Bioinformatics for High-throughput Sequencing

An Overview

Simon Anders
Nicolas Delhomme
EMBL-EBI

EBI is an Outstation of the European Molecular Biology Laboratory.

Overview

In recent years, new sequencing schemes, also called

- high-throughput sequencing
- massively parallel sequencing
- flow-cell sequencing

have been proposed.

Commercially available are devices from

- Roche (formerly: 454)
- Illumina (formerly: Solexa): “GenomeAnalyzer”
- Applied Biosystems: “SOLiD system”
- Helicos: “Helicoscope”

Core ideas

Two core differences of HTS to Sanger capillary sequencing:

- The library is not constructed by cloning, but by a novel way of doing PCR, where the fragments are separated by physico-chemical means (emulsion PCR or bridge PCR).
- Very many fragments are sequenced in parallel in a flow cell (as opposed to a capillary), observed by a microscope with CCD camera.

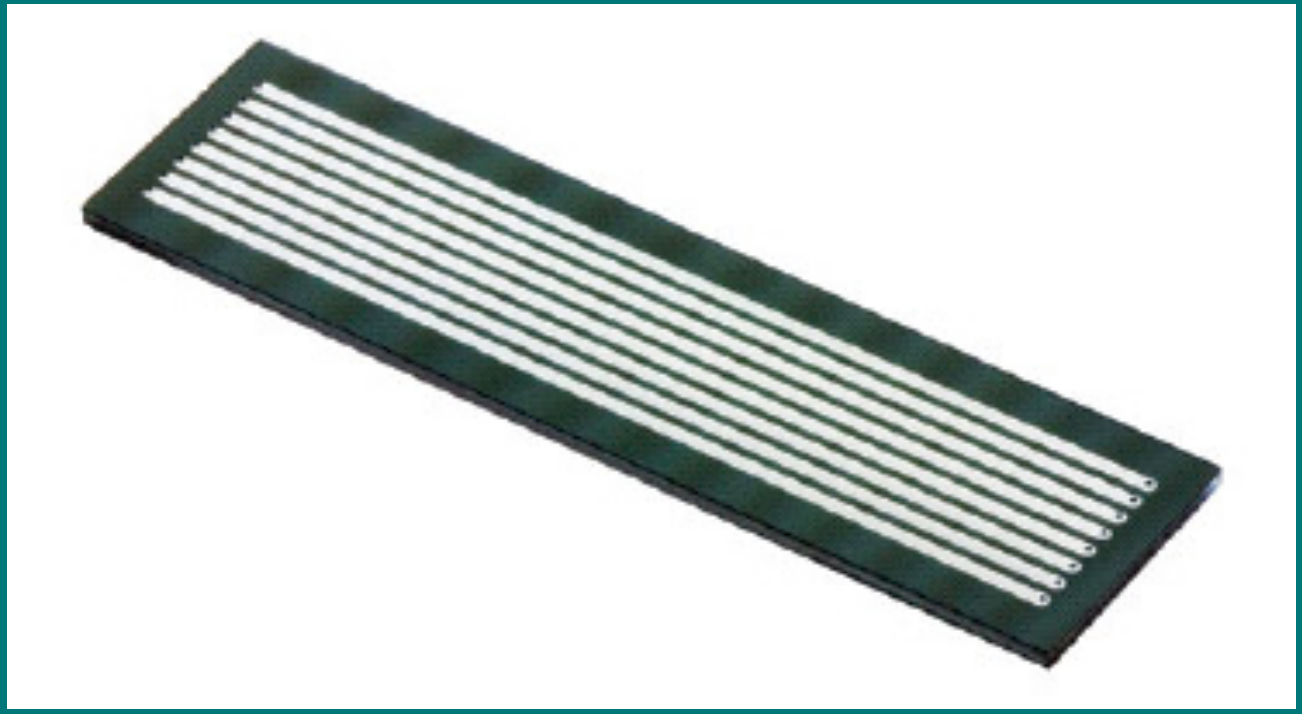
Solexa workflow

- Bridge PCD to prepare “clusters”
- Sequencing: 35 or more cycles x 4 bases, with micrographs taken in 300 tiles x 8 lanes

-> more than 1 terabyte of image data

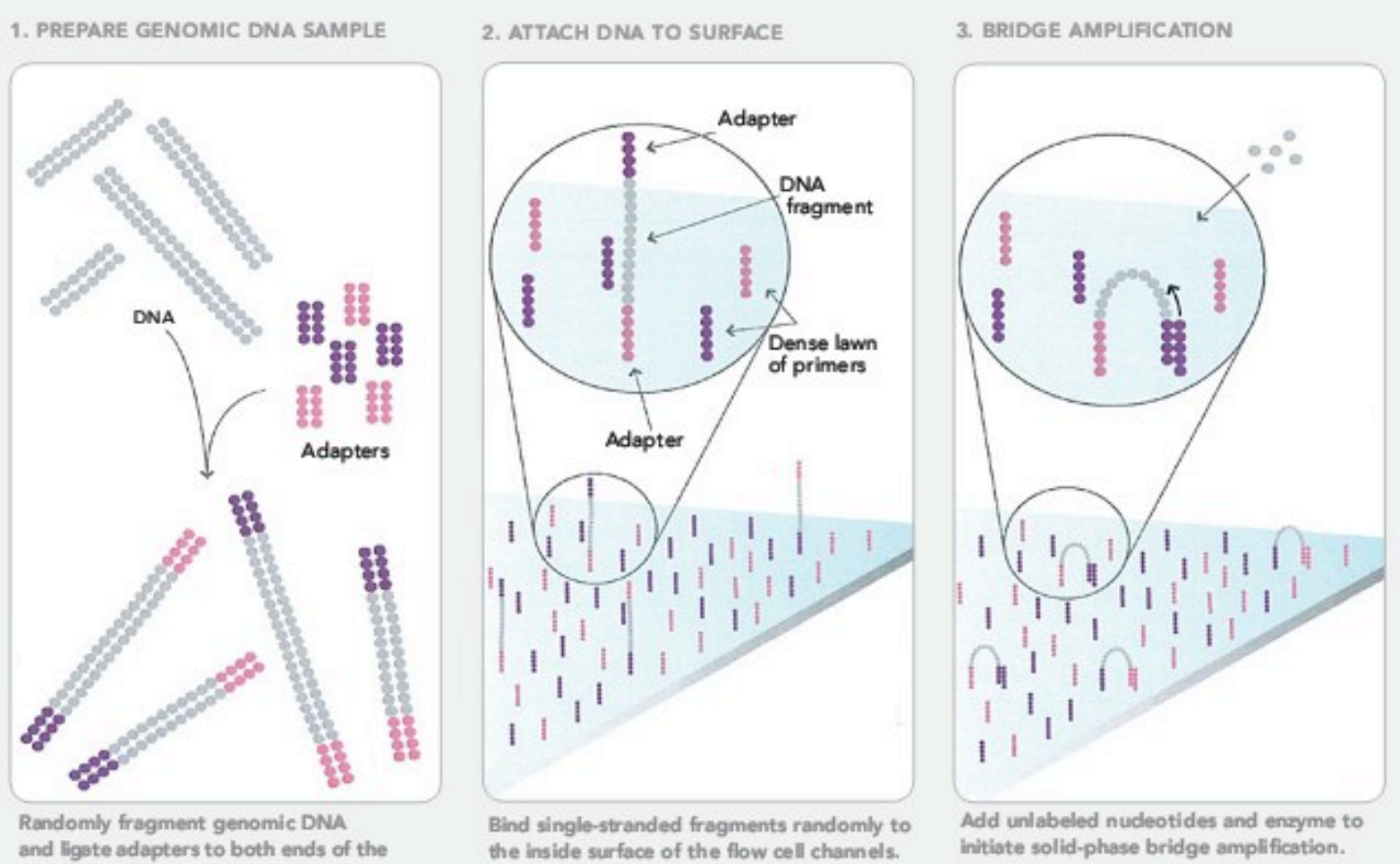
- “SolexaPipeline”: Sequences and alignment

