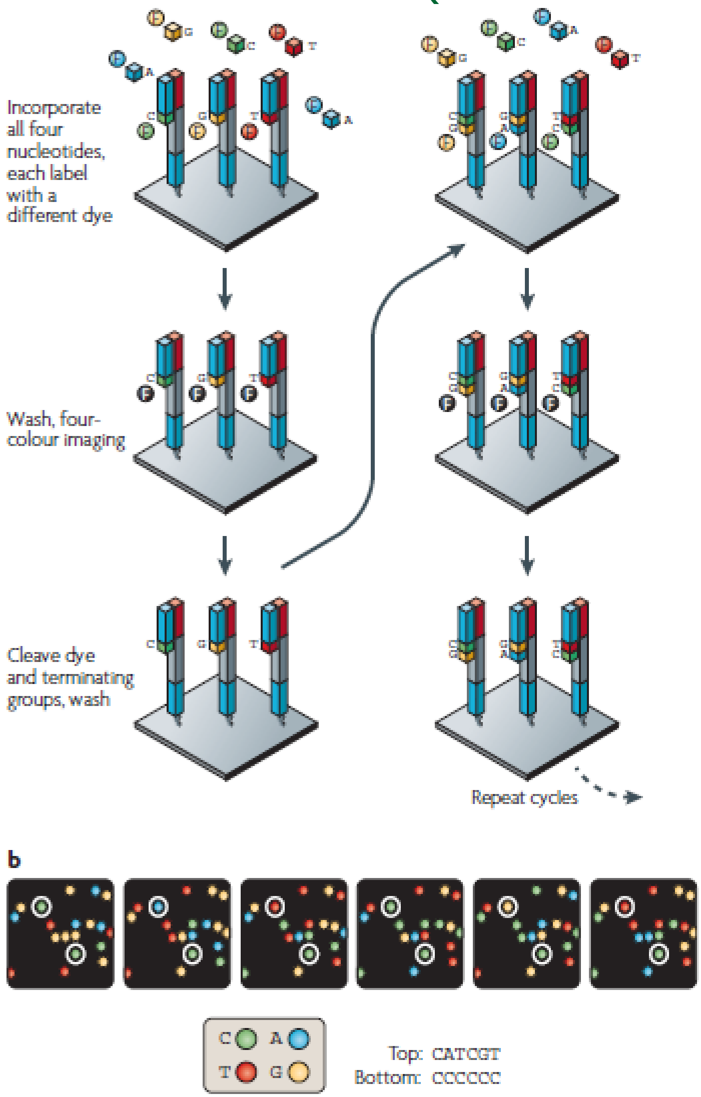
Illumina (Solexa)

- Current market leader
- Based on *sequencing by synthesis*
- Current read length 100-150bp
- Paired-end easy, longer matepairs harder
- Error ~0.1%
  - Mismatch errors dominate
- Throughput: 4 Tbp in one run (5 days)
- Cheapest sequencing technology
  - Cost: ~$1000 per human genme

Illumina (Solexa)


Incorporate all four nucleotides, each label with a different dye
Wash, four-colour imaging
Cleave dye and terminating groups, wash
Repeat cycles
b
C A
T G
Top: CATCGT
Bottom: CCCCCC

GA IIx

MiSeq

HiSeq 2000

Illumina (Solexa)

•Read length and quality string length are the same

Read and Quality (1)

```
@FC81ET1ABXX:3:1101:1215:2154/1
TTTTTCAAATGTTTGTTGCCTATTTTTATATCTTCTTTTGAGAATTGTCTGTTCATGTCNTNNGNNCNCNNTNTCANGGGATTGTTTGTT
+
HHGHHHHHGHHHHDHFHHHHHHHFHHHHHHHEHHEHHHHEGGDEF2CGDCDFB0>DA###################################
```

Read and Quality (2)

```
@FC81ET1ABXX:3:1101:1215:2154/2
AAGCCANNTNNNNNNNNNNNNNACTGGATCCTCATAGCTCACCTTATGCAAAAATCAACTCAAGATGGATGAAGGTCTTAAACCTAATAC
+
HHHBH?##;#############:83<9:;7FDFBFEFE;BEEBE8C>2D8@BBACDFG=E@=CDDHEGGDB;<,:19*23?=@#######
```

- Read length and quality string length are the same
- All read/1s are the same length in the same run
- All read/2s are the same length in the same run

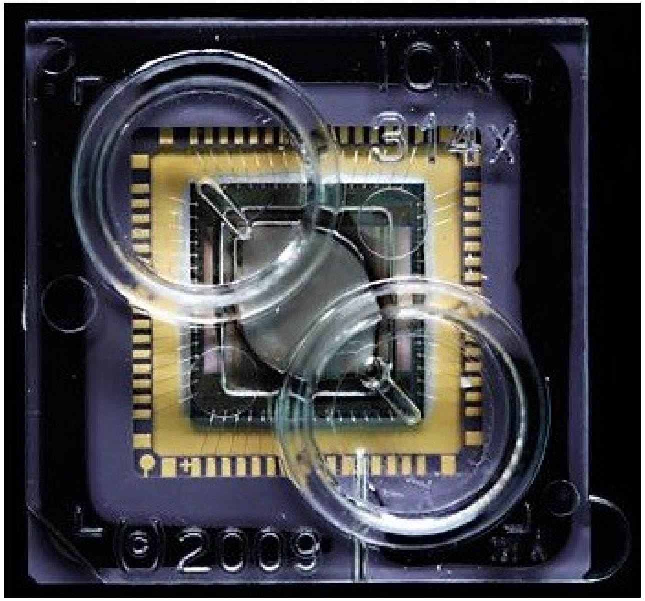
Illumina (Solexa)

