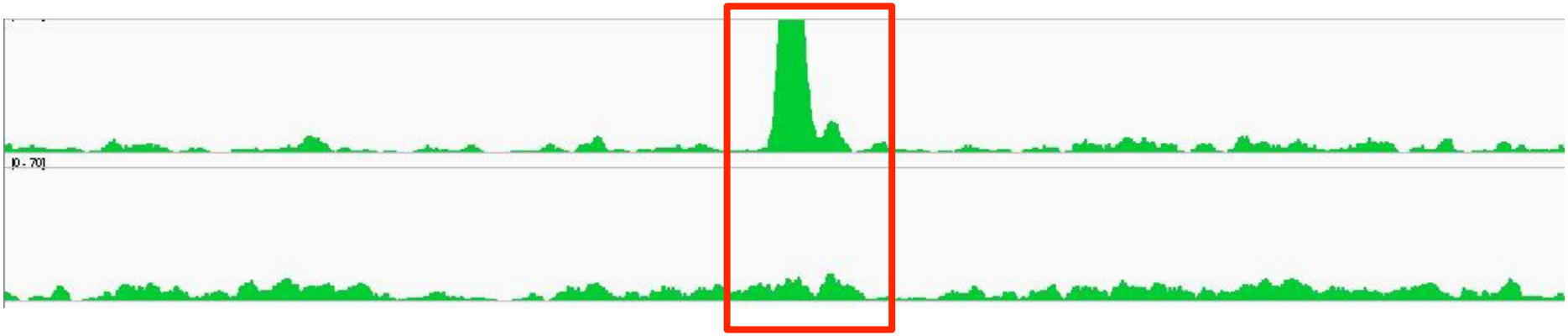
Similar to the situation with read-depth analysis in genome sequencing, local characteristics of the genome can lead to a bias in the number of reads being mapped.

The tag count in ChIP versus control in 10 kb windows across the genome. Each dot represents a 10 kb window; red dots are windows containing ChIP peaks and black dots are windows containing control peaks



MACS: background bias

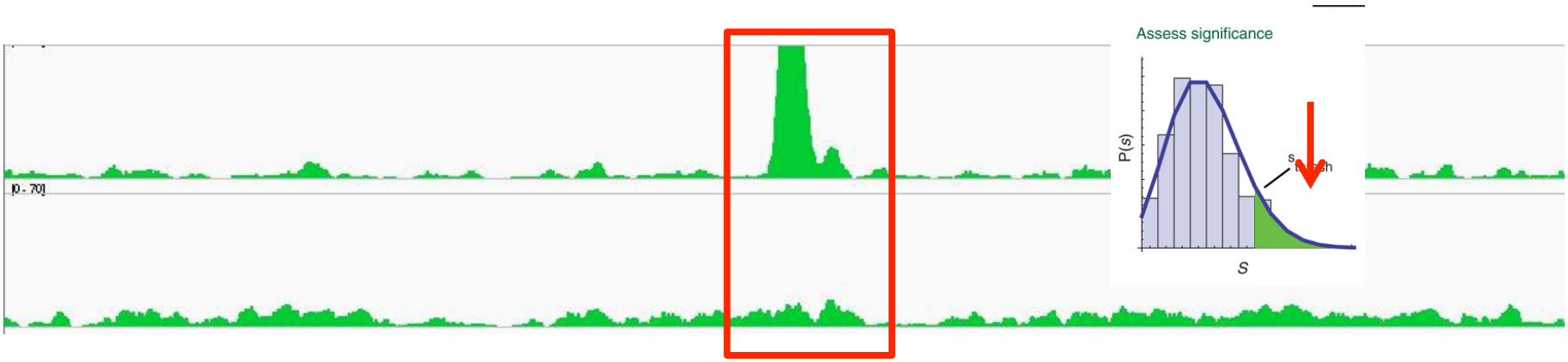
- slide a window of size 2*d across treatment and input
- estimate parameter $\lambda\downarrow local$ of Poisson distribution



MACS: peak calling

$\lambda\downarrow local$ reduces the influence of local biases, and is robust against occasional low tag counts at small local regions. MACS uses $\lambda\downarrow local$ to calculate the p-value of each candidate peak.