Solexa: Flow cell

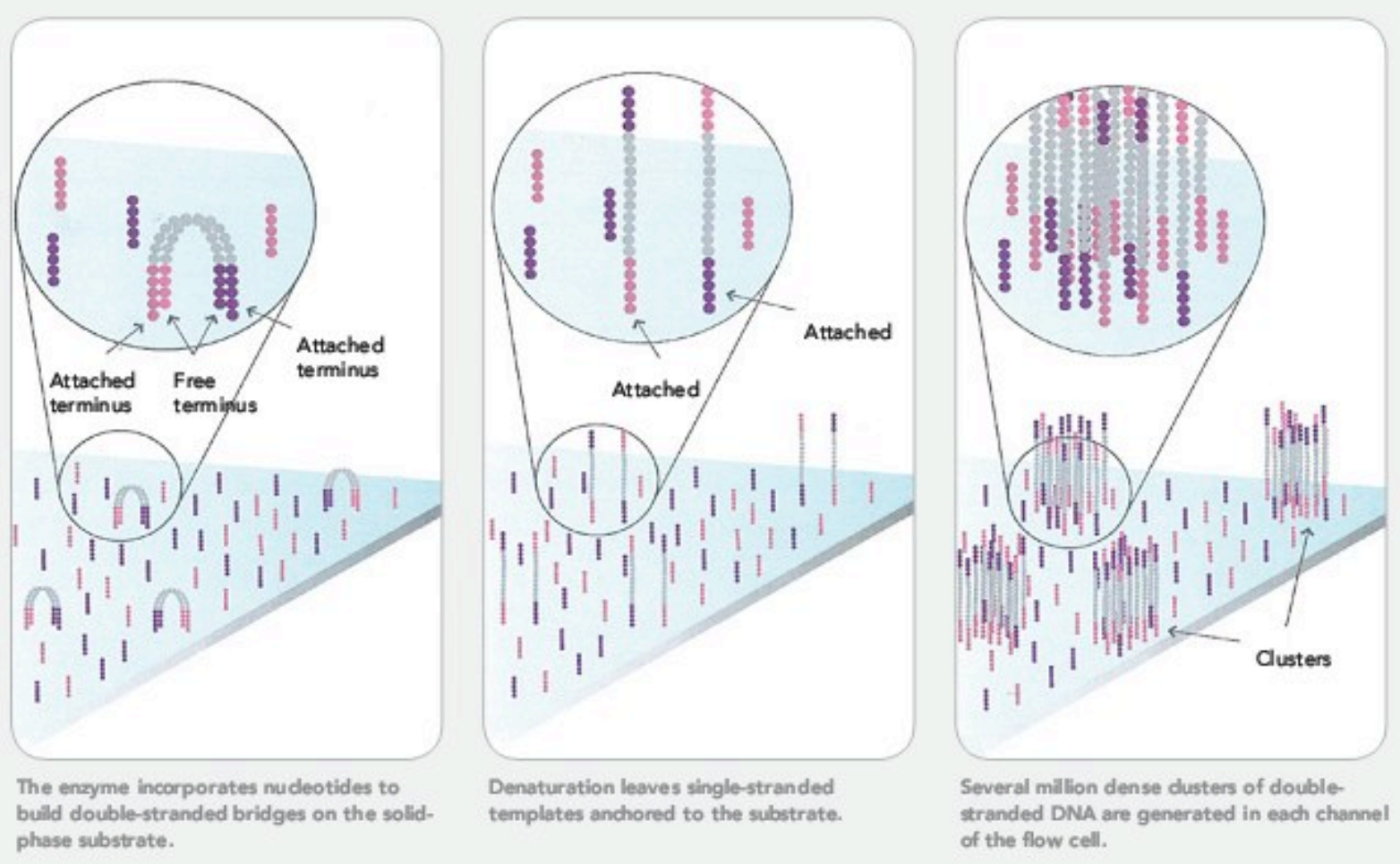


EMBL-EBI

Solexa: sample prepararation

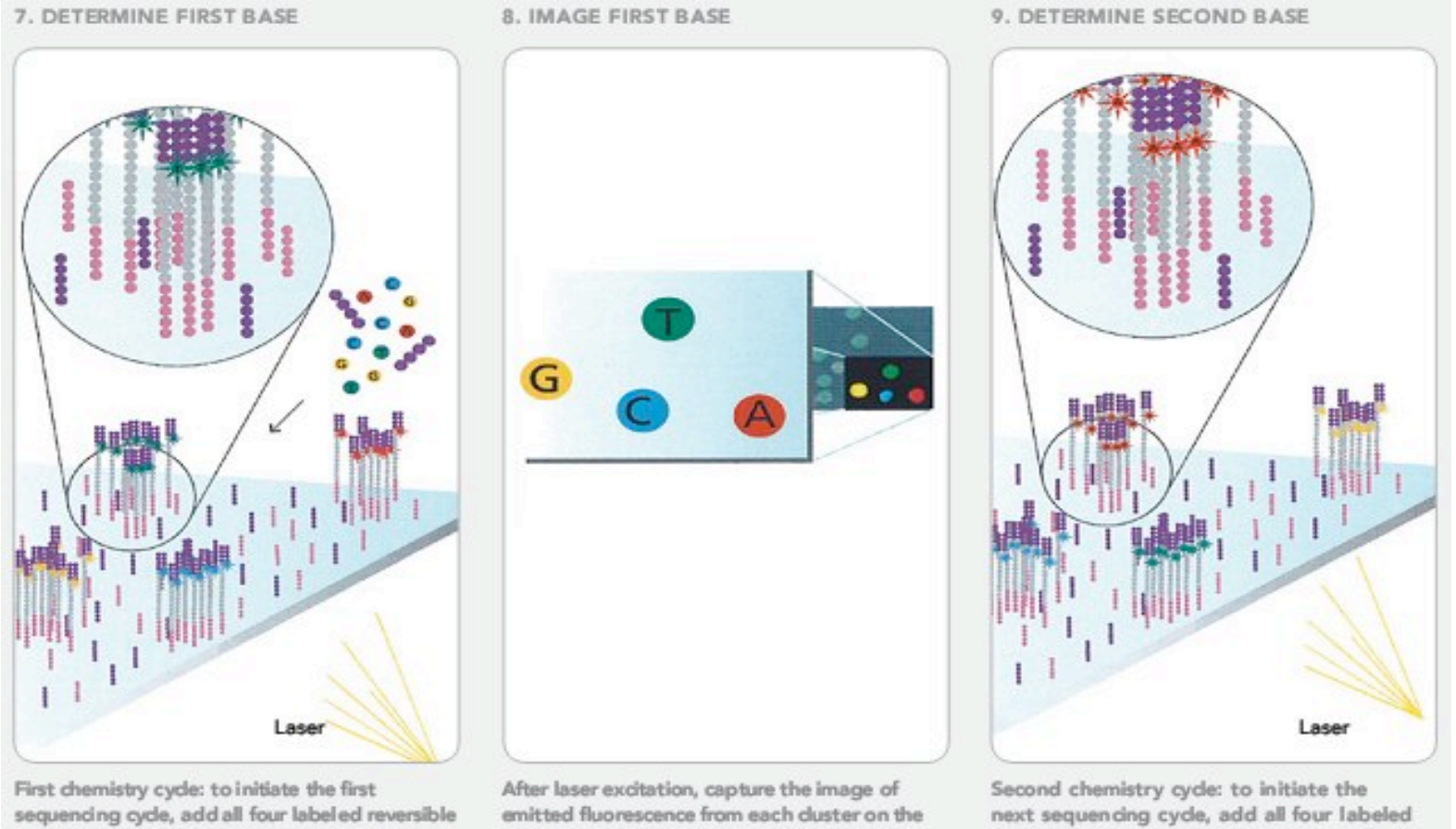


Solexa: sample prepararation

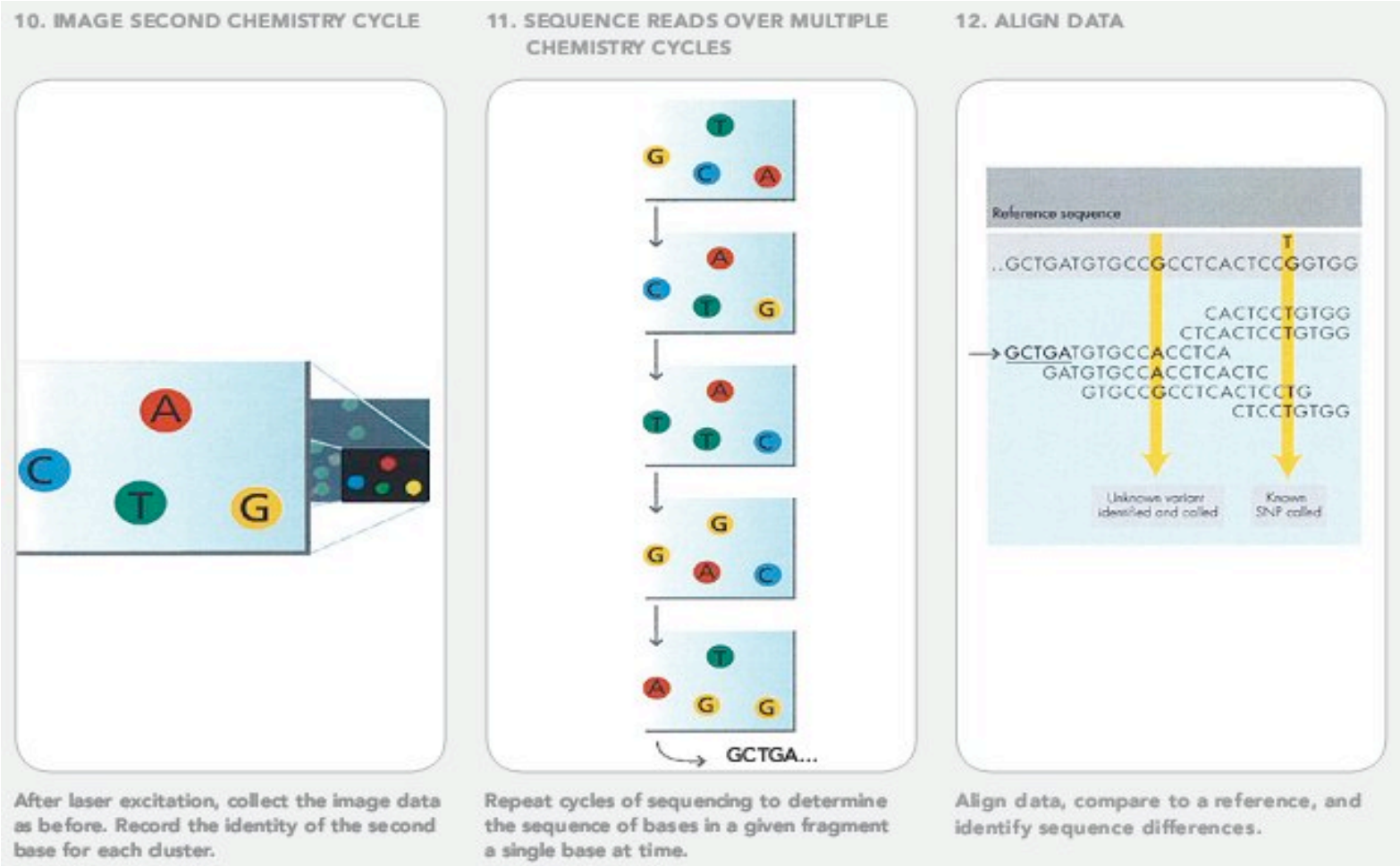


EMBL-EBI

Solexa: sequencing



Solexa: sequencing



Roche 454

- presented 2005, first on market
- emulsion PCR
- pyrosequencing (polymerase-based)
- read length: 250 bp
- paired read separation: 3 kb
- 300 Mb per day
- $60 per Mb
- error rate: around 5% per bp
- dominant type of error: indels, especially in homopolymers

Illumina / Solexa

- second on the market
- bridge PCR
- polymerase-based sequencing-by-synthesis
- 32..40 bp (newest models: up to 100 bp)
- paired read separation: 200 bp
- 400 Mb per day (getting better)
- $2 per Mb
- error rate: 1% per bp (good reads: 0.1%)
- dominant error type: substitutions

Applied Biosystems SOLiD

- third on market (since late 2007)
- emulsion PCR
- ligase-based sequencing
- read length: 50bp
- paired read separation: 3 kb
- 600 Mb per day (colour space)
- $1 per Mb
- very low error rate: <0.1% per bp
  (still high compared to Sanger capillary sequencing: 0.001%)
- dominant error type: substitutions (colour shift)

Helicos ("Helicoscope")

- on the market since a year
- no amplification
- single-molecule polymerase-based sequencing
- read length: 25..45 bp
- 1200 Mb per day
- $1 per Mb
- error rate: <1% (manufacturer claim)

Comparison data from:
- E Mardis, Trends in Genetics 24 (2008) 133
- R A Holt, S J M Jones, Genome Res 18 (2008) 839
- J Shendure, H Ji, Nature Biotech 26 (2008) 1135

Polonator

- on the market since less than a year
- emulsion PCR
- ligase-base sequencing
- very short read-length: 13 bp
- but: low-cost instrument ($150,000)
- <$1 per Mb

Use-cases for HTS

- de-novo sequencing and assembly of small genomes
- transcriptome analysis (RNA-Seq, sRNA-Seq, ...)
  - identifying transcripted regions
  - expression profiling
- Resequencing to find genetic polymorphisms:
  - SNPs, micro-indels
  - CNVs
- ChIP-Seq, nucleosome positions, etc.
- DNA methylation studies (after bisulfite treatment)
- environmental sampling (metagenomics)
- reading bar codes

Multiplexing

- Solexa now generates 6-12 mio. 36bp reads per lane.
- Using a lane for a single sample is often wasteful.
- *Multiplexing*: incorporate tags between sequencing primer and sample fragments to distinguish several samples in the same lane

Coming soon: Targeted sequencing

- Currently, one always samples the whole genome, which is wasteful if one is interested in only a specific region.
- Microarrays allow to select fragments of interest.

Paired-end sequencing: Principle

The two ends of the fragments get different adapters.

Hence, one can sequence from one end with one primer, then repeat to get the other end with the other primer.

This yields “pairs” of reads, separated by a known distance (200bp for Illumina).

For large distances, “circularisation” might be needed and generates “mate pairs”.

Paired ends: Uses

Paired-end sequencing is useful

- to find micro-indels
- to find copy-number variations
- for assembly tasks
- to look for splice variants

but of little value for

- standard ChIP-Seq
- “normal” RNA-Seq (not looking for “unknown” transcripts)

Use cases for HTS: Bioinformatics challenges

Established procedures may not be suitable.

New algorithms are required for

- assembly
- alignment
- statistical tests (counting statistics)
- visualization
- segmentation
- ...

Solexa standard workflow

EMBL-EBI

SolexaPipeline

- "Firecrest": Identifying clusters

⇒ typically 3..5 mio clusters per lane

- "Bustard": Base calling

⇒ sequence for each cluster,
with Phred-like scores

- "Eland": Aligning to reference

Firecrest output

Large tab-separated text files with one row per identified cluster, specifying

- lane index and tile index
- x and y coordinates of cluster on tile
- for each cycle a group of four number, specifying the flourescence intensity for A, C, G, and T.

Bustard output

Two tab-seperated text files, with one row per cluster:

- "seq.txt" file:
  - lane and tile index, x and y coordinates
  - the called sequence as string of A, C, G, T
- "prb.txt" file:
  - Phred-like scores, ranging from -40 to 40;
  - one value per called base

Fastq format

"FASTA with Qualities"

Example:

```
@HWI-EAS225:3:1:2:854#0/1
GGGGGGAAGTCGGCAAAATAGATCCGTAACTTCGGG
+HWI-EAS225:3:1:2:854#0/1
a`abbbbabaabbbabab^`[aaa`_N]b^ab^``a
@HWI-EAS225:3:1:2:1595#0/1
GGGAAGATCTCAAAAACAGAAGTAAAACATCGAACG
+HWI-EAS225:3:1:2:1595#0/1
a`abbbabab bbabbbbbbabb`aaababab\aa_`
```

Fastq format

Each read is represented by four lines:

- '@', followed by read ID
- sequence
- '+', optionally followed by repeated read ID
- quality string:
  - same length as sequence
  - each character encodes the base-call quality of one base

Fastq format: Base-call quality strings

- If *p* is the probability that the base call is wrong, the (standard Sanger) Phred score is:

$$Q_{\text{Phred}} = -10 \log_{10} p$$

  - The score is written with the character whose ASCII code is $Q_{\text{Phred}}+33$
- Solexa uses a different convention:

$$Q_{\text{Solexa}} = -10 \log_{10} (p / (1-p))$$

  - For high-quality bases, $Q_{\text{Phred}} \approx Q_{\text{Solexa}}$
  - The character has ASCII code $Q_{\text{Solexa}}+64$
- SolexaPipeline 1.3 changed it again

FASTQ: Phred base-call qualities

quality score Q_{phred}	error prob. *p*	characters
0 .. 9	1 .. 0.13	!”#$%&'()*
10 .. 19	0.1 .. 0.013	+,-./01234
20 .. 29	0.01 .. 0.0013	56789:;<=>
30 .. 39	0.001 .. 0.00013	?@ABCDEFGH
40	0.0001	I

The Sanger / Solexa FASTQ confusion

- Solexa has chosen to base their quality on the odds instead of the proability of an error.
- Also, they use a different offset, and hence different characters:

- Most tools (e.g., Maq, Bowtie, BWA) expect Sanger scores by default, so you have to either convert the scores or tell the tool.
- New: SolexaPipeline 1.3 has changed the definition of *Q* again!

