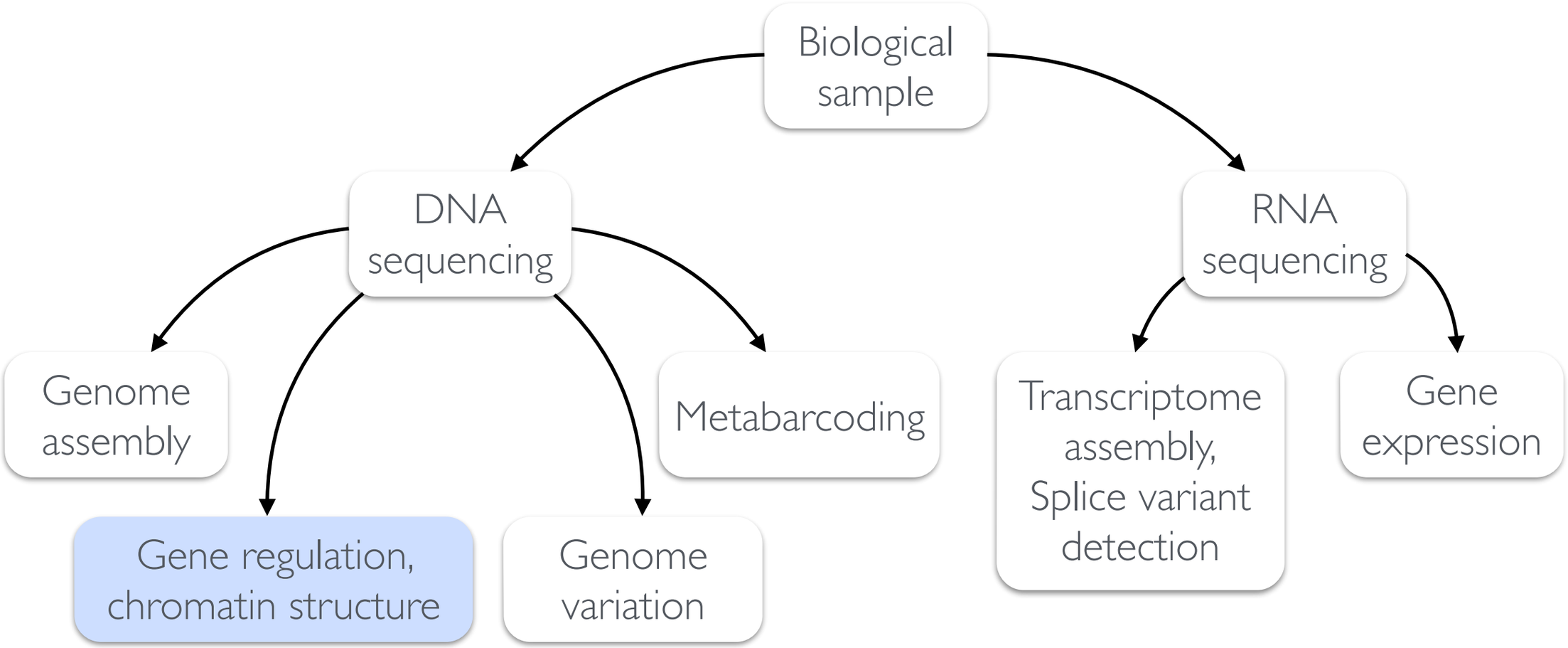
MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

ChIP - Data processing

Sebastian Schmeier
s.schmeier@gmail.com
http://sschmeier.github.io/bioinf-workshop/
2015

THE ENGINE
OF THE NEW
NEW ZEALAND

Common analyses overview

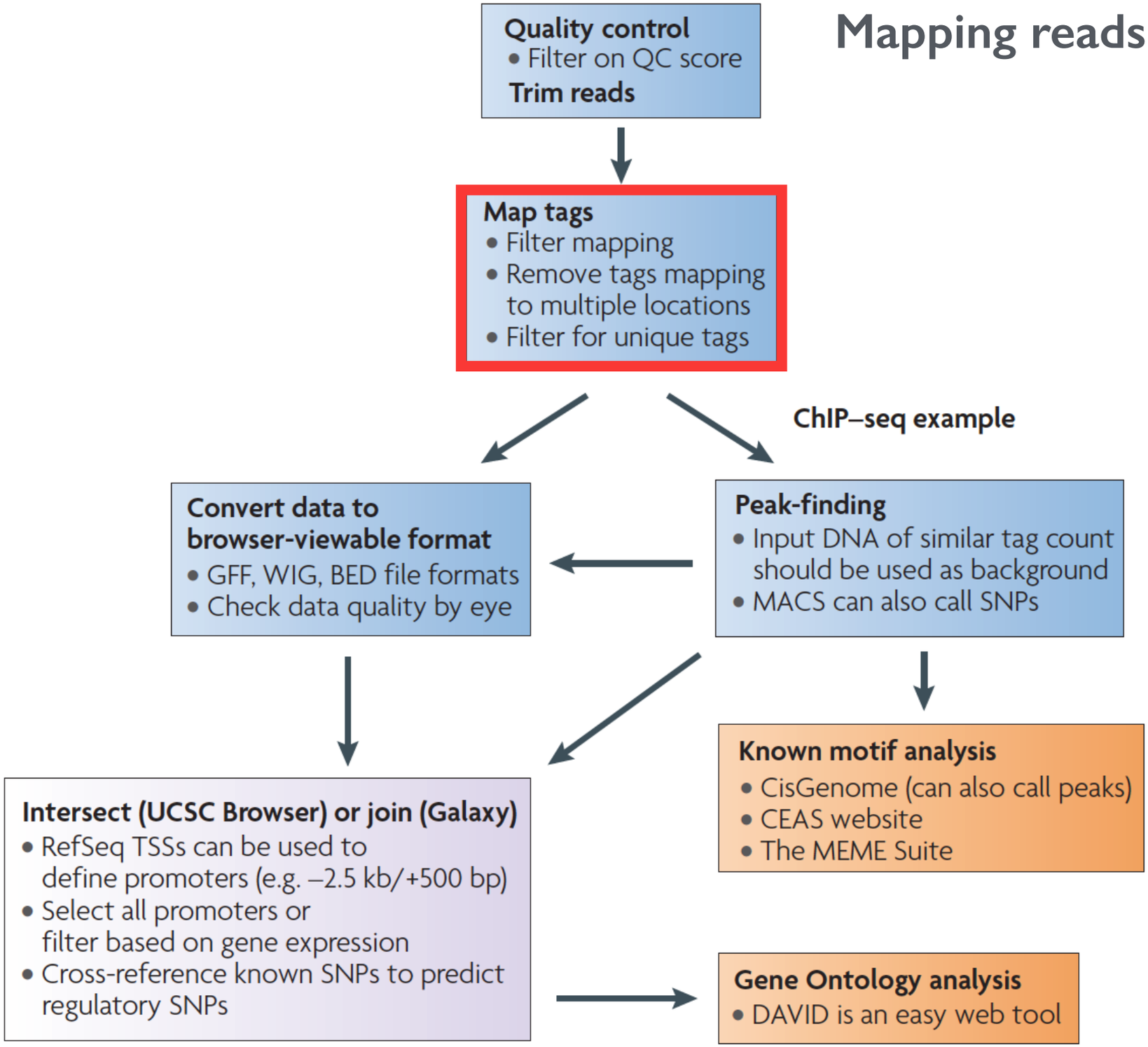


Gene regulation, chromatin structure

- How do we analyse it?
  - Mapping reads to a reference genome
  - Calling peaks

Chromatin immunoprecipitation (ChIP)

Mapping reads



http://www.nature.com/nrg/journal/v11/n7/full/nrg2795.html

Mapping reads

- Challenges
- Approximate String Matching Problem
- Burrows-Wheeler transform
- Bowtie