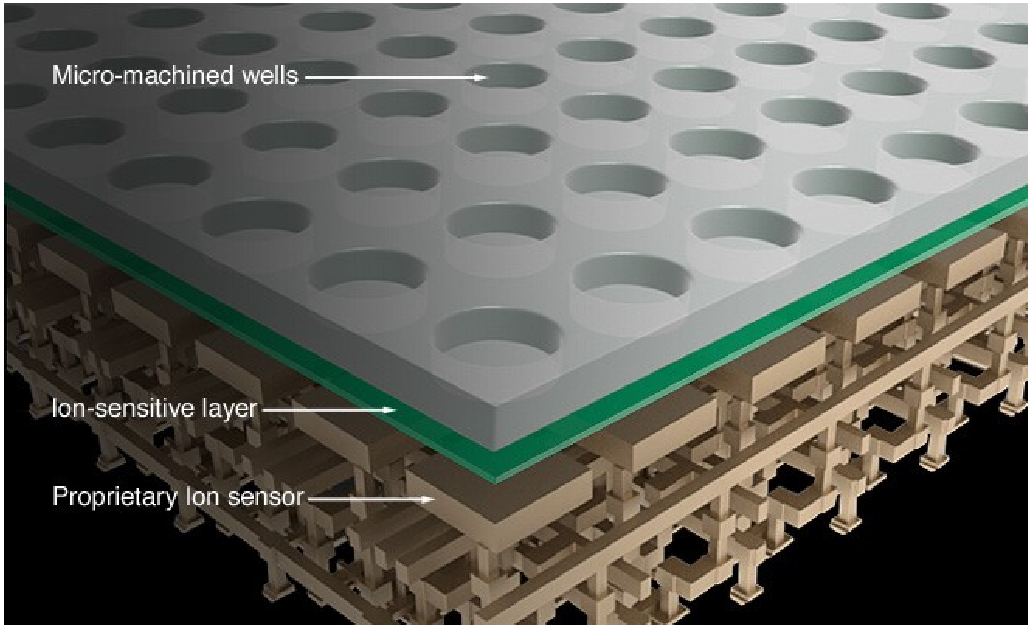
- Read mapping:
  - mrFAST, mrsFAST, BWA, MAQ, BFAST, MOSAIK, Bowtie, SOAP, SHRiMP, many more
- *De novo* assembly:
  - EULER, Velvet, ABySS, Hapsembler, SGA, ALLPATHS, ….

Ion Torrent

- No laser, no image processing:
  - Sequencing is done on a microprocessor that measures pH level changes as bases incorporate
- Error ~1%
  - Indel dominated & homopolymers (454 Life Sci.)
- Matepair sequencing possible, but difficult