Raw vs. calibrated base-call qualities

- The raw quality values reported by the Bustard base caller are estimates based on the intensity values reported by Firecrest.
  - If one colour is clearly brighter, a good quality is given.
  - If the two brightest colours are comparable in intensity, a bad quality is indicated.
- To interpret the scores reliably as probabilities, they need to be calibrated.
- This can be done by looking at mismatches in aligned reads.

FASTQ and paired-end reads

Convention for paired-end runs:

The reads are reported in two FASTQ files, such that the n^{th} read in the first file is paired to the n^{th} read in the second file. The read IDs must match.

Short Read Alignment: an overview

Simon Anders
Nicolas Delhomme

EMBL-EBI

Purpose

New generation DNA sequencers provide billions of bases rapidly and inexpensively

Millions of short reads (25-100bp) rather than a "few" long ones

Their position within a reference sequence has to be determined → "read mapping"

Typical input data for alignment

- Illumina/Solexa: 100 million 75+75bp read pairs in a run
- ABI/SOLiD: similar in scale (50+50bp)
- Roche/454: ~300-500bp reads, 100Mbp a run

✓ Currently a little more expensive in terms of money/ base-pair

Courtesy of Heng Li (Welcome Trust Sanger Institute)

Challenges of mapping short reads

Speed: if the genome is large and we have billions of reads?

Using traditional tools like BLAST/BLAT would require 100s CPU hours.

Table 2. Mapping efficiency compared to BLAST, BLAT, RMAP and Mosaik on BAC data

Program	BAC on MHC-162k	BAC on chr6	BAC on all
BLAST	06:56:11 (51M)	>5 days	>8 days
BLAT	00:04:06 (32M)	06:33:03 (32M)	7 days+22:47:16(32M)
RMAP	00:00:51 (1.9G)	00:27:54 (1.9G)	10:09:03 (1.9G)
Mosaik	00:05:33 (214M)	00:07:41 (3.4G)	02:11:15 (3.5G)
ZOOM	00:00:37 (1.1G)	00:06:09 (1.1G)	01:33:03 (1.1G)

Time is represented as hh:mm:ss.

BAC dataset: 3 415 291 reads; Lin, H. *et al.*, 2008

Memory:

- Suffix array requires 12GB for human genome
- Indexing reads or better in-memory index

Additional Challenges

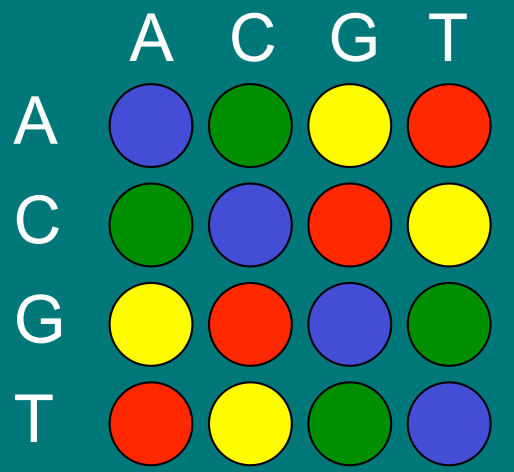
- Read errors
  - dominant cause for mismatches
  - detection of substitutions?
  - Importance of the base-call quality (“phred scores”)
- Unknown reference genome
  - “de-novo” assembly
- Repetitive regions / Accuracy
  - ~20% of the human genome are repetitive to 32bp reads
  - Importance of paired-end information

Technical Challenges

Sequencer difference:

- 454: longer reads require different tools

- SOLiD:
  - use color space
  - distinct seq. error from polymorphism



  - deletion shift the colors
  - not easy to convert into “base” space
    - ➔ has to be aligned against color space reference

Additional Technical Challenges

Atypical reference treatment: e.g. bisulfite treatment

Alignment software

In the last two years, many tools for short-read alignments have been published:

- Eland
- Maq
- Bowtie
- BWA
- SOAP(2)
- Biostrings
- SSAHA2, RMAP, SHRiMP, ZOOM, Novoalign, Mosaik, Slider, ...

Table 1 A selection of short-read analysis software

Program	Website	Open source?	Handles ABI color space?	Maximum read length
Bowtie	http://bowtie.cbcb.umd.edu	Yes	No	None
BWA	http://maq.sourceforge.net/bwa-man.shtml	Yes	Yes	None
Maq	http://maq.sourceforge.net	Yes	Yes	127
Mosaik	http://bioinformatics.bc.edu/marthlab/Mosaik	No	Yes	None
Novoalign	http://www.novocraft.com	No	No	None
SOAP2	http://soap.genomics.org.cn	No	No	60
ZOOM	http://www.bioinfor.com	No	Yes	240

Trapnell, C. & Salzberg, S.L., 2009

Short read aligners: Differences

Alignment tools differ in

- speed
- suitability for use on compute clusters
- memory requirements
- accuracy
  - Is a good match always found?
  - What is the maximum number of allowed mismatches?
- ease of use
- available down-stream analysis tools
  - Are there SNP and indel callers that can deal with the tool's output format?
  - Is there an R package to read in their output?

Short read aligners: Differences

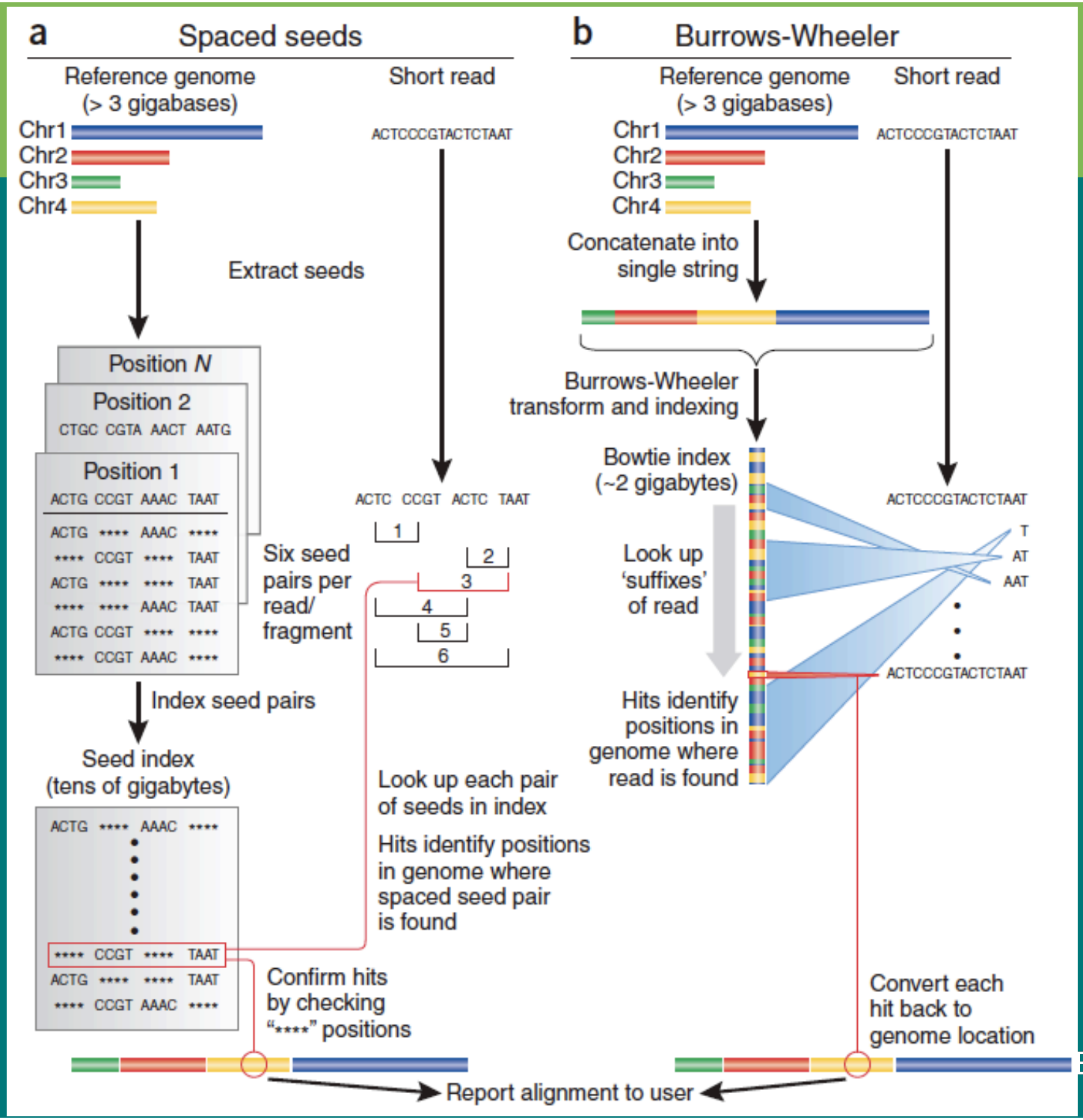
Alignment tools also differs in whether they can

- make use of base-call quality scores
- estimate alignment quality
- work with paired-end data
- report multiple matches
- work with longer than normal reads
- match in colour space (for SOLiD systems)
- align data from methylation experiments
- deal with splice junctions

Review of alignment algorithms

- Hashing the reference genome:
  - ✓ Pros: straightforward; easy multi-threading
  - ✓ Cons: large memory
- Hashing read sequences:
  - ✓ Pros: flexible memory footprint
  - ✓ Cons: multi-threading is hard
- Alignment by merge sorting:
  - ✓ Pros: flexible memory
  - ✓ Cons: hard for pairing?
- Indexing genome by BWT: (BWT = Burrows-Wheeler Transform)
  - ✓ Pros: fast and relatively small memory footprint
  - ✓ Cons: not applicable to long reads at the moment

Courtesy of Heng Li (Welcome Trust Sanger Institute)



Trapnell, C. & Salzberg, S.L., 2009

EMBL-EBI

Short read alignment: Algorithms

The Burrows-Wheeler transform seems to be the winning idea:

- very fast
- sufficiently accurate
- used by the newest tools (Bowtie, SOAPv2, BWA).

Popular alignment tools

- Eland (Solexa)
  - supplied by Illumina as part of the Solexa Pipeline
  - very fast
  - does not make use of quality scores
- Maq (Li *et al.*, Sanger Institute)
  - widely used
  - interprets quality score and estimates alignment score
  - downstream analysis tools (SNP, indel calling)
  - can deal with SOLiD colour space data
  - being replaced by BWA
- Bowtie (Langmead *et al.*, Univ of Maryland)
  - based on Burrows-Wheeler transform
  - very fast, good accuracy
  - downstream tools available

Short-read algorithms: Seed matches

Maq claims to find all alignments with up to 2 mismatches and may find alignments with more than two mismtaches.

How comes?