Challenges of mapping short reads

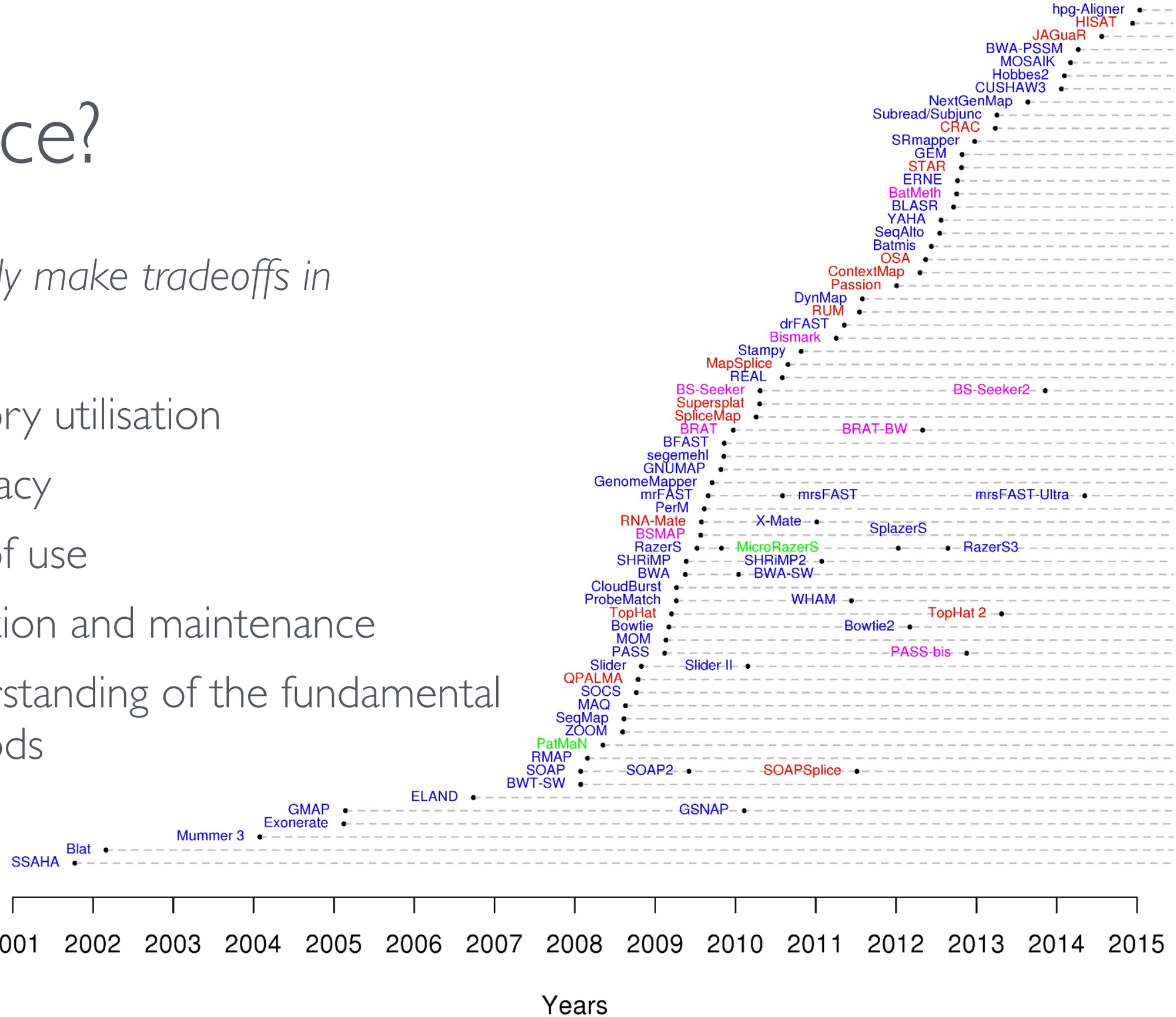
- If the reference genome is very large, and if we have billions of reads, how quickly can we align the reads to the genome?
  - The task of mapping billions of sequences to a mammalian-sized genome calls for extraordinarily efficient algorithms, in which every bit of memory is used optimally or near optimally.

Challenges of mapping short reads

- If a read comes from a repetitive element in the reference, a program must pick which copy of the repeat the read belongs to
  - The program may choose to report multiple possible locations or to pick a location heuristically
- Sequencing errors or variations between the sequenced chromosomes and the reference genome exacerbate this problem, because the alignment between the read and its true source in the genome may actually have more differences than the alignment between the read and some other copy of the repeat

Choice?

- *Intelligently make tradeoffs in*
  - Speed
  - Memory utilisation
  - Accuracy
  - Ease of use
  - Adoption and maintenance
  - Understanding of the fundamental methods



https://www.ebi.ac.uk/~nf/hts_mappers/

Mapping algorithms

- One could find the true locations using **exact matching**, assuming:
  - a genome had no repeats and a sequencing experiment introduced no errors
  - a sufficient read length relative to the genome size
- Assumption do **NOT** hold

Mapping algorithms

- **Approximate String Matching Problem**
  - Searching for occurrences of the read sequence within the reference sequence but allowing for *some* mismatches and gaps between the two
- Standard algorithm: *dynamic programming*
  - Too slow
  - Too much memory required

Mapping algorithms

- **Approximate String Matching Problem**
- Two main ideas for addressing large input sizes (in # of reads and size of the reference):
  - **filtering**
    - quickly exclude large regions of the reference where no approximate match can be found
  - **indexing**
    - *Preprocessing* the reference sequence and/or the set of reads to establish string indices
    - Benefit of preprocessing into string indices is that it typically does not require scanning the whole reference, and it can therefore conduct queries much faster at the expense of larger memory consumption.
    - The string indices that are currently used are:
      - Suffix array
      - Enhanced suffix array
      - **FM-index (Full-text index in Minute space) + Burrows-Wheeler transform**

Burrows-Wheeler transform (BWT)
Creation

- Write down all rotation of the string
- Sort the matrix lexicographically
- Last column is the BWT(T)
  - The rows in the matrix are essentially the sorted suffixes of the text
- SA(T) is the start offset in the original string

T = abaaba$

							SA(T)
$	a	b	a	a	b	a	6
a	$	a	b	a	a	b	5
b	a	$	a	b	a	a	4
a	b	a	$	a	b	a	3
a	a	b	a	$	a	b	2
b	a	a	b	a	$	a	1
a	b	a	a	b	a	$	0

							SA(T)
$	a	b	a	a	b	a	6
a	$	a	b	a	a	b	5
a	a	b	a	$	a	b	2
a	b	a	$	a	b	a	3
a	b	a	a	b	a	$	0
b	a	$	a	b	a	a	4
b	a	a	b	a	$	a	1

BWT(T) = abba$aa

Burrows-Wheeler transform (BWT)
LF mapping

T = abaaba$

BWT(T) = abba$aa

							SA(T)
$	a	b	a	a	b	a	6
a	$	a	b	a	a	b	5
a	a	b	a	$	a	b	2
a	b	a	$	a	b	a	3
a	b	a	a	b	a	$	0
b	a	$	a	b	a	a	4
b	a	a	b	a	$	a	1

- We *rank* according to how many times the same character occurred previously in BWT(T)
- We keep an array of positions in the rotation SA(T)
- We keep an index of occurrences starting at zero

	F	*L*	*rank*	SA(T)
	$	a	0	6
0	a	b	0	5
1	a	b	1	2
2	a	a	1	3
3	a	$	0	0
0	b	a	2	4
1	b	a	3	1

Burrows-Wheeler transform (BWT)