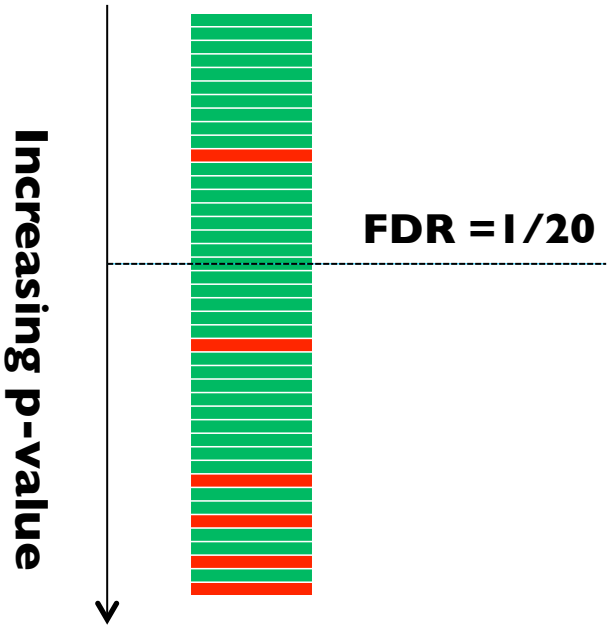
- Candidate peaks with p-values below a user-defined threshold p-value (default 10-5) are called (Poisson distribution)
- The ratio between the ChIP-Seq tag count and $\lambda\downarrow local$ is reported as the fold enrichment.



MACS: estimate FDR

- positive peaks (P-values)
- swap treatment and input; call negative peaks (P-value)

- $FDR(p) = \#\ negative\ peaks\ with\ p-value < p / \#\ positive\ peaks\ with\ p-value < p$

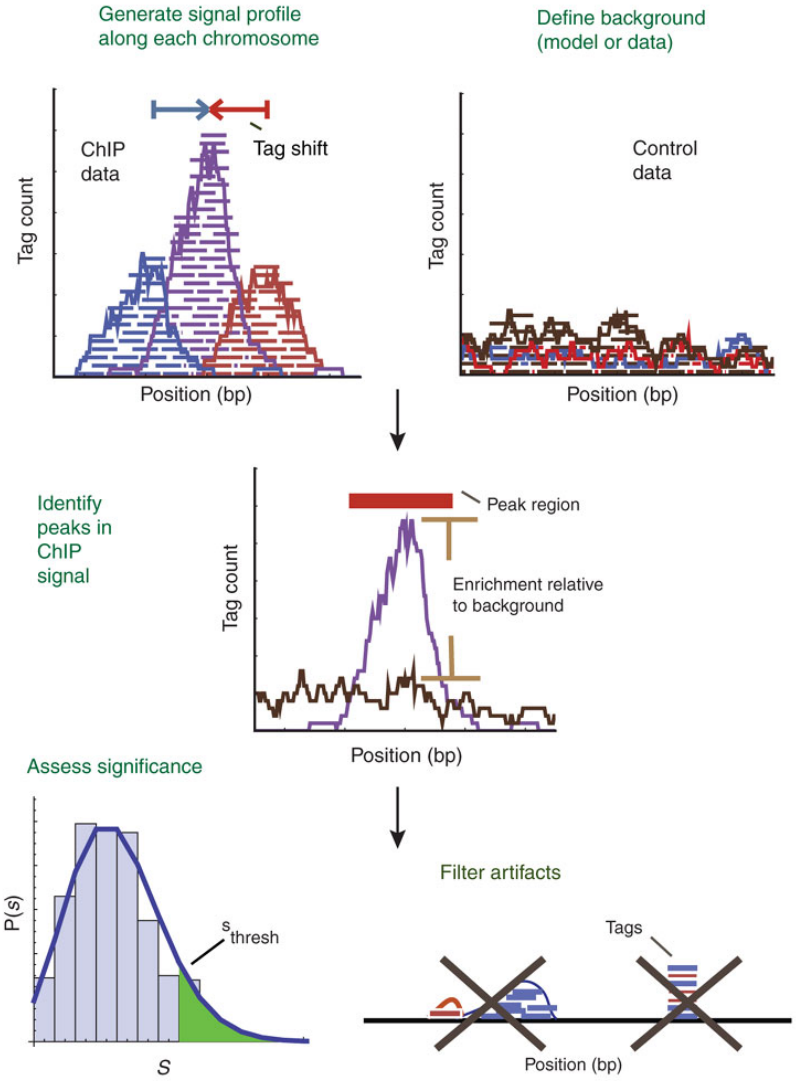


ChIP Seq - Artifacts

It may also be useful to filter out certain classes of peaks that are likely to be artifacts