Ion Torrent

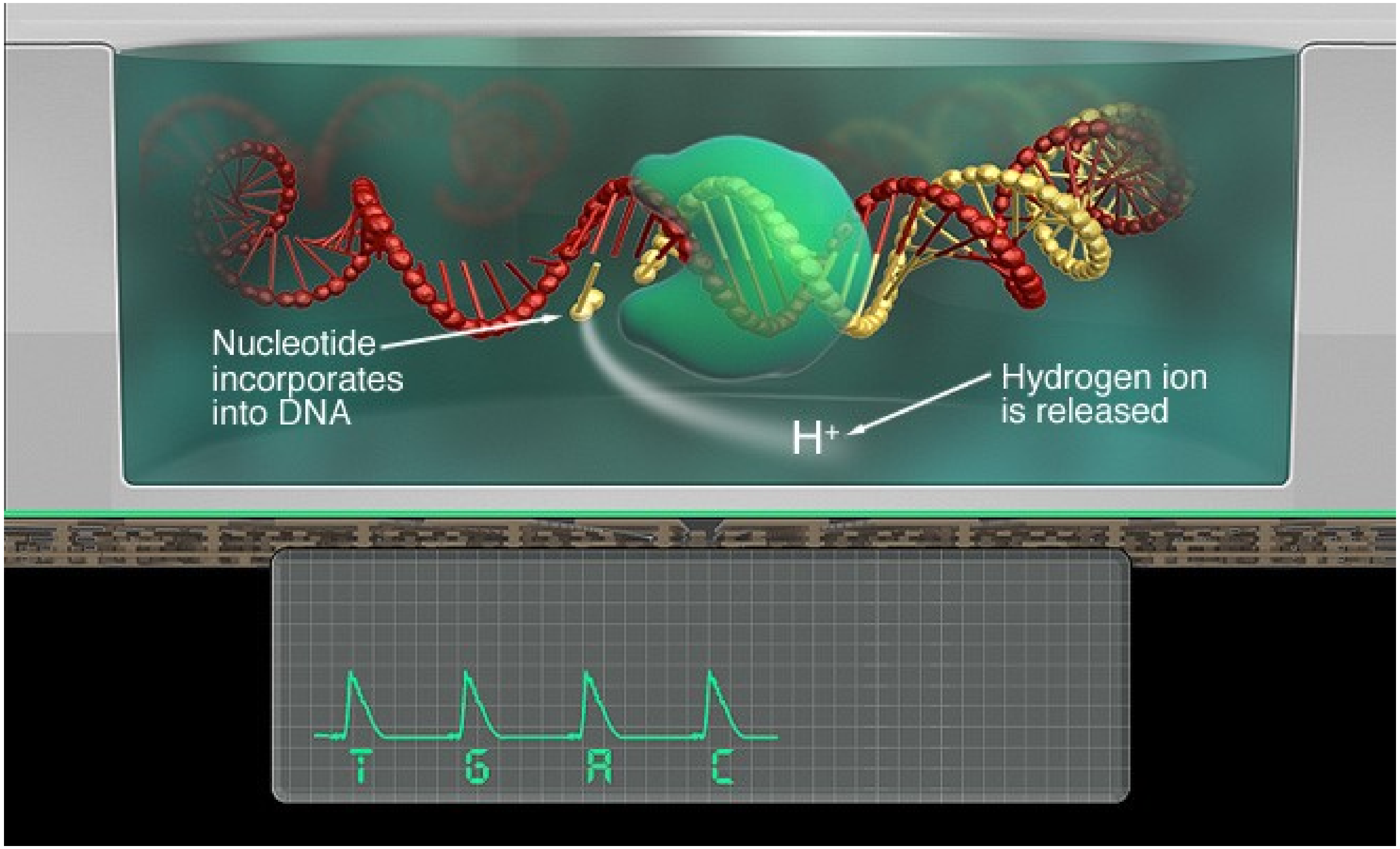

ION
314x
©2009


Micro-machined wells
Ion-sensitive layer
Proprietary Ion sensor

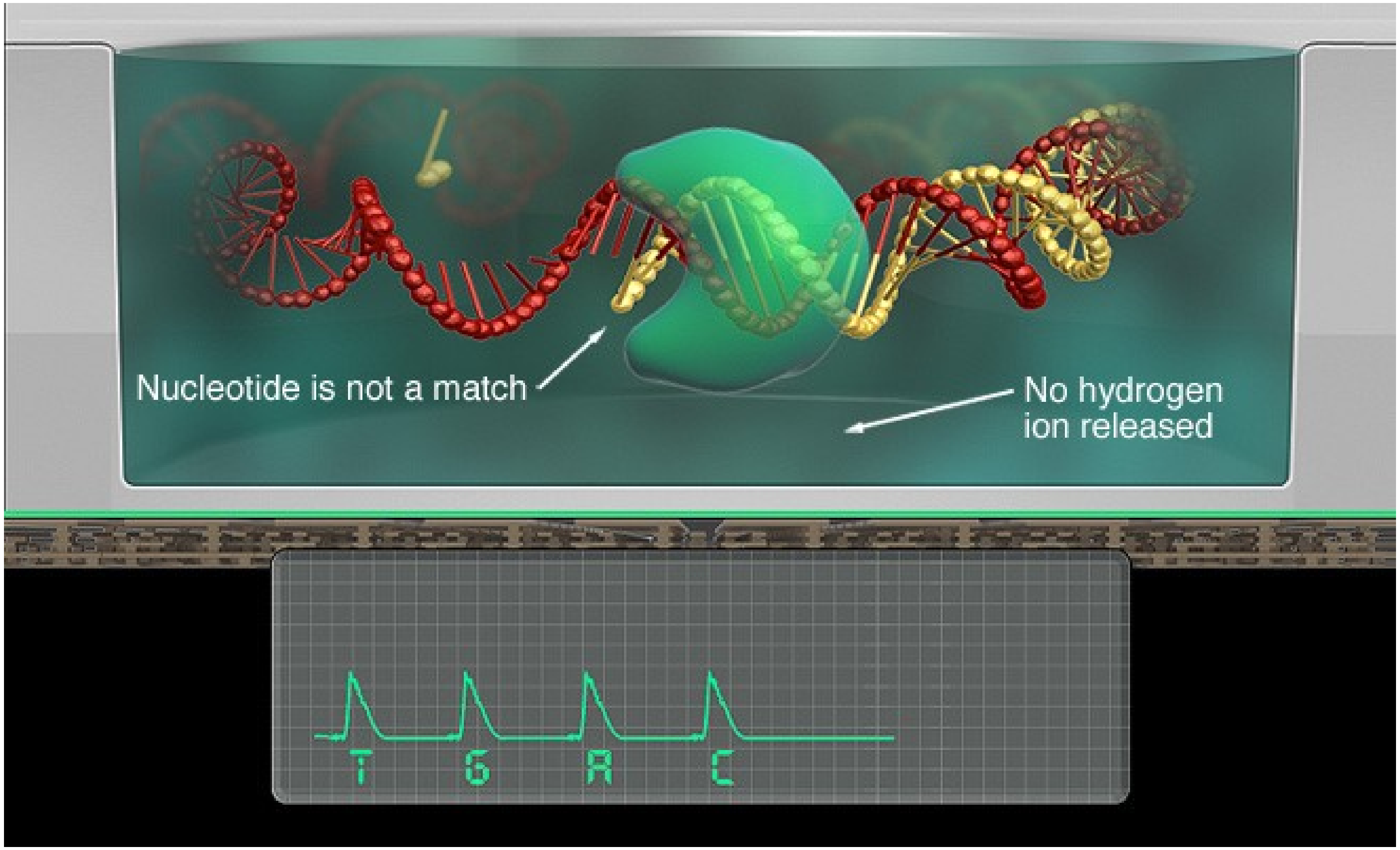
personal genome machine

ion torrent
Proton
Semiconductor Sequencer

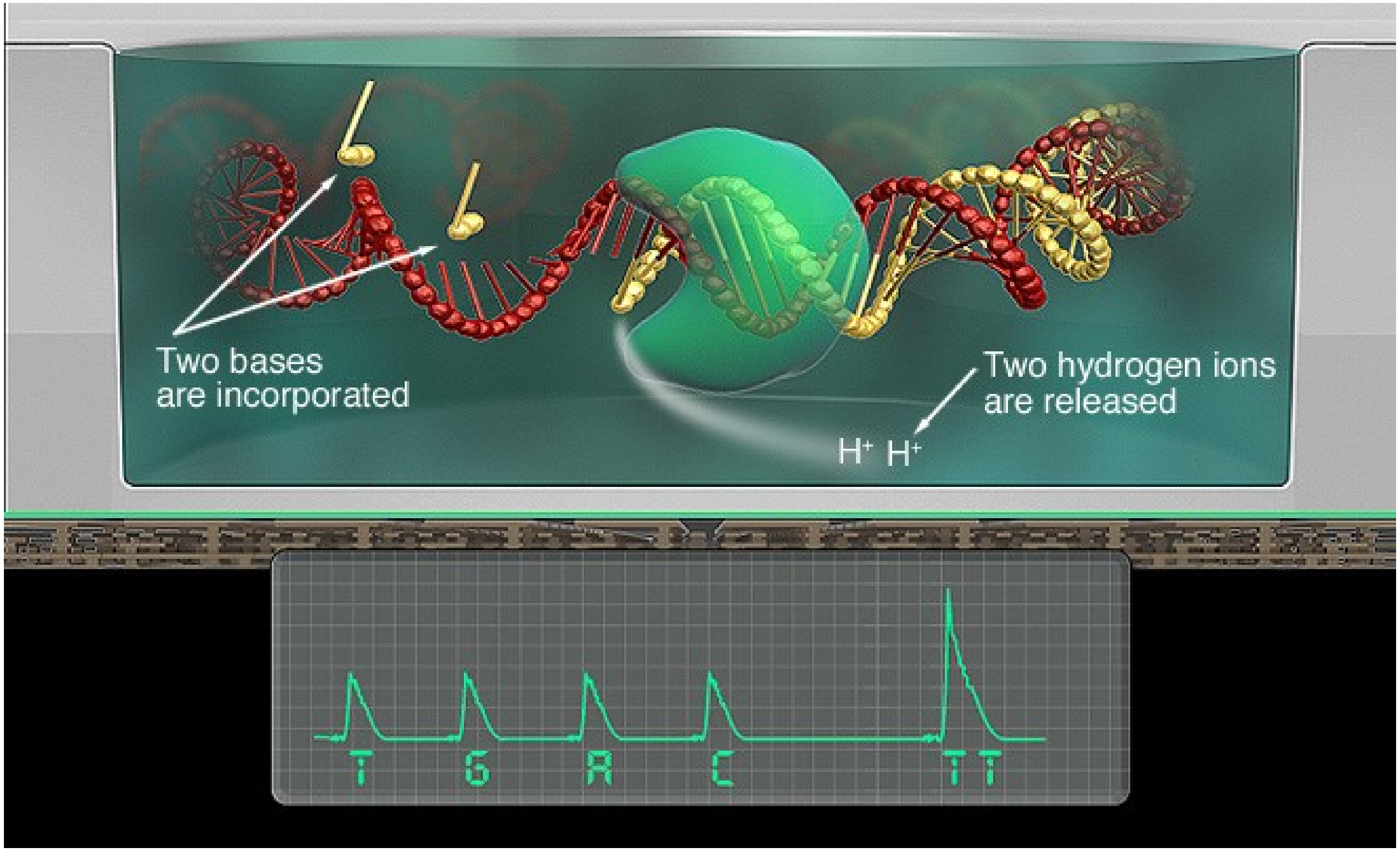
Ion Torrent


Nucleotide incorporates into DNA
Hydrogen ion is released
H+
T
G
A
C

Ion Torrent

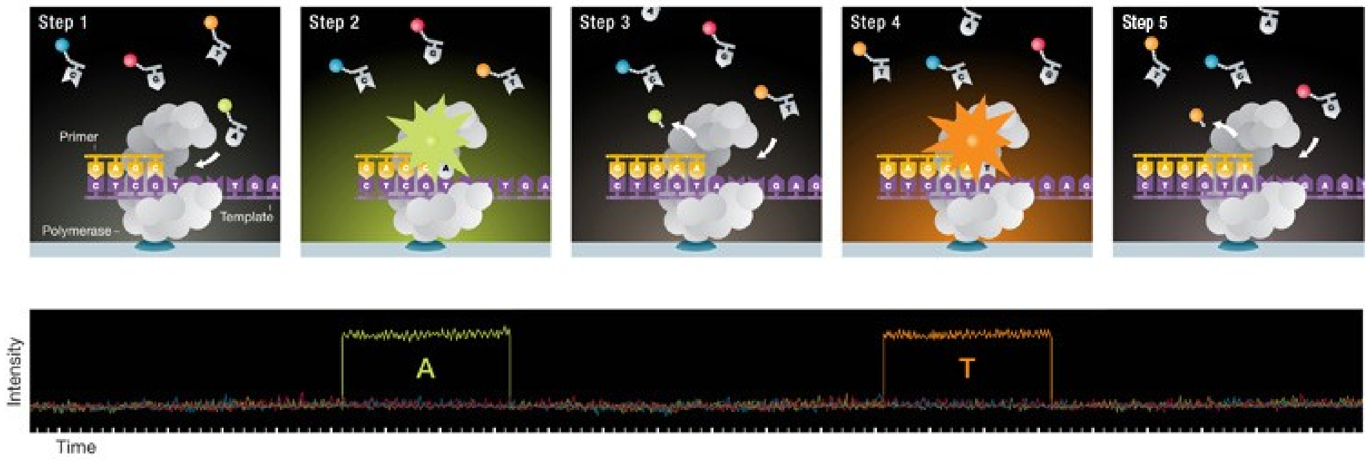


Ion Torrent


Two bases
are incorporated
Two hydrogen ions
are released
H+ H+
T
G
A
C
T T

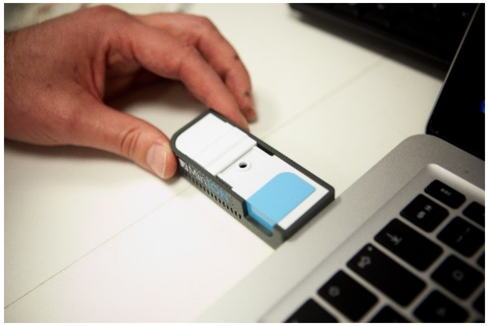
Pacific Biosciences

- “Third generation”; single molecule real time sequencing (SMRT)
- No replication with PCR
- Phosphates are labeled. Watches DNA polymerase in real-time while it copies single DNA molecules.