Aligning hashed reads

Naive algorithm:

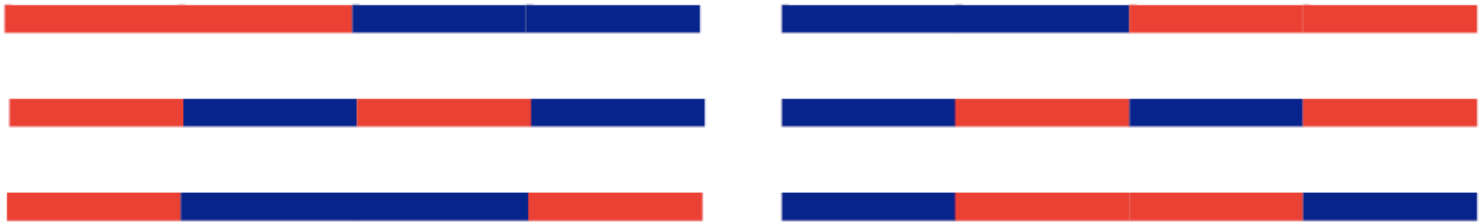
- Make a hash table of the first 28mers of each read, so that for each 28mer, we can look up quickly which reads start with it.
- Then, go through the genome, base for base. For each 28mer, look up in the hash table whether reads start with it, and if so, add a note of the current genome position to these reads.

Problem: What if there are read errors in the first 28 base pairs?

MAQ: basic algorithm

- Index reads and scan the genome.
  - ✓ Avoid aligning too few reads
- 28bp seed; Eland-like indexing
  - ✓ Able to find more mismatches beyond the seed
- Guarantee to find 2-mismatch seed hits

Seed templates:



Spaced seeds

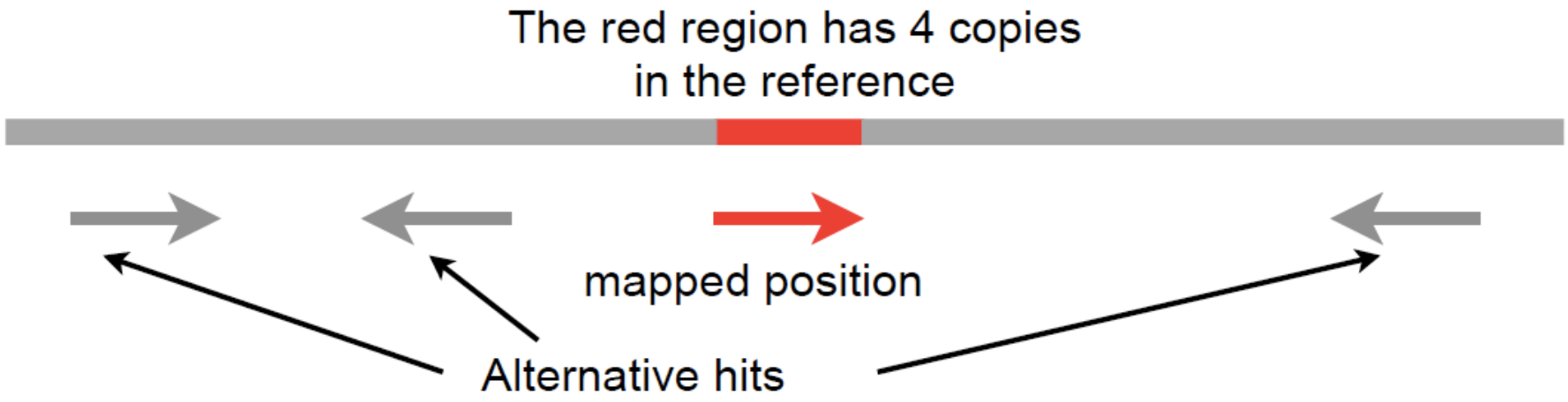
Maq prepares six hash table, each indexing 28 of the first 36 bases of the reads, selected as follows:

Hence, Maq finds all alignments with at most 2 mismatches in the first 36 bases.

EMBL-EBI

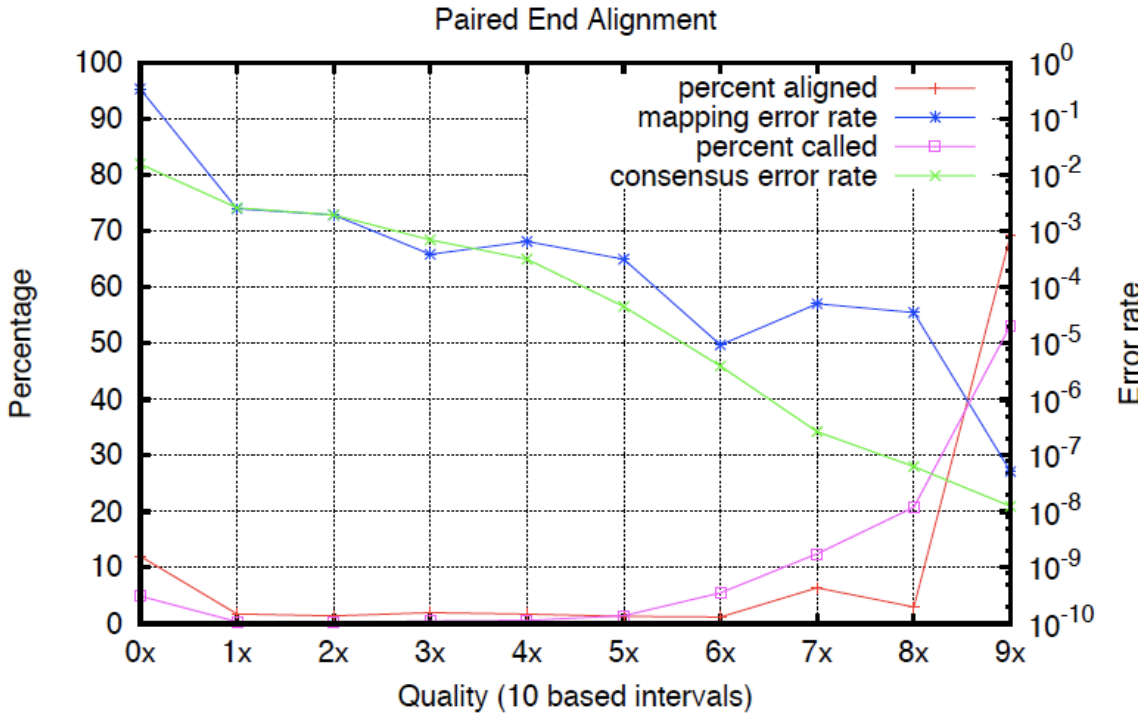
MAQ: random mapping

- Randomly place a read if it has multiple equally best hits
- Advantages:
  - ✓ tell if a read is mapped
  - ✓ tell if a region has reads mapped (avoid holes due to repeats)



MAQ: mapping quality

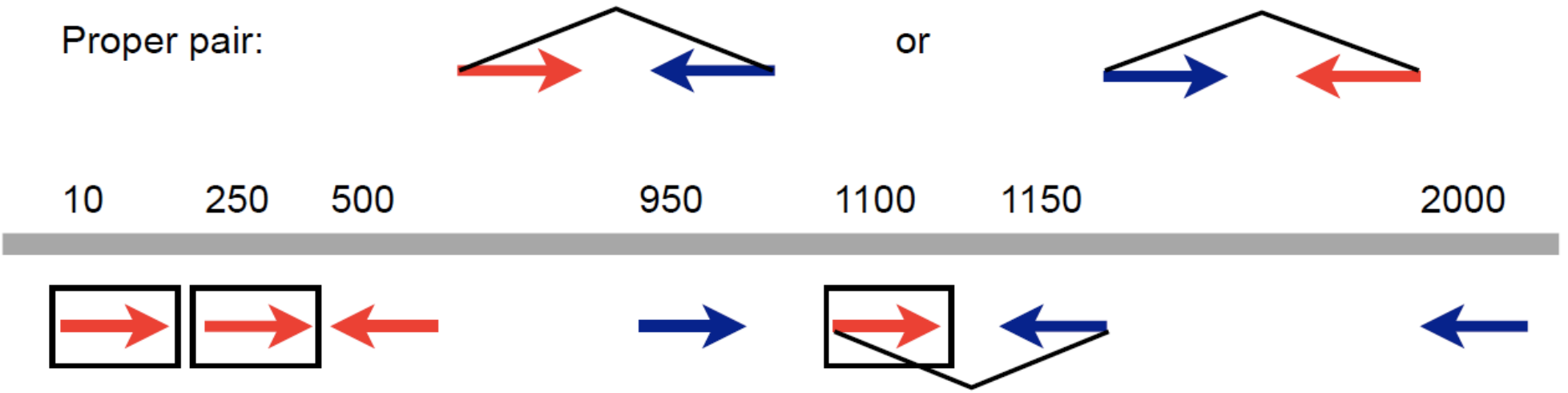
- Mapping quality is the phred-scaled probability of the alignment being wrong.
- Discriminate good mappings from bad ones, e.g.:
  - ✓ repetitive reads
  - ✓ top hit is perfect but there are 100 1-mismatch hits
  - ✓ top hit is perfect but the second best hit has one Q5 mismatch
- Proved to be effective for SV detections where wrong alignments dominate.



Courtesy of Heng Li (Welcome Trust Sanger Institute)

MAQ: PET mapping

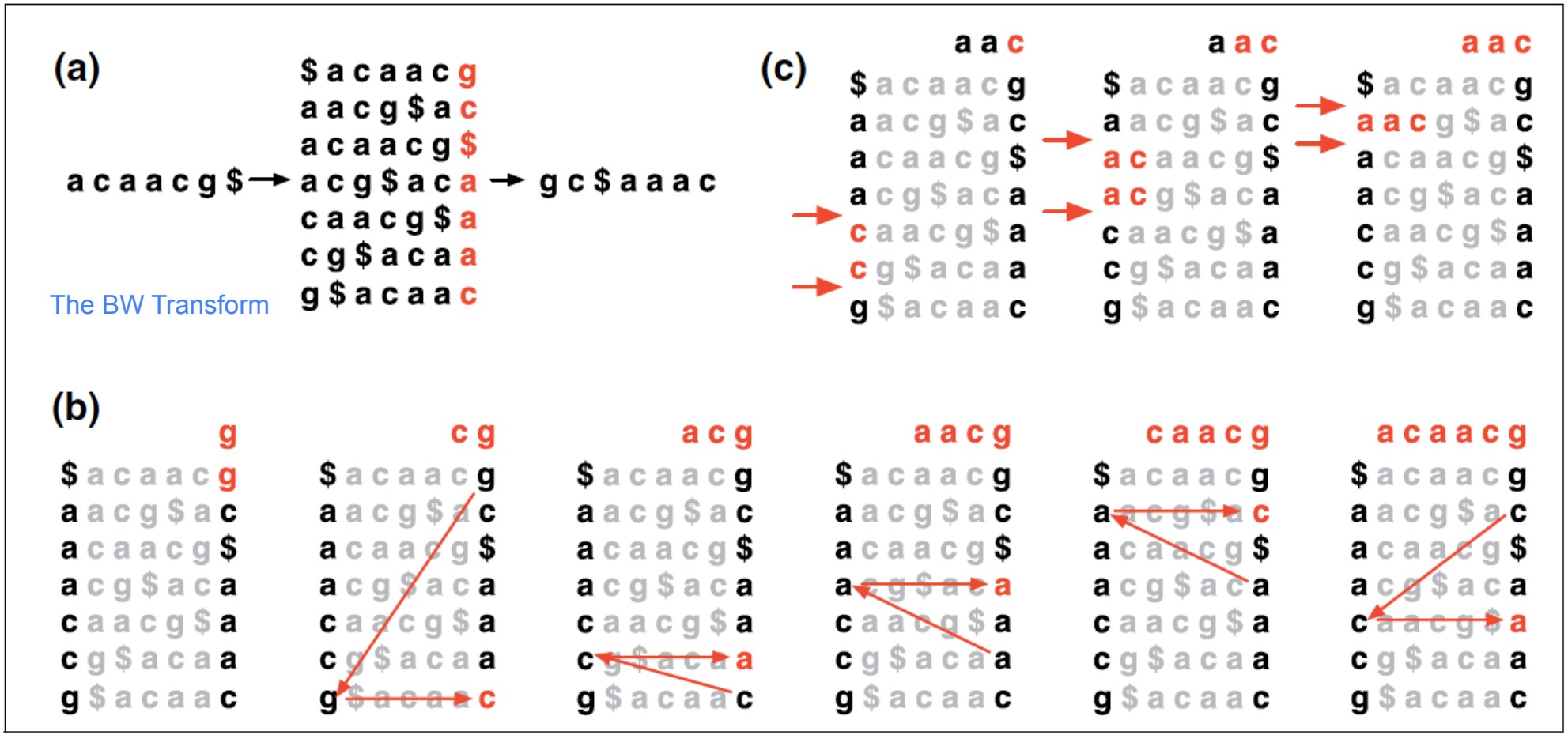
- Hit to a read found on the forward strand: keep the position in a 2-element queue
- Hit to a read found on the reverse strand: check the positions in the queue of its mate



