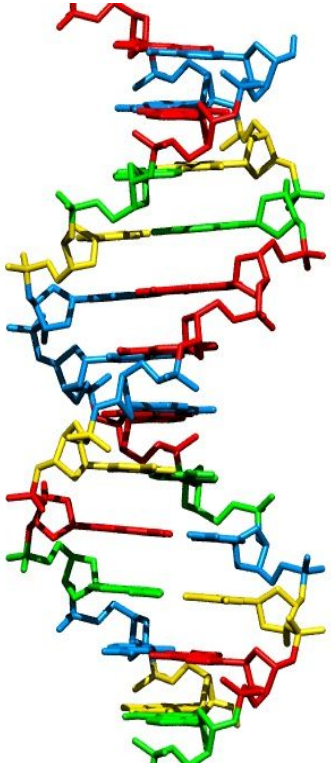
CS481/CS583: Bioinformatics Algorithms

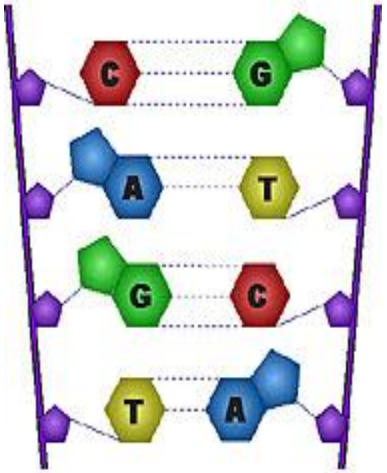
Can Alkan
EA509
calkan@cs.bilkent.edu.tr

http://www.cs.bilkent.edu.tr/~calkan/teaching/cs481/

DNA sequencing

How we obtain the sequence of nucleotides of a species





…ACGTGACTGAGGACCGTG
CGACTGAGACTGACTGGGT
CTAGCTAGACTACGTTTTA
TATATATATACGTCGTCGT
ACTGATGACTAGATTACAG
ACTGATTTAGATACCTGAC
TGATTTTAAAAAAATATT…

DNA Sequencing

GENERAL CONCEPTS AND CAPILLARY (SANGER) SEQUENCING

DNA Sequencing

Goal:

Find the complete sequence of A, C, G, T's in DNA

Challenge:

There is no machine that takes long DNA as an input, and gives the complete sequence as output

DNA Sequencing: History

Sanger method (1977): labeled ddNTPs terminate DNA copying at random points.

Gilbert method (1977): chemical method to cleave DNA at specific points (G, G+A, T+C, C).

Both methods generate labeled fragments of varying lengths that are further electrophoresed.

History of DNA Sequencing

Adapted from Eric Green, NIH; Adapted from Messing & Llaca, *PNAS* (1998)

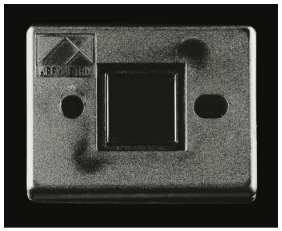
Efficiency (bp/person/year)		
1	1870	Miescher: Discovers DNA
	1940	Avery: Proposes DNA as 'Genetic Material'
		Watson & Crick: Double Helix Structure of DNA
15	1953	Holley: Sequences Yeast $tRNA^{Ala}$
150	1965	Wu: Sequences λ Cohesive End DNA
1,500	1970	Sanger: Dideoxy Chain Termination Gilbert: Chemical Degradation
15,000	1977	Messing: M13 Cloning
25,000	1980	Hood et al.: Partial Automation
50,000	1986	
200,000	1990	• Cycle Sequencing • Improved Sequencing Enzymes • Improved Fluorescent Detection Schemes
50,000,000	2002	
100,000,000,000	2009	• Next Generation Sequencing •Improved enzymes and chemistry •New image processing

Sequencing by Hybridization (SBH): History

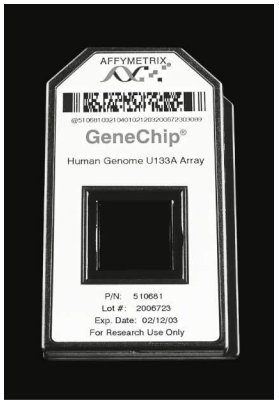
- **1988:** SBH suggested as an an alternative sequencing method.
- **1991:** Light directed polymer synthesis developed by Steve Fodor and colleagues.
- **1994:** Affymetrix develops first 64-kb DNA microarray

First microarray prototype **(1989)**

First commercial DNA microarray prototype w/16,000 features **(1994)**



500,000 features per chip **(2002)**



How SBH Works

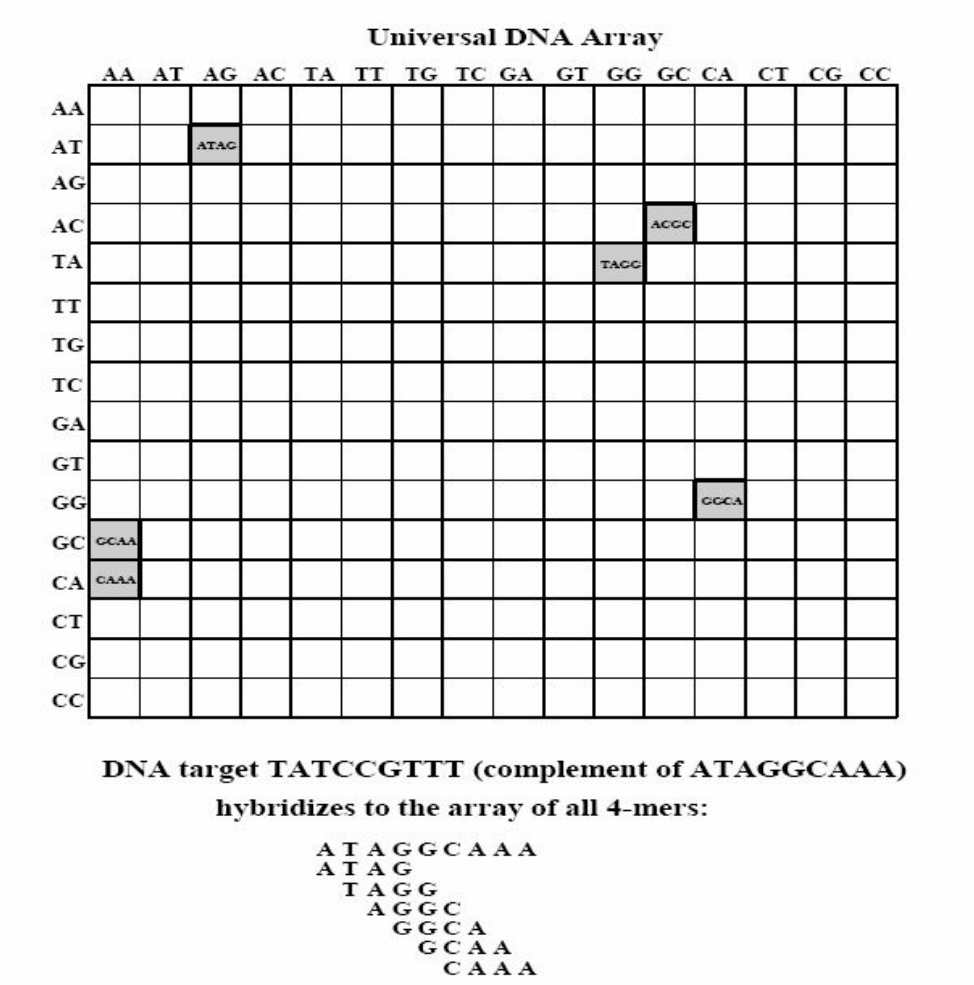
- Attach all possible DNA probes of length *l* to a flat surface, each probe at a distinct and known location. This set of probes is called the DNA array.
- Apply a solution containing fluorescently labeled DNA fragment to the array.
- The DNA fragment hybridizes with those probes that are complementary to substrings of length *l* of the fragment.

How SBH Works (cont'd)

- Using a spectroscopic detector, determine which probes hybridize to the DNA fragment to obtain the *l*–mer composition of the target DNA fragment.

- Apply the combinatorial algorithm (below) to reconstruct the sequence of the target DNA fragment from the *l* – mer composition.

Hybridization on DNA Array



l-mer composition

- ***Spectrum (s, l)*** - *unordered* multiset of all possible *(n – l + 1) l*-mers in a string *s* of length *n*
- The order of individual elements in *Spectrum (s, l)* does not matter
- For *s* = TATGGTGC all of the following are equivalent representations of *Spectrum (s, 3):*

  {TAT, ATG, TGG, GGT, GTG, TGC}

  {ATG, GGT, GTG, TAT, TGC, TGG}

  {TGG, TGC, TAT, GTG, GGT, ATG}

Different sequences – the same spectrum

- Different sequences may have the same spectrum:

  Spectrum(GTATCT,2)=

  Spectrum(GTCTAT,2)=

  {AT, CT, GT, TA, TC}

The SBH Problem

- Goal: Reconstruct a string from its *l*-mer composition
- Input: A set *S*, representing all *l*-mers from an (unknown) string *s*
- Output: String *s* such that *Spectrum (s,l)* = *S*

l-mer composition

- ***Spectrum (s, l)*** - *unordered* multiset of all possible *(n – l + 1)* *l*-mers in a string *s* of length *n*
- The order of individual elements in *Spectrum (s, l)* does not matter
- For *s* = TATGGTGC all of the following are equivalent representations of *Spectrum (s, 3):*

  {TAT, ATG, TGG, GGT, GTG, TGC}

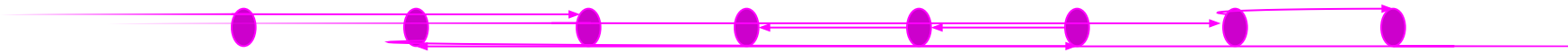
  {ATG, GGT, GTG, TAT, TGC, TGG}

  {TGG, TGC, TAT, GTG, GGT, ATG}

SBH: Hamiltonian Path Approach

S = { ATG AGG TGC TCC GTC GGT GCA CAG }

H ATG AGG TGC TCC GTC GGT GCA CAG



ATG CAG G TCC

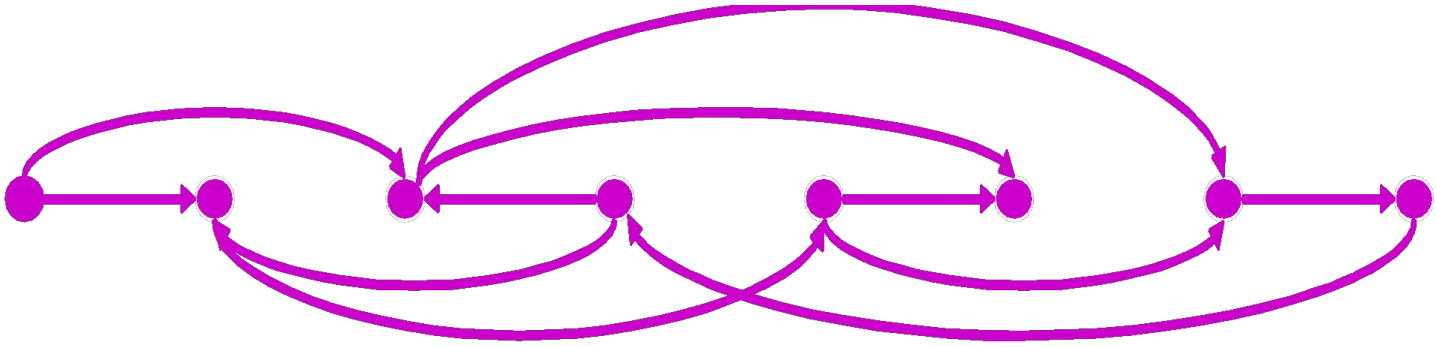
Path visited every VERTEX once

SBH: Hamiltonian Path Approach

A more complicated graph:

S = { ATG TGG TGC GTG GGC GCA GCG CGT }

H

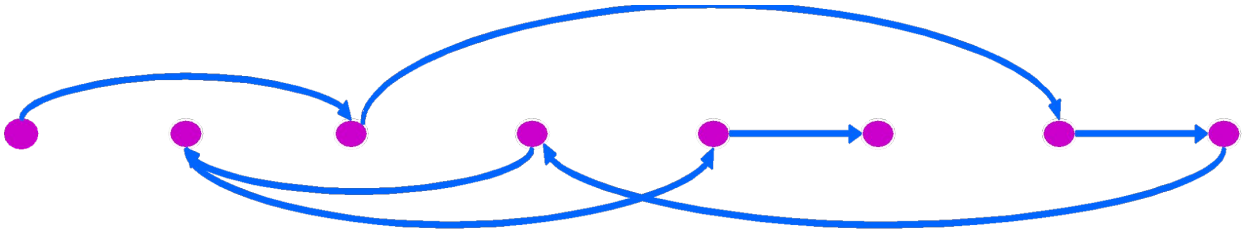


SBH: Hamiltonian Path Approach

S = { ATG TGG TGC GTG GGC GCA GCG CGT }

Path 1:

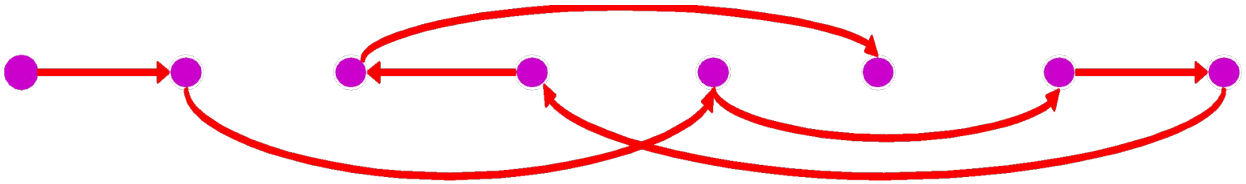
H



ATGCGTGGCA

Path 2:

H



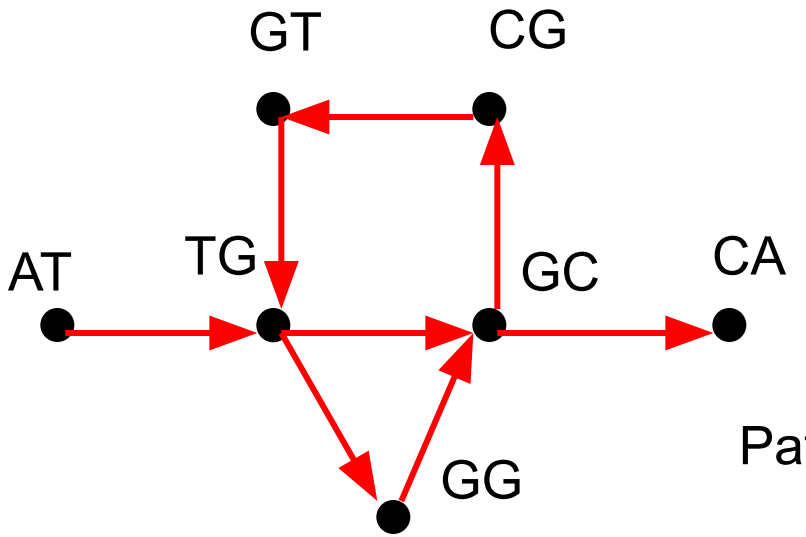
ATGGCGTGCA

SBH: Eulerian Path Approach

S = { ATG, TGC, GTG, GGC, GCA, GCG, CGT }

Vertices correspond to (l – 1) – mers : { AT, TG, GC, GG, GT, CA, CG }

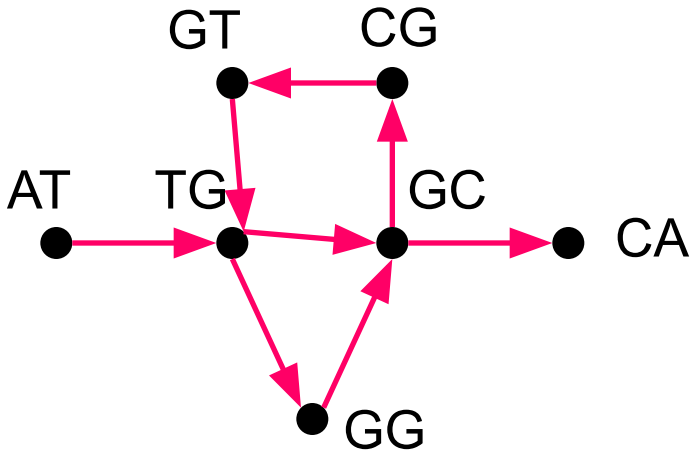
Edges correspond to l – mers from S



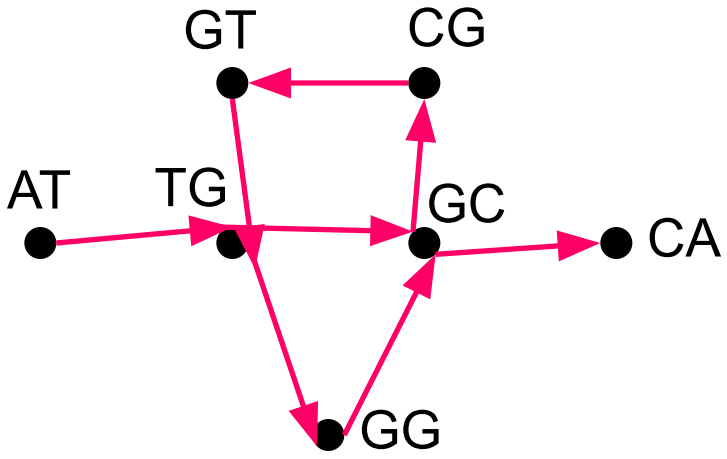
Path visited every EDGE once

SBH: Eulerian Path Approach

S = { AT, TG, GC, GG, GT, CA, CG } corresponds to two different paths:



ATGGCGTGCA



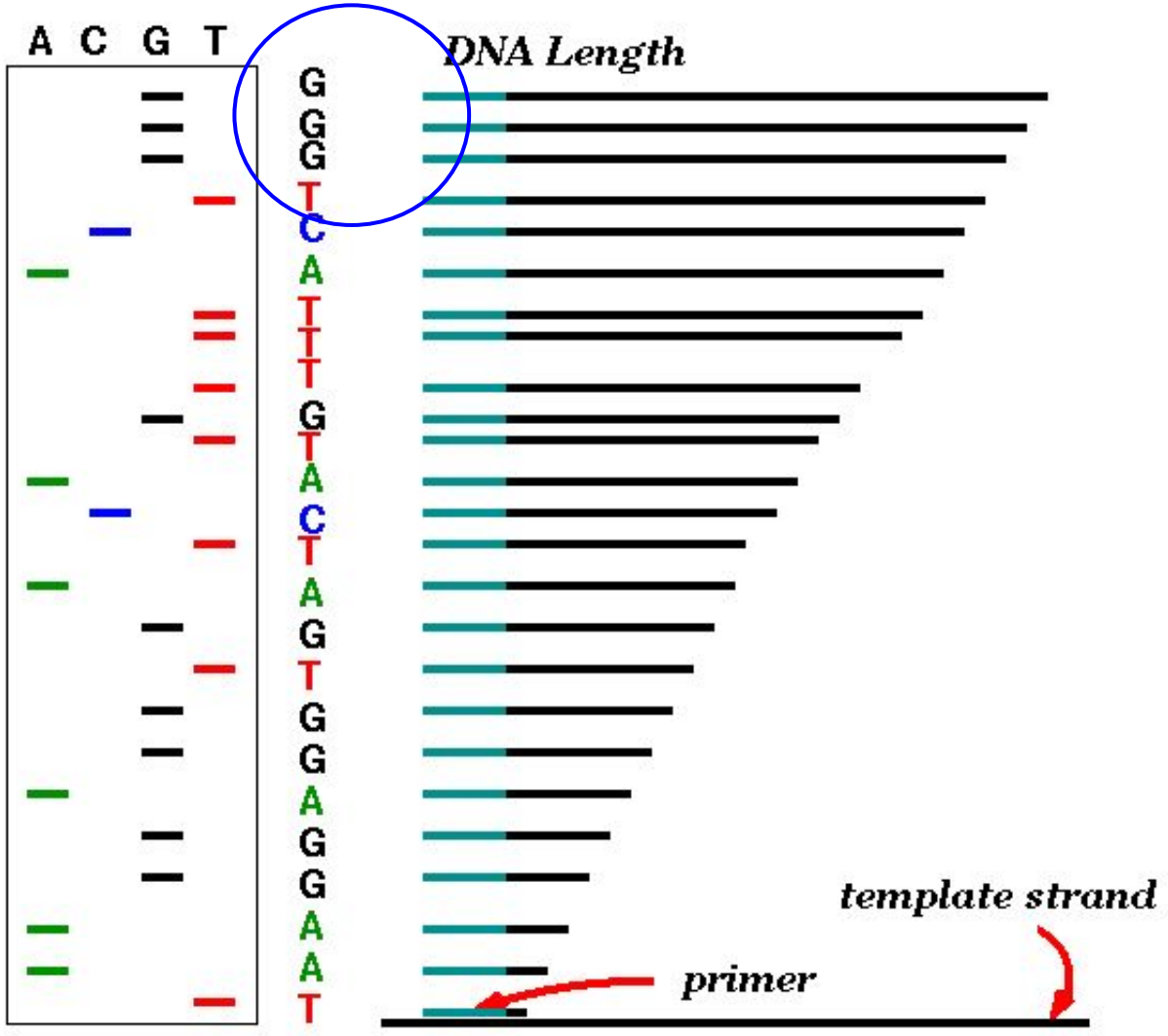
ATGCGTGGCA

Some Difficulties with SBH

- **Fidelity of Hybridization:** difficult to detect differences between probes hybridized with perfect matches and 1 or 2 mismatches
- **Array Size:** Effect of low fidelity can be decreased with longer *l*-mers, but array size increases exponentially in *l.* Array size is limited with current technology.
- **Practicality:** SBH is still impractical.
- **Practicality again**: Although SBH is still impractical, it spearheaded expression analysis and SNP analysis techniques

DNA sequencing – gel electrophoresis

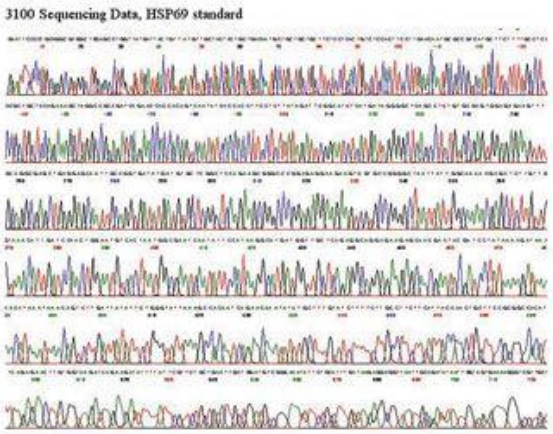
1. Start at primer (restriction site)
2. Grow DNA chain
3. Include dideoxynucleotide (modified a, c, g, t)
4. Stops reaction at all possible points
5. Separate products with length, using gel electrophoresis



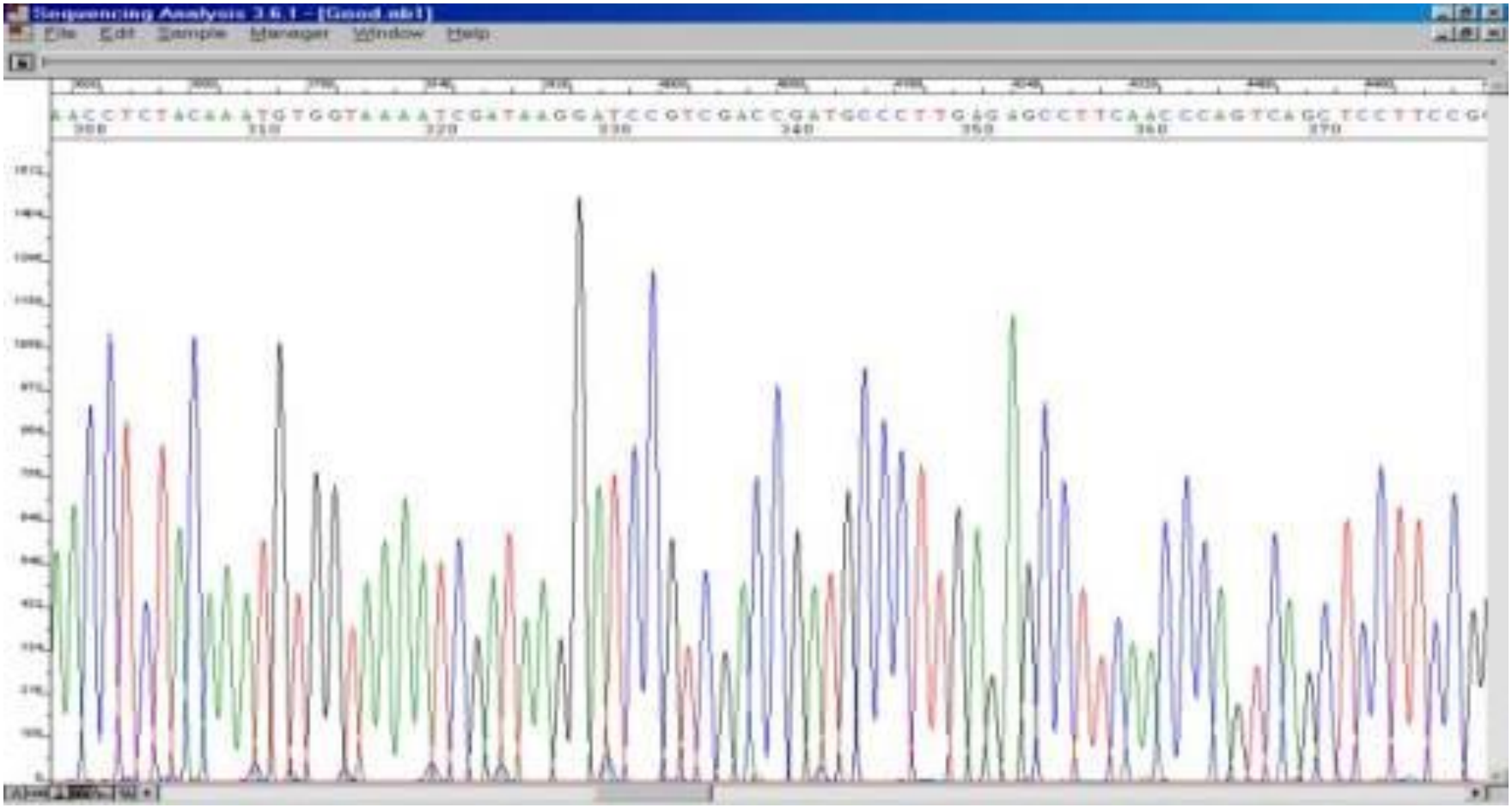
Capillary (Sanger) sequencing

Capillary sequencing (Sanger):