Exact matching

T = abaaba$

BWT(T) = abba$aa

P=aba

	F	L	$rank$
	$	a	0
0	a	b	0
1	a	b	1
2	a	a	1
3	a	$	0
0	b	a	2
1	b	a	3

P=aba

	F	L	$rank$
	$	a	0
0	a	b	0
1	a	b	1
2	a	a	1
3	a	$	0
0	b	a	2
1	b	a	3

P=aba

	F	L	$rank$	SA(T)
	$	a	0	6
0	a	b	0	5
1	a	b	1	2
2	a	a	1	3
3	a	$	0	0
0	b	a	2	4
1	b	a	3	1

Burrows-Wheeler transform (BWT)

Exact matching

T = abaaba$

BWT(T) = abba$aa

P=aba

	F	L	$rank$
	$	a	0
0	a	b	0
1	a	b	1
2	a	a	1
3	a	$	0
0	b	a	2
1	b	a	3

P=aba

	F	L	$rank$
	$	a	0
0	a	b	0
1	a	b	1
2	a	a	1
3	a	$	0
0	b	a	2
1	b	a	3

P=aba

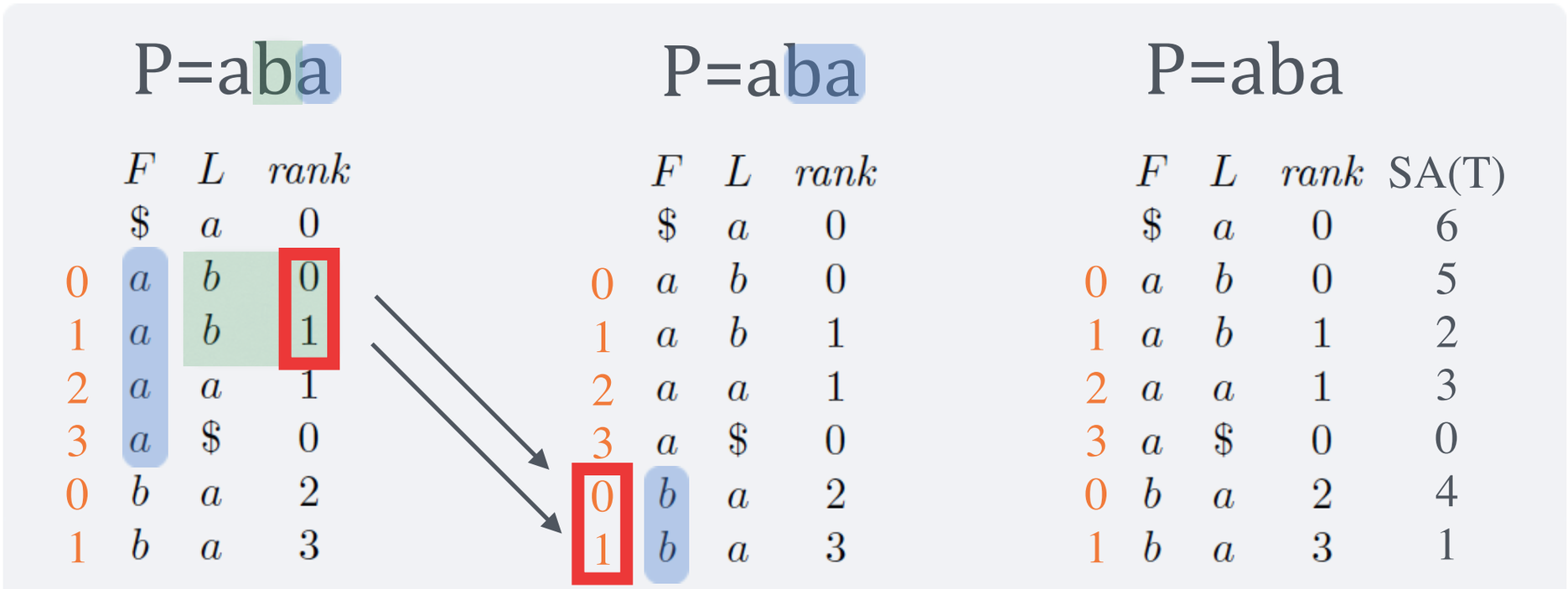
	F	L	$rank$	SA(T)
	$	a	0	6
0	a	b	0	5
1	a	b	1	2
2	a	a	1	3
3	a	$	0	0
0	b	a	2	4
1	b	a	3	1

Burrows-Wheeler transform (BWT)

Exact matching

T = abaaba$

BWT(T) = abba$aa



Burrows-Wheeler transform (BWT)

Exact matching

T = abaaba$

BWT(T) = abba$aa

P=aba

	F	L	$rank$
	$	a	0
0	a	b	0
1	a	b	1
2	a	a	1
3	a	$	0
0	b	a	2
1	b	a	3

P=aba

	F	L	$rank$
	$	a	0
0	a	b	0
1	a	b	1
2	a	a	1
3	a	$	0
0	b	a	2
1	b	a	3

P=aba

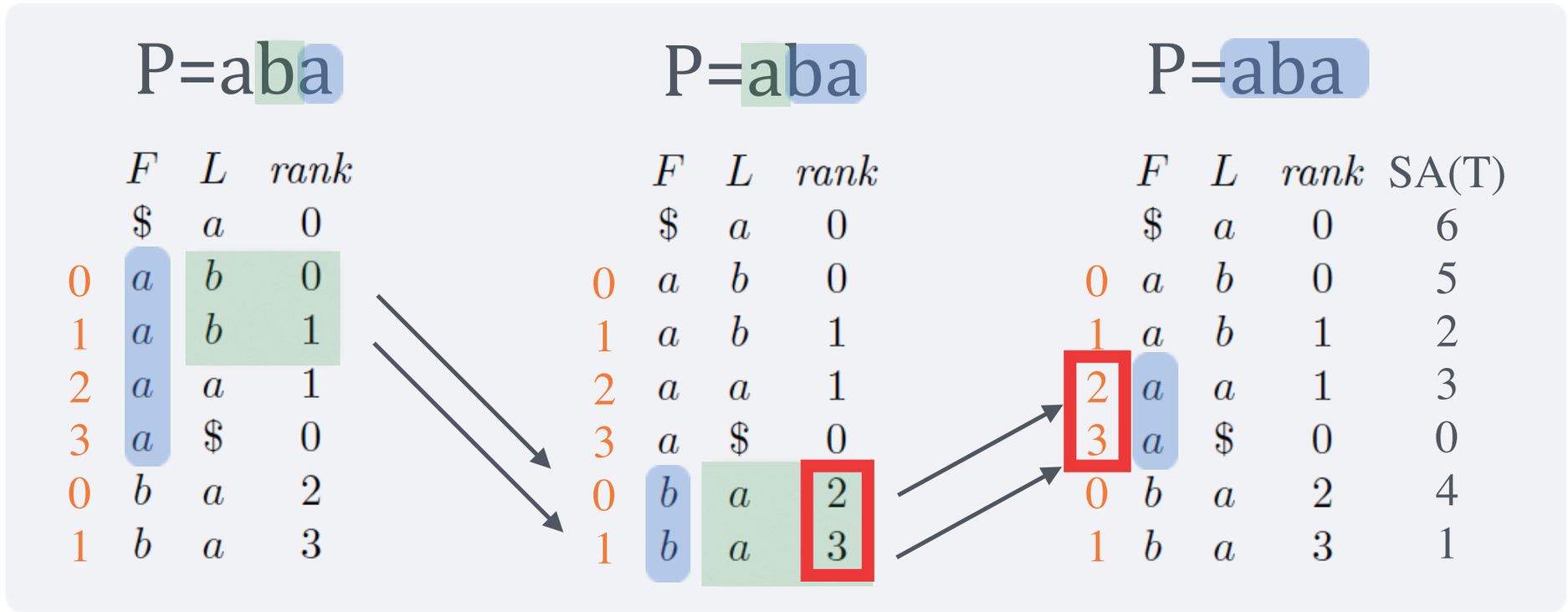
	F	L	$rank$	SA(T)
	$	a	0	6
0	a	b	0	5
1	a	b	1	2
2	a	a	1	3
3	a	$	0	0
0	b	a	2	4
1	b	a	3	1

