DNA sequencing, FASTQ format, tools.

BFX Workshop Week 3

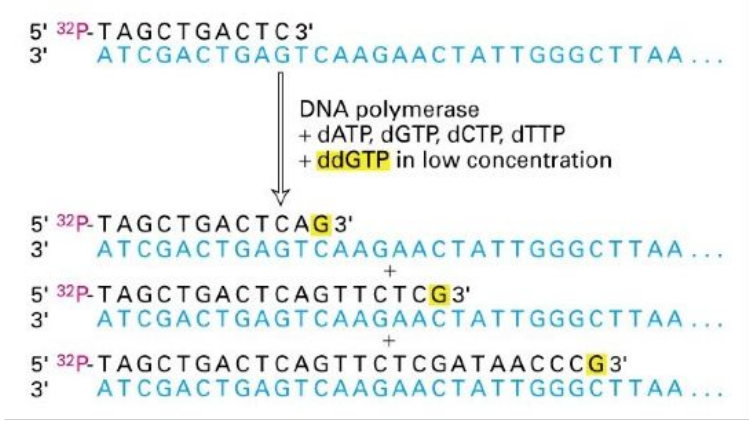
Chris Miller

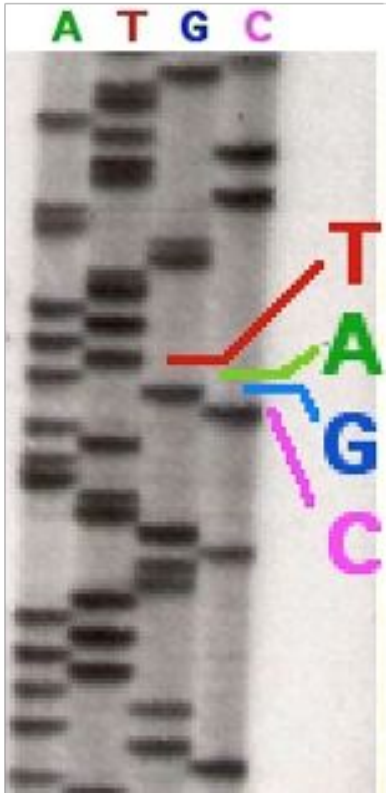
Many slides adapted from:
Applied Computational Genomics
https://github.com/quinlan-lab/applied-computational-genomics
Aaron Quinlan

How to sequence a human genome: Sanger method

Key points:

1) sequencing by synthesis (not degradation)
2) primers hybridize to DNA
3) polymerase + dNTPS + ddNTP terminators at low concentration
4) 1 lane per base, visually interpret ladder





Sanger sequencing: technological advances

1977: Fred Sanger → 1 hard working technician = 700 bases per day = 118,000 years to sequence the human genome

↓

1985: ABI 370 (first automated sequencer) → 5000 bases per day = 16,000 years

↓

1995: ABI 377 (Bigger gels, better chemistry & optics, more sensitive dyes, faster computers) → 19,000 bases per day = 4,400 years

↓

1999: ABI 3700 (96 capillaries, 96 well plates, fluid handling robots) → 400,000 bases per day = 205 years

Slide from Ira Hall

The next wave of DNA sequencing technologies

whiz-bang terms

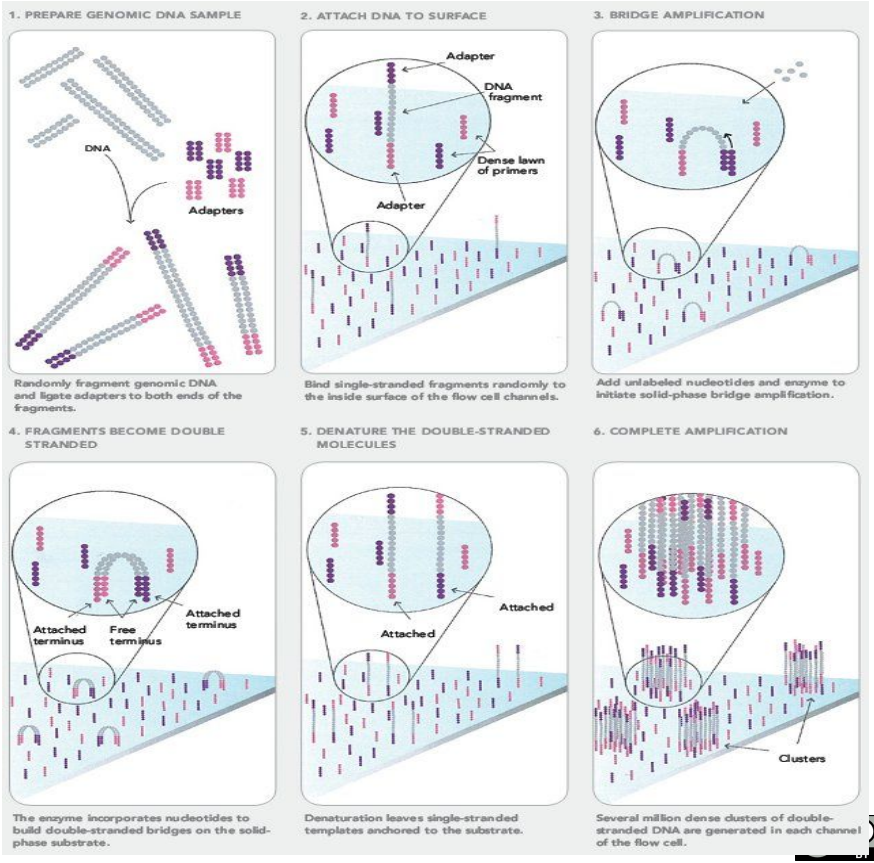
- “Massively parallel” sequencing
- “High-throughput” sequencing
- “Ultra high-throughput” sequencing
- “Next generation” sequencing (NGS)
- “Second generation” sequencing

- **2005: 454 (Roche)**
- **2006: Solexa (Illumina)**
- **2007: ABI/SOLiD (Life Technologies)**
- **2010: Complete Genomics**
- **2011: Pacific Biosciences**
- **2010: Ion Torrent (Life Technologies)**
- **2015: Oxford Nanopore Technologies**