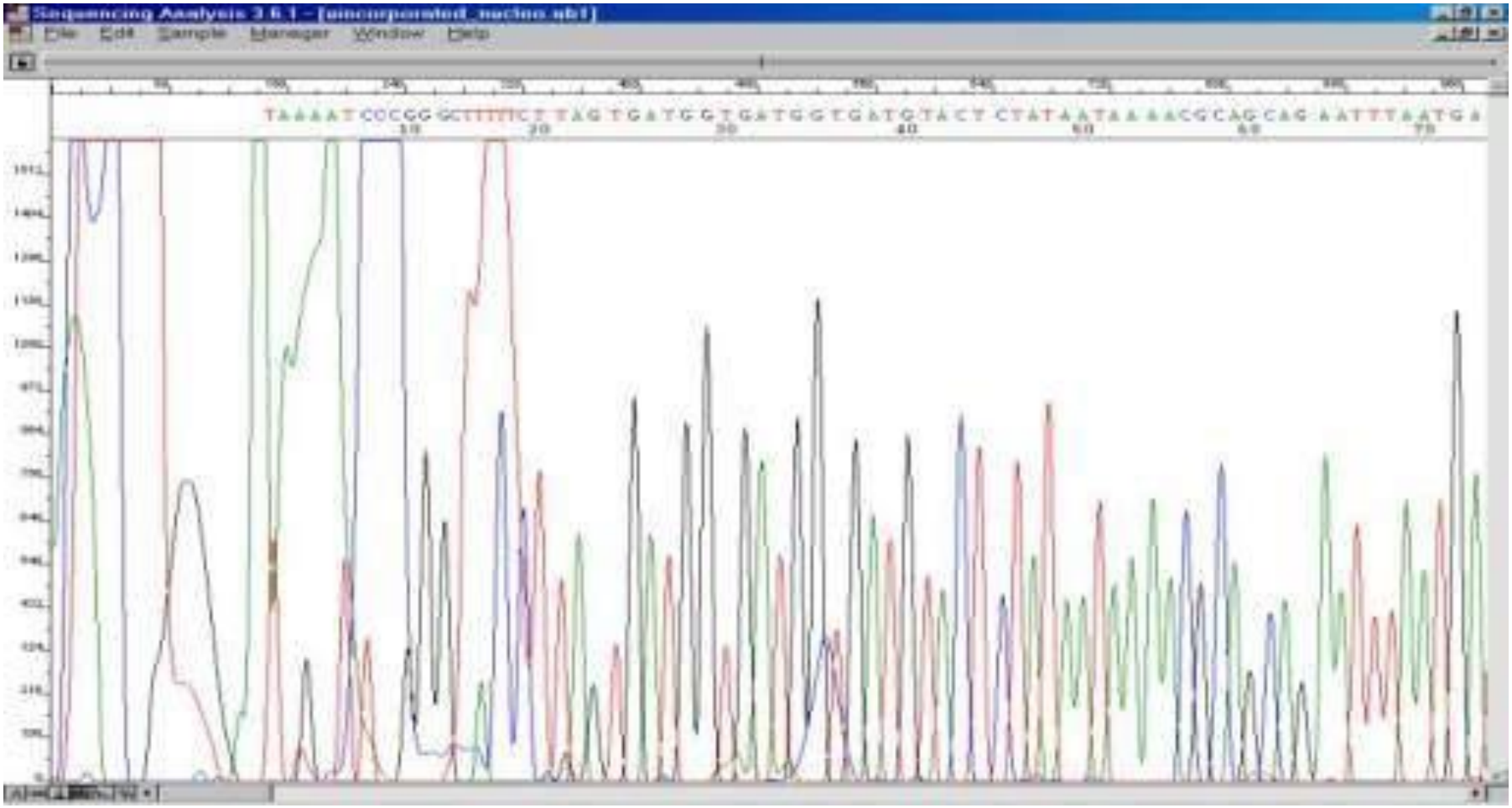
Can only sequence ~1000 letters at a time



Electrophoresis diagrams



Challenging to Read Answer



Reading an electropherogram

1. Filtering
2. Smoothening
3. Correction for length compressions
4. A method for calling the letters – **PHRED**

PHRED – **PH**il's **R**evised **ED**itor (by Phil Green)

Based on dynamic programming

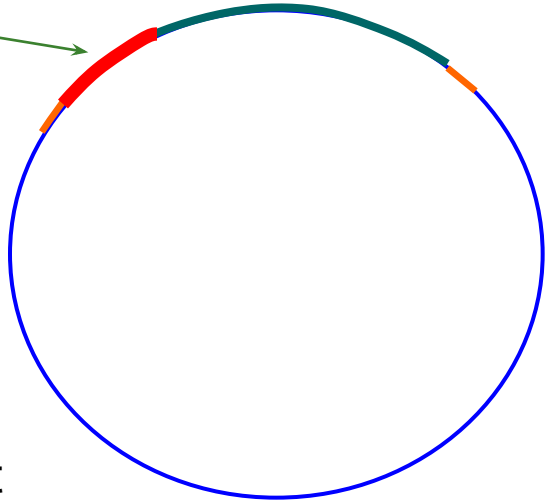
PHRAP – **PH**il's **R**evised **A**ssembly **P**rogram (by Phil Green)

(small) genome assembler

Output of PHRED: a read

A **read**: ~1000 nucleotides

```
A  C  G  A  A  T  C  A  G  …A
16 18 21 23 25 15 28 30 32 …21
```



Quality scores: -10*$\log_{10}$Prob(Error)

"FASTQ format": ASCII character that corresponds to *q+33* (or 64)

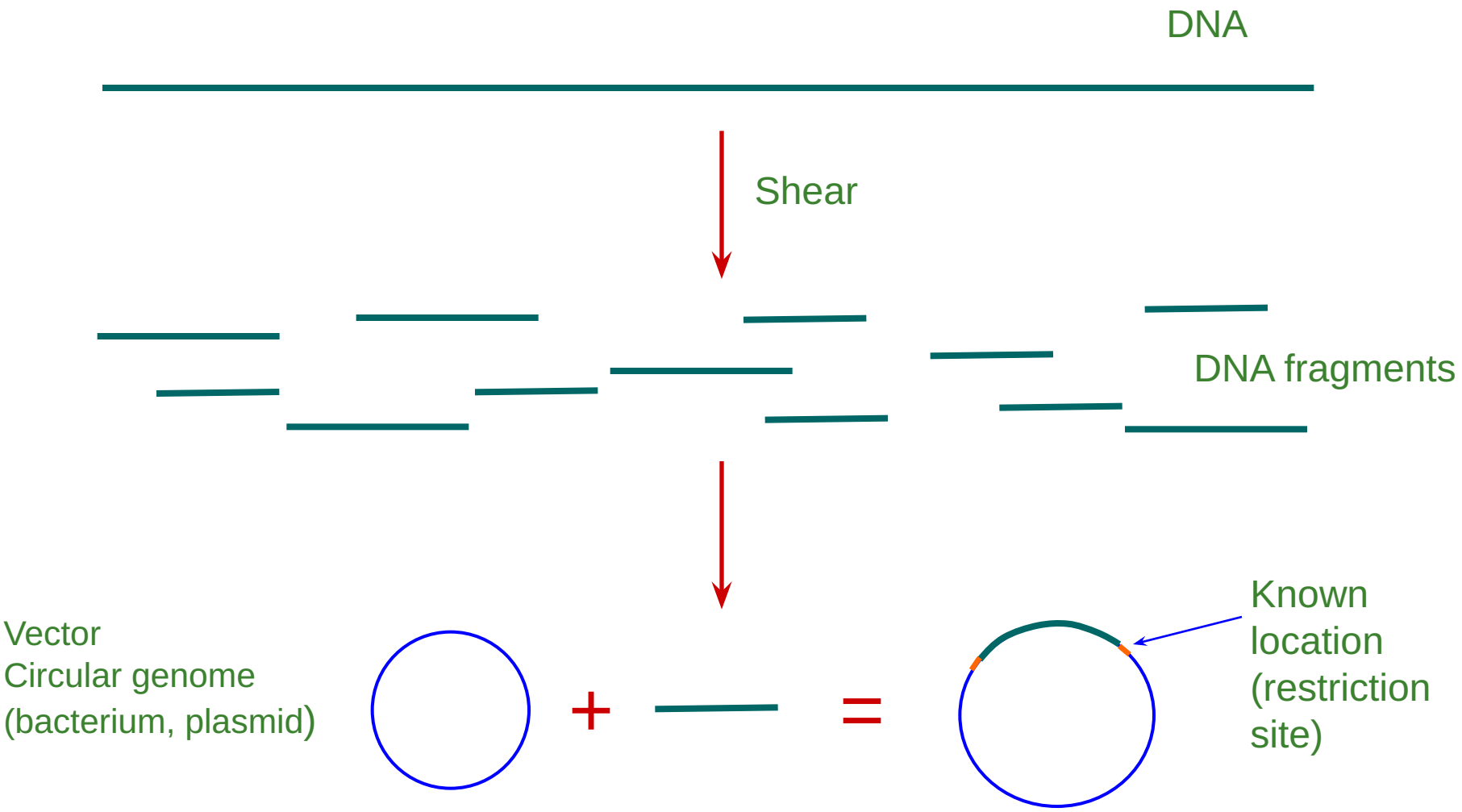
(I = 73; 73-33 = 40 = *q;* q40-> 0.01% error)

Reads can be obtained from leftmost, rightmost ends of the insert

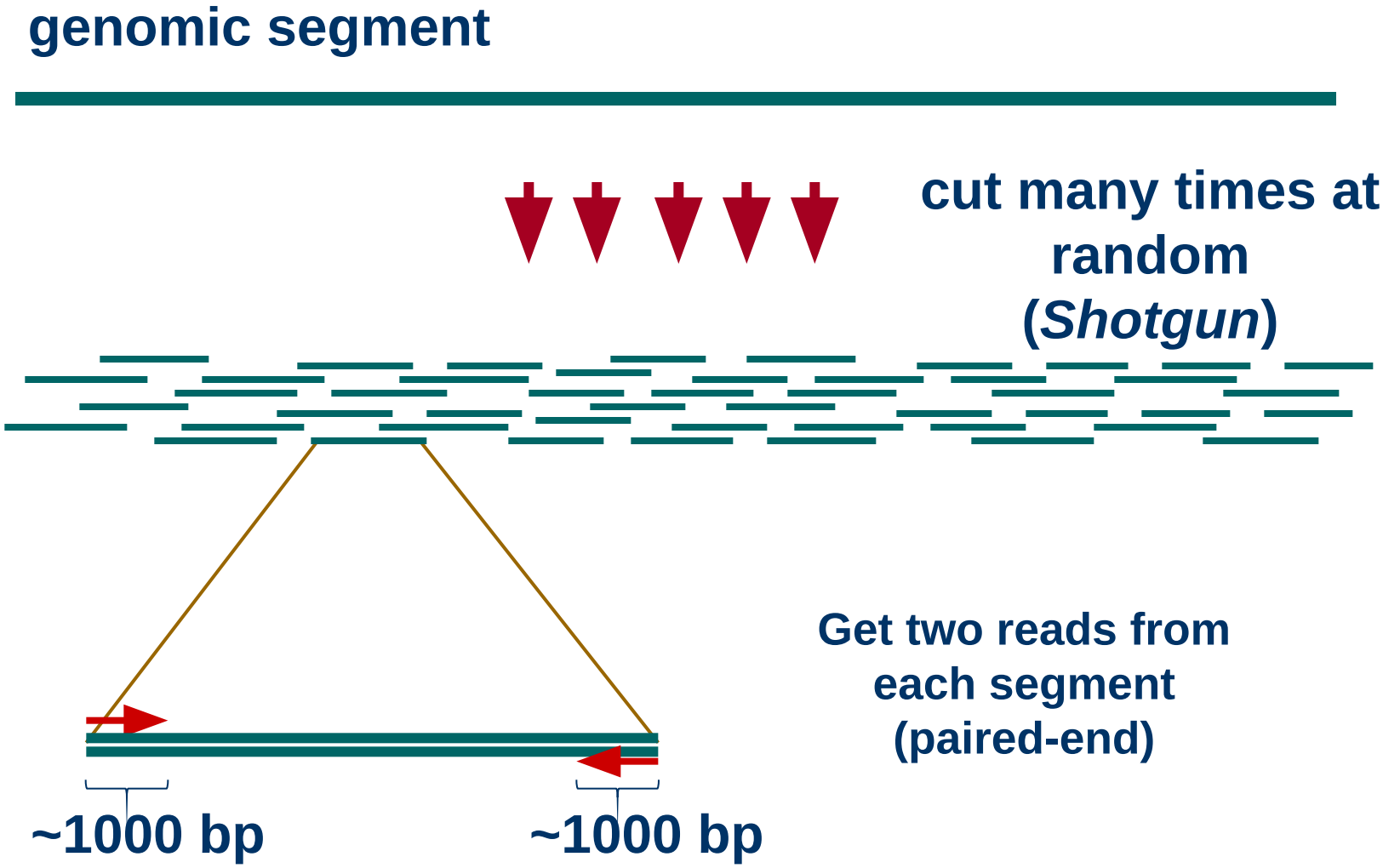
Double-barreled (paired-end, matepair) sequencing:

Both leftmost & rightmost ends are sequenced

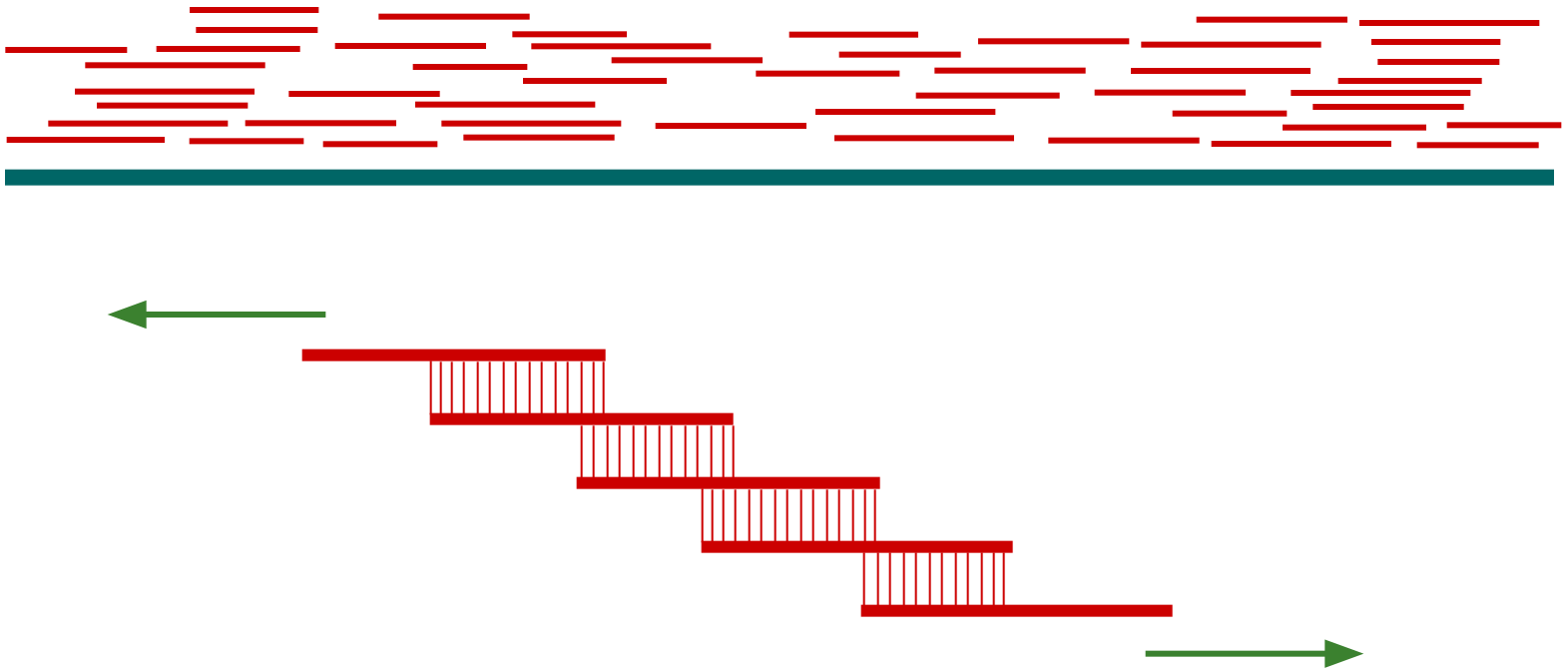
Traditional DNA Sequencing



Double-barreled sequencing



Reconstructing The Sequence



Need to cover region with >7-fold redundancy (7X) if you use Sanger technology

Overlap reads and extend to reconstruct the original genomic region

Definition of Coverage

Length of genomic segment: **L**
Number of reads: **n**
Length of each read: **l**

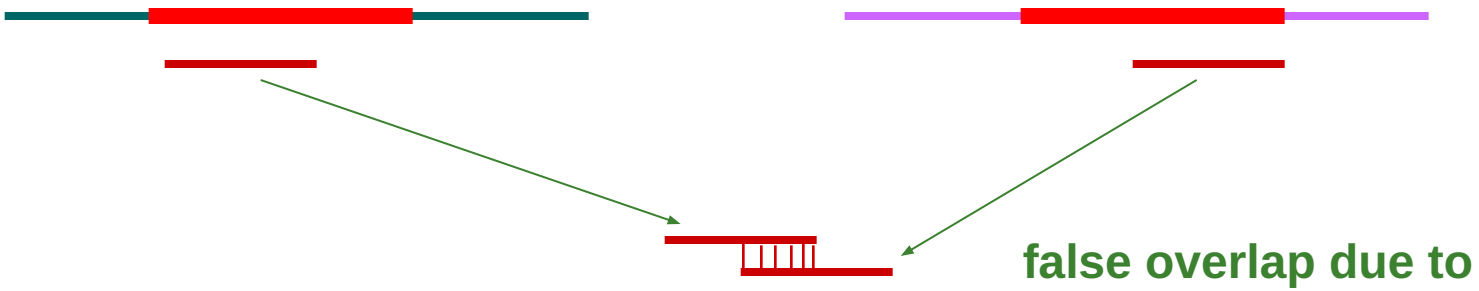
Definition: Coverage **C = n l / L**

How much coverage is enough?

Lander-Waterman model:
Assuming uniform distribution of reads, C=10 results in 1 gapped region /1,000,000 nucleotides

Challenges with Fragment Assembly

- **Sequencing errors**
  ~0.1% of bases are wrong
- **Repeats**



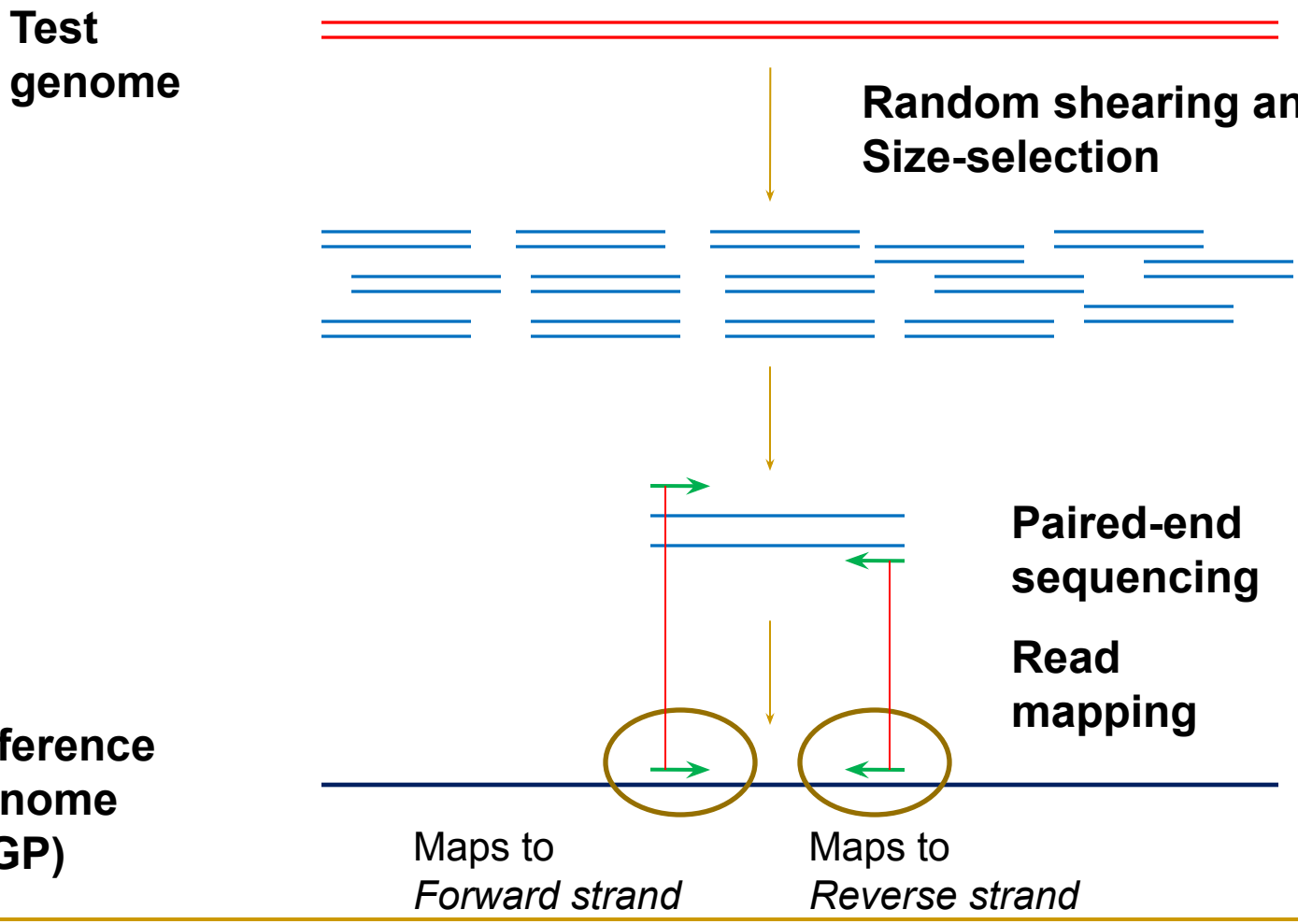
- **Computation: ~ O(N^2) where N = # reads**