Burrows-Wheeler transform (BWT)

Exact matching

T = abaaba$

BWT(T) = abba$aa

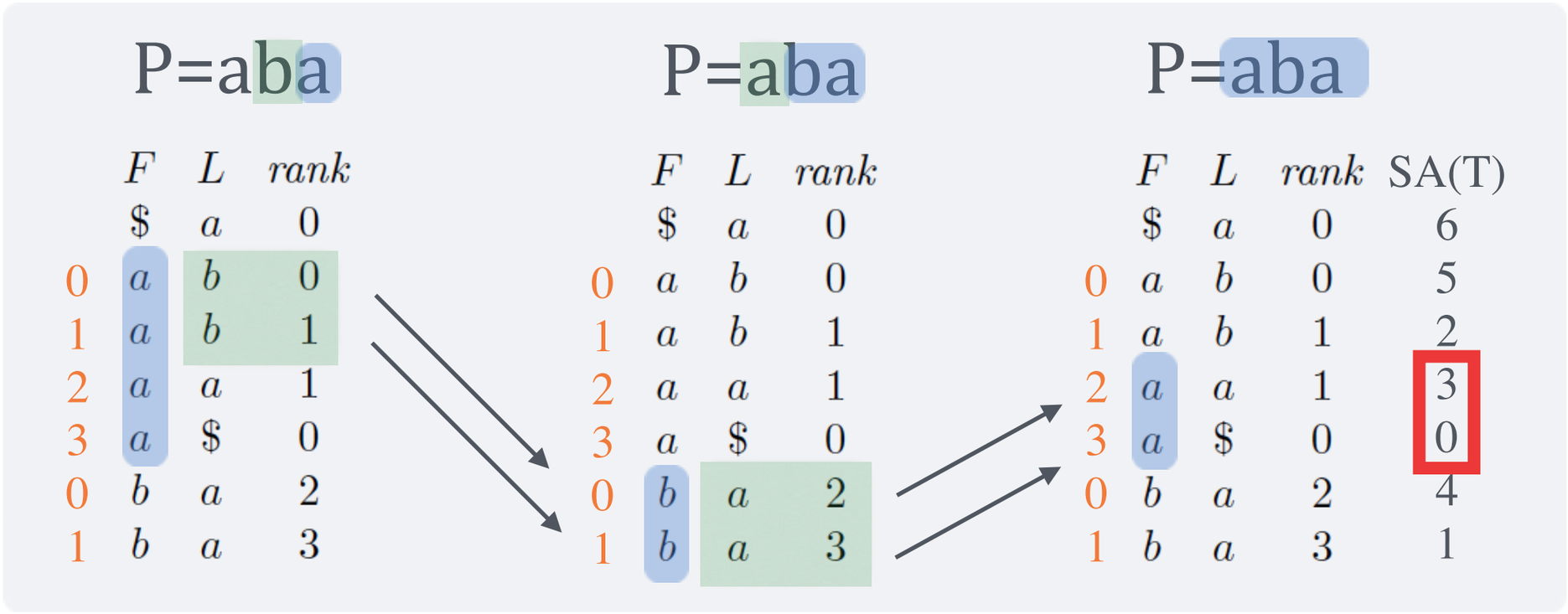


Burrows-Wheeler transform (BWT)

Exact matching

T = abaaba$

BWT(T) = abba$aa



- Allows for matching in constant time

T = abaaba$

0 3

Resource

Mapping short DNA sequencing reads and calling variants using mapping quality scores

Heng Li,[1] Jue Ruan,[2] and Richard Durbin[1,3]

[1]The Wellcome Trust Sanger Institute, Hinxton CB10 1SA, United Kingdom; [2]Beijing Genomics Institute, Chinese Academy of Science, Beijing 100029, China

Software

Open Access

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

Ben Langmead, Cole Trapnell, Mihai Pop and Steven L Salzberg

Address: Center for Bioinformatics and Computational Biology, Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742, USA.

Correspondence: Ben Langmead. Email: langmead@cs.umd.edu

Published: 4 March 2009

Genome ***Biology*** 2009, **10:**R25 (doi:10.1186/gb-2009-10-3-r25)

Received: 21 October 2008
Revised: 19 December 2008
Accepted: 4 March 2009

Bowtie

- FM Index finds exact sequence matches quickly in small memory, but short read alignment demands more:
  - Allowances for mismatches
  - Consideration of quality values

Bowtie

- Bowtie's solution: *backtracking quality-aware* search
  - if a particular base is not found in the index, while traversing the matrix, backtrack and try another "base" based on quality and continue with the search string

T = abaaba$

P=aaaa

	F	L	rank
	$	a	0
0	a	b	0
1	a	b	1
2	a	a	1
3	a	$	0
0	b	a	2
1	b	a	3

P= aaaa

	F	L	rank
	$	a	0
0	a	b	0
1	a	b	1
2	a	a	1
3	a	$	0
0	b	a	2
1	b	a	3

P=aaaa

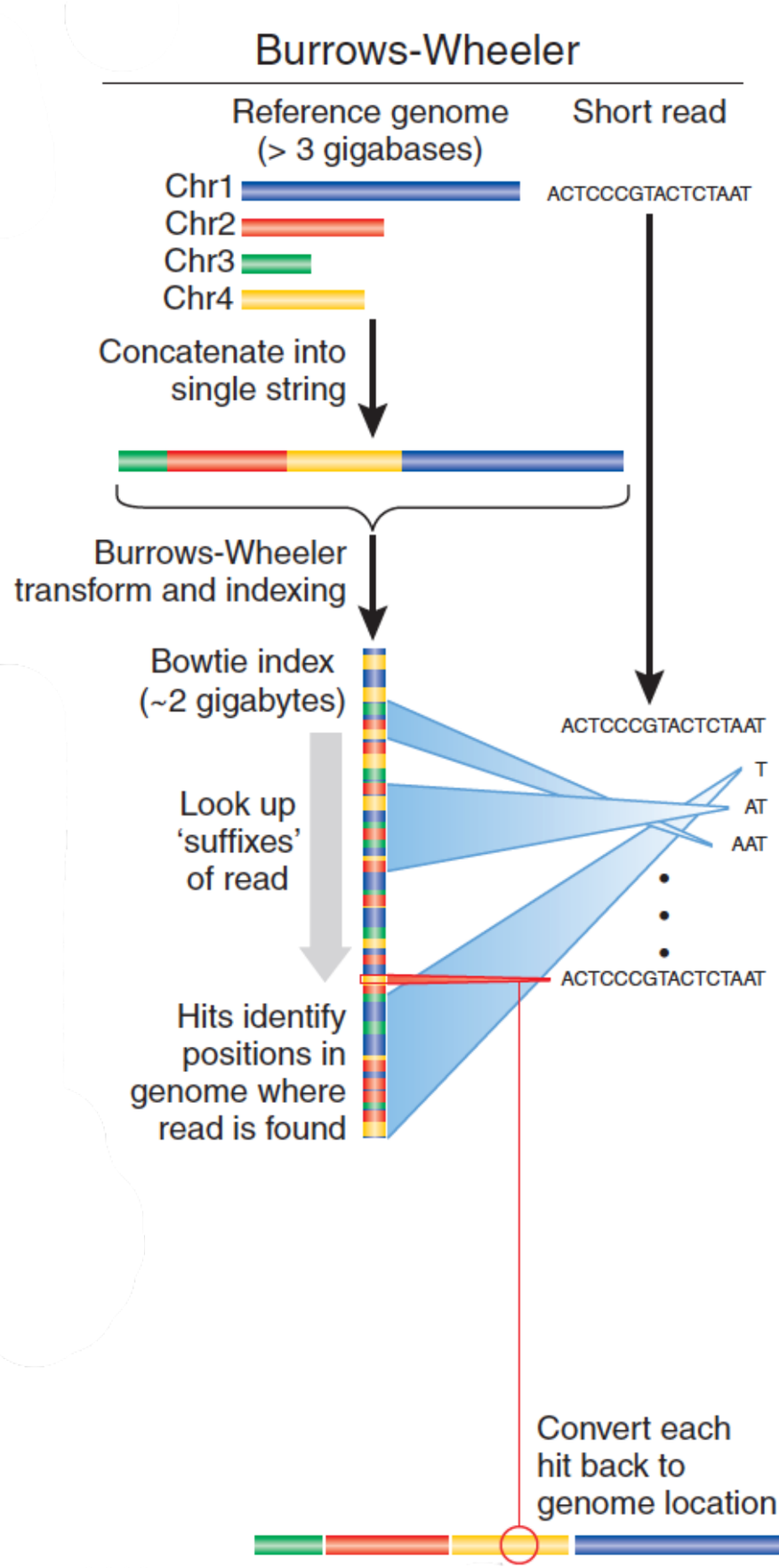
	F	L	rank
	$	a	0
0	a	b	0
1	a	b	1
2	a	a	1
3	a	$	0
0	b	a	2
1	b	a	3