MAQ

- Pros
  - paired-end able (gapped alignment)
  - adapter trimming
  - SOLiD support
    - alignment decoding for color reads
    - correct color error after the alignment
- Cons
  - “slow”
  - does not support Helicos
  - no gapped alignment for single reads

Bowtie: Burrows-Wheeler indexing



First Last Mapping

first occurrence of g → second occurrence of c

third occurrence of a → first occurrence of a

second occurrence of c → third occurrence of a

and so on…

EMBL-EBI

Bowtie

- Reference genome suffix arrays are BW transformed and indexed
- Model organism genome indexes are available for download from the Bowtie webpage

Bowtie alignment performance versus SOAP and Maq

	Platform	CPU time	Wall clock time	Reads mapped per hour (millions)	Peak virtual memory footprint (megabytes)	Bowtie speed-up	Reads aligned (%)
Bowtie -v 2	Server	15 m 7 s	15 m 41 s	33.8	1,149	-	67.4
SOAP		91 h 57 m 35 s	91 h 47 m 46 s	0.10	13,619	351×	67.3
Bowtie	PC	16 m 41 s	17 m 57 s	29.5	1,353	-	71.9
Maq		17 h 46 m 35 s	17 h 53 m 7 s	0.49	804	59.8×	74.7
Bowtie	Server	17 m 58 s	18 m 26 s	28.8	1,353	-	71.9
Maq		32 h 56 m 53 s	32 h 58 m 39 s	0.27	804	107×	74.7

Bowtie

- Pros
  - small memory footprint (1.3GB for the human genome)
  - fast (8M reads aligned in 8 mins against the Drosophila genome)
  - paired-end able (gapped alignment)
- Cons
  - less accurate than MAQ
  - does not support SOLiD, Helicos
  - no gapped alignment

Other commonly used aligners

- SOAP and SOAP2 (Beijing Genomics Institute)
  - with downstream tools
  - SOAP2 uses BWT
- SSAHA, SSAHA2 (Sanger Institute)
  - one of the first short-read aligners
- Exonerate (EBI)
  - not really designed for short reads but still useful
- Biostrings (Bioconductor)
  - R package, see M. Morgan's talk

References

Langmead, B. et al., 2009. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology, 10(3), R25.*

Lin, H. et al., 2008. ZOOM! Zillions of oligos mapped. Bioinformatics, 24(21), 2431-2437.

Trapnell, C. & Salzberg, S.L., 2009. How to map billions of short reads onto genomes. Nat Biotech, 27(5), 455-457.

Short Read Alignment: extended usage

Simon Anders
Nicolas Delhomme

EMBL-EBI

What next?

NGS offers the possibility to sequence anything and aligning the reads against “reference” genome is straightforward.

But what if there is no such “reference” genome?

➔ “de novo” assembly

Aligning the reads is the only first step

Assembly

- Solexa reads are too short for *de novo* assembly of large genomes.
- However, for prokaryotes and simple eukaryotes, reasonably large contigs can be assembled.
- Using paired-end reads with very large end separation is crucial.
- Assembly requires specialized software, typically based on so-called de-Brujin graphs
- Most popular assembly tool:
  - Velvet (Zerbino et al.)
  - ABySS (Simpson et al.)

Recommended alignment procedures

- Long reads: BLAT/SSAHA2/cross_match/Mosaik
- Long reference (e.g. human genome), short reads:
  - ✓ BWA for initial mapping (for speed)
  - ✓ NovoAlign for unmapped/unpaired reads (for accuracy, in particular for detecting structural variations)
  - ✓ Local *de novo* assembly with PE reads for structural variations
- Short reference (e.g. bacterial genome), short reads:
  - ✓ Speed/memory is less critical
  - ✓ *De novo* assembly + cross_match contigs (to find variants in highly variable regions, but require deep coverage)

Courtesy of Heng Li (Welcome Trust Sanger Institute)

Paired-end alignment

When aligning matepaired-end data, the aligner can use the information that mate-paired reads have a known separation:

- Try to align the reads individually
- Then, for each aligned read, attempt to align the mate in a small window near the first read's position with a more sensitive algorithm, e.g., Smith-Waterman to allow for gaps.
  - Be sure to tell the aligner the minimal and maximal separation.
- This allows to find small indels.

Resequencing: SNP calling

HTS is ideally suited for re-sequencing

- If a base differs from the reference in most reads that are aligned to this locus, it is a likely SNP
- If the difference occurs in half of the reads, it is a heterozygous SNP.
- If it appears in only a few reads, it could also be a read error.

- Calculating a *p*-value for a SNP call is straight-forward
  - Complication: Include base-call and alignment qualities as priors; interdependence of bases causes bias

Software for SNP calling

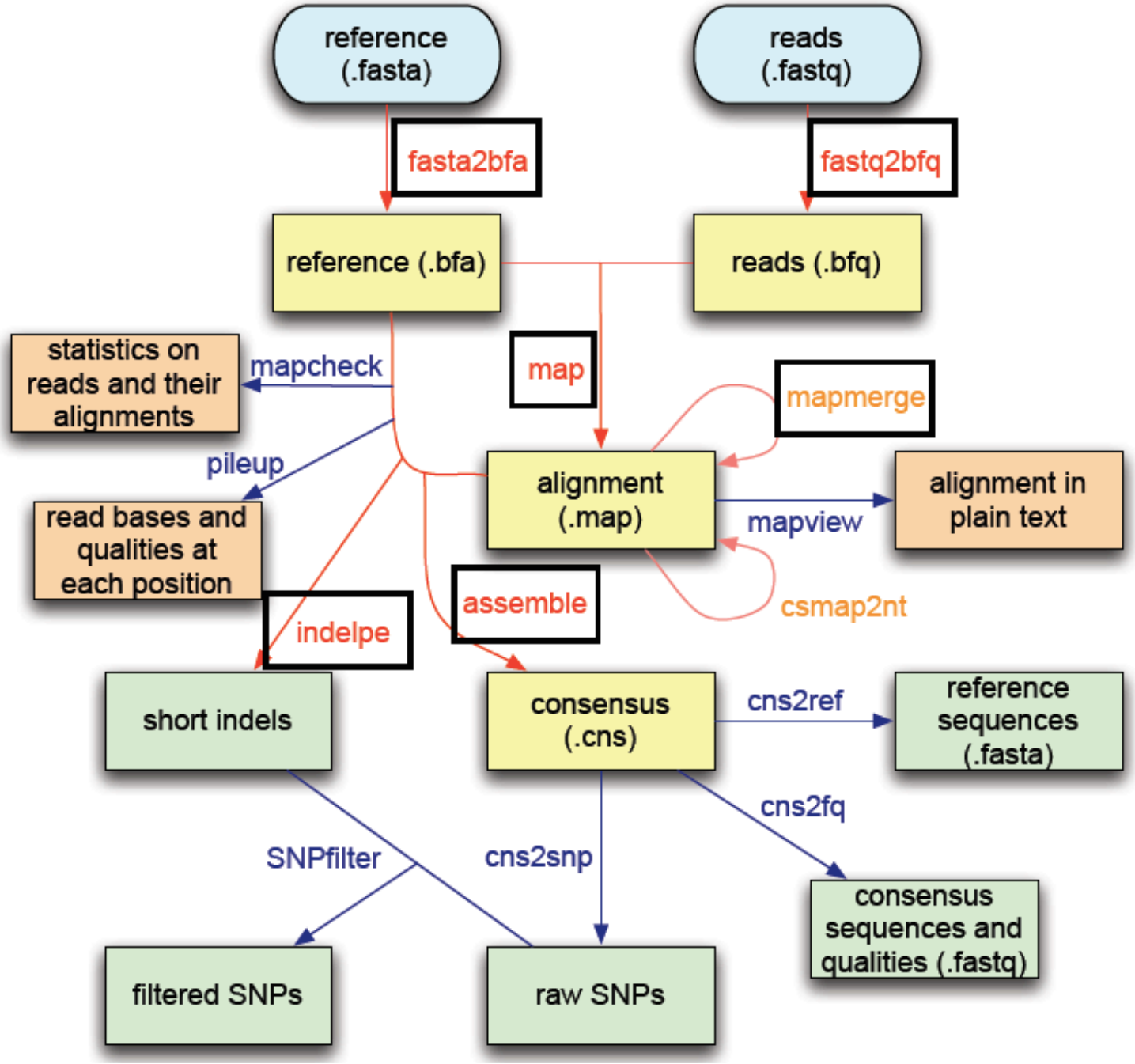
- Some aligners com with SNP calling functionality
  - Maq
  - SOAP
  - Bowtie has a converter to Maq's format to allow to use Maq's facilities
  - For BWA, the SAMtools can be used
- Output is a list of SNPs, if possible with *p* values
- The many different alignment have prevented modularization so far.
  - SAM (the Sequence/Alignment Map format) may become a standard.

The two pipelines

- MAQ:
  - ✓ Publication proved
  - ✓ More matured
- BWA+SAMtools:
  - ✓ 10X faster for human alignment with similar alignment accuracy
  - ✓ Gapped alignment for single-end reads
  - ✓ Improved short indel caller
  - ✓ Bleeding-edge

Courtesy of Heng Li (Welcome Trust Sanger Institute)

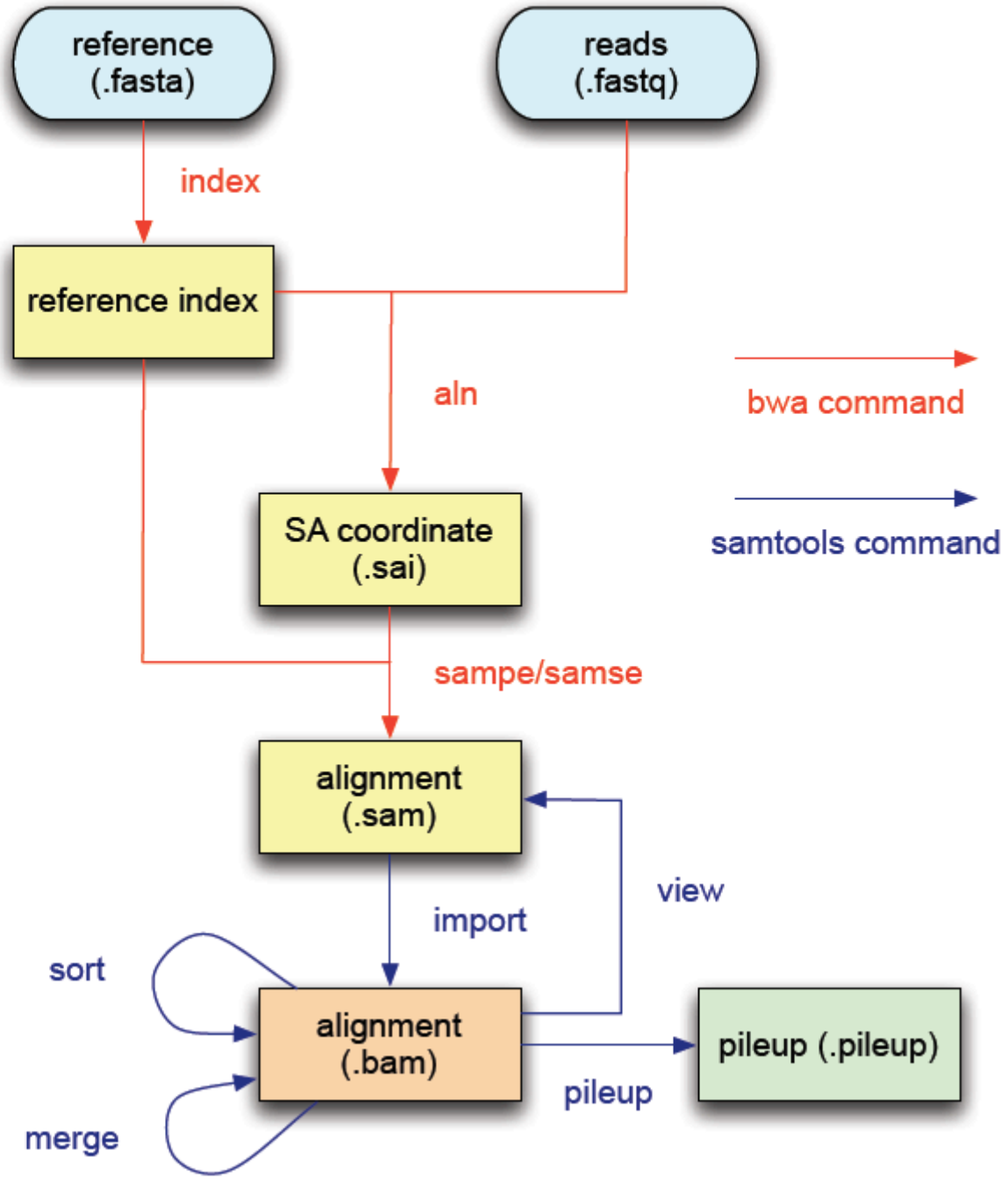
MAQ Work Flow



Courtesy of Heng Li (Welcome Trust Sanger Institute)