Sanger sequencing

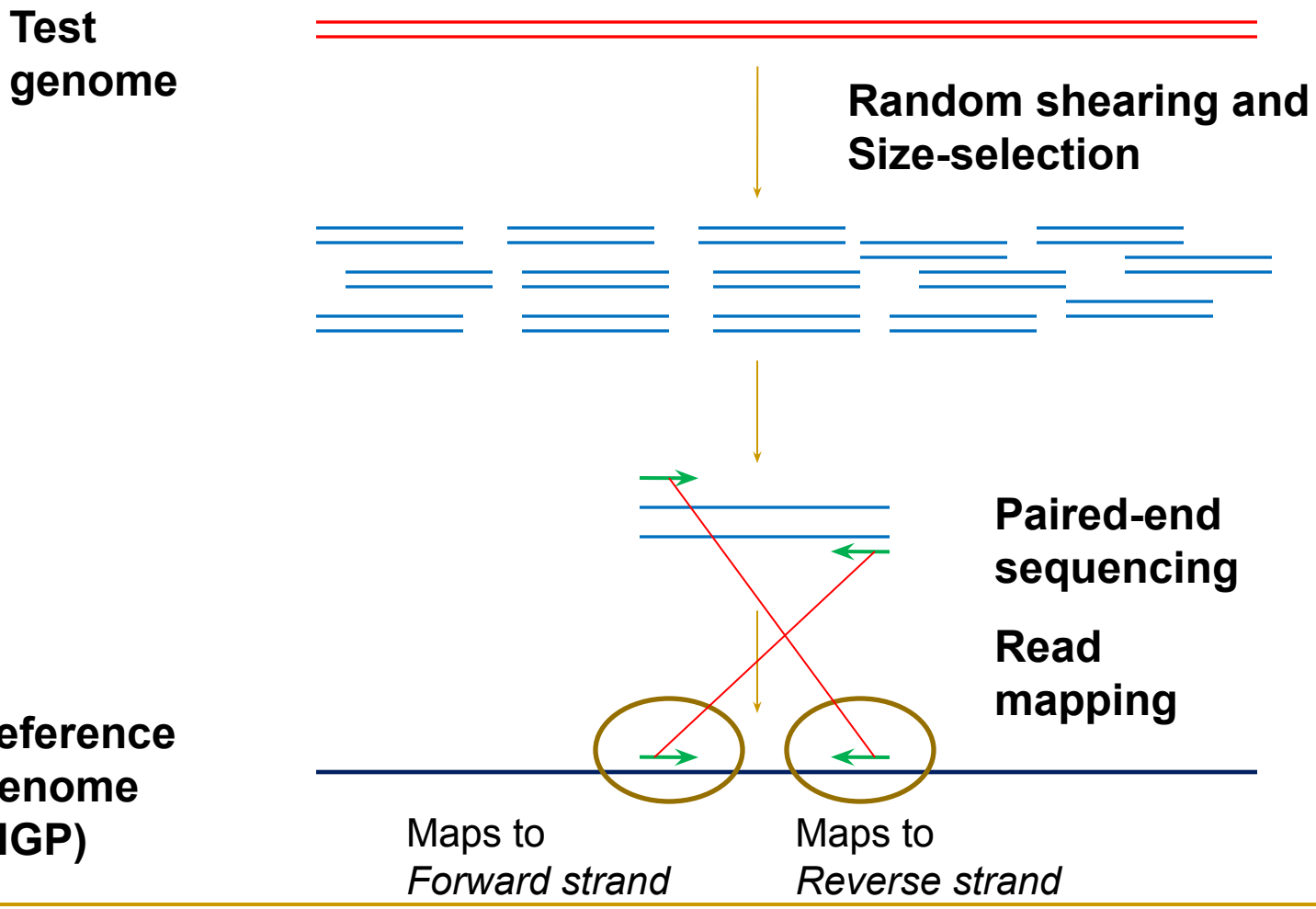
- Advantages
  - Long read lengths (>1000 bp)
  - Highest sequence accuracy (error < 0.1%)
  - Clone libraries can be used in further processing
- Disadvantages
  - The most expensive technology
    - $1500 per Mb
  - Building and storing clone libraries is hard & time consuming

HIGH THROUGHPUT SEQUENCING

Whole Genome Sequencing



Whole Genome Sequencing



HTS Technologies

- ~~454 Life Sciences: the first, acquired by Roche~~
  - ~~*Pyrosequencing*~~
- Illumina (Solexa): current market leader
  - *GAIIx, HiSeq2000-2500-3000-4000, X Ten, NextSeq, MiSeq, NovaSeq*
  - *Sequencing by synthesis*
- Applied Biosystems:
  - ~~*SOLiD: "color-space reads"*~~
  - *Ion Torrent*
- Pacific Biosciences (PacBio)
- Oxford Nanopore (ONT)

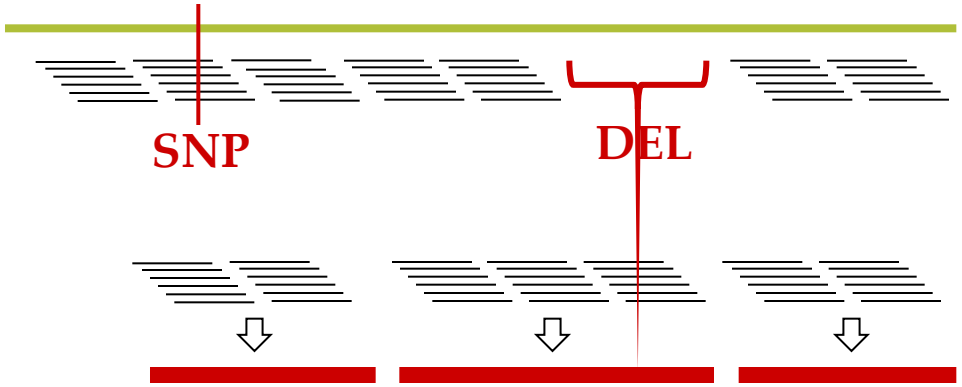
Features of HTS data

- Short sequence reads
  - –~500 bp: 454 (Roche)
  - – 100 – 150 bp Solexa(Illumina), SOLiD(AB)
- Longer
  - •PacBio: 8-20 Kb
  - •ONT: 10-100 Kb
- •Huge amount of sequence per run
  - –Gigabases per run (4 Tbp for Illumina/HiSeq4000)
- Huge number of reads per run
  - •Up to billions
- Bias against high and low GC content (Illumina and Ion Torrent)
  - •GC% = (G + C) / (G + C + A + T)
- Higher error (compared with Sanger)
  - –Different error profiles
  - –10% PacBio, 1% PacBio CCS (HiFi), 5% ONT

Current and future application areas

Genome re-sequencing: somatic mutation detection, organismal SNP discovery, mutational profiling, structural variation discovery

reference genome



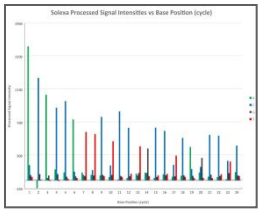
De novo genome sequencing

Also:

- **DNA-protein interaction analysis (CHiP-Seq)**
- **novel transcript discovery**
- **quantification of gene expression**
- **epigenetic analysis (methylation profiling)**

Fundamental informatics challenges

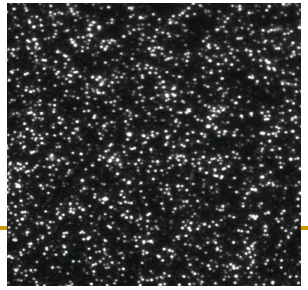
1. Interpreting machine readouts – base calling, base error estimation

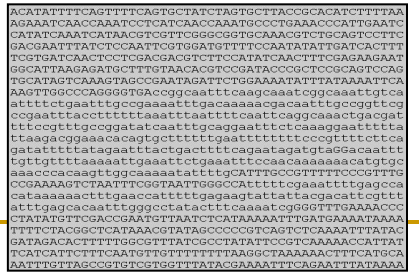


```
>B_TITR_1_1_668_35 TIME: Tue Feb 20 02:26:06 2007
ATATCGGATGACACAATATGGGAGGTTGAC
>B_TITR_1_2_843_403 TIME: Tue Feb 20 02:26:06 2007
TGTAGCTTTTCATGACAATTTTATAGGTGT
>B_TITR_1_1_668_35
27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 27 6
>B_TITR_1_2_843_403
26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 26 24
>B_TITR_1_3_618_922
```

2. Data visualization

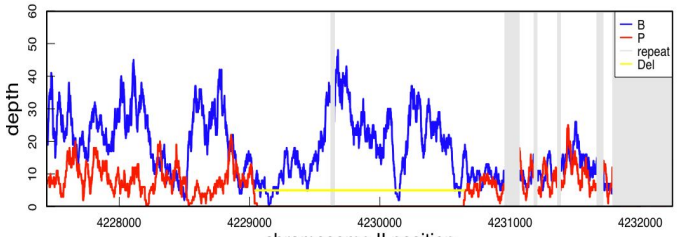
3. Data storage & management
Gzip compressed raw data for one human genome > 100 GB

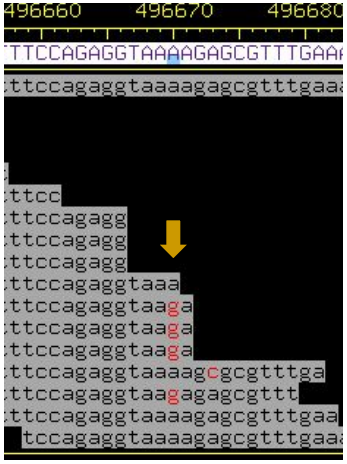




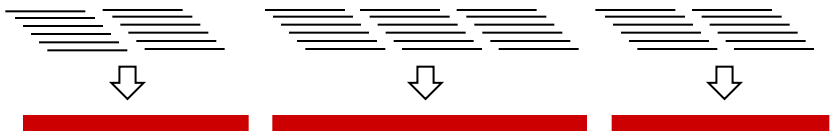
Informatics challenges (cont'd)

4. SNP, indel, and structural variation discovery





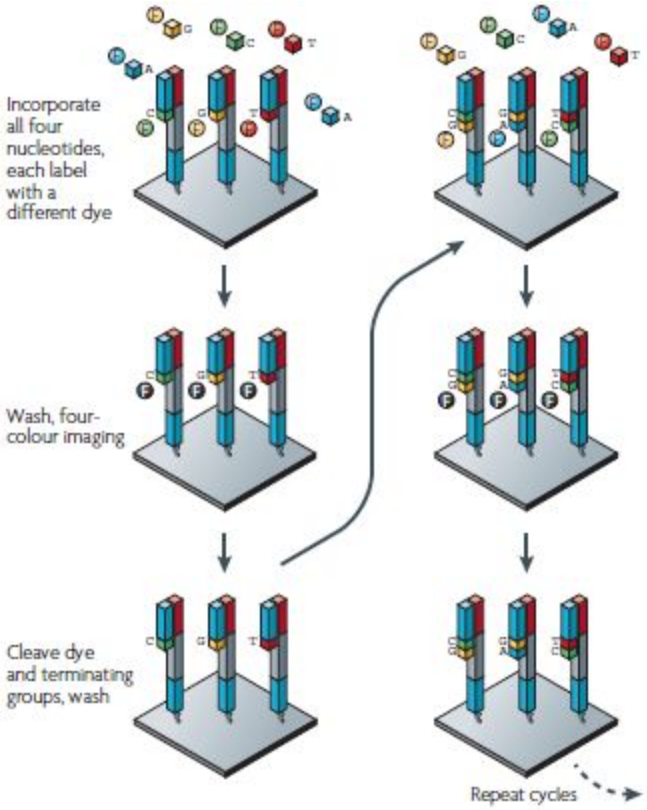
5. *De novo* Assembly

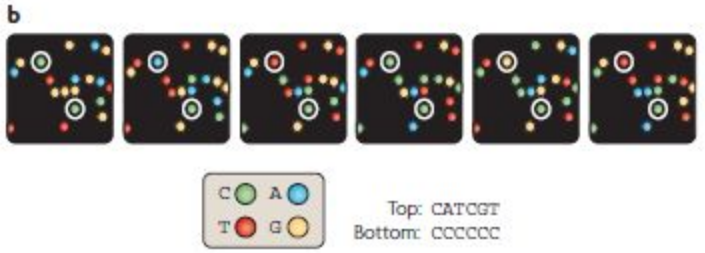


Illumina (Solexa)

- Current market leader
- Based on *sequencing by synthesis*
- Current read length 100-150bp (up to 300 bp with more errors)
- Paired-end
- Error ~0.1%
  - Mismatch errors dominate
- Throughput: 6 Tbp in one run (2 days)
- Cheapest sequencing technology
  - Cost: ~$1000 per human genome

Illumina





NovaSeq

MiSeq

HiSeq 2000/2500/4000

Illumina (Solexa)

•Read length and quality string length are the same

Read and Quality (1)

@FC81ET1ABXX:3:1101:1215:2154/1

TTTTTCAAATGTTTGTTGCCTATTTTTATATCTTCTTTTGAGAATTGTCTGTTCATGTCNTNNGNNCNCNNTNTCANGGGATTGTTTGTT

+

HHGHHHHHGHHHHDHFHHHHHHHFHHHHHHEHHEHHHHEGGDEF2CGDCDFB0>DA###################################

Read and Quality (2)

@FC81ET1ABXX:3:1101:1215:2154/2

AAGCCANNTNNNNNNNNNNNNACTGGATCCTCATAGCTCACCTTATGCAAAAATCAACTCAAGATGGATGAAGGTCTTAAACCTAATAC

+

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

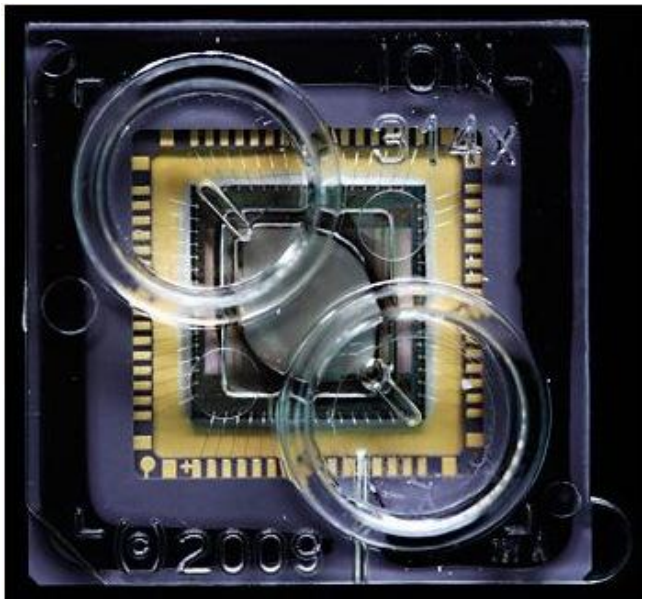
Illumina (Solexa)

