
DTU

Sequencing technologies, quality, trimming and *de novo* assembly of WGS data

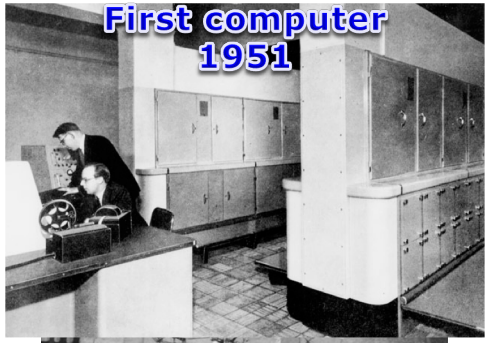
Rolf Sommer Kaas
Philip Thomas Lanken Conradsen Clausen

Overview

1. Quick history lesson
2. Sequencing technologies
3. Next Generation Sequencing (NGS) data
4. Trimming NGS data
5. De novo assembly
6. Quality Control

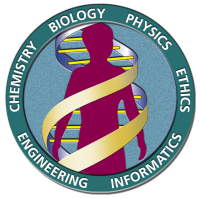
Quick History Lesson

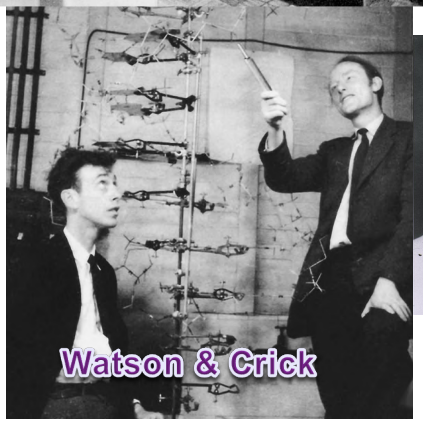

World Wide Web

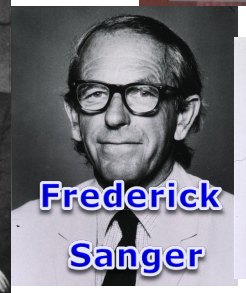

First computer
1951

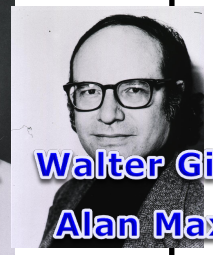

1980


First Portable
computer
IBM 5100


BIOLOGY
PHYSICS
ETHICS
INFORMATICS
ENGINEERING
CHEMISTRY

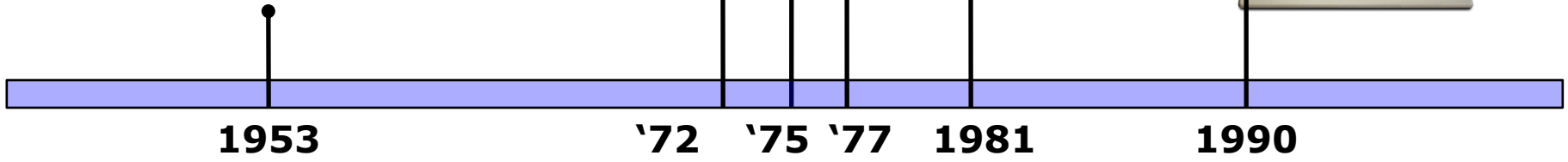

Watson & Crick


Frederick
Sanger


Walter Gilbe
Alan Maxam


First Laptop
Osborne 1 (11kg)


Amiga 500


1953
'72
'75
'77
1981
1990

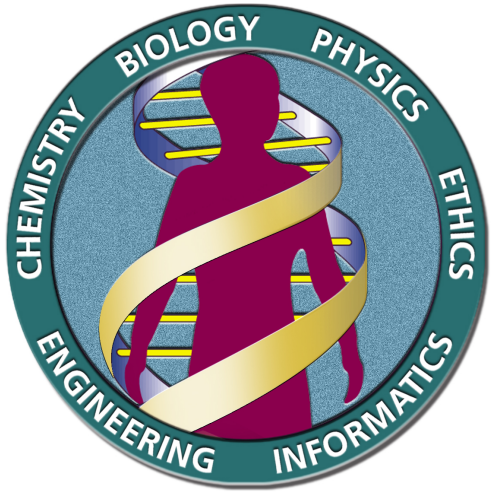
History

1990-2003

Human genome project

1998

CELERA



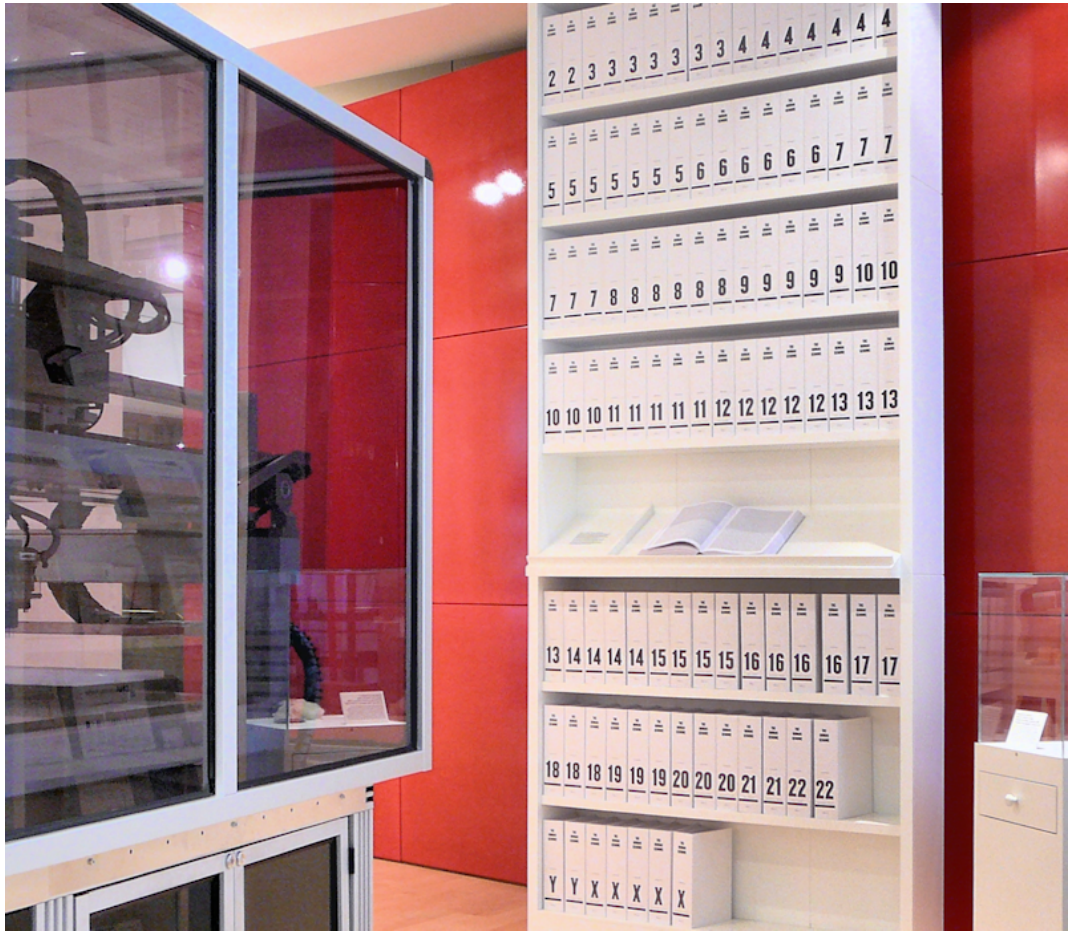
- Random Shotgun Sequencing
- Fast
- 300 mill. $