P=aaaa

	F	L	rank	SA(T)
	$	a	0	6
0	a	b	0	5
1	a	b	1	2
2	a	a	1	3
3	a	$	0	0
0	b	a	2	4
1	b	a	3	1

Burrows-Wheeler transform *genome scale*

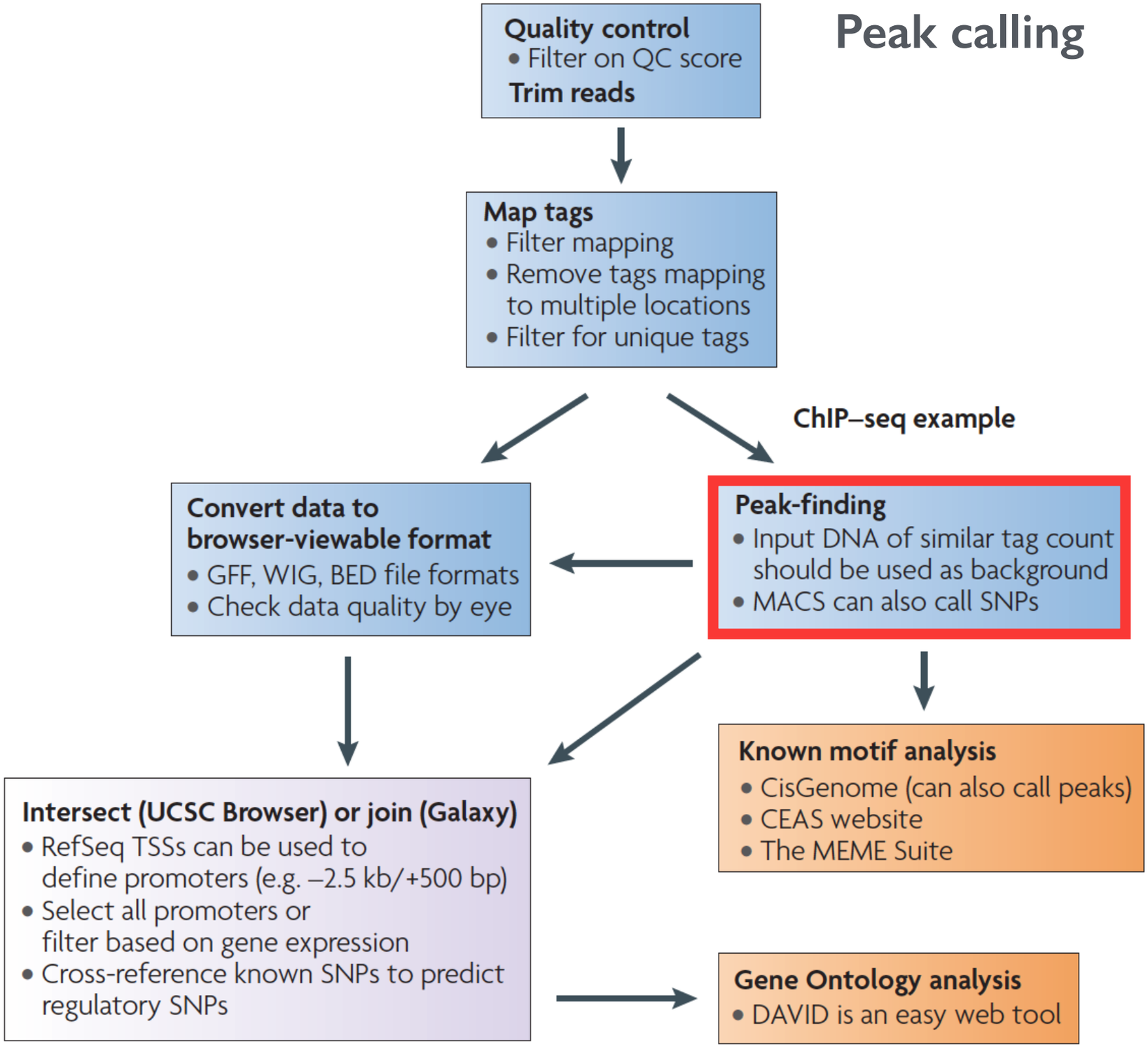
- Some clever tricks involved to achieve more compression of the data structures (FM-Index*)
- Use BWT on the reference genome to build the index
- Look up each read
- Convert to genome locations



How to map billions of short reads onto genomes. Trapnell & Salzberg. Nature Biotechnology 2009

*Ferragina & Manzini (2000). Opportunistic Data Structures with Applications. Proc. of the 41st Annual Symposium on Foundations of Computer Science