Slide from Ira Hall

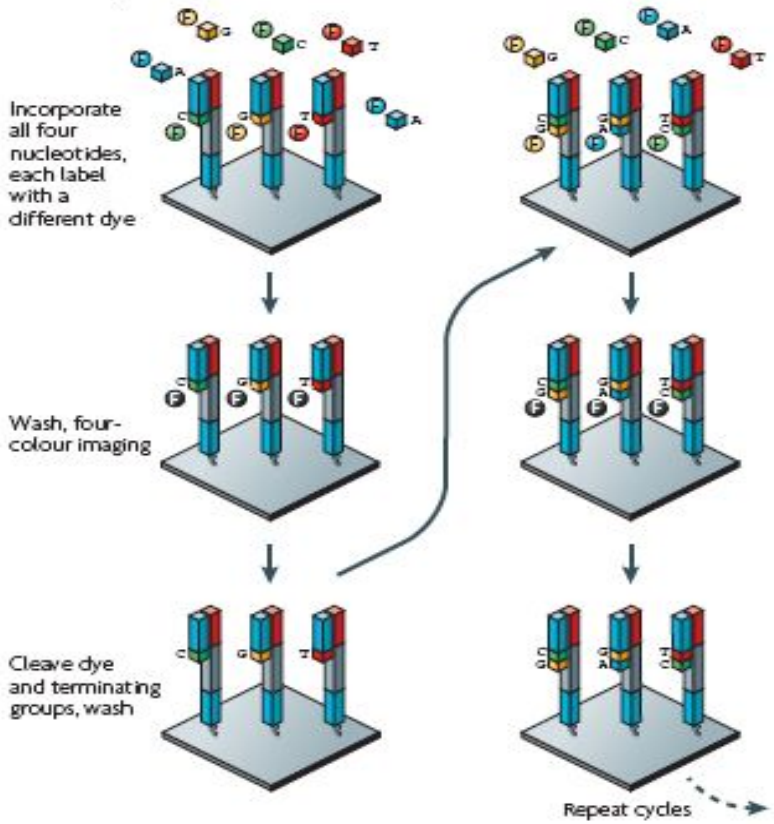
Solexa (Illumina) sequencing (2006)

- **PCR amplify sample (opt.)**
- **Immobilize and amplify single molecules on a solid surface**
- **<u>Reversible</u> terminator sequencing with 4 color dye-labelled nucleotides**

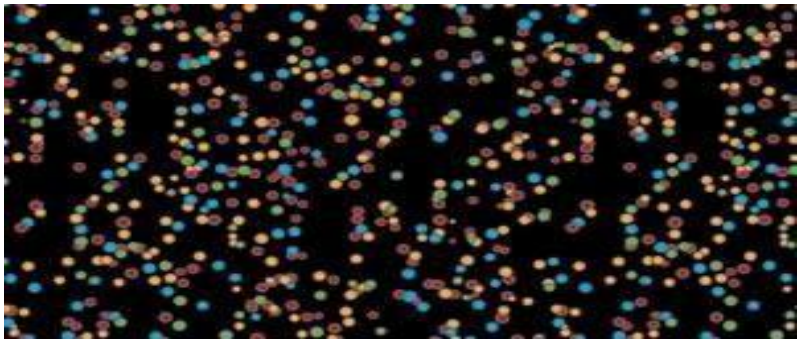


Slide from Ira Hall

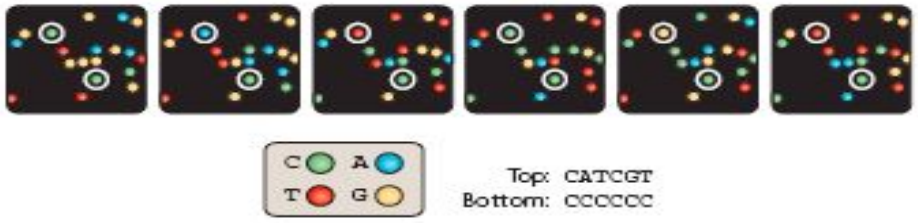
Cluster amplification by "bridge" PCR



4 different images merged



6 cycles w/ base-calling



Slide from Ira Hall

Illumina sequencing summary

Advantages:

- Best throughput, accuracy and read length for any 2nd gen. sequencer
- Fast & robust library preparation

Disadvantages:

- Inherent limits to read length (practically, 150bp)
- Some runs are error prone