- Hierarchical Shotgun Sequencing
- 3 billion $

History

1990-2003

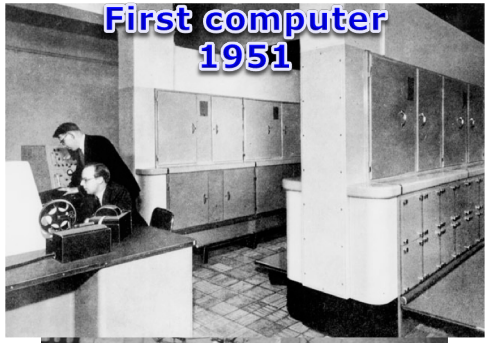
Human genome project

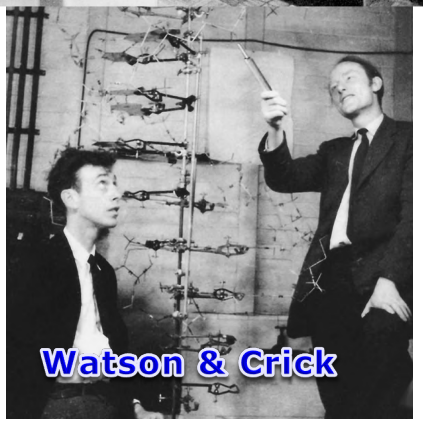


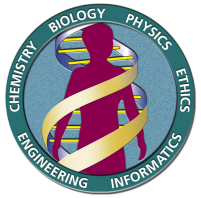
2001: Draft

2003: Complete

History







1953 '72 '75 '77 1981 1990 2003

History

2004

Next Generation Sequencing

454 Life Sciences: Parallelized pyrosequencing

Reduce costs 6 fold

Sequencing Technologies

Sequence Platforms

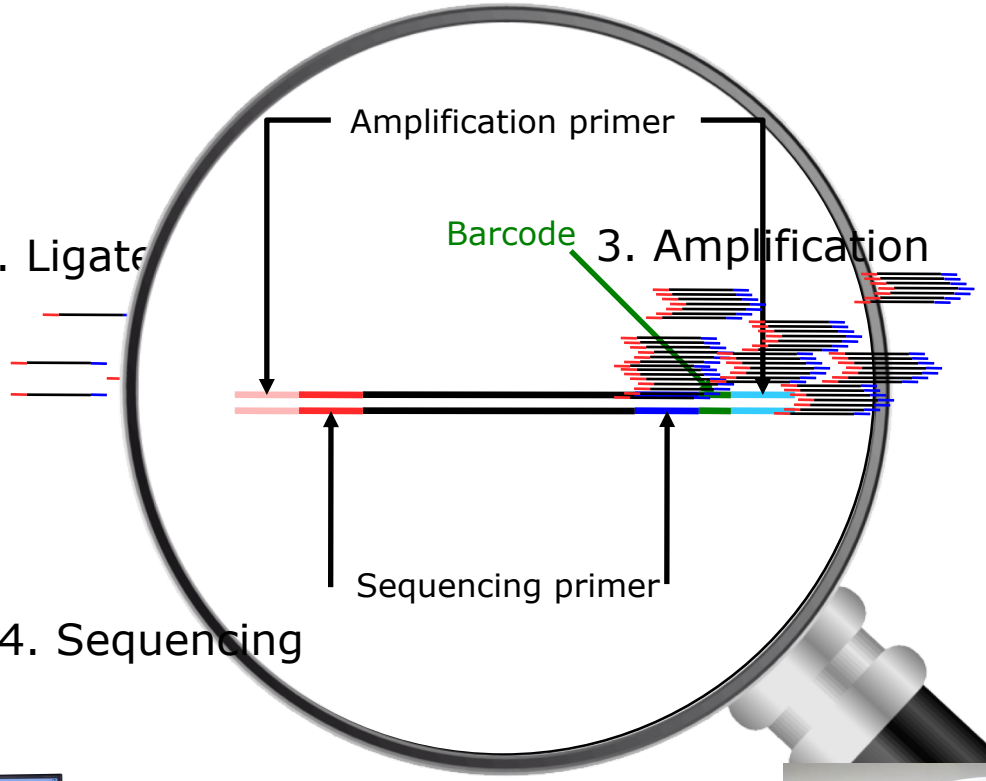
- Roche, 454 Life Sciences (GS FLX Titanium)
- Life Technologies (Ion Torrent & Ion Proton)
- Illumina (NovaSeq, HiSeq, NextSeq, MiSeq MiniSeq, GenomeAnalyzer)
- Pacific Biosciences (PacBio RS, PacBio Sequel)
- Oxford Nanopore (MinION, PromethION, GridION)