- Long sequence reads (7-20 Kb)
- Errors: ~12%; indel dominated



Nanopore Sequencing

- Nanopore sequencing:
  - Oxford Biosciences
    - 100 Kb reads
    - 20% error rate



MinION

SmidgION

PromethION

PromethION

NGS: Computational Challenges

- Data management
  - Files are very large; compression algorithms needed
- Read mapping
  - Finding the location on the reference genome
  - All platforms have different data types and error models
  - Repeats!!!!
- Variation discovery
  - Depends on mapping
  - Again, all platforms has strengths and weaknesses
- *De novo* assembly
  - It’s very difficult to assemble short sequences with high errors

Compression

- 1 – Reference based
  - Coding/decoding rather than real compression
  - Very high compression rate
  - Fast to encode
  - Slow to decode
  - Needs a reference genome
    - None, or poor quality for most species
    - Use same version of reference genome in decompression
  - Needs mapping (takes a long time)
    - Unmapped reads should be treated separately
  - CRAMtools, SlimGene, etc.
    - *Very* lossy

Compression

- 2 – Reference free
  - Less compression rate
  - No need for reference, applicable to any dataset from any species
  - Slower to compress, faster to decompress
  - Can be lossy or lossless
  - Multipurpose compressors:
    - gzip, bzip2, 7-zip, etc.
  - Specialized FASTQ compressors
    - SCALCE, ReCoil, G-SQZ, etc.