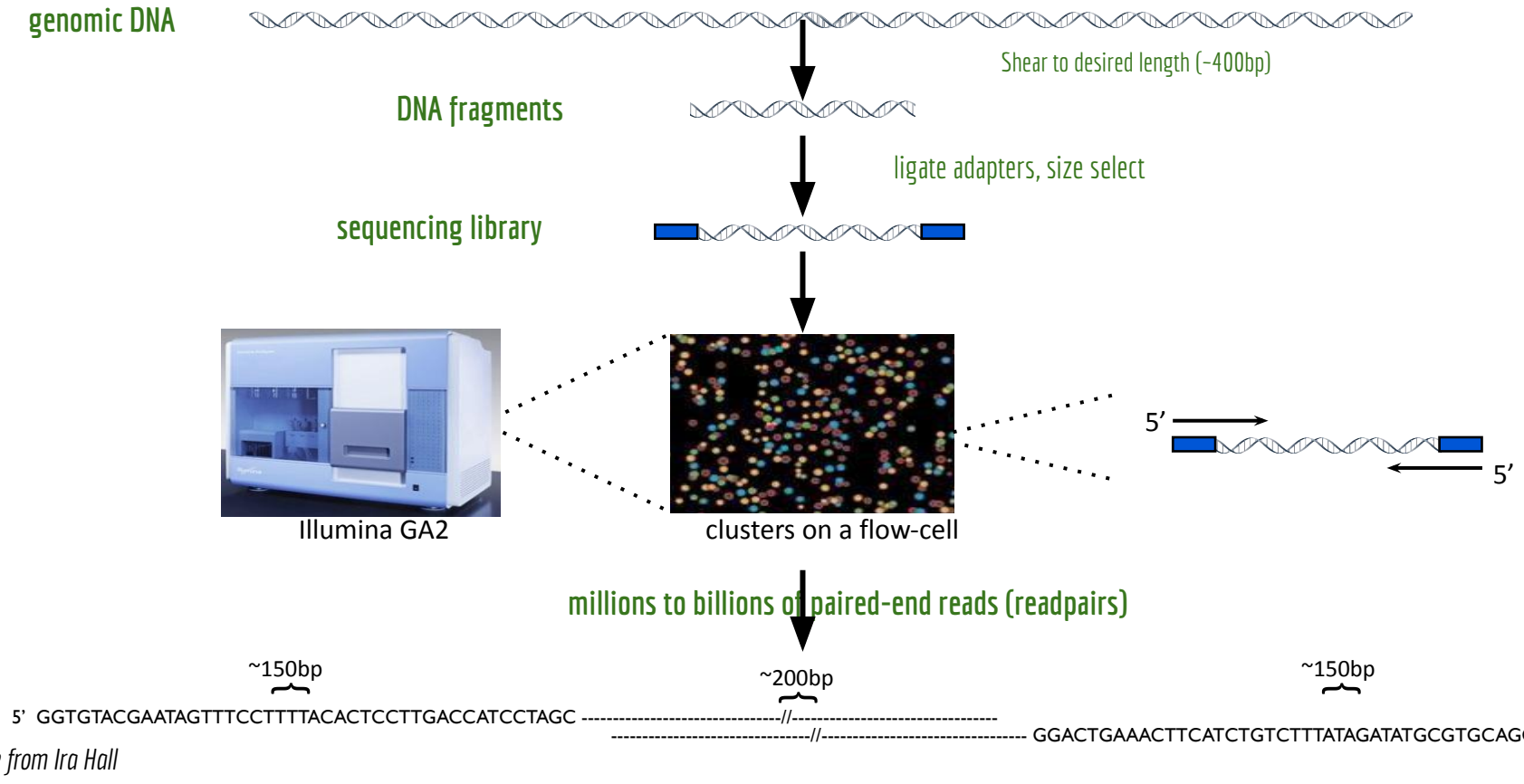
Illumina X-plus

8-16 Tb per run
64 human genomes per run
48 hours

~$400 per genome

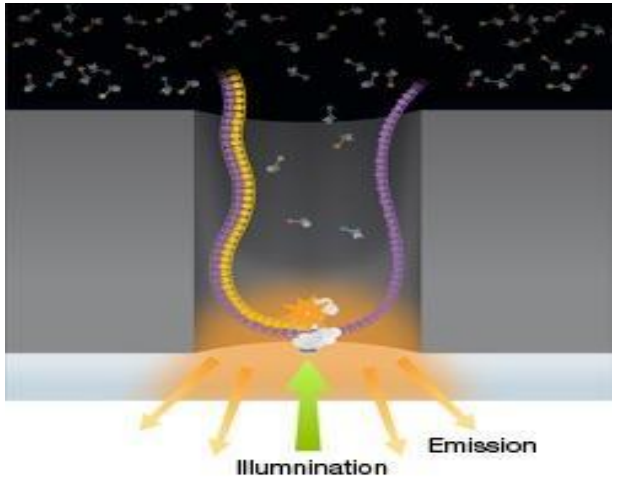
Slide from Ira Hall

Paired-end sequencing:
A molecular hack to sequence longer fragments



Slide from Ira Hall

Pacific Biosciences

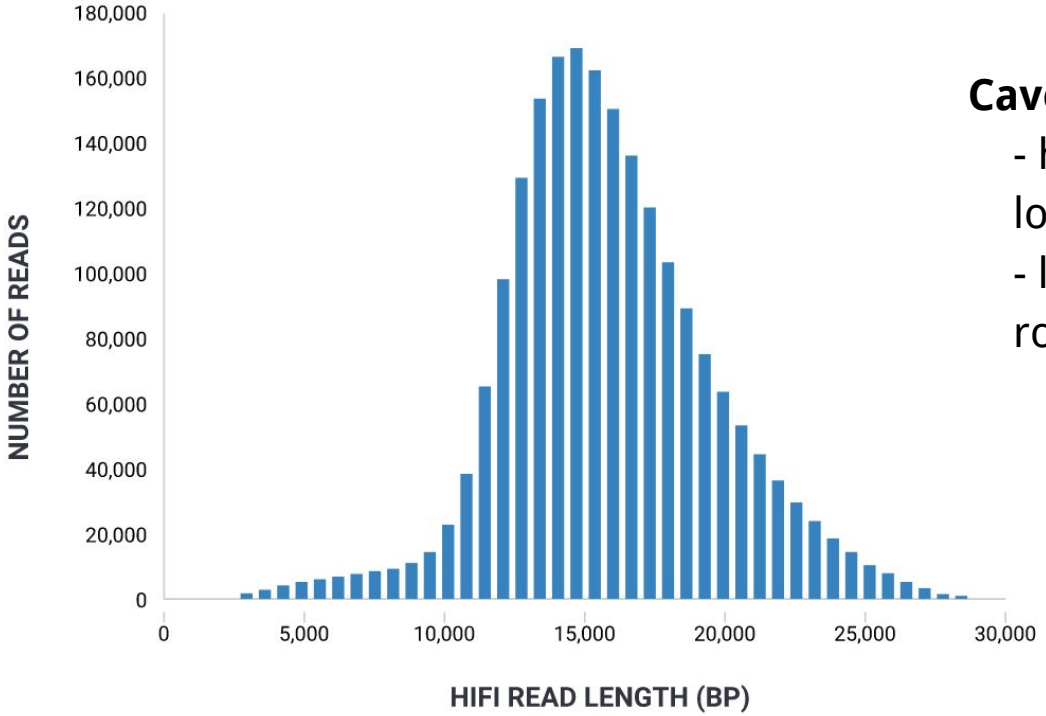


Key Points:

- 1 DNA molecule and 1 polymerase in each well (zero-mode waveguide)
- 4 colors flash in real time as polymerase acts
- Methylated cytosine has distinct pattern
- No *theoretical* limit to DNA fragment length