Sequencing

How it works

1. Fragment DNA

2. Ligate

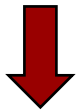


4. Sequencing

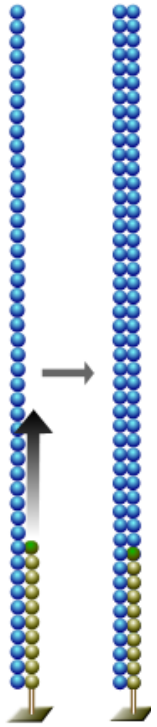
Template Generation & Amplification

Short read sequencing technologies

Fragment DNA



Ligate adapters

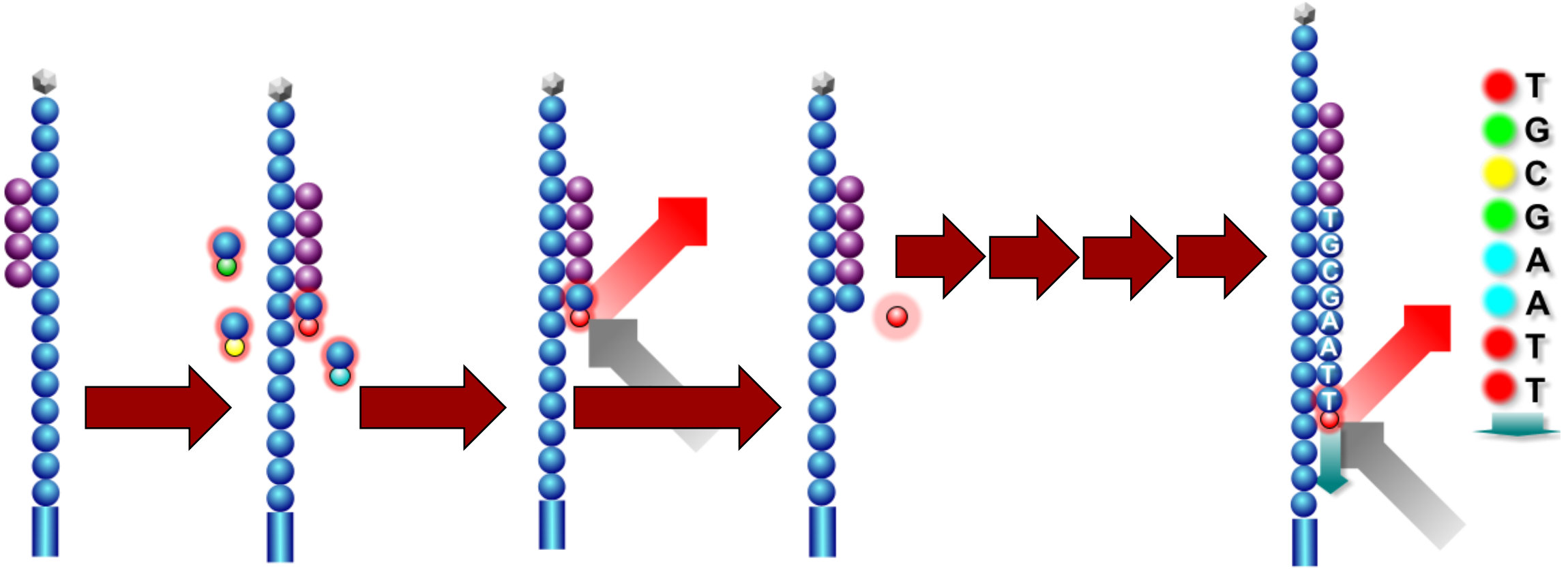


Reference: en.wikipedia.org

https://en.wikipedia.org/wiki/File:DNA_Sequencing_Bridge_Amplification.png

Sequencing by Synthesis - Illumina

Short read sequencing technologies



Reference: https://en.wikipedia.org/wiki/File:Sequencing_by_synthesis_Reversible_terminators.png (Abizar Lakdawalla)