Reference-free compression

- Easy task (or gzip, etc.): Concatenate all sequences, then run Lempel-Ziv algorithm
- Problem: Locality

http://scalce.sourceforge.net

Hach et al., unpublished

Lempel-Ziv Compression

a b b a a b b a a b a b b a a a a b a a b b a
0 1 1 0 2--- 4--- 2--- 6------- 5--- 5--- 7------- 3--- 0

Index	Entry	Index	Entry
0	a	7	baa
1	b	8	aba
2	ab	9	abba
3	bb	10	aaa
4	ba	11	aab
5	aa	12	baab
6	abb	13	bba

NGS/algorithms

READ MAPPING

Read Mapping

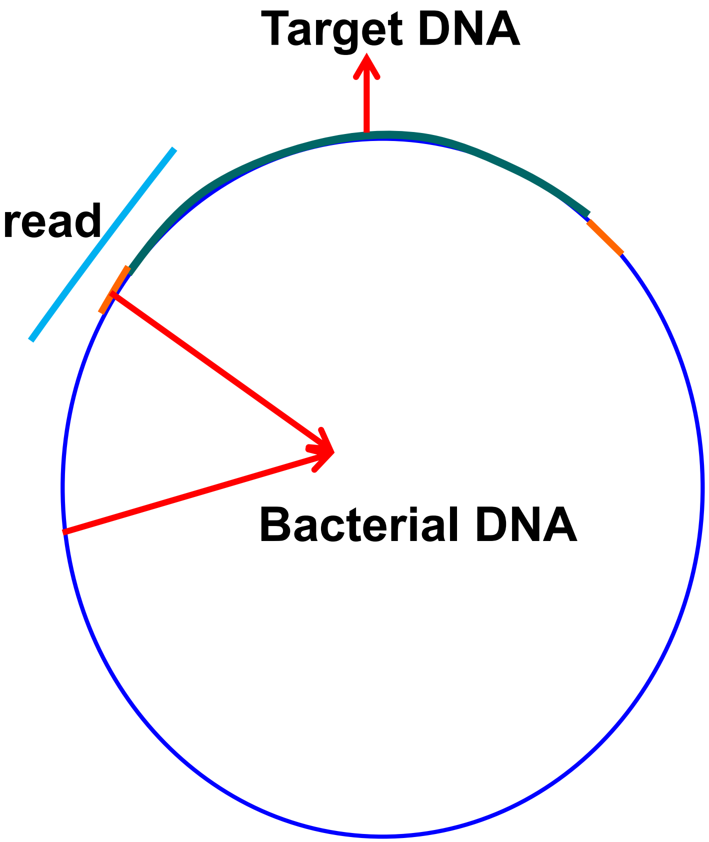
- When we have a reference genome & reads from DNA sequencing, which part of the genome does it come from?
- Challenges:
  - Sanger sequencing
    - Cloning vectors
    - Millions of long (~1000 bp reads)
  - Next-Gen sequencing:
    - Billions of short reads with low error
    - OR: hundreds of millions of long reads with high error
  - Common: sequencing errors
    - More prevalent in NGS
  - Common: contamination
    - Typically ~2-3% of reads come from different sources; i.e. human resequencing contaminated with yeast, E. coli, etc.
  - Common: Repeats & Duplications

Read Mapping

- Accuracy
  - Due to repeats, we need a confidence score in alignment
- Sensitivity
  - Don’t lose information
- Speed
- Think of the memory usage
- Output
  - Keep all needed information, but don’t overflow your disks
- All read mapping algorithms perform alignment at some point (read vs. reference)

Sanger vs NGS: cloning vectors

- Sanger reads may contain sequence from the cloning vector; thus mapping needs *local alignment.*
- No cloning vectors in NGS, *global alignment* is fine.



Mapping Reads

Problem: We are given a read, *R,* and a reference sequence, *S*. Find the best or all occurrences of *R* in *S*.

Example:

R = AAACGAGTTA

S = TTAATGC*AAACGAGTTA*CCCAATATATATAT*AAACCAGTTA*TT

Considering no error: one occurrence.

Considering up to 1 substitution error: two occurrences.

Considering up to 10 substitution errors: many meaningless occurrences!

Don’t forget to search in both forward and reverse strands!!!

Mapping Reads (continued)

Variations:

- Sequencing error
  - No error: *R* is a perfect subsequence of *S.*
  - Only substitution error: *R* is a subsequence of *S* up to a few substitutions.
  - Indel and substitution error: *R* is a subsequence of *S* up to a few short indels and substitutions.
- Junctions (for instance in alternative splicing)
  - Fixed order/orientation

    $R = R_1R_2\ldots R_n$ and R_i map to different non-overlapping loci in *S*, but to the same strand and preserving the order.
  - Arbitrary order/orientation

    $R = R_1R_2\ldots R_n$ and R_i map to different non-overlapping loci in *S.*

Mapping algorithms

- Two main “styles”:
  - Hash based seed-and-extend (hash table, suffix array, suffix tree)
    - Index the k-mers in the genome
      - Continuous seeds and gapped seeds
    - When searching a read, find the location of a k-mer in the read; then extend through alignment
    - Requires large memory; this can be reduced with cost to run time
    - More sensitive, but slow
  - Burrows-Wheeler Transform & Ferragina-Manzini Index based aligners
    - BWT is a data compression method used to compress the genome index
    - Perfect hits can be found very quickly, memory lookup costs increase for imperfect hits
    - Reduced sensitivity
  - Today’s standard: hybrid
    - Seed with BWT-FM then extend

“Long” read mappers

- BLAST, MegaBLAST, BLAT, LASTZ can be used for Sanger, 454, Ion Torrent
  - Hash based
  - Extension step is done using Smith-Waterman algorithm
  - BLAST and MegaBLAST have additional scoring scheme to order hits and assign confidence values
  - 454/Ion Torrent only: PASH, Newbler