Slide from Ira Hall

Pacific Biosciences: long reads. Great for genome assembly

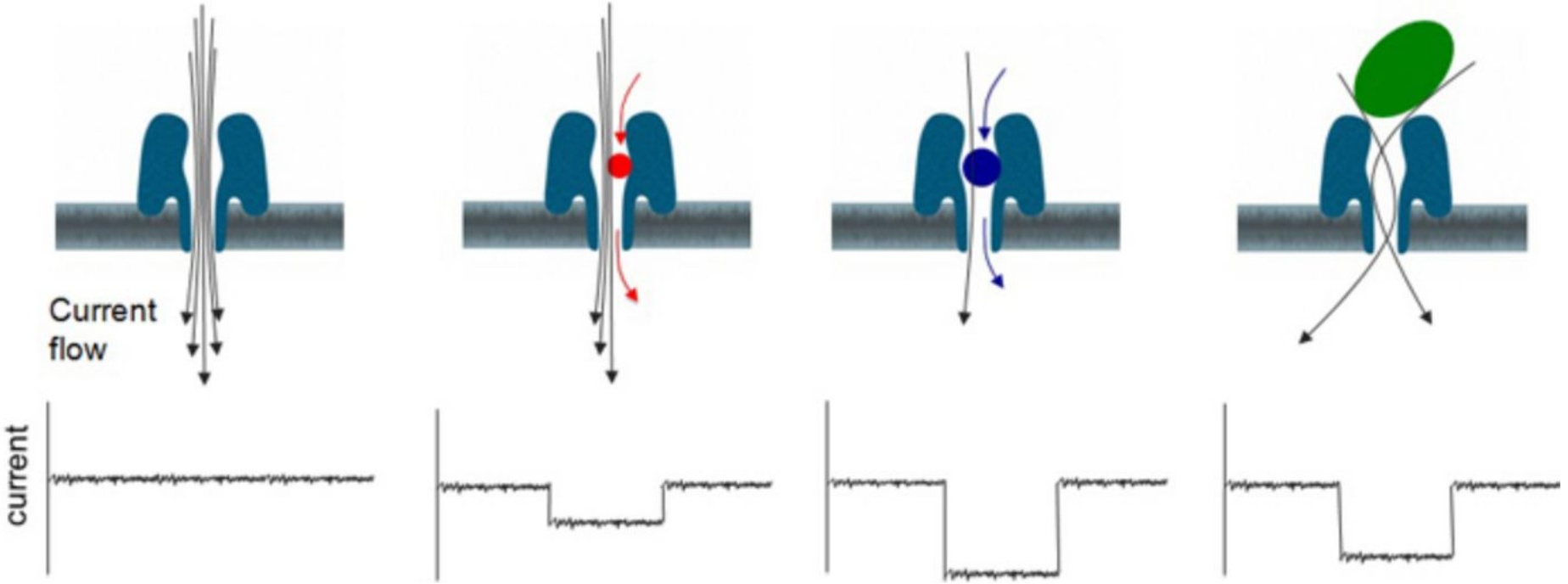


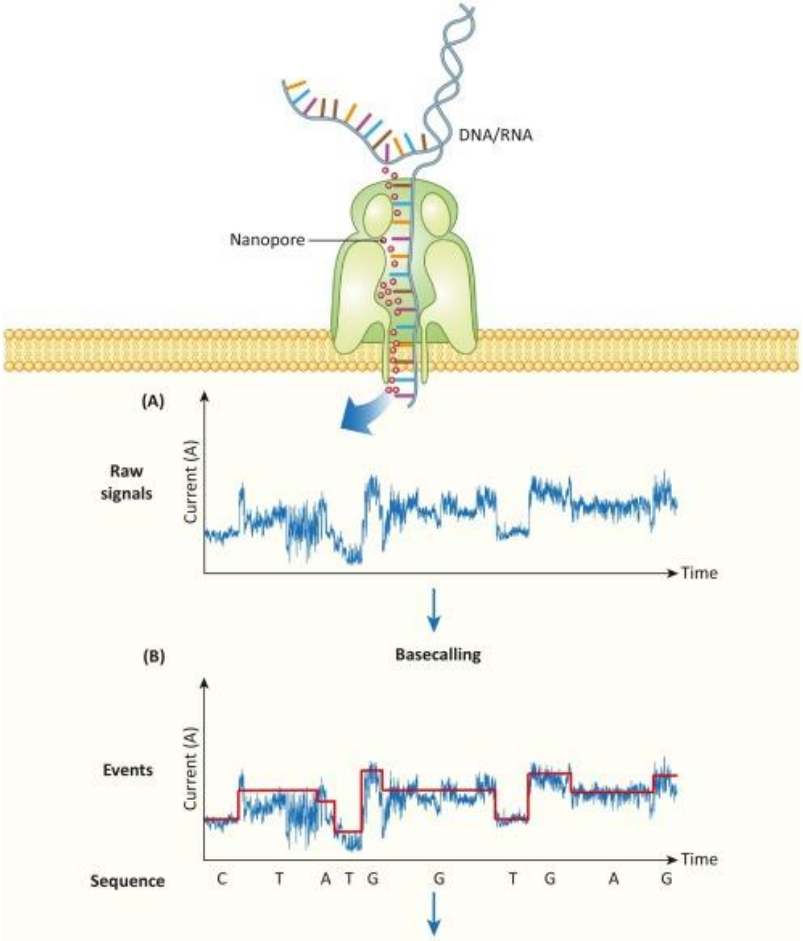
Caveats:

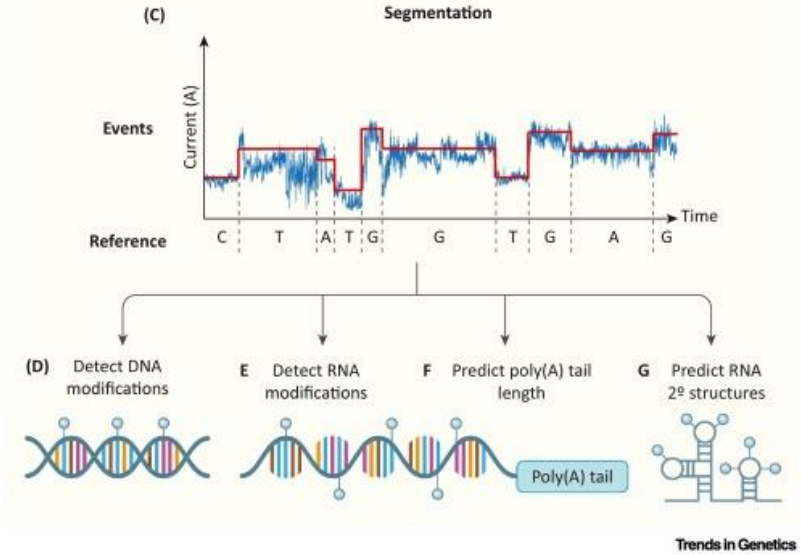
- higher error rate (1-2%), lower with Duplex runs
- lower throughput : roughly 90 gigabases per run