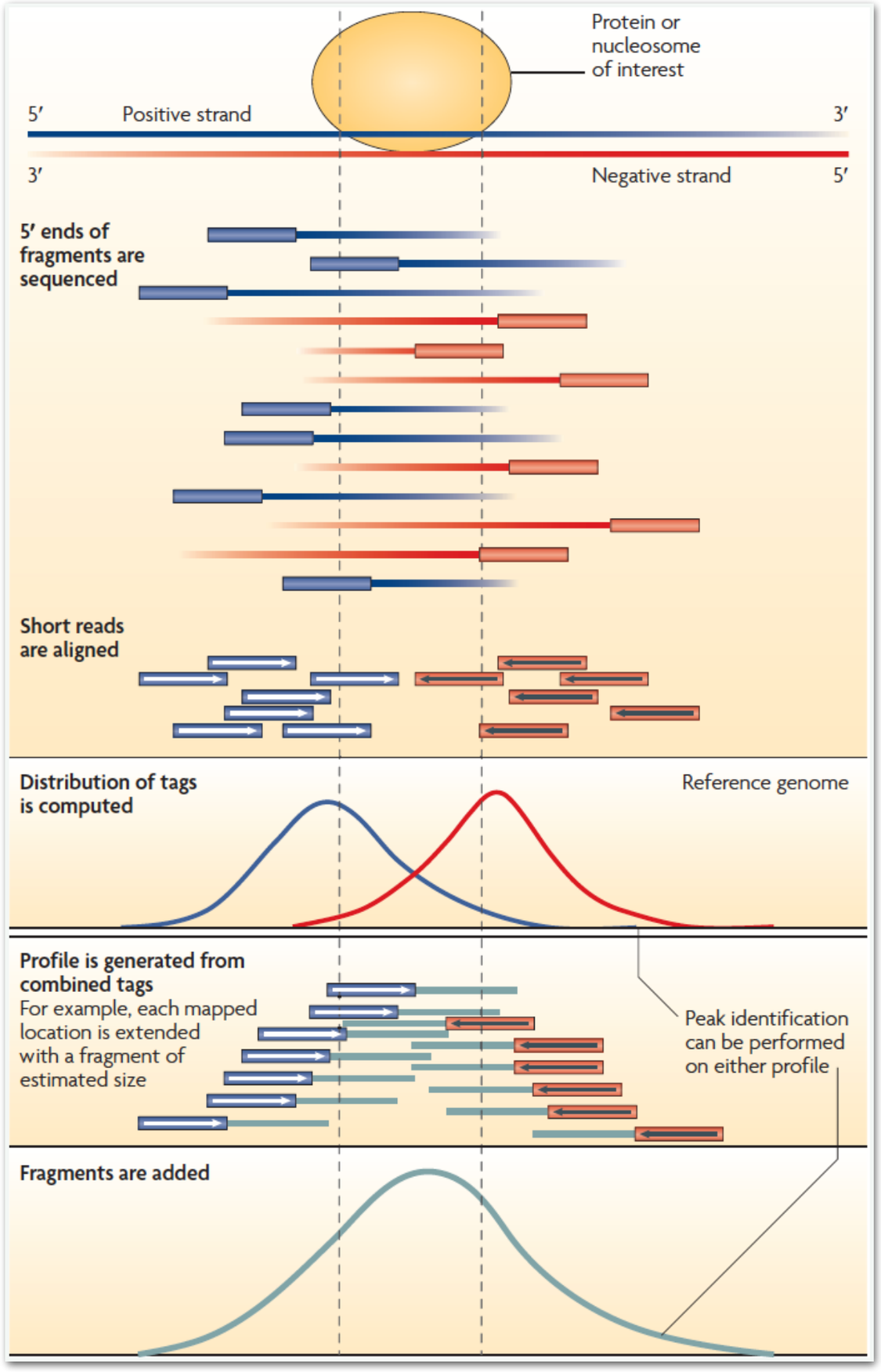
Peak calling



http://www.nature.com/nrg/journal/v11/n7/full/nrg2795.html

Peak calling

- ChIP profile
- Challenges
- MACS



ChIP profile

- Only 5' ends of ChIPed fragments are sequenced
  - Shifted read distribution
  - Expected symmetry between Watson/Crick read distributions

http://www.nature.com/nrg/journal/v10/n10/abs/nrg2641.html

Peak calling challenges

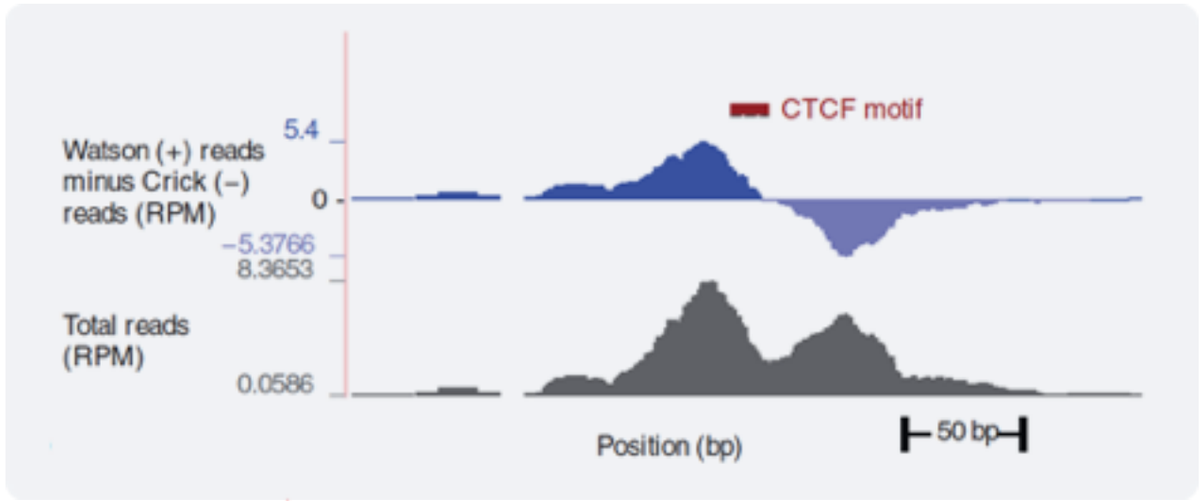
- Adjust for sequence mappability - regions that contain repetitive elements have different expected tag count

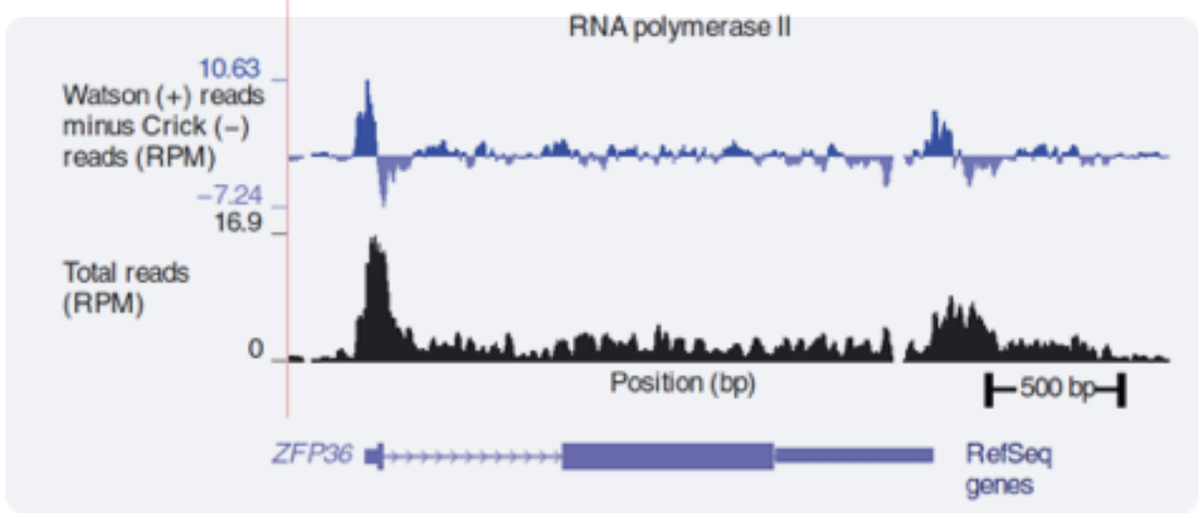
		Nonrepetitive sequence		Mappable sequence	
Organism	Genome size (Mb)	Size (Mb)	Percentage	Size (Mb)	Percentage
Caenorhabditis elegans	100.28	87.01	86.8%	93.26	93.0%
Drosophila melanogaster	168.74	117.45	69.6%	121.40	71.9%
Mus musculus	2,654.91	1,438.61	54.2%	2,150.57	81.0%
Homo sapiens	3,080.44	1,462.69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

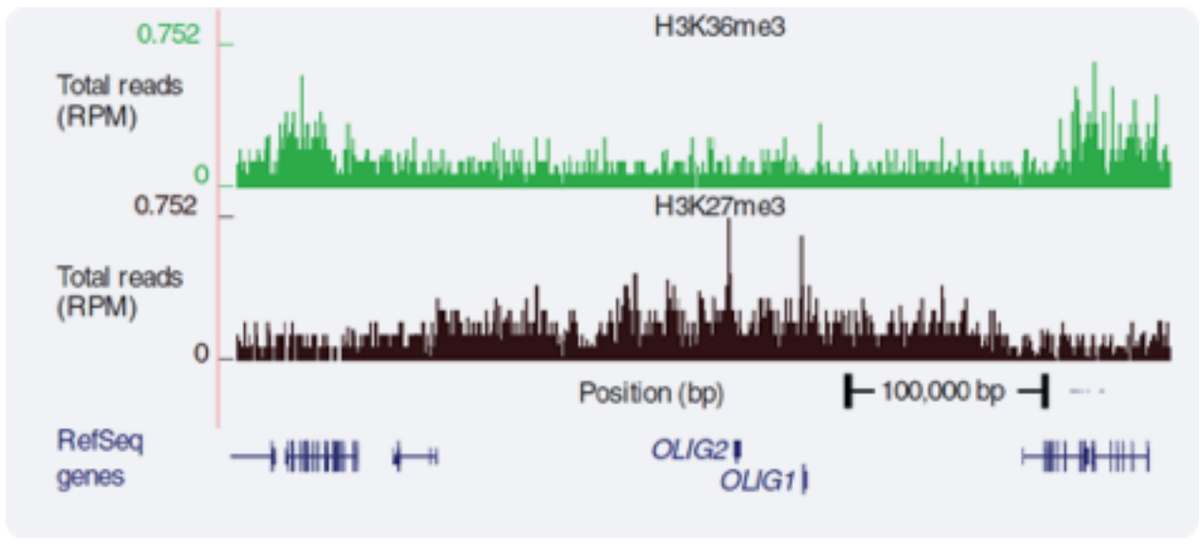
http://www.nature.com/nbt/journal/v27/n1/full/nbt.1518.html