EMBL-EBI

BWA/SAMtools Work Flow



Courtesy of Heng Li (Welcome Trust Sanger Institute)

EMBL-EBI

MAQ pitfalls: alignment (I)

- Too many reads in a batch
  - ✓ MAQ's memory is linear in the #reads in the alignment
  - ✓ Too many reads make MAQ use too much memory
- Too few reads in a batch
  - ✓ MAQ's speed is similar given 100 reads and 100,000 reads
  - ✓ Recommendation: 2 million reads or 1 million pairs in a batch
- Assertion failure for paired end reads
  - ✓ In PET mapping, i-th read in the first file and i-th in the second file constitute a read pair
  - ✓ Two reads in a pair must have identical read names OR only differ at the tailing "/[12]": read001/1 and read001/2

MAQ pitfalls: alignment (II)

- Wrong '-e' option for long reads:
  - ✓ -e controls the tolerance of mismatches across the full read
  - ✓ -n controls the max-mismatches in the 28bp seed
- Improper '-a' option:
  - ✓ -a sets the maximum insert size
  - ✓ Excessively small -a leads to more wrong alignments
  - ✓ Recommendation: it is safer to use larger -a, although the resultant mapping quality would be a little conservative.
- Highly inaccurate base qualities:
  - ✓ Lead to inaccurate mapping quality (though not highly)
  - ✓ Recommendation: calibrate base qualities

Courtesy of Heng Li (Welcome Trust Sanger Institute)

Tools for RNA-Seq

RNA-Seq has additional challenges:

- Reads may straddle splice junctions
- Paralogy between genes prevent unique mappings
- One may want to incorporate or amend known gene models

Specialized tools for RNA-Seq alignment are

- ERANGE — to call differential expression
- TopHat — edgeR
- G-Mo.R-Se — BayesSeq

A worked example: Yeast resequencing

Samples: 3 closely related strains derived from the standard *S. cervesiae* lab strain S288c.

- Strain A: spontaneously evolved resistance against antibiotic X.
- Strain B: spontaneously evolved resistance against antibiotic Y.
- Strain C: engineered to have resistance against antibiotic B

Worked example: Data

For each strain, genomic DNA was extracted, fragemented and sequenced in one Solexa lane in paired-end mode, 36 bp from either end.

From the core facility, we got six data files

- StrainA_1_sequence.txt
- StrainA_2_sequence.txt
- StrainB_1_sequence.txt
- StrainB_2_sequence.txt
- StrainC_1_sequence.txt
- StrainC_2_sequence.txt

Worked example: Raw data files

Content of the _sequence.txt data files:

```
[...]
@HWI-EAS225_309MTAAXX:5:1:80:1842/1
GCAAACAATGTTTTGTTGTCGTATTTCTTTGTGAAG
+HWI-EAS225_309MTAAXX:5:1:80:1842/1
bbbbbbbbbbbbbbbbb`bb[bbbbb`bbbS`[`WK
@HWI-EAS225_309MTAAXX:5:1:1214:1711/1
GAAACCACATCAAAAACTTTTCTGTTGACAGTCCAC
+HWI-EAS225_309MTAAXX:5:1:1214:1711/1
bbbbbbbbbbbbbbbbbbbbbbbbbbb`bbb[[`^`]
[...]
```

Small letters in quality lines

=> FASTQ file in *Solexa* scale

EMBL-EBI

Step 1: Get reference and build index

- Download the reference genome (the Saccharomyces Genome Database's assembly for the S288c reference strain, as provided by Ensembl version 54)

wget ftp://ftp.ensembl.org/*[...]*revisiae.SGD1.01.54.dna.toplevel.fa.gz

gunzip Saccharomyces_cerevisiae.SGD1.01.54.dna.toplevel.fa.gz

We use one big FASTA file containing all chromosomes, the mitochondrial genome and the 2-micron plasmid.

- Let Bowtie build its index (i.e., permorf the Burrows-Wheeler transformation on the genome):

bowtie-build Saccharomyces_cervesia*[...]*toplev*e*l.fa Scerv

This takes a while and produces six files, named Scerv*ebwt.

- For a vertebrate genome, it may take many hours. Hence, prebuilt indices for many genomes can be downloaded from the Bowtie web site.

Step 2: Perform the alignment

- Bowtie is called as follows:

```
bowtie –-solexa-quals –-threads 4                  \
        Scerv                                      \
        -1 StrainA_1_sequence.txt                  \
        -2 StrainA_2_sequence.txt                  \
      --unfq StrainA.unaligned.fq                  \
      StrainA.bwtout
```

- With a single-core CPU, this takes a couple of hours. Using a multi-core CPU ("--threads") speeds things up.

- We get three output files:
  - StrainA.bwtout: the alignment,
  - StrainA_unaligned_1.fq and
  - StrainA_unaligned_2.fq: all reads that could not be aligned.

Bowtie output file

The output file looks like this (with lines wrapped)

```
[...]
HWI-EAS225_309MTAAXX:5:1:419:260/2      -       XII     1027676
        CATATTCTTGAATCAATGACGTGGTCAAAGACTCTG
        5AA<AA?:AAAA?AAAAAAAA:AAAAAAAAAAAAAA    0
HWI-EAS225_309MTAAXX:5:1:1436:122/1     +       II      177542
        GTTGTGTTATACTTCTTAGAAAAGAGGCAAAGAGGT
        CCCCCCCCCCCCCCCCCCC<CABCACAACCC:?4?<<    0
HWI-EAS225_309MTAAXX:5:1:1436:122/2     -       II      177649
        TTCGTTTCTCGAAATTTTTCGTTGTCCTATTTTCTT    ?
        AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA    0
HWI-EAS225_309MTAAXX:5:1:1088:946/2     +       XII     457916
        TACACTCTTGACCAGCGCACTCCGTCACCCTACGCT
        ????<1?+?8;??4<?0:<;;??40:&87&*(8,&,    1
        29:A>C,33:C>G,34:A>C
[...]
```

EMBL-EBI

Bowtie output

The 'bowtie' aligner outputs each alignment on a separate line. Each line is a collection of 8 fields separated by tabs; from left to right, the fields are:

1. Name of read that aligned
2. Orientation of read in the alignment, '-' for reverse complement, '+' otherwise
3. Name of reference sequence where alignment occurs, or ordinal ID if no name was provided
4. 0-based offset into the reference sequence where leftmost character of the alignment occurs
5. Read sequence (reverse-complemented if orientation is '-')
6. Read qualities (reversed if orientation is '-')
7. Reserved
8. Comma-separated list of mismatch descriptors. If there are no mismatches in the alignment, this field is empty. A single descriptor has the format offset:reference-base>read-base. The offset is expressed as a 0-based offset from the high-quality (5') end of the read.

Bowtie: Useful options

Bowtie has a number of useful optional features, e.g.:

- Report more than one match
- Convert from various quality scales
- Give maximum Q distance
- Specify seed length
- Maq- or SOAP-like alignment policy
- Collect unmapped reads in a file
- Multithread to make use of multi-core CPUs
- ...

Step 3a: Convert to SAM format

- Bowtie does not come with a SNP caller. We convert to the SAM (Sequence Alignment/Map) format, so that we can use the SAMtools.

  bowtie2sam.pl StrainA.bwtout >StrainA_u.sam

(The conversion script is distributed with SAMtools.)

Step 3b: Further preparations for SAMtools

- Index the reference FASTA file, so that SAMtools can access it quickly at random positions:

samtools faidx Sac*[...]*el.fa

(This adds a short file Sac*[...]*el.fa.fai)
We can now easily get subsequences, e.g.

samtools faidx Sac*[...]*el.fa III:10000-12000

- Convert the SAM file into the binary SAM (BAM) format for fast processing:

**samtools import Sac*[...]*el.fa **
StrainA.sam StrainA.bam

- Sort the reads by genomic position:
  samtools sort StrainA.bam StrainA_sorted.bam

Step 4: Inspect the alignment

SAMtools offers two tools to inspect the alignment

- pileup (streaming output)
- tview (interactive tool)

Pileup performs three functions:

- display the alignment
- call the consensus sequence and calculate SNP quality scores (i.e., *p* values)
- Call small indels

SAMtools pileup output

samtools pileup -c -f Sac*[...]*el.fa StrainA_sorted.sam

```
[...]
I     25514   G      G      42     0      25     5      ....^:.

        CCCCC
I     25515   T      T      42     0      25     5      .....                                                CC?CC
I     25516   A      G      48     48     25     7      GGGGG^:G^:g

        CCCCCC5
I     25517   G      G      51     0      25     8      ......,^:,

        CCCCCC1?
I     25518   T      T      60     0      25     11     .......,^:.^:.,^:,

        CCCCCC3A<:;
I     25519   T      T      60     0      25     11     ......,,.,,

        CCCCCC>A@AA
I     25520   G      G      60     0      25     11      ......,,.,,

        CCCACC>A@<A
I     25521   T      T      60     0      25     11     ......,,.,,

        CCCCCC?ACAA
I     25522   A      A      60     0      25     11     ......,,.,,

        CCCCCC>ACAA
I     25523   A      A      72     0      25     15     .......,.,,^:.^:.,^:.,^:.
```

EMBL-EBI

Step 5: Get a list of SNPs

- Filter out all SNPs with good quality from the pileup with a short Python (or Perl, or awk) script:

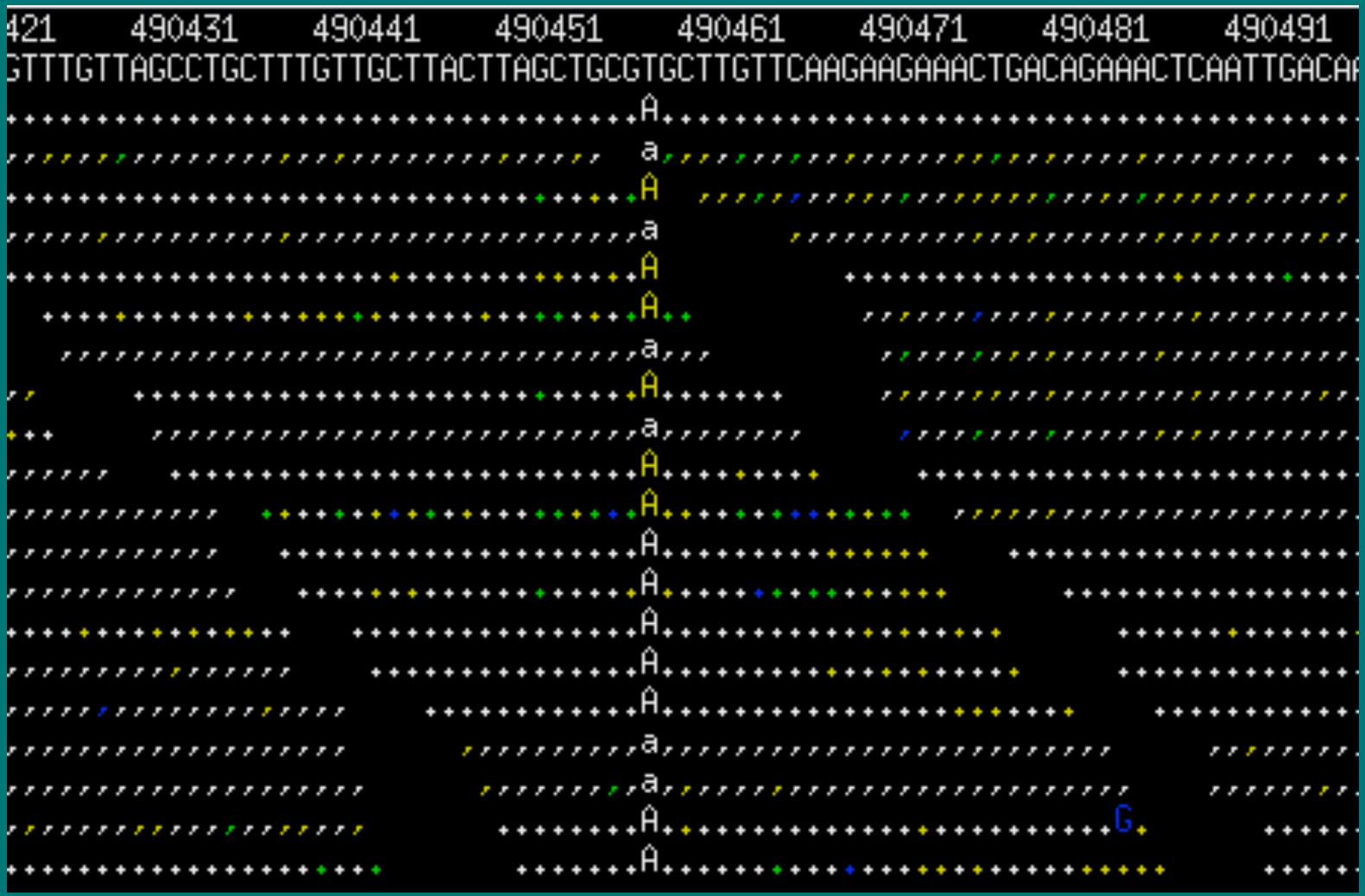
```
import fileinput
for line in fileinput.input():
    fields = line.split( "\t" )
    if int(fields[5]) >= 20:
       print line,
```

- Figure out which SNPs are not common to all three strains, e.g., with R.