About $4,000 for a 30x human genome on the RevIO machine

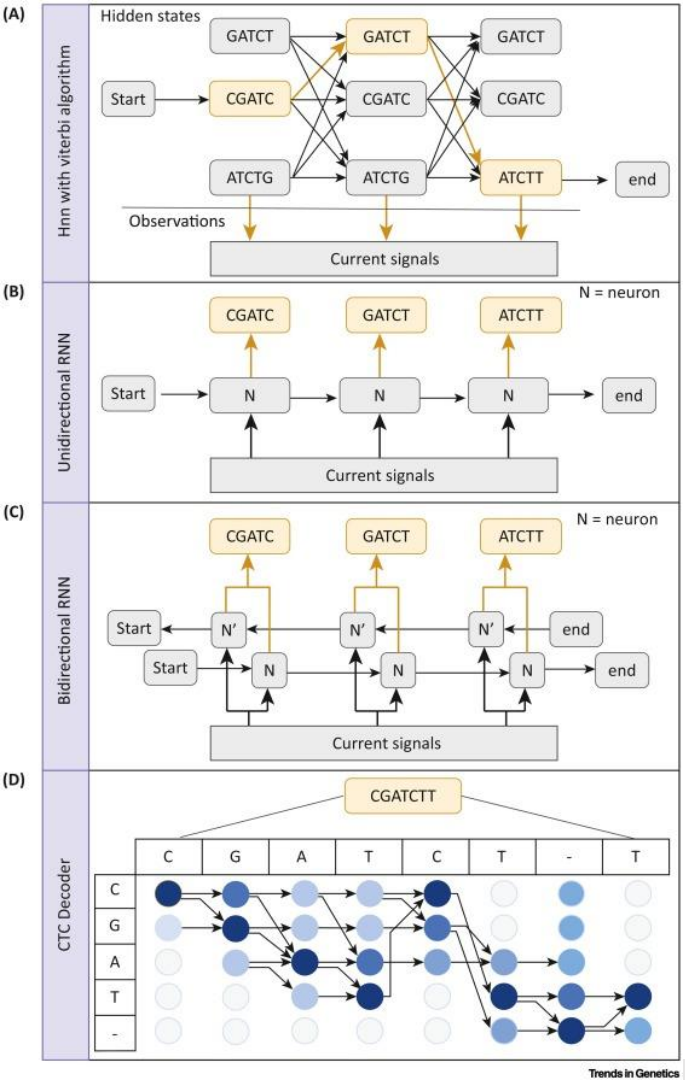
Oxford Nanopore Technologies







doi.org/10.1016/j.tig.2021.09.001



Neural networks to translate signal into base calls

- Guppy (many versions)

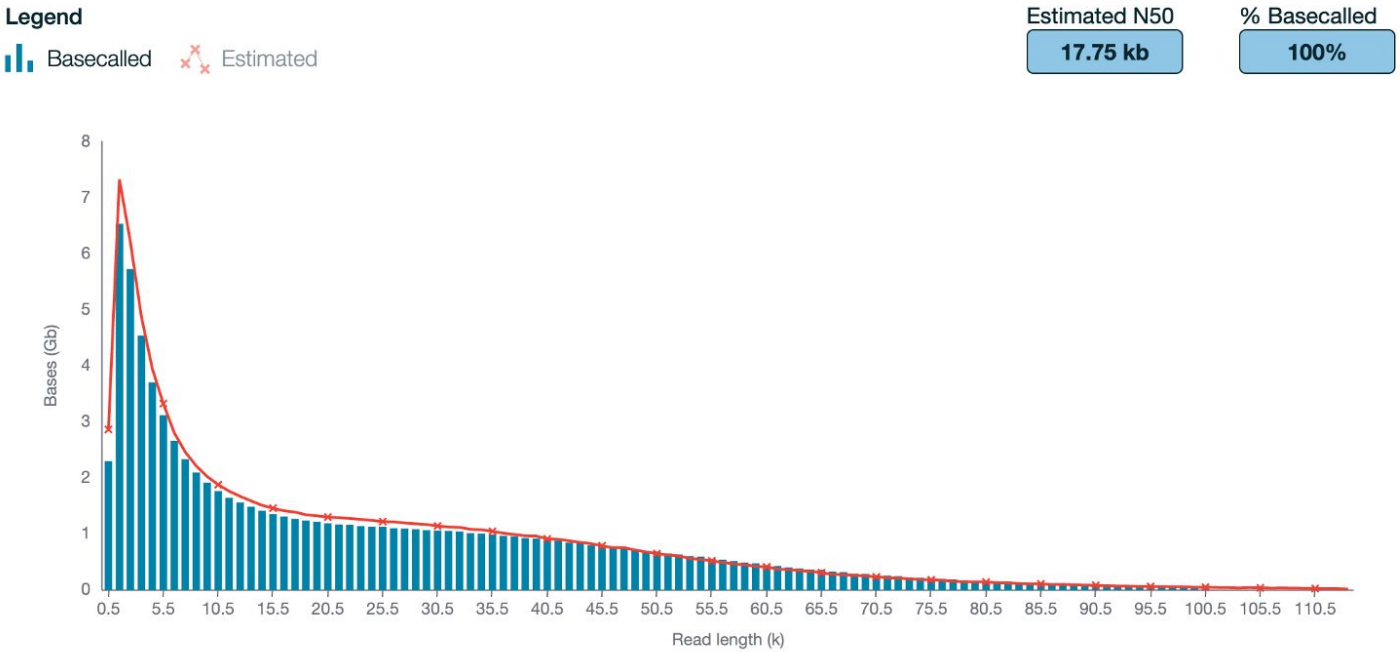
- Dorado (v0.4, eventual guppy replacement)