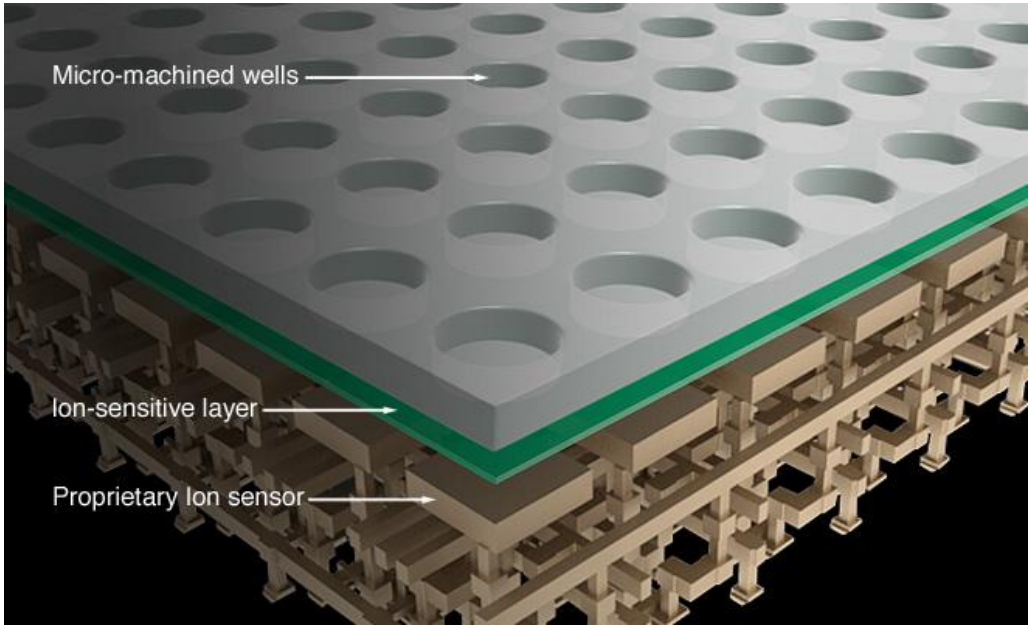
- Read mapping:
  - mrFAST, mrsFAST, BWA, MAQ, BFAST, MOSAIK, Bowtie, SOAP, SHRiMP, many more
- *De novo* assembly:
  - EULER, Velvet, ABySS, Hapsembler, SGA, ALLPATHS, ….