Peak calling challenges

- Different ChIP-seq applications produce different type of peaks.
- Most current tools have been designed to detect sharp peaks (TF binding, histone modifications at regulatory elements)



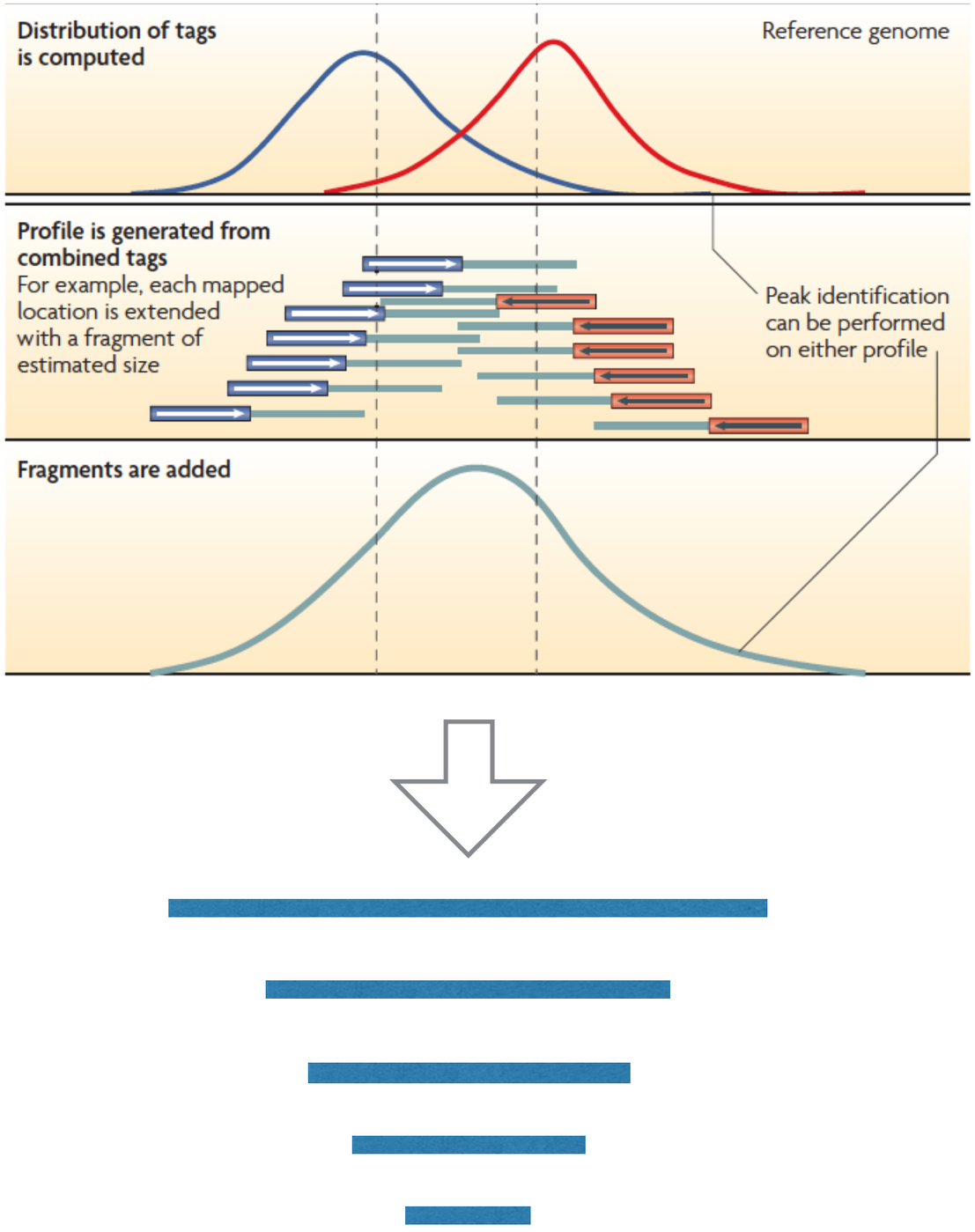




Computation for ChIP-seq and RNA-seq studies Pepke et al. Nat. Methods 2009

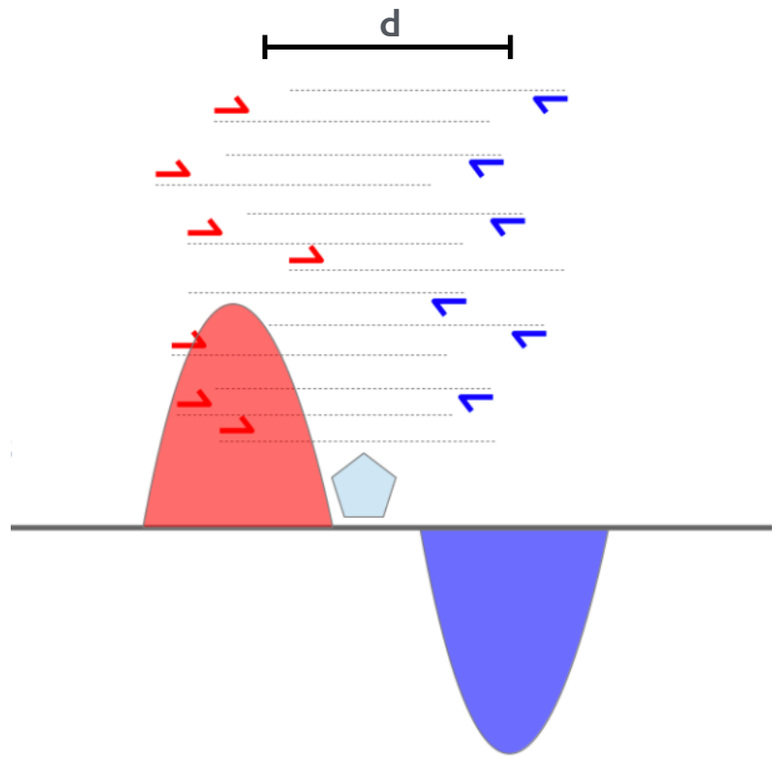
Peak calling challenges

- Definition of enriched regions/peaks:
  - Which statistic to used?
  - What boundaries should be reported?
  - What score to use (ratio, p-val, q-val)?
  - Compute/estimate a FDR?