License: https://creativecommons.org/licenses/by-sa/3.0/deed.en

Image has been altered from the original

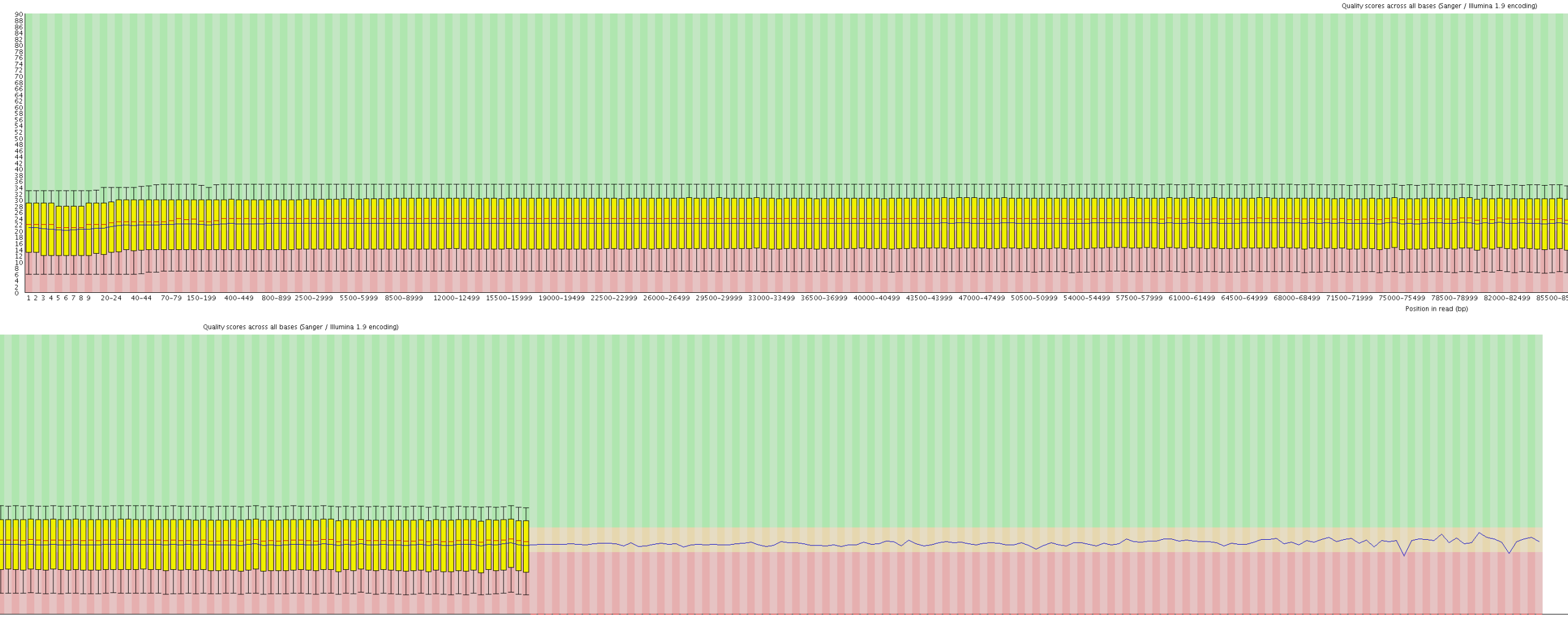
Illumina

- Good accuracy
- Error rate ~0.1%
- Some underrepresentation in AT and GC rich regions
- High throughput
- Small DNA fragments

Nanopore (ONT)

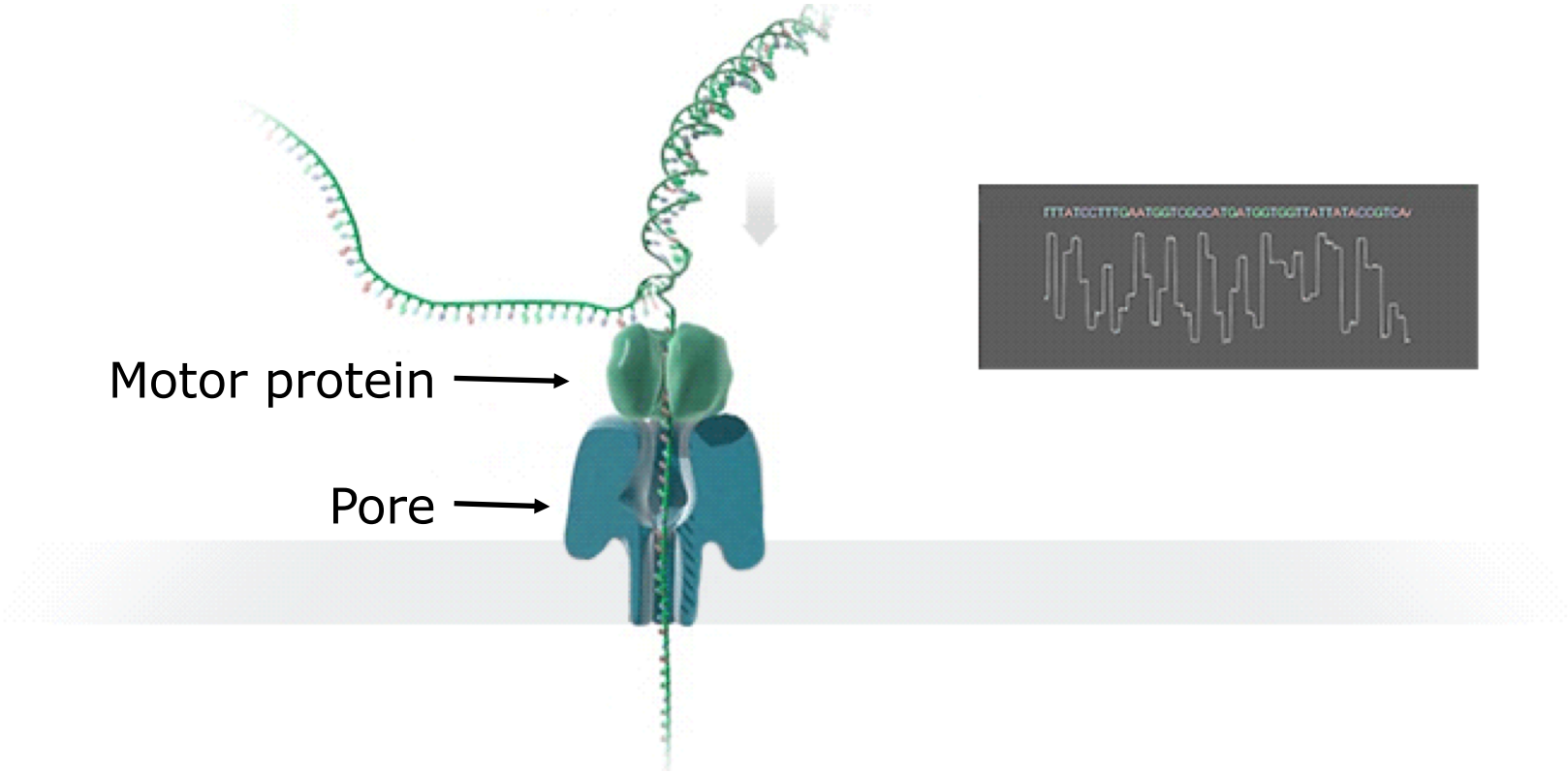
- Short turnaround (<24h).
- Fieldable.
- Long reads (>10 000 bp).
- Uniform error profile.