- many others

Practically, that means that we can't yet throw away our raw signal intensities. (1 Tb or more per run)

doi.org/10.1016/j.tig.2021.09.001

ONT sequence length distribution

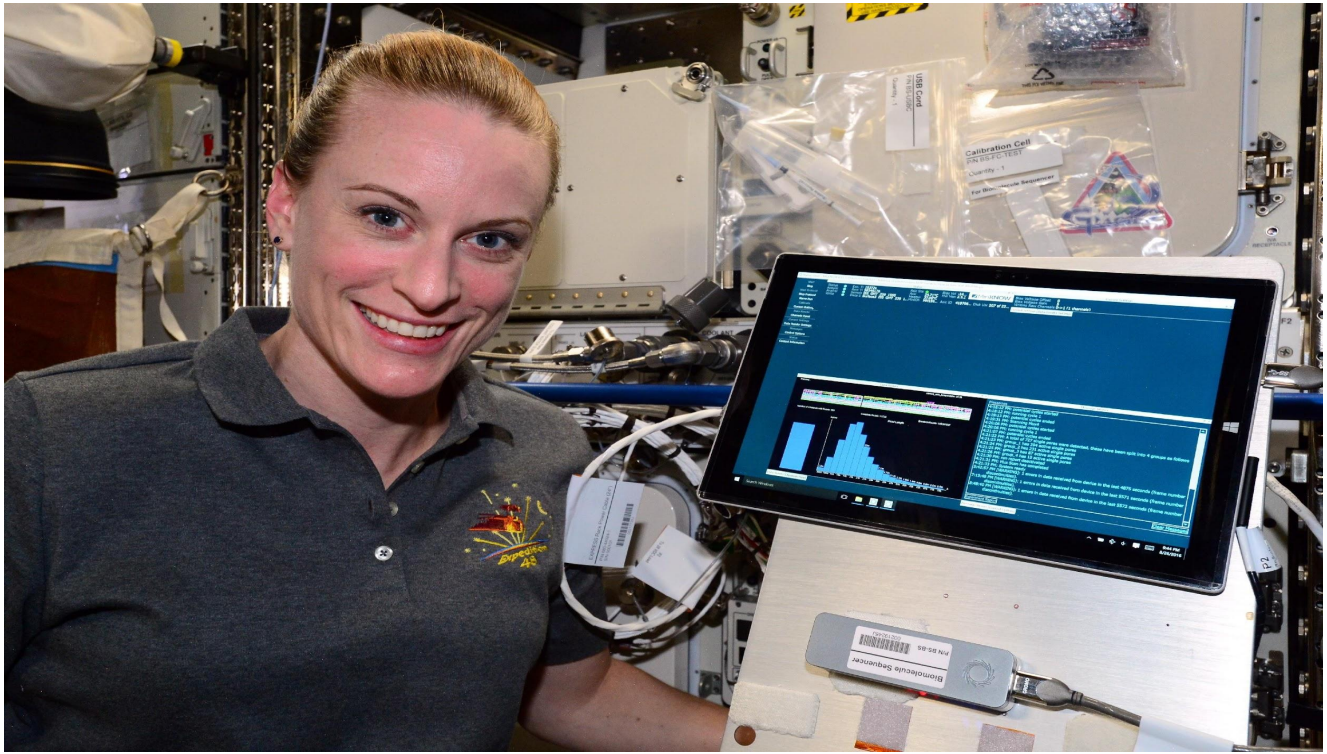


their relative amounts.

Read length (kb)	Aggregated reads (Mb)
100 - 164	886.98
164 - 228	36.06
228 - 292	4.02
292 - 344	0.35

Recent run of a tumor sample

Nanopore sequencing is *extremely* portable



Kate Rubins sequencing DNA on the ISS

Long reads. Great for genome assembly

Single haplotype assembly of the human genome from a hydatidiform mole

Karyn Meltz Steinberg,[1] Valerie A. Schneider,[2] Tina A. Graves-Lindsay,[1] Robert S. Fulton,[1] Richa Agarwala,[2] John Huddleston,[3,4] Sergey A. Shiryev,[2] Aleksandr Morgulis,[2] Urvashi Surti,[5] Wesley C. Warren,[1] Deanna M. Church,[6] Evan E. Eichler,[3,4] and Richard K. Wilson[1]

[1]*The Genome Institute at Washington University, St. Louis, Missouri 63108, USA; [2]National Center for Biotechnology Information, National Library of Medicine, National Institutes [illegible] University of Washington, Seattle, Washington 98 Washington 98195, USA; [5]Department of Patholo USA; [6]Personalis, Inc., Menlo Park, California 940*

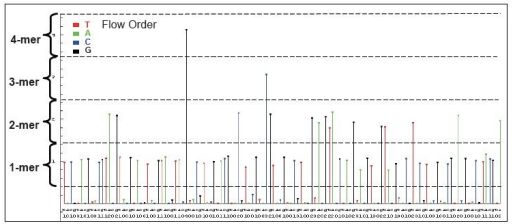
The complete sequence of a human genome

SERGEY NURK, SERGEY KOREN, ARANG RHIE, MIKKO RAUTIAINEN, ANDREY V. BZIKADZE, ALLA MIKHEENKO, MITCHELL R. VOLLGER, NICOLAS ALTEMOSE, LEV URALSKY, [...], AND ADAM M. PHILLIPPY +90 authors Authors Info & Affiliations

SCIENCE • 31 Mar 2022 • Vol 376, Issue 6588 • pp. 44-53 • DOI: 10.1126/science.abj6987

T2T consortium made heavy use of PacBio and ONT long reads

Base calling: the conversion of signal to a nucleotide sequence



Raw signal
(e.g., 454 Life Sciences)

Base calling algorithms

Errors happen.
Hopefully infrequently

ACCTTCGAACGGCGGGGGGTTACAA

(Mostly) all technologies yield DNA sequences in FASTQ format

DNA

```
@seq1
ACCTTCGAACGGCGGGGGGTTACAA
+
!''*((((***+))%%%++).1***
@seq2
TGGAACCGAACGGCCCCGGTTACAT
+
!''*!!!!***+))+++++).1***
And so on...
```

The FASTQ format. Welcome to a minor hell.