- Check if there are any indels:

  samtools pileup **-i** -f Sac*[...]*el.fa StrainA_sorted.sam

Screenshot of SAMtools tview

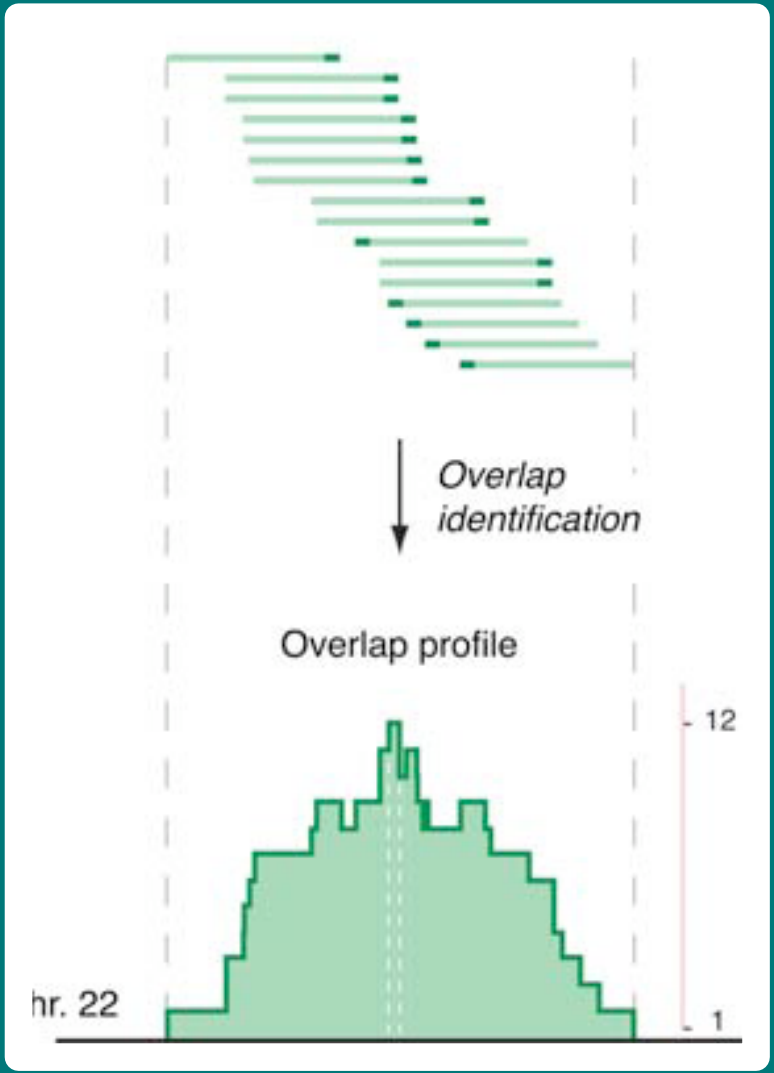


EMBL-EBI

ChIP-Seq and related techniques

EMBL-EBI

Coverage vectors



<-- Solexa reads,
aligned to genome

<-- coverage
vector

Figure taken from Zhang et al., PLoS Comp. Biol. 2008

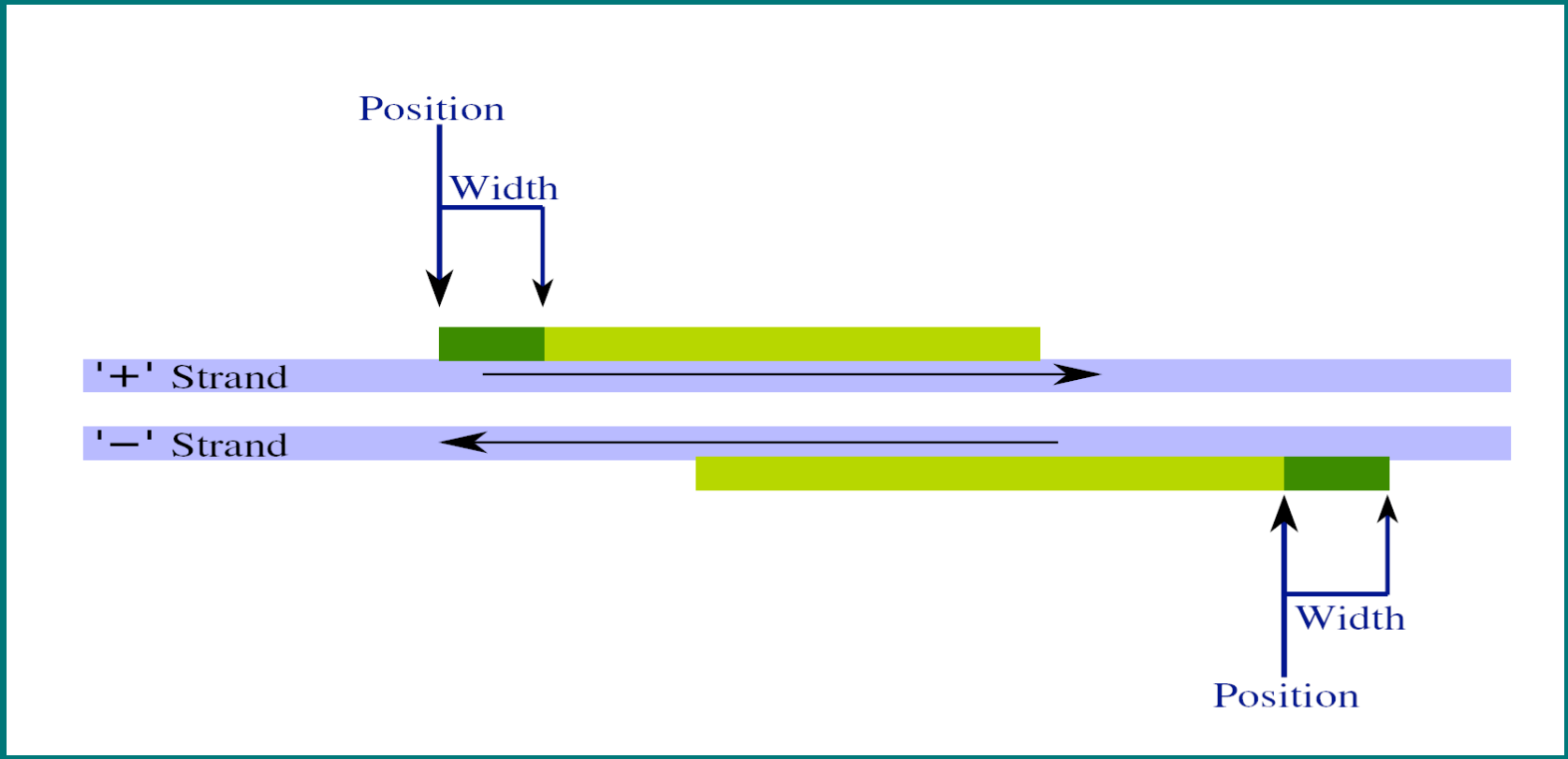
ChIP-Seq: Coverage vectors

- A coverage (or: “pile-up”) vector is an integer vector with on element per base pair in a chromosome, tallying the number of reads (or fragments) mapping onto each base pair.
- It is the essential intermediate data type in assays like ChIP-Seq or RNA-Seq

- One may ever count the coverage by the reads themselves, or extend to the length of the fragments

Calculating coverage vectors

Extending reads to fragments:



Chip-Seq coverage: examples

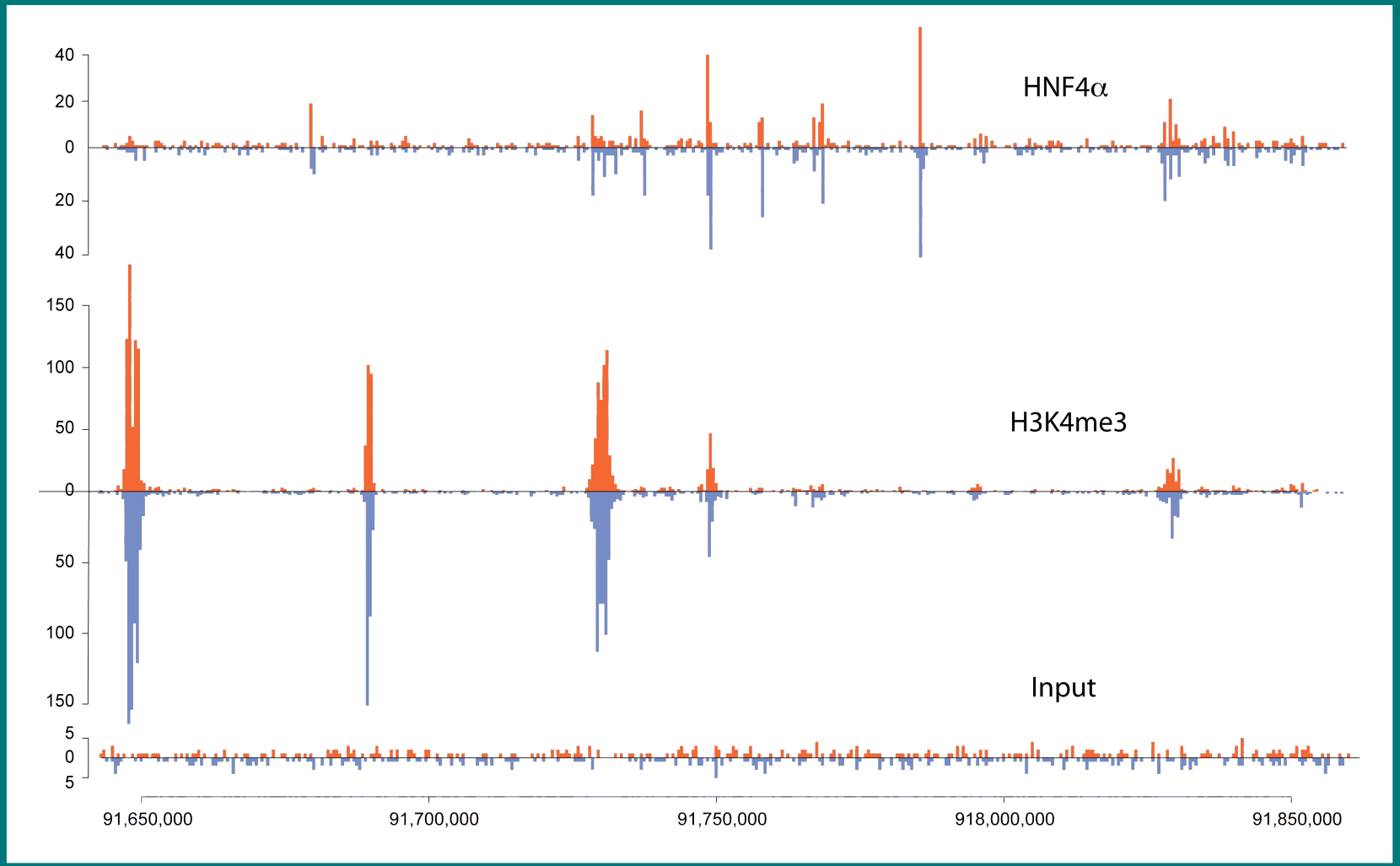
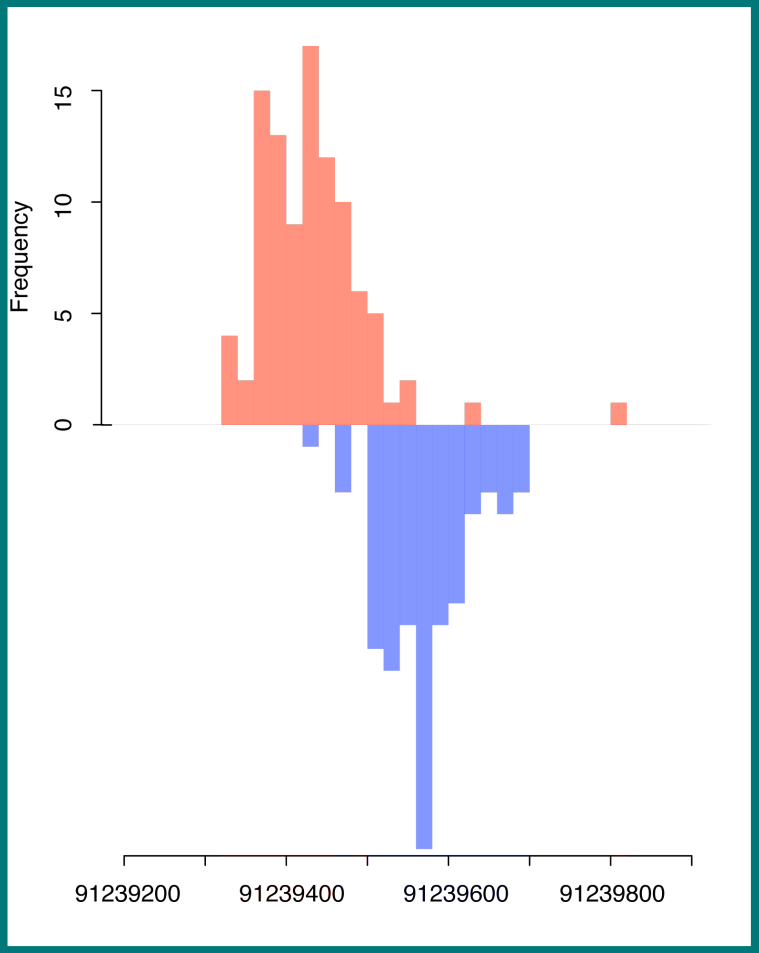