FastQC Report (16 threads)



Oxford Nanopore

Specific Errors

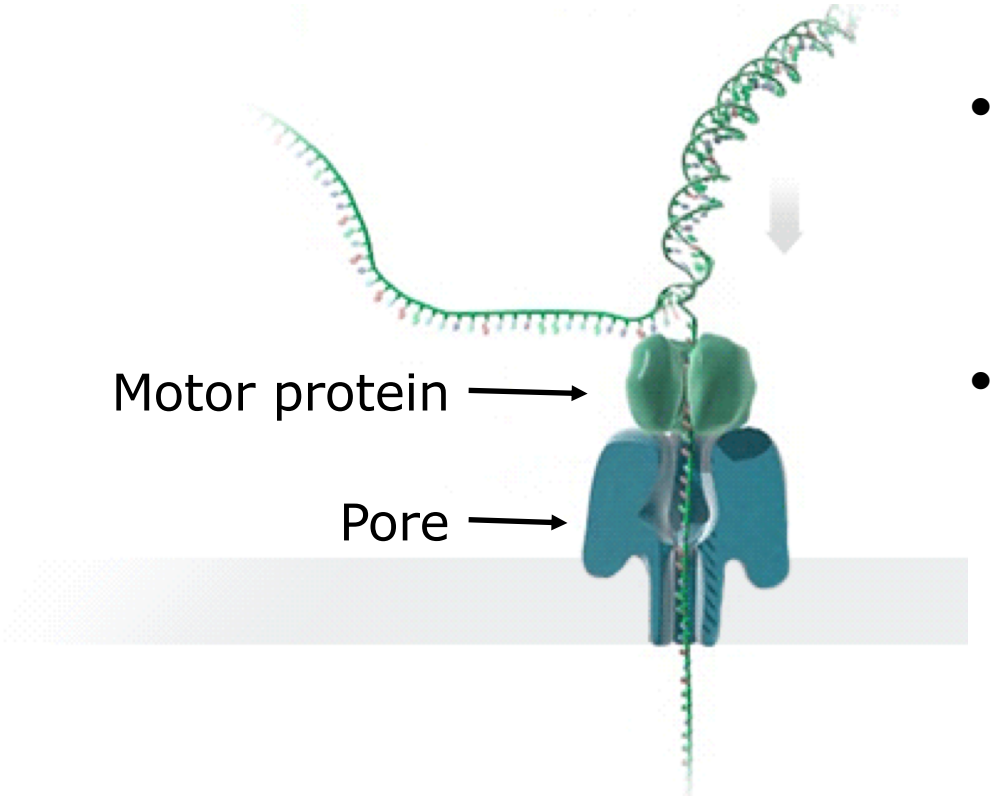


Reference: https://nanoporetech.com/sites/default/files/s3/galeries/2017-06/sequencing-animated.gif

License: Image used with permission from the manufacture (Oxford Nanopore Technologies)

Oxford Nanopore

Specific Errors

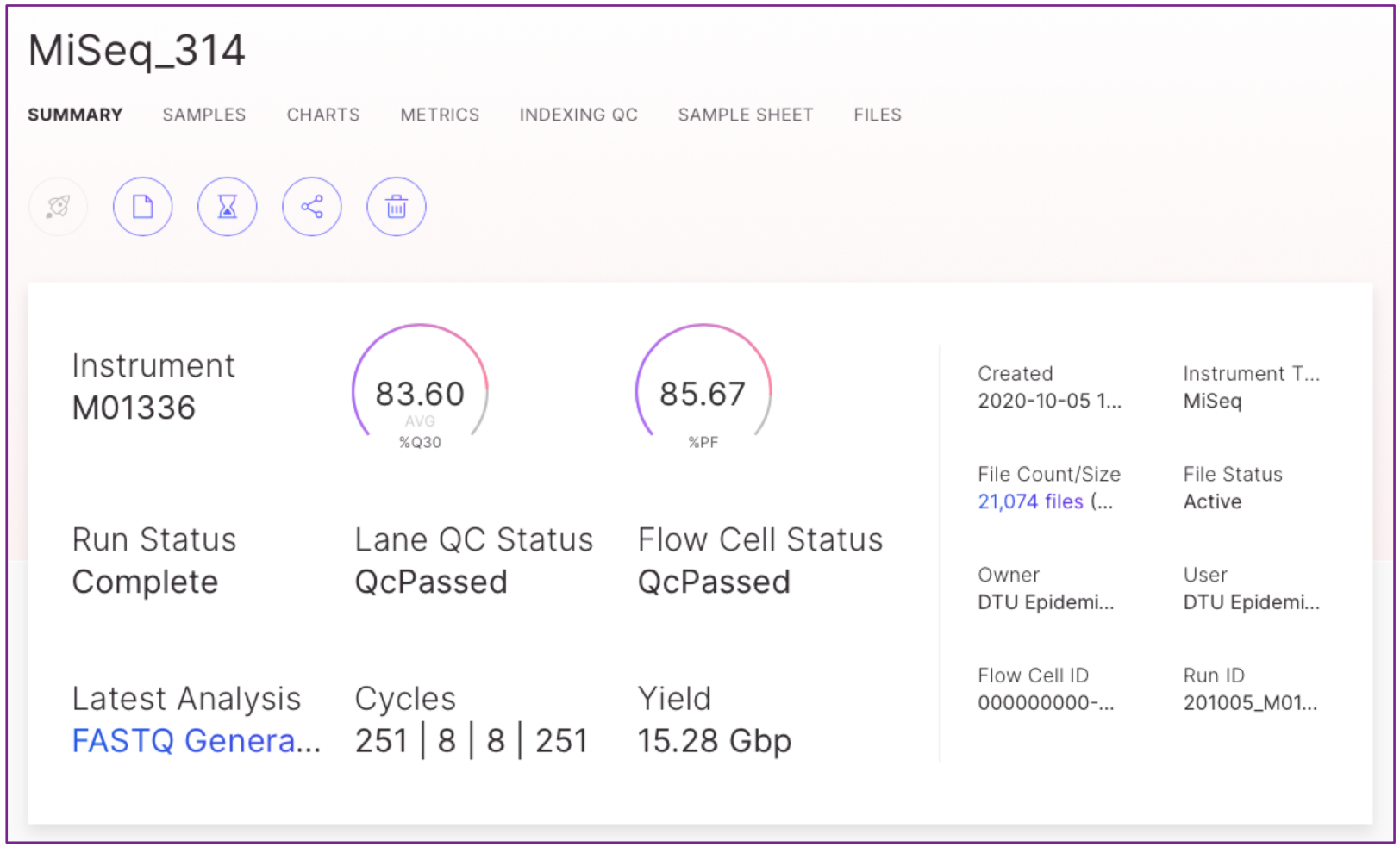


- Homopolymer errors
  - Revealed with lower quality
- Methylation
  - Altered size of nucleotides
  - Different current through pore