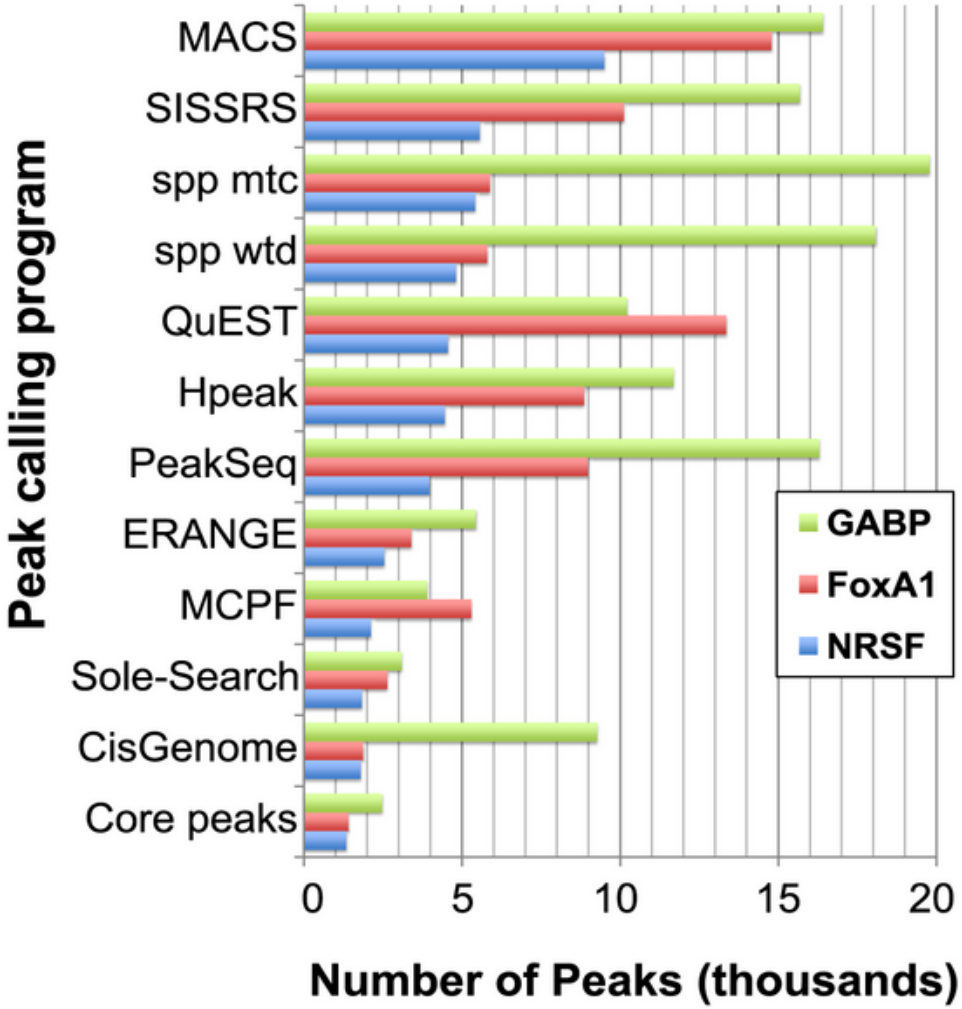
- Peaks with many reads starting from the same position
- Peaks with reads mainly from only one strand

[Pepke S et al. (2009) Computation for ChIP-seq and RNA-seq studies Nature Methods 6:S22{S32]



ChIP-Seq: An unsolved problem

Programs report different numbers of peaks, when run with their default or recommended settings on the same dataset.



Wilbanks EG, Facciotti MT (2010) Evaluation of Algorithm Performance in ChIP-Seq Peak Detection. PLoS ONE 5(7): e11471. doi:10.1371/journal.pone.0011471
http://www.plosone.org/article/info:doi/10.1371/journal.pone.0011471