Short read mappers

- Hash based
  - Illumina: mrFAST, mrsFAST, MAQ, MOSAIK, SOAP, SHRiMP, etc.
    - MOSAIK requires ~30GB memory
    - Others limit memory usage by dividing genome into chunks
    - mrFAST, SHRiMP have SSE-based implementation
    - MAQ: Hamming distance only

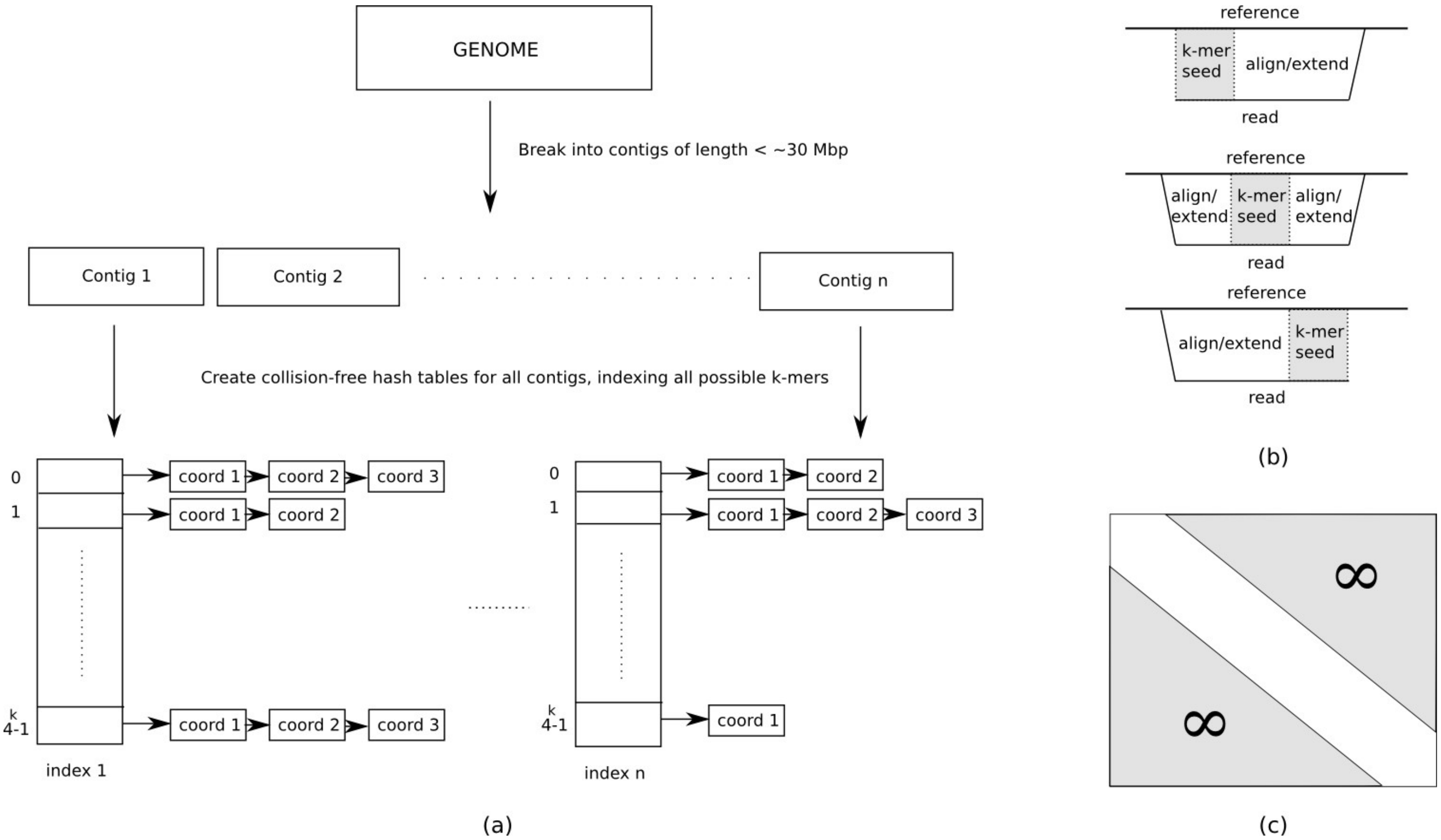
Short read mappers

- BWT-FM based
  - Illumina: BWA, Bowtie, SOAP2
  - Human genome can be compressed into a 2.3 GB data structure through BWT
  - Extremely fast for perfect hits
  - Increased memory lookups for mismatch
    - Indels are found in postprocessing when paired-end reads are available
  - GPGPU implementations: SOAP3 (poor performance due to memory lookups)
- Hybrid: BWA-MEM

Read mappers: PacBio

- BLASR aligner; tuned for PacBio error model (indel dominated, ~15%)
- Two versions:
  - Hash based
  - BWT-FM based