Reference: https://nanoporetech.com/sites/default/files/s3/galeries/2017-06/sequencing-animated.gif

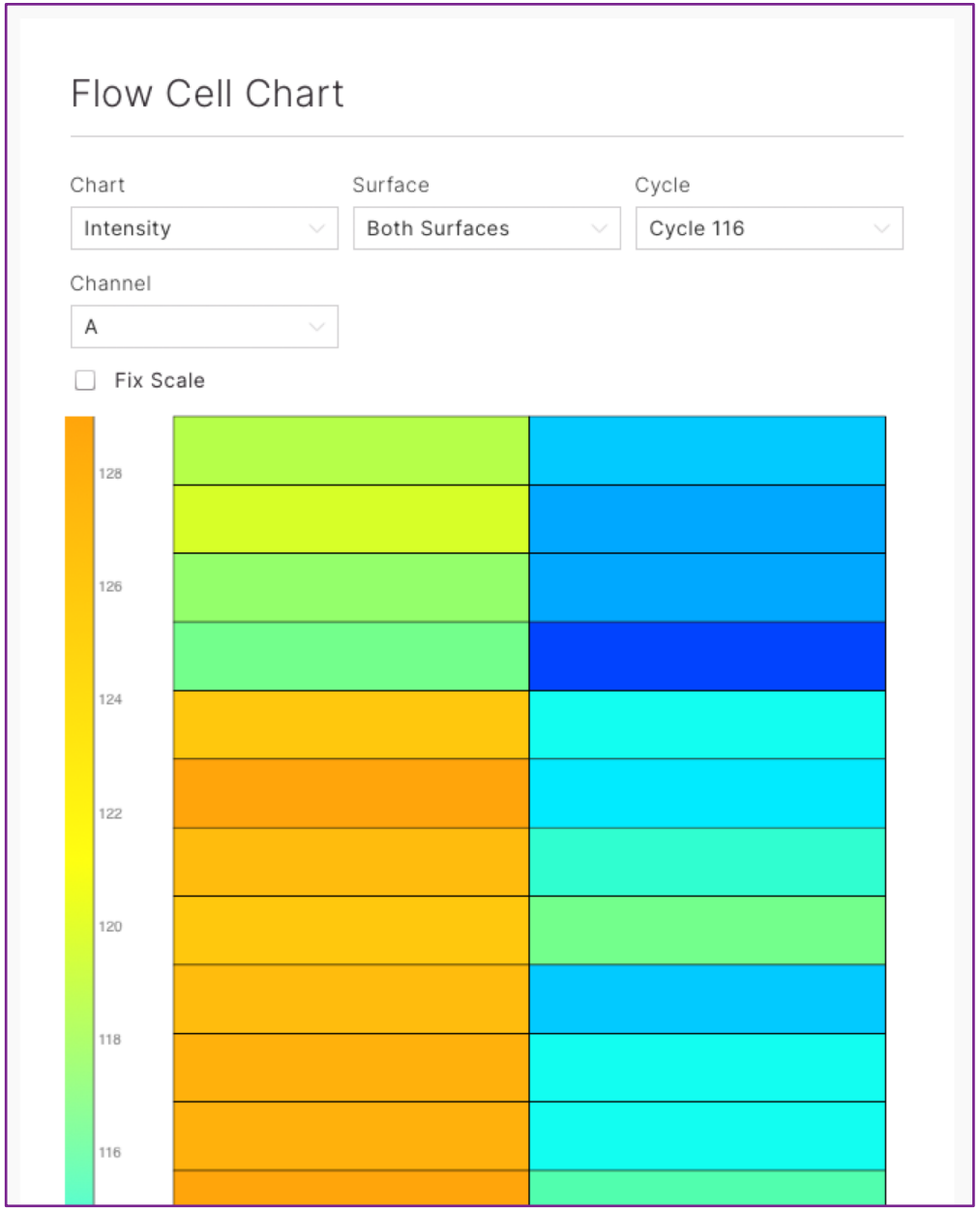
Illumina Run Results

Basespace charts



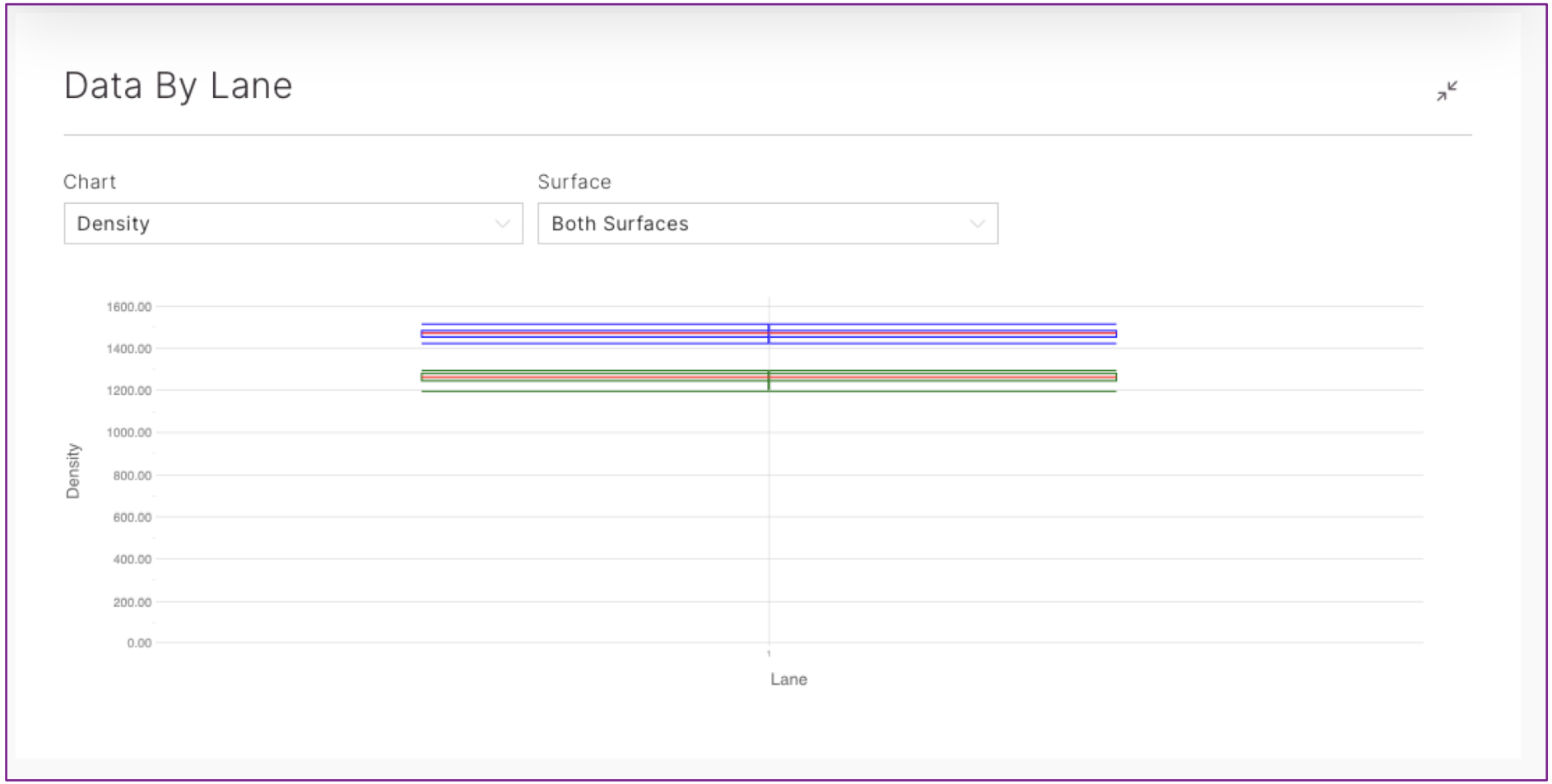
Illumina Run Results

Basespace charts



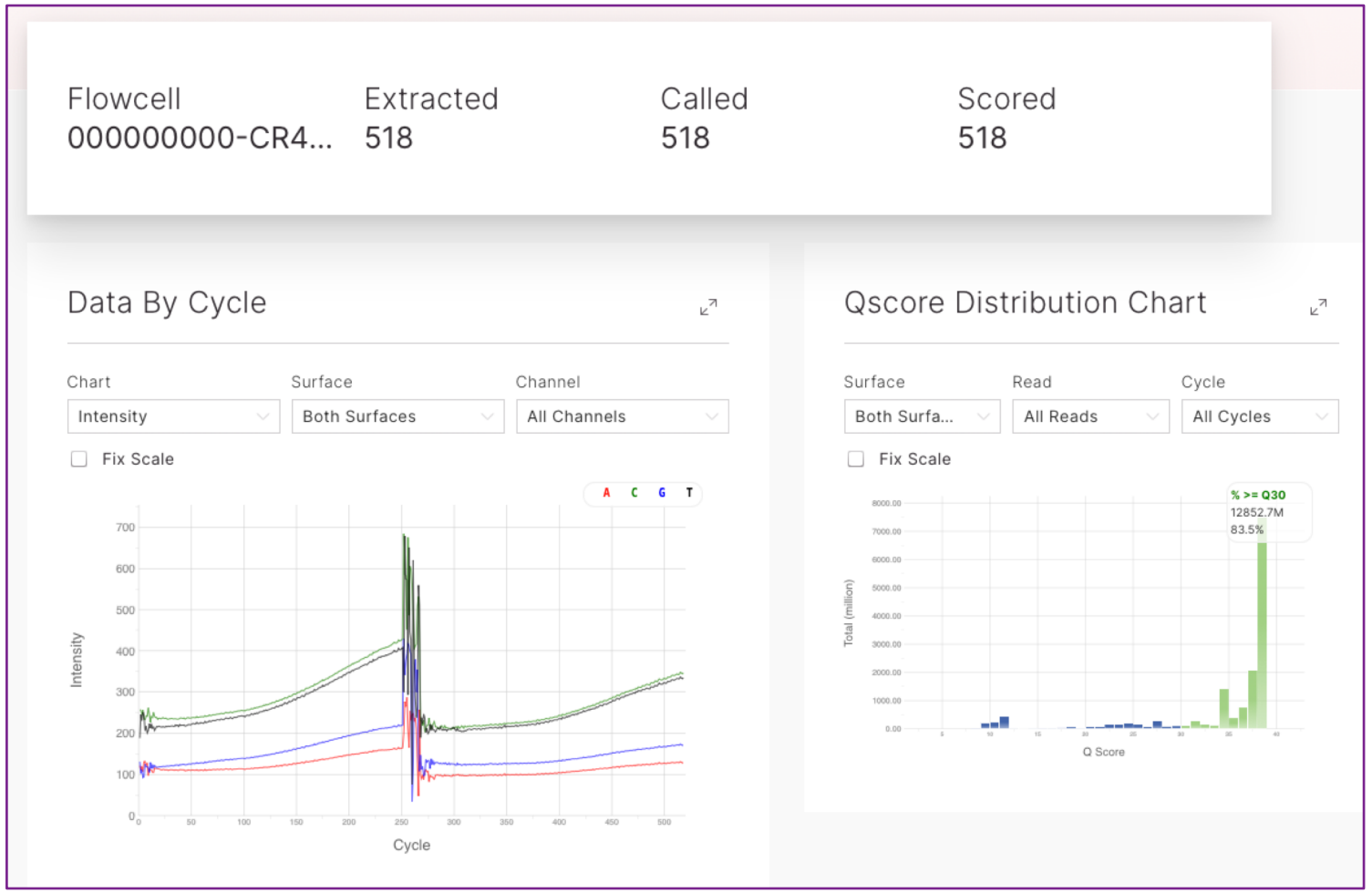
Illumina Run Results

Basespace charts



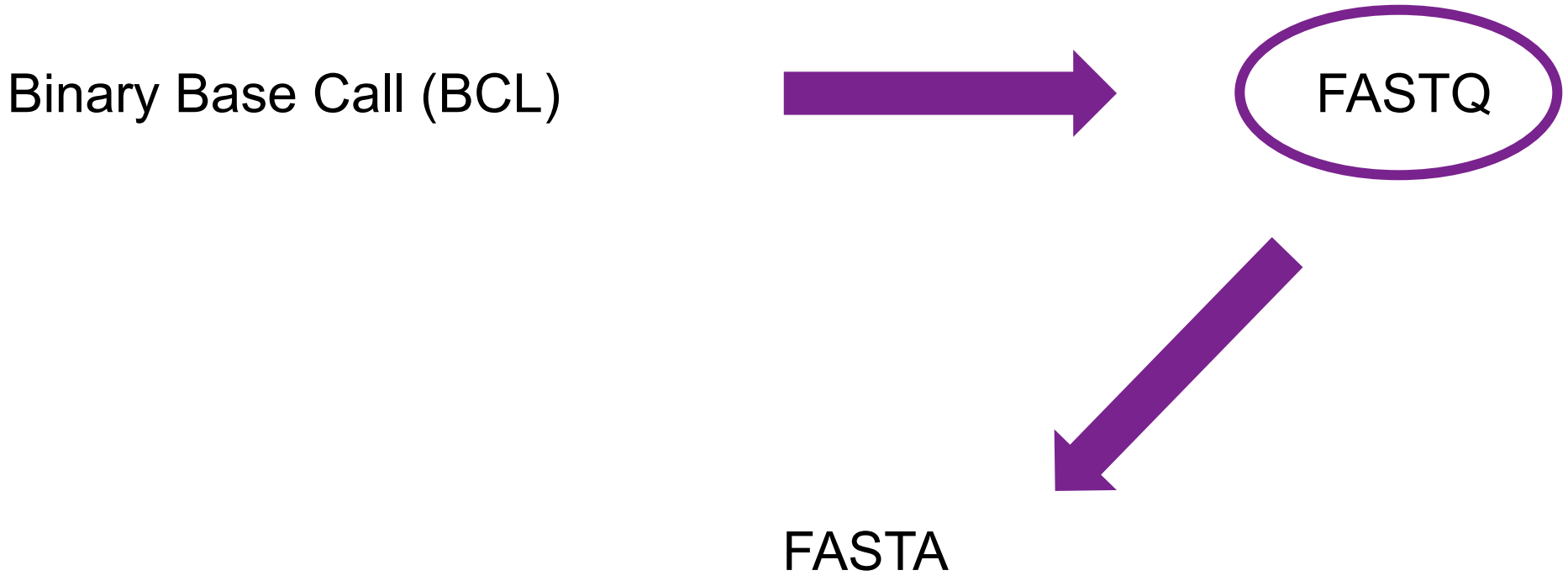
Illumina Run Results

Basespace charts



Next Generation Sequencing (NGS) data

Do you want “raw data” ?



Give me those FASTQ files

FASTQ = FASTA + Quality

1 read = 4 lines

```
@FCC0CD5ACXX:1:1101:1103:2048#ACCGT/1
ACNGTGTTTTTAGTTATTGTTTTGTTAAGTTGGGTTTTTTGTACCCAATAGCCAACAAGCCGCCTTTATGGCGGTTTTTTTGTGCCTGAAAAGTGGGCGCA