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

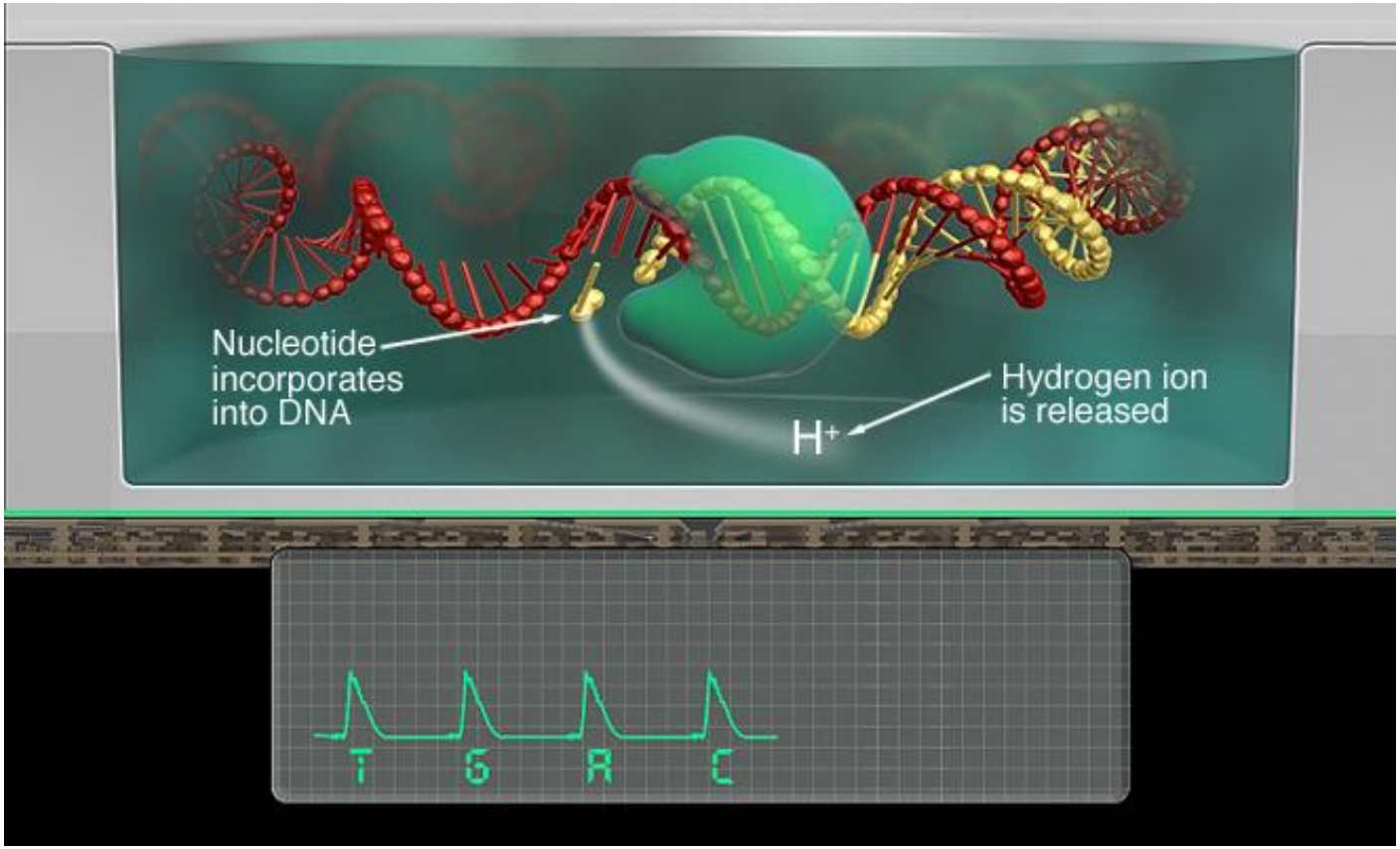
Ion Torrent


ION
314
©2009

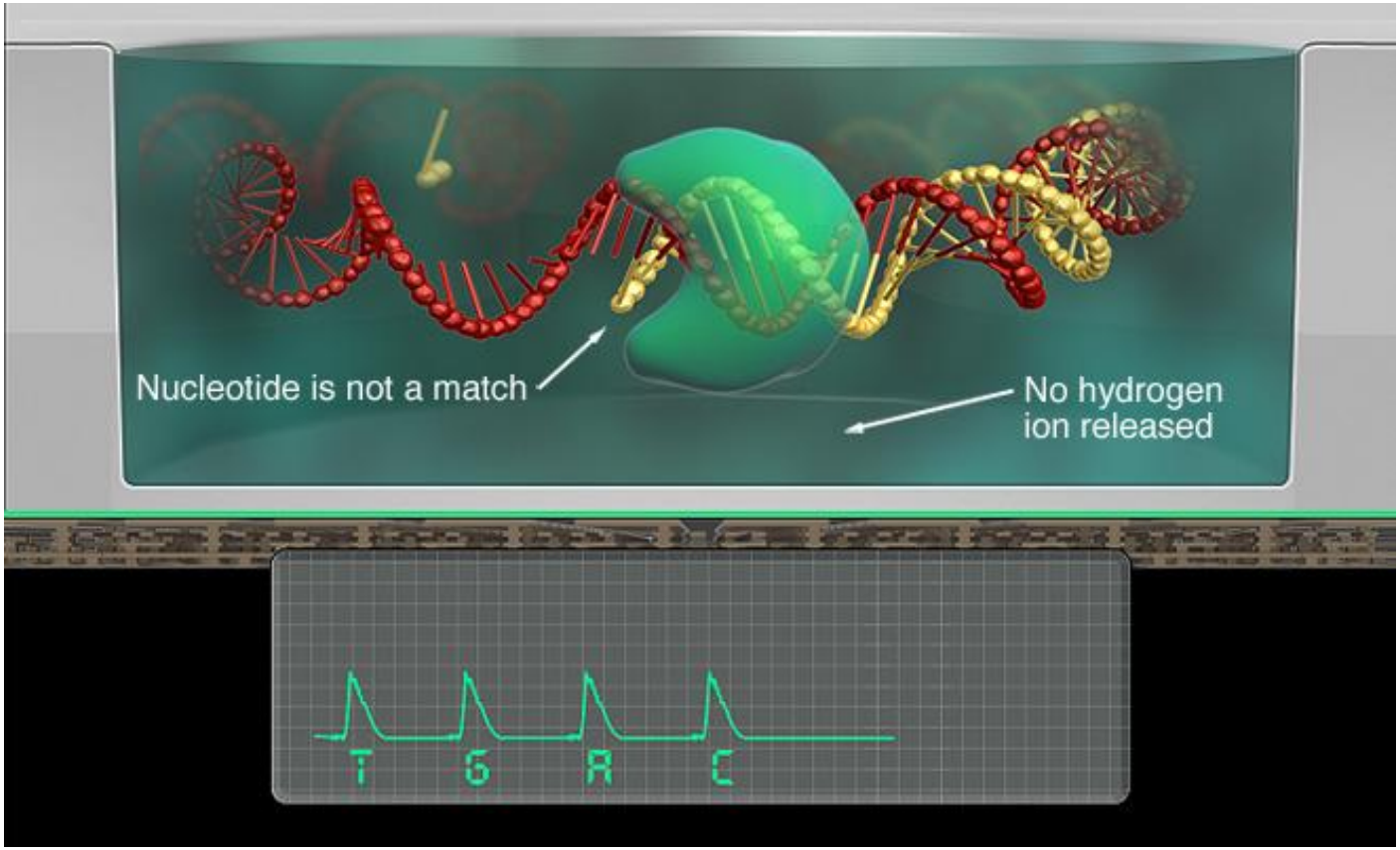

Micro-machined wells
Ion-sensitive layer
Proprietary Ion sensor


ion torrent
Proton

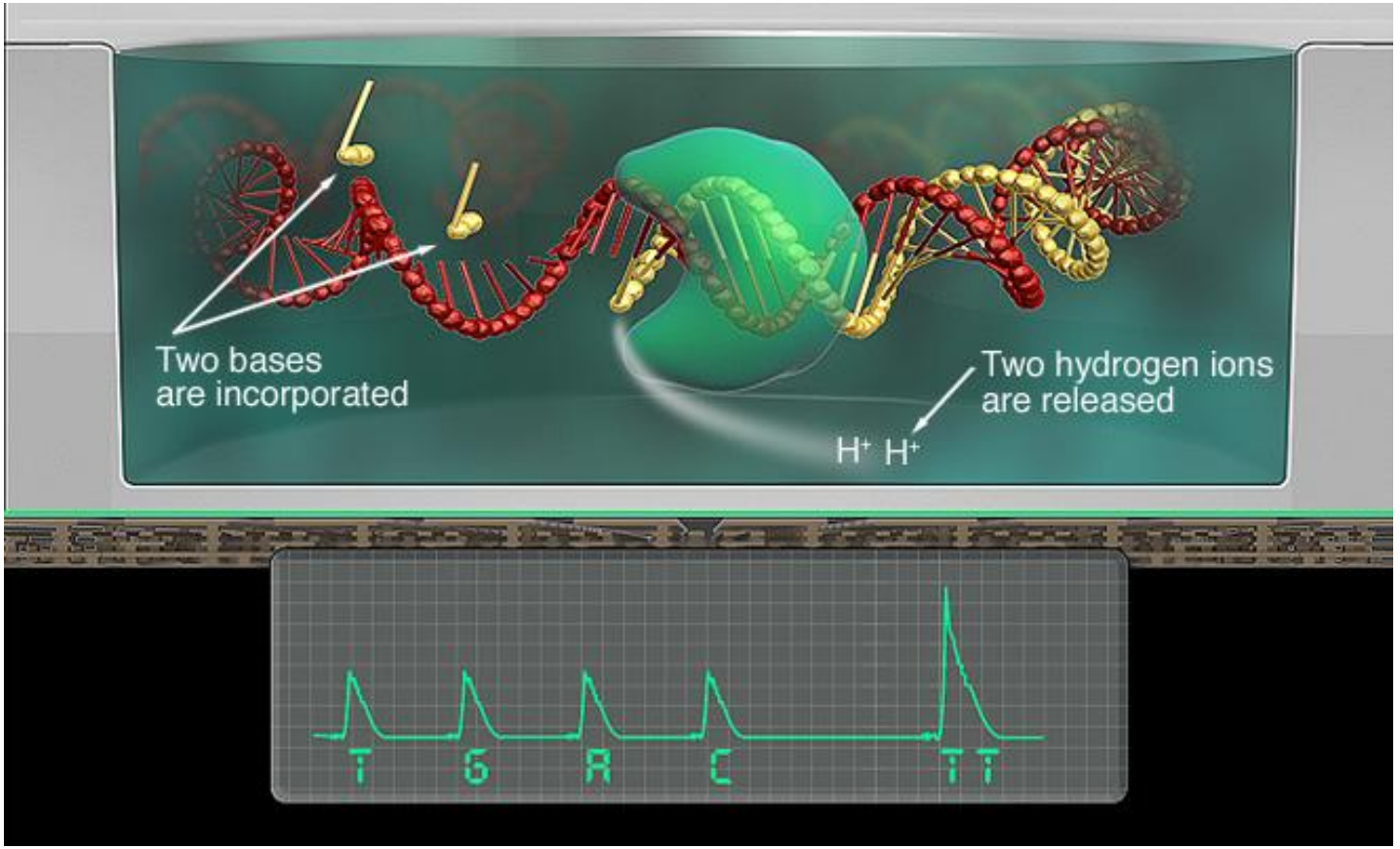
Ion Torrent


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

Ion Torrent

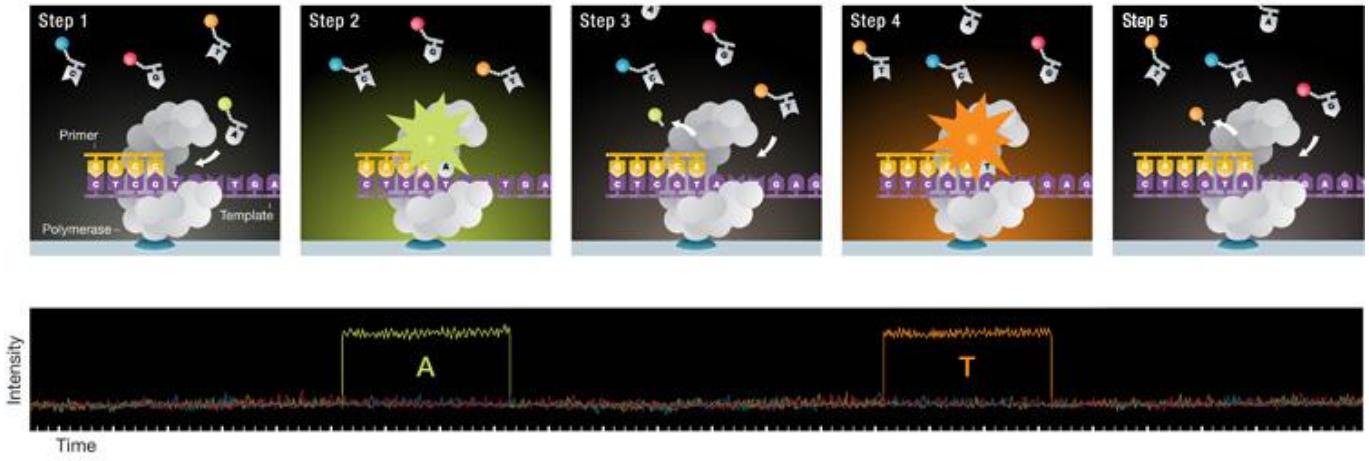


Ion Torrent


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 (10-80 Kb)
- Errors: ~10%; indel dominated

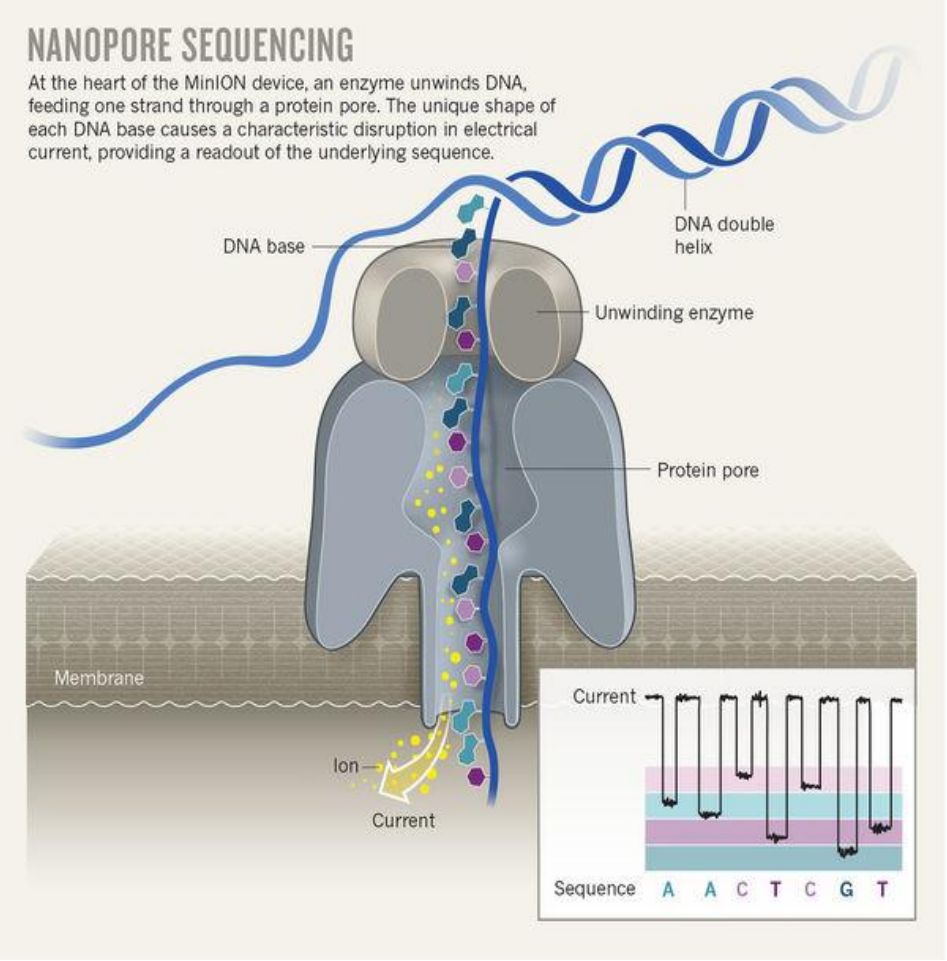


Pacific Biosciences

- For any DNA polymerase you can read a total of ~70 kb (median) sequence
- Two sequencing protocols:
  - CLR: single read
  - CCS: Make a circle, re-read the same molecule 5-6 times
    - Renamed as HiFi
    - Multiple sequence alignment to correct errors
    - Median length = 60000 / 6= 10 Kbp
    - > 99% accuracy

Nanopore sequencing

- Up to 2 Mbp reads
  - ~5% error, indel dominated
- Real-time analysis supported
- RNN-based basecallers



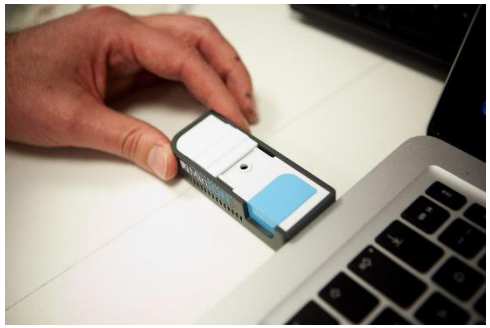
Nanopore sequencing technology and tools for genome assembly: computational analysis of the current state, bottlenecks and future directions FREE

Damla Senol Cali ✉, Jeremie S Kim, Saugata Ghose, Can Alkan, Onur Mutlu

Briefings in Bioinformatics, bby017, https://doi.org/10.1093/bib/bby017
Published: 02 April 2018 **Article history** ▾

Nanopore Sequencing

- Nanopore sequencing:
  - Oxford Nanopore Technologies (ONT)
    - 100 Kb reads
    - 20% error rate
      - Latest: 5% error rate



MinION

SmidgION

PromethION

HTS: 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

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

Index	Entry	Index	Entry

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

Index	Entry	Index	Entry
0	a		

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

Index	Entry	Index	Entry
0	a		
1	b		
2	ab		

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

Index	Entry	Index	Entry
0	a		
1	b		
2	ab		

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

Index	Entry
0	a
1	b
2	ab
3	bb
4	ba

Index	Entry

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---

Index	Entry	Index	Entry
0	a		
1	b		
2	ab		
3	bb		
4	ba		
5	aa		

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---

Index	Entry	Index	Entry
0	a		
1	b		
2	ab		
3	bb		
4	ba		
5	aa		
6	abb		

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---

Index	Entry	Index	Entry
0	a		
1	b		
2	ab		
3	bb		
4	ba		
5	aa		
6	abb		

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---

Index	Entry	Index	Entry
0	a	7	baa
1	b		
2	ab		
3	bb		
4	ba		
5	aa		
6	abb		

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-------

Index	Entry	Index	Entry
0	a	7	baa
1	b		
2	ab		
3	bb		
4	ba		
5	aa		
6	abb		

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-------

Index	Entry	Index	Entry
0	a	7	baa
1	b	8	aba
2	ab		
3	bb		
4	ba		
5	aa		
6	abb		

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---

Index	Entry	Index	Entry
0	a	7	baa
1	b	8	aba
2	ab		
3	bb		
4	ba		
5	aa		
6	abb		

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---

Index	Entry	Index	Entry
0	a	7	baa
1	b	8	aba
2	ab	9	abba
3	bb		
4	ba		
5	aa		
6	abb		

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---

Index	Entry	Index	Entry
0	a	7	baa
1	b	8	aba
2	ab	9	abba
3	bb		
4	ba		
5	aa		
6	abb		

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---

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

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-------

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

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-------

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		
6	abb		

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---

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		
6	abb		

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---

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		

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

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		

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

Reordering improves locality

File Size: 250MB, 5Mil 51bp Bacterial Genome

Pre-processing	Time (s)	Gzip time	Size (MB)	Comp. Factor	Boosting
-	-	70	65	4	-
Mapping	180	21	20	12.5	3.25
Lexo. Sorting	10	30	26	9.61	2.5
Cores*	10	21	21	11.9	3.1

*** Idea behind SCALCE**

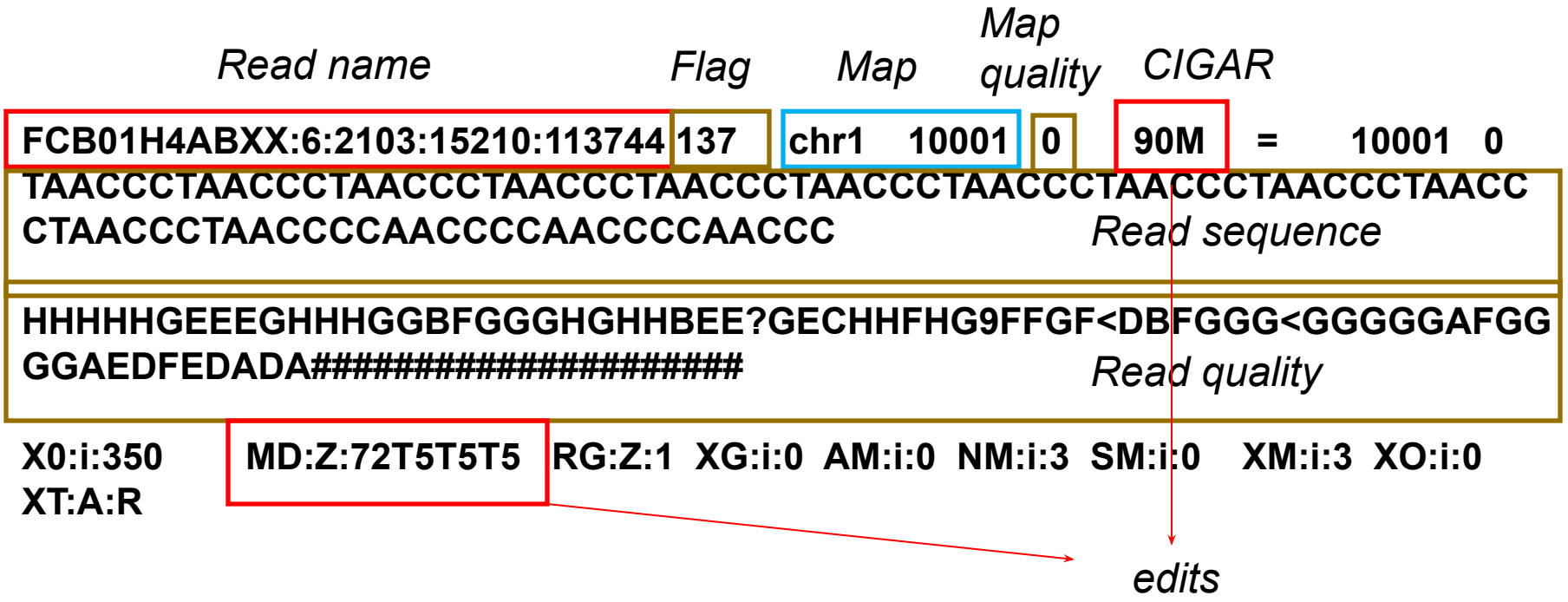
Reordering example

Ref: AAAAAATGACGTCTCTCCTCCTTTTTTAAAACCT

Original	Mapping	Sorting	Cores
CTTTTT	AAAAAA	AAAAAA	AAAAAA
GATGAC	TAATGA	ATGACG	TAAAAC
CCCCCT	GATGAC	CCCCCT	CCCCCT
AAAAAA	ATGACG	CTTTTT	CTTTTT
ATGACG	CCCCCT	GATGAC	TAATGA
TAAAAC	CTTTTT	TAAAAC	GATGAC
TAATGA	TAAAAC	TAATGA	ATGACG