Hash Based Aligners

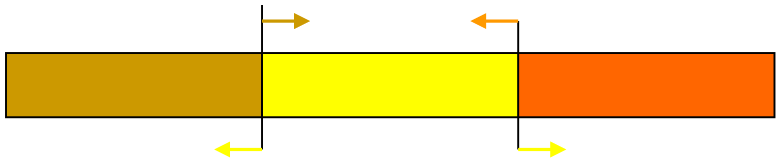

GENOME

Break into contigs of length < ~30 Mbp

Create collision-free hash tables for all contigs, indexing all possible k-mers

Seed and extend

- Break the read into *n* segments of k-mers.
  - For perfect sensitivity under edit distance *e*
    - There is at least one *l*-mer where l = floor(*L*/(e+1)); *L*=read length
    - For fixed *l*=*k*; *n* = *e*+1 and k ≤ *L* / *n*
  - Large k -> large memory
  - Small k -> more hash hits
- Lets consider the read length is 36 bp, and k=12.



- if we are looking for 2 edit distance (mismatch, indel) this would guaranty to find all of the hits

Mapping Quality

- MAPQ = -10 * $\log_{10}$(Prob(mapping is wrong))

For reference sequence *x;* read sequence *z:*
***p(z | x,u)* =** probability that *z* comes from position *u*
= multiplication of p_e of mismatched bases of *z*

For posterior probability **p(u | x,z)** assume uniform prior distribution **p(u|x)**
L=|x| and *l*=|z|. Apply Bayesian formula:

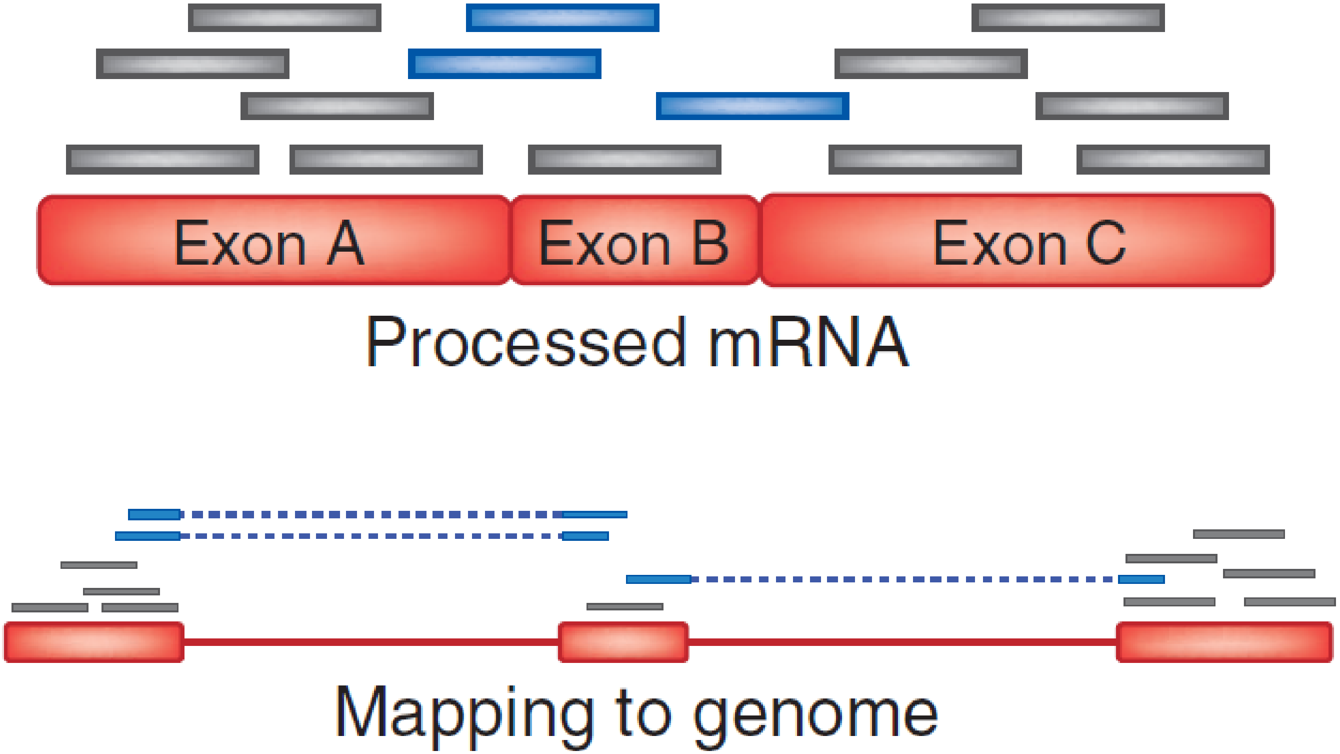
$$p_s(u|x,z) = \frac{p(z|x,u)}{\sum_{v=1}^{L-l+1} p(z|x,v)}$$

$$Q_s(u|x,z) = -10 \log_{10}[1 - p_s(u|x,z)].$$

Calculated for one "best" hit

Li et al., Genome Research, 2008

Spliced-read mapping



- Used for processed mRNA data
- Reports reads that span introns.
- Examples: TopHat, ERANGE