A “standard” format for storing and defining sequences from next-generation sequencing technologies.

```
   Sequence ID @SEQ_ID
      Sequence GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT
   <separator> +
Quality scores !''*(((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65
```

http://en.wikipedia.org/wiki/FASTQ_format

The FASTQ format's sequence identifier (first line of each record)

Old format

```
@HWUSI-EAS100R:6:73:941:1973#0/1
```

HWUSI-EAS100R	the unique instrument name
6	flowcell lane
73	tile number within the flowcell lane
941	'x'-coordinate of the cluster within the tile
1973	'y'-coordinate of the cluster within the tile
#0	index number for a multiplexed sample (0 for no indexing)
/1	the member of a pair, /1 or /2 *(paired-end or mate-pair reads only)*

New format

```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
```

EAS139	the unique instrument name
136	the run id
FC706VJ	the flowcell id
2	flowcell lane
2104	tile number within the flowcell lane
15343	'x'-coordinate of the cluster within the tile
197393	'y'-coordinate of the cluster within the tile
1	the member of a pair, 1 or 2 *(paired-end or mate-pair reads only)*
Y	Y if the read is filtered, N otherwise
18	0 when none of the control bits are on, otherwise it is an even number
ATCACG	index sequence

FASTQ quality scores: estimate of confidence in each base (sequencing technologies make errors!)

Sequence ID	`@SEQ_ID`
Sequence	`GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT`
‹separator›	`+`
Quality scores	`!''*(((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65`

FASTQ quality scores: estimate of confidence in each base (sequencing technologies make errors!)

Sequence ID	`@SEQ_ID`
Sequence	`GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT`
‹separator›	`+`
Quality scores	`!''*(((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65`

Qualities are based on the Phred scale and are *encoded*

$$Q = -10 * \log_{10}(P_{err})$$

Phred quality score calculation

$$Q = -10 * \log_{10}(P_{err})$$

Error probability (P_{err})	$\log_{10}(P_{err})$	Phred quality score
1	0	0
0.1	-1	10
0.01	-2	20
0.001	-3	30
0.0001	-4	40

FASTQ quality scores: estimate of confidence in each base (sequencing technologies make errors!)

Sequence ID	`@SEQ_ID`
Sequence	`GATTTGGGGTTCAAAGCAGTATCGATCAAATAGTAAATCCATTTGTTCAACTCACAGTTT`
‹separator›	`+`
Quality scores	`!''*(((***+))%%%++)(%%%%).1***-+*''))**55CCF>>>>>>CCCCCCC65`

Quality score encoding based on ASCII table

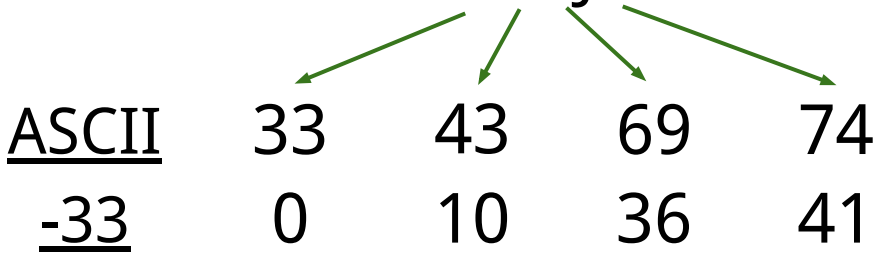
Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Formula for getting PHRED quality from encoded quality:

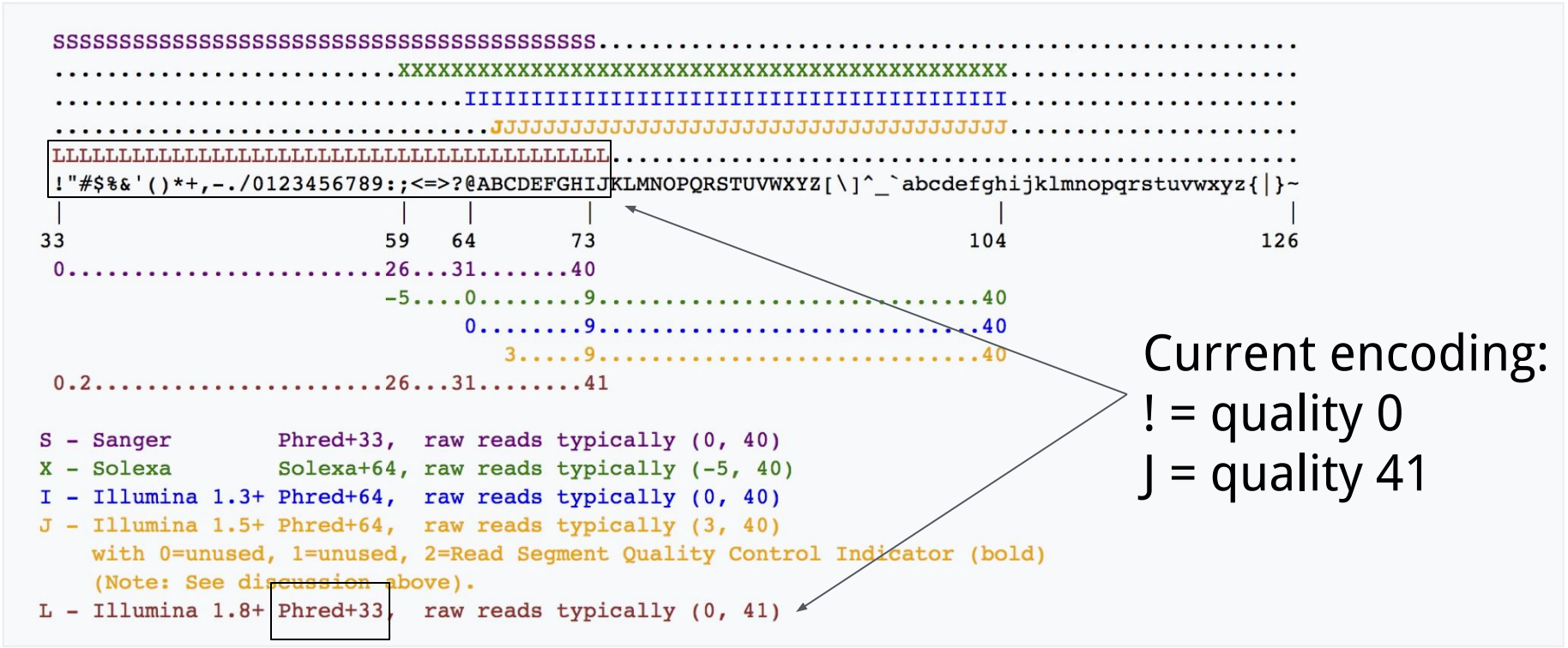
Q = ascii(char) - 33

Example:

!+EJ



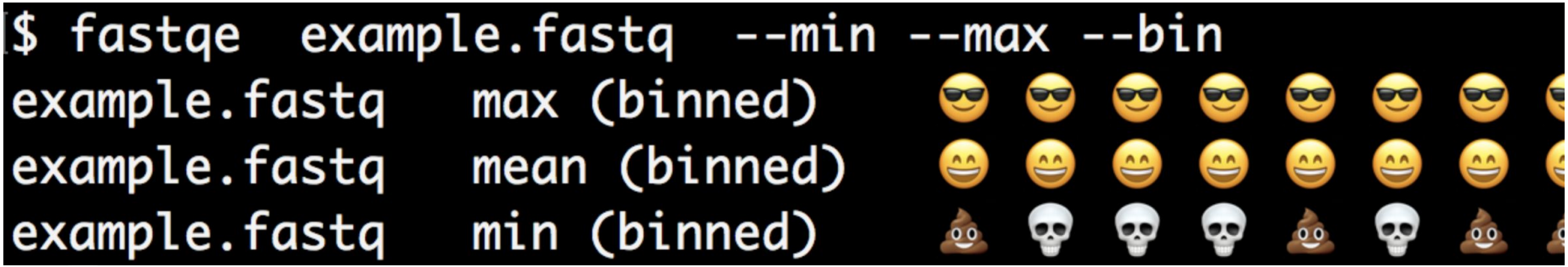
Historically, FASTQ has had different encoding schemes for encoding PHRED quality scores. Ouch.



Current encoding:
! = quality 0
J = quality 41

FASTQE

https://github.com/fastqe/fastqe



Quality score binning

Table 1: Q-Score Bins for an Optimized 8-Level Mapping

Quality Score Bins	Example of Empirically Mapped Quality Scores*
N (no call)	N (no call)
2–9	6
10–19	15
20–24	22
25–29	27
30–34	33
35–39	37
≥ 40	40

https://www.illumina.com/documents/products/whitepapers/whitepaper_datacompression.pdf

FASTQ vs FASTA

```
>seq1
ACGACGAGACCTTCATCAAAAACATCATCATCCAGGACTGTATGTGGAGCGGCTTCTCGG
CCGCCGCCAAGCTCGTCTCAGAGAAGCTGGCCTCCTACCAGGCTGCGCGCAAAGACAGCG
GCAGCCCGAACCCCGCCCGCGGCCACAGCGTCTGCTCCACCTCCAGCTTGTACCTGCAGG
ATCTGAGCGCCGCCGCCTCAGAGTGCATCGACCCCTCGGTGGTCTTCCCCTACCCTCTCA
ACGACAGCAGCTCGCCCAAGTCCTGCGCCTCGCAAGACTCCAGCGCCTTCTCTCCGTCCT
CGGATTCTCTGCTCTCCTCGACGGAGTCCTCCCCGCAGGGCAGCCCCGAGC
>seq2
TCCATGAGGAGACACCGCCCACCACCAGCAGCGACTCTGGTAAGCGAAGCCCGCCCAGGC
CTGTCAAAAGTGGGCGGCTGGATACCTTTCCCATTTTCATTGGCAGCTTATTTAACGGGC
CACTCTTATTAGGAAGGAGAGATAGCAGATCTGGAGAGATTTGGGAGCTCATCACCTCTG
AAACCTTGGGCTTTAGCGTTTCCTCCCATCCCTTCCCCTTAGACTGCCCATGTTTGCAGC
CCCCCTCCCCGTTTGTCTCCCACCCCTCAGGAATTTCATTTAGGTTTTTAAACCTTCTGG
CTTATCTTACAACTCAATCCACTTCTTCTTACCTCCCGTTAACATTTTAATTGCCCTGGG
GCGGGGTGGCAGGGAGTGTATGAATGAGGATAAGAGAGGATTGATCTCTGAGAGTGAATG
AATTGCTTCCCTCTTAACTTCCGAGAAGTGGTGGGATTTAATGAACTATCTACAAAAATG
```

FASTQC: Is my sequence data any good?

http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

Using docker on your laptop

0) Make sure docker is installed

1) Pull the docker image you would like to use

```
docker pull chrisamiller/genomic-analysis:0.2
```

2) Run the docker container interactively

```
docker run -it chrisamiller/genomic-analysis:0.2 /bin/bash
```

3) Run a container while mounting the current directory as /data

```
docker run -v $(pwd -P):/data -it chrisamiller/genomic-analysis:0.2 /bin/bash
```

Some useful UNIX commands

- **`head`**print the first 10 lines of a file

- **`tail`** print the last 10 lines of a file
  getting fancy: **`tail -n +2`**

- **`wc`** count the number of characters/words/lines in a file
  `wc -l` for only lines

- **`less`**because you don't want 3 million lines scrolling through your terminal
  `q` to exit, **`-S`** to wrap lines (lots more useful options here)

- **`grep`**to search through a file (**`-v`** to search for lines *without* pattern)

| (pipes)

You cannot be a productive command line user until you really understand the power of pipes

```
grep TP53 genes.txt | grep “exon” | wc -l
```

This kind of construction allows you to get answers quickly!

Working with compressed data

- **`tar`** work with a “bundle” of data

  create: **tar -cvf output.tar infile1 infile2**
  extract: **tar -xvf output.tar**

- **`gzip`** compress a single file

  create: **gzip mydata.txt** (creates mydata.txt.gz)
  extract: **gunzip mydata.txt.gz** (creates mydata.txt)

Often these operations are combined

```
tar -czvf myfile.tar.gz <list of files>
tar -xzvf myfile.tar.gz
```

sed and awk

sed is most commonly used for find and replace operations:

```
cat file.txt | sed 's/foo/bar/g' >file_fixed.txt
```

Awk can be used to reorder particular columns (here, third, first, then second):

```
awk '{print $3,$1,$2}' file.txt >file2.txt
```

Or to print only certain lines of a file - here, every third line, starting at line 0

```
awk 'NR % 3 == 0' file > file2.txt
```

(both are very powerful, if somewhat opaque tools, this is just scratching the surface!)

Homework