Reference-based compression: CRAMtools

Post mapping; SAM format:



- Read name is unnecessary
- Flag tells you whether /1 or /2
- Map location and edit fields (CIGAR & MD) can be used to regenerate reads
- Don't store quality if edit distance = 0; otherwise only keep the qualities of changed bases

Fritz et al. Genome Research, 2011

CRAMtools

Post mapping; SAM format:

Read name *Flag* *Map* *Map quality* *CIGAR*

FCB01H4ABXX:6:2103:15210:113744 137 chr1 10001 0 90M = 10001 0
TAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACCCTAACC
CTAACCCTAACCCCAACCCCAACCCCAACCC *Read sequence*

HHHHHGEEEGHHHGGBFGGGHGHHBEE?GECHHFHG9FFGF<DBFGGG<GGGGGAFGG
GGAEDFEDADA###################### *Read quality*

X0:i:350 MD:Z:72T5T5T5 RG:Z:1 XG:i:0 AM:i:0 NM:i:3 SM:i:0 XM:i:3 XO:i:0
XT:A:R

Keep: **137** ; **chr1:10001** ; **0** ; **90M**; **72T5T5T5** ; **(#,#,#)**
Add a layer of Huffman encoding

Fritz et al. Genome Research, 2011

CRAMtools: test case

- One human genome
  - 40X coverage
  - 134 GB gzipped = 479 GB raw text
  - Mapped with BWA; >1 day with 200 CPUs
  - SAM format converted to BAM file: 112 GB
  - BAM to CRAM: 7.5 GB
  - Decode CRAM to BAM: 33 GB (lossy!!!)

HTS/algorithms

READ MAPPING

Read Mapping

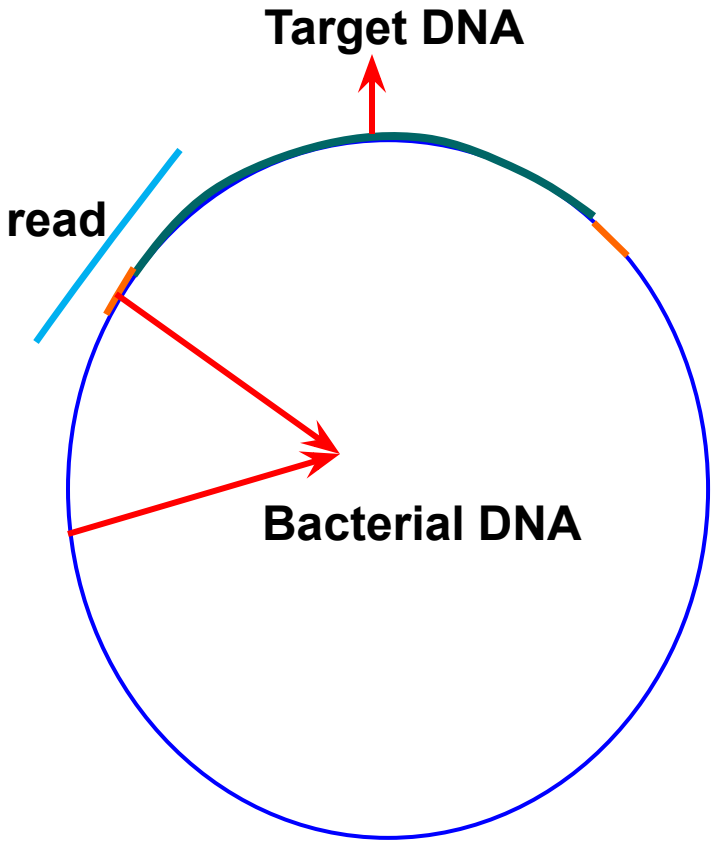
- When we have a reference genome & reads from DNA sequencing, which part of the genome does it come from?
- Challenges:
  - Sanger sequencing
    - Cloning vectors
    - Millions of long (~1000 bp reads)
  - HTS sequencing:
    - Billions of short reads with low error
    - OR: hundreds of millions of long reads with high error
  - Common: sequencing errors
    - More prevalent in HTS
  - 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 HTS: cloning vectors

- Sanger reads may contain sequence from the cloning vector; thus mapping needs *local alignment.*
- No cloning vectors in HTS, *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*CCCAATATATAT*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

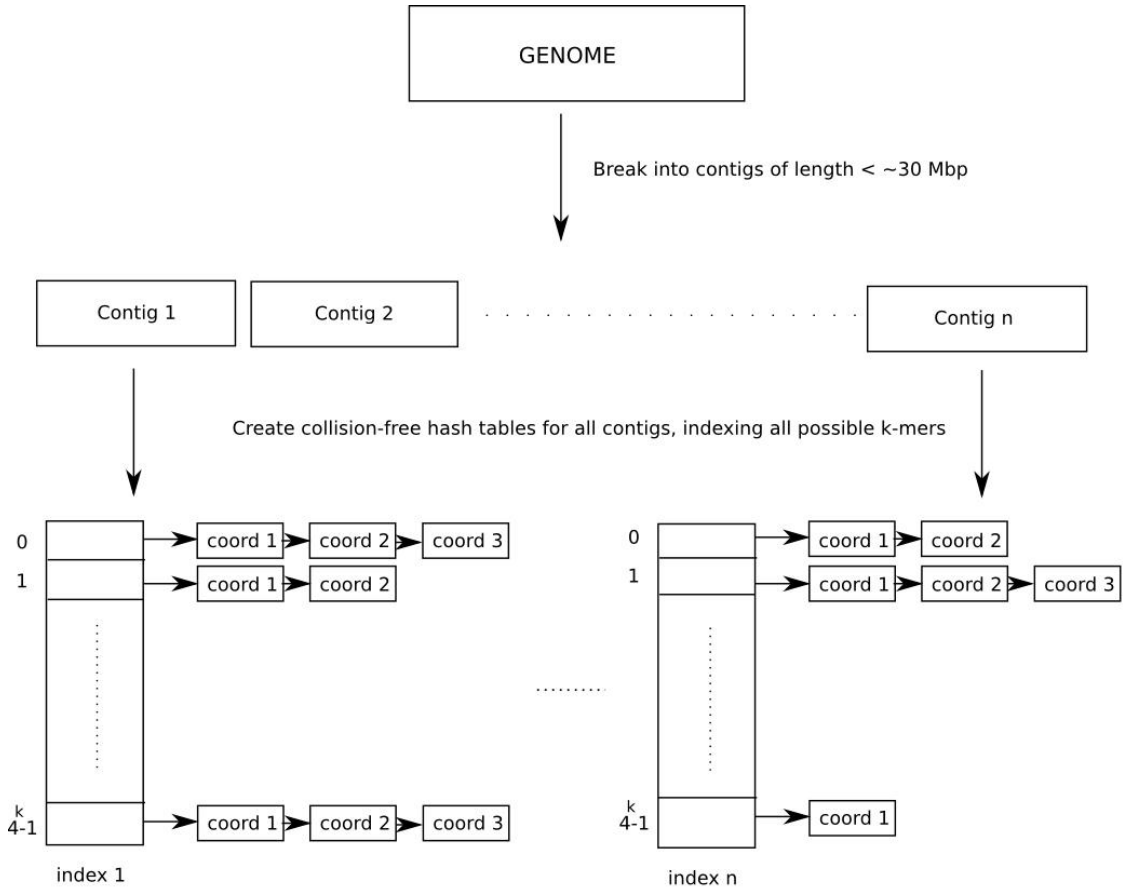
Short read mappers

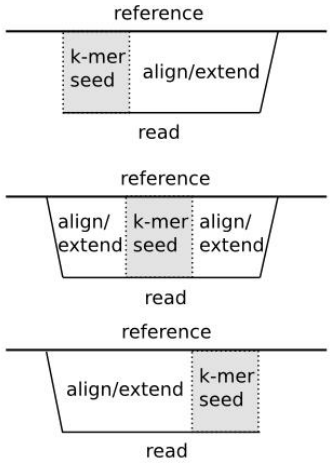
- BWT-FM based
  - Illumina: BWA, Bowtie, SOAP2
  - Human genome can be compressed into a 2.3 GB data structure through BWT
  - Extremely fast for perfect hits
  - Increased memory lookups for mismatch
    - Indels are found in postprocessing when paired-end reads are available
  - GPGPU implementations: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

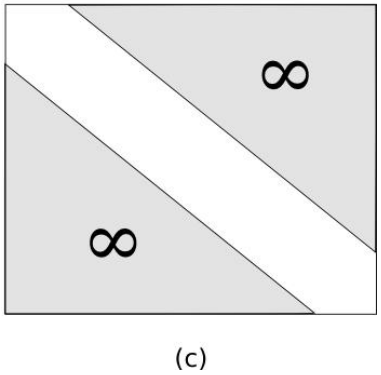
Long read mappers

- PacBio and ONT:
  - BLASR (suffix-tree based indexing)
  - MashMap and Minimap2 (minimizers + chaining + Smith-Waterman)
    - Paper presentation candidate
  - NGM-LR (hash table + chaining + alignment w/ convex gap penalty model
    - Paper presentation candidate

Hash Based Aligners







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:

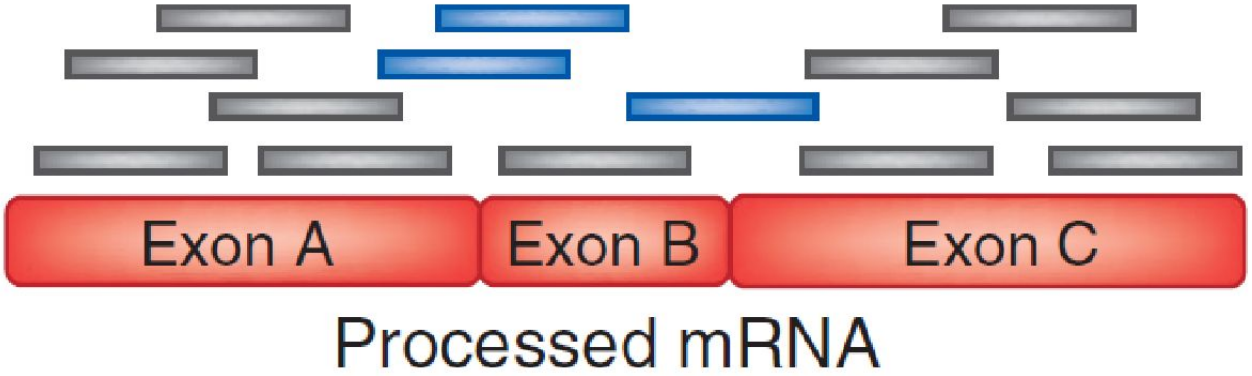
$$p_s(u|x,z) = \frac{p(z|x,u)}{\sum_{v=1}^{L-l+1} p(z|x,v)}$$

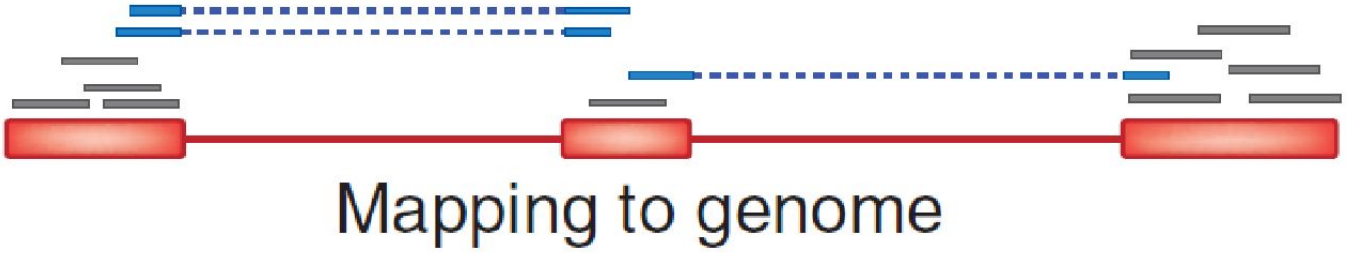
$$Q_s(u|x,z) = -10 \log_{10}[1 - p_s(u|x,z)].$$

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