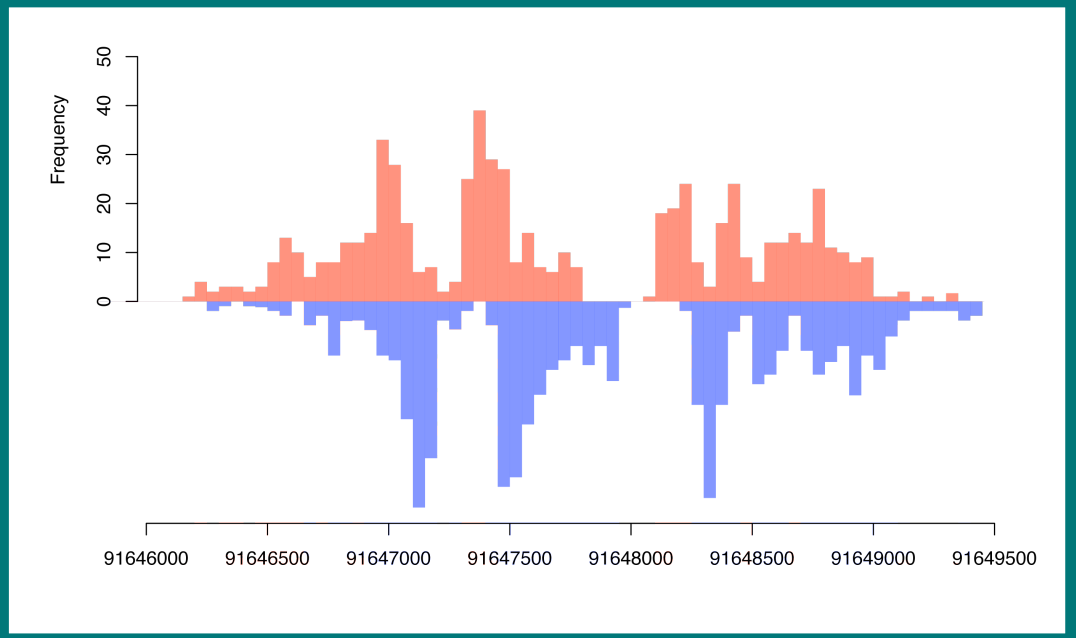
Figure coutesy of Christiana Spyrou (CR UK)

EMBL-EBI

ChIP-Seq: Different peak shapes



Transcription factor binding site



Histone modification

Figures coutesy of Christiana Spyrou (CR UK)

EMBL-EBI

The issue with multiple reads

If one finds several reads with the exact same sequenche, does this mean

- that many fragments from this locus were precipitated and often got got cut at the exact same place, or
- that there was only a single fragment, but it was amplified more efficiently than fragments from other loci in the PCR (or more efficiently transcribed to cDNA)?
  - If you consider the latter more likely, you should count these reads only once. However, this dramatically compresses your dynamic range.

Peak-finding software

- In principle, peak finders developed for tiling arrays can still be used.
- There are, however, tools taylored to HTS data:
  - [Put Mali's list here]

- These use a large variety of quite different algorithms

Peak-finding software: Comparison

	IP only, & control, either	read features	data from different strands	masks genomic repeats	scoring criteria	confidence in results, FDR estimates sensitivity / specificity	for both TF & HM
CSPF	& / or	read length no orientation	merges strands	N	simple height criteria	empirically: ROC curve	both
XSET	& / or	mean fragment length orientation	merges strands	N	simple height criteria	FDR based on randomised sample and Poisson probabilities	both
Mikkelsen et al.	IP only	no orientation	no merge / shift	Y	p-values produced by randomising the datasets	no official FDR	both
MACS	& / or	mean fragment length orientation ignores duplicated reads	shifts reads merges strands	N	Poisson p-values	FDR = no. peaks in control : IP	both
QuEST	&	orientation	shifts reads merges strands	N	kernel density estimation	FDR based on calling peaks in 1/2 the control sample	TF
FindPeaks	IP only	mean fragment length orientation	no merge / shift	N	simple height criteria	Monte-Carlo based FDR (ie. from randomised sample)	both
SISSR	& / or	mean fragment length orientation	no merge / shift	N	compares read density on different strands	FDR comparing simulated background peaks to real data	better for TF
Kharchenko et al.	&	orientation	no merge / shift	N	Poisson probabilities	FDR based on different randomised versions of the input sample	better for TF
PeakSeq	&	mean fragment length orientation	merges strands	Y	pre-processing: normalisation Binomial p-values	FDR: q-values after multiple correction adjustment	both
BayesPeak	& / or	mean fragment length orientation	no merge / shift	N	Negative Binomial distribution Bayesian posterior probabilities	posterior probabilities of enrichment presence	both

Table coutesy of Christiana Spyrou (CR UK)

EMBL-EBI

Peak finding software: Comparison

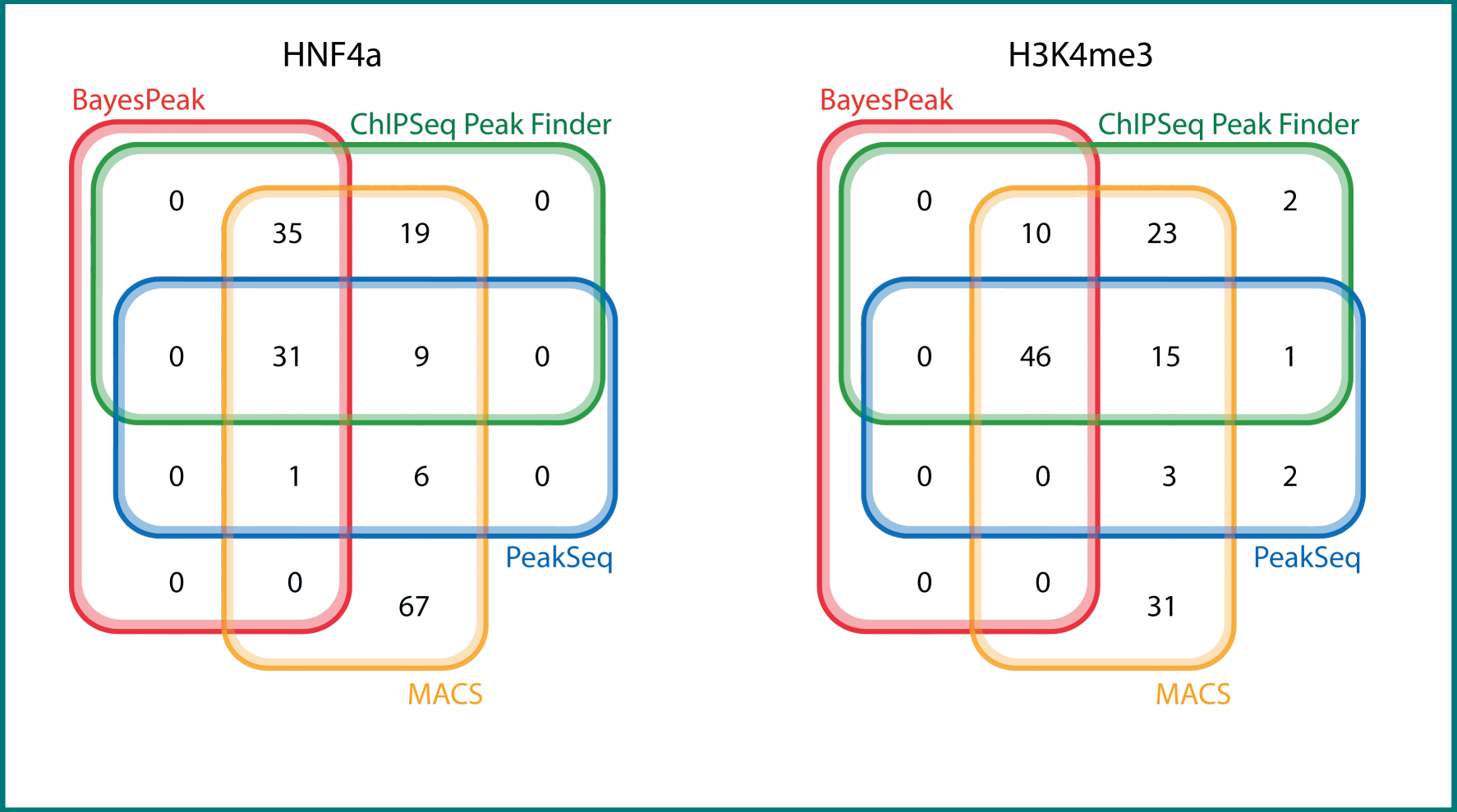


Figure coutesy of Christiana Spyrou (CR UK)

EMBL-EBI

Writing you own software

- The “glue” to combine the available tools is mostly missing.
- You will have to write your own scripts.
- Often used languages:
  - Perl
  - Python
  - R
  - Java
  - C/C++

Pros and cons for using R

Pro:

- Huge statistical library
- Large bioinformatics library
- Good plotting facilities
- Convenient interactive shell

Con:

- Call-by-value semantics not well suited for very large amounts of data
- Slow due to lack of bytecode compiler
- Poor string-handling abilities
- Outdated OOP paradigm

Bioconductor packages for HTS

- Biostrings
- BSGenome
- ShortRead
- TileQC
- GenomeGraphs
- HilbertVis
- TileQC
- ChipSeq
- edgeR