http://www.nature.com/nrg/journal/v10/n10/abs/nrg2641.html

MACS



Step 1: Modelling the tag shift

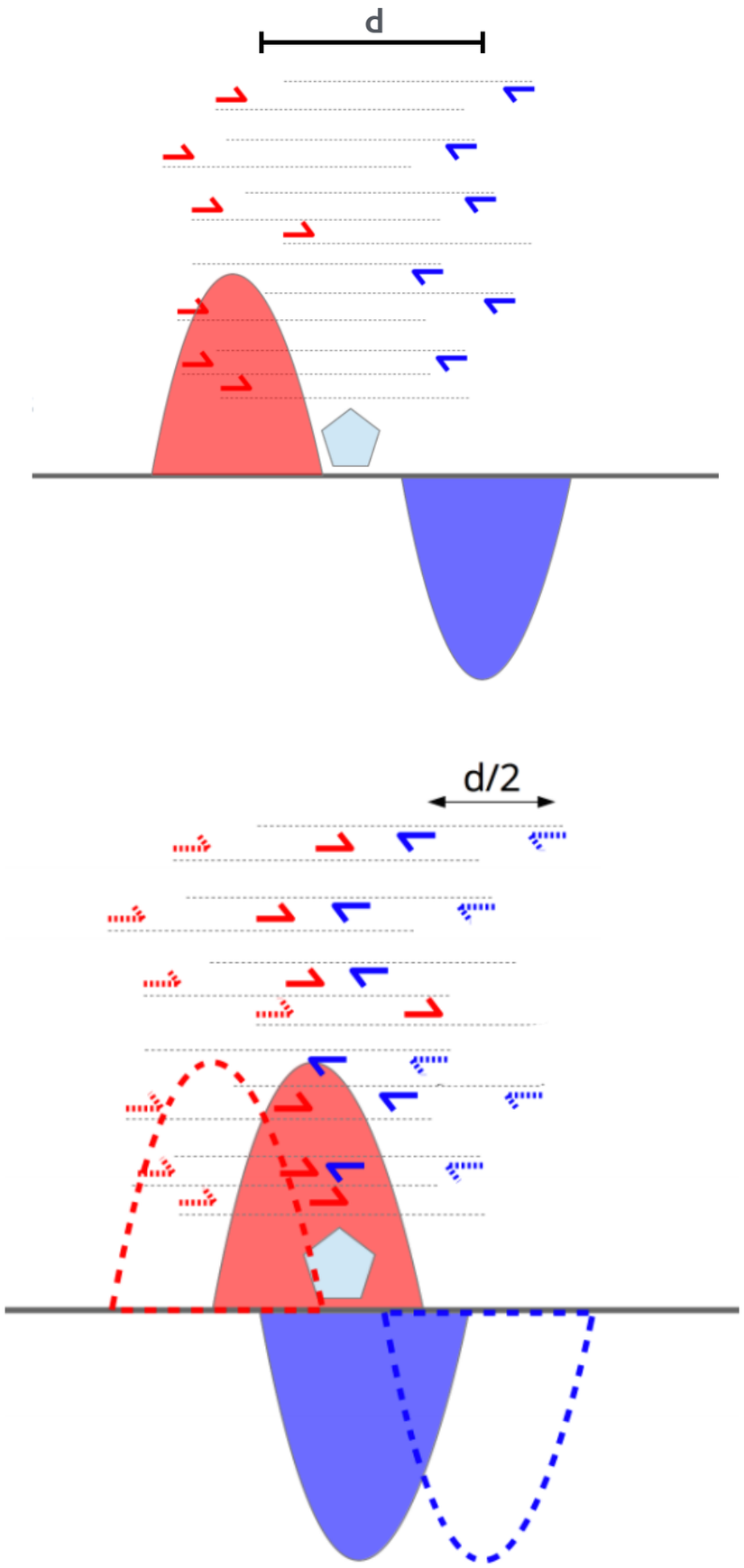
1. Scan genome with a window of user-defined sonication size
2. Keep the best 1000 (or less) peaks having a fold enr. > mfold (default 32, relative to random model)
3. Separate Watson/Crick tags
4. Shift size is modelled as the distance *d* between the modes of the Watson and Crick peaks

Model-based Analysis of ChIP-Seq (MACS). Zhang. et al. Genome Biology 2008
http://www.biologie.ens.fr/~mthomas/other/chip-seq-training/booklet/booklet_chip-seq.pdf

MACS

Step 2: Peak detection

1. Shift every tag by *d*/2
2. Slide a 2*d* window across the genome to find candidate peaks with significant tag enrichment (according to Poisson distribution, default p-value = 10^{-5})
3. Merge overlapping peaks
4. Report:
   - fold enrichment for called peaks: ratio between tag counts and expected using Poisson distribution (using input data if provided)
   - Position with highest pile-up is defined as the summit of peak
   - Empiric FDR if control sample is provided (sample swap), FDR = #control peaks / #ChIP peaks



Model-based Analysis of ChIP-Seq (MACS). Zhang. et al. Genome Biology 2008
http://www.biologie.ens.fr/~mthomas/other/chip-seq-training/booklet/booklet_chip-seq.pdf

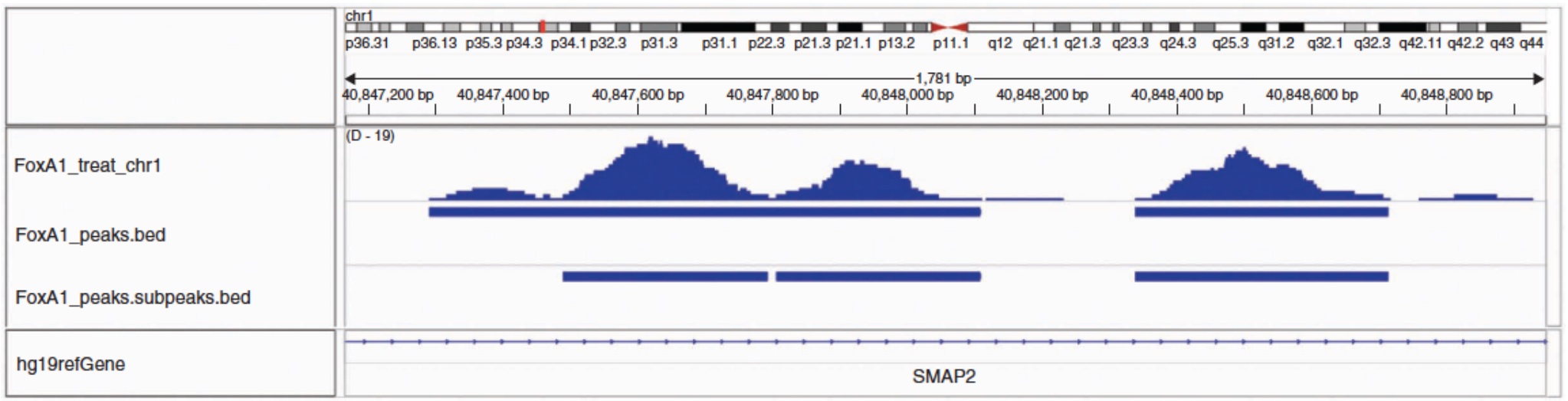
Again lots of choice

	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

and more…

Visualise to assess quality

- Assess the data quality e.g. positive controls, background
- Determine cutoffs (looking at positive controls)
- Compare different peak finder outputs
- Integration of data / co-visualization



Identifying ChIP-seq enrichment using MACS. Feng et al. Nat. Protocols, 2012

MASSEY UNIVERSITY
TE KUNENGA KI PŪREHUROA
UNIVERSITY OF NEW ZEALAND

References

Introduction to the Burrows-Wheeler Transform and FM Index. Ben Langmead, Department of Computer Science, JHU

How to map billions of short reads onto genomes. Trapnell & Salzberg. Nature Biotechnology 2009

Practical Guidelines for the Comprehensive Analysis of ChIP-seq Data. Bailey et al. PLoS Comp. Bio. 2013

Model-based Analysis of ChIP-Seq (MACS). Zhang. et al. Genome Biology 2008

Computation for ChIP-seq and RNA-seq studies Pepke et al. Nat. Methods 2009

Sebastian Schmeier
s.schmeier@gmail.com
http://sschmeier.github.io/bioinf-workshop/

THE ENGINE
OF THE NEW
NEW ZEALAND