+
_BP`ccceggcegihiiighiifhihfddgfhi^efgfhhhhhegiiiiiiiihiihihggeeccdddcccacWTT^acc[ab_`]`[_b`^BBBBBBBB
@FCC0CD5ACXX:1:1101:1165:2058#ACGTT/1
ACGTTAGCAGAATCGCTTTCTGTTCGTTTTCCACCTGCGACAGACGCACCGGACCACGGTTGGCGAGATCGTCGCGCAGAATATCGGCGGCACGCTGCGAC
+
bb_eeceefeggehhdagfghhiihfghighhffhifhhcghfdhiihafgdceba`a\aaccc^V]^baccaccXaaX^bbcccaac[_X]]a[aacXT
@FCC0CD5ACXX:1:1101:1135:2082#AGCGT/1
AGCGTGACAAACATTTTATTGCGCCCGGTTTTATCCAGCTTGAATGCCTGACGAAAGAAGATGATGGTGACGACGATGGAGAGAACAATCAGCACCAGATT
+
bbbeeeeefggfgiihgiigiiiiiiiffgifgeghiiihhfefffhhhfgh_fhggdgegeaceeacbdcbcc\^aa]``_^bb]bcccccbac_a^bc
@FCC0CD5ACXX:1:1101:1239:2083#AGCGT/1
AGCGTCTGACTCACACAAAAACGGTAACACAGTTATCCACAGAATCAGGGGATAAGGCCGGAAAGAACATGTGAGCAAAAAGGCAAAGCCAGGACAAAAGG
+
bbbeeeeeggggiiiiiiiiiigifhhiiighiiihhiiiiiiihiiiiiiiiiihiigcdbbdcdcccccdcccccccccacccccccbcccacccccc
```

Give me those FASTQ files

Header / ID

```
@FCC0CD5ACXX:1:1101:1103:2048#ACCGT/1
ACNGTGTTTTTAGTTATTGTTTTGTTAAGTTGGGTTTTTTGTACCCAATAGCCAACAAGCCGCCTTTATGGCGGTTT
+
_BP`ccceggcegihiiighiifhihfddgfhi^efgfhhhhhegiiiiiiiihiihihggeeccdddcccacWTT
```

Give me those FASTQ files

DNA sequence

```
@FCC0CD5ACXX:1:1101:1103:2048#ACCGT/1
ACNGTGTTTTTAGTTATTGTTTTGTTAAGTTGGGTTTTTTGTACCCAATAGCCAACAAGCCGCCTTTATGGCGGTTT
+
_BP`ccceggcegihiiighiifhihfddgfhi^efgfhhhhhegiiiiiiiihiihihggeeccdddcccacWTT
```

Give me those FASTQ files

Name field (optional)

```
@FCC0CD5ACXX:1:1101:1103:2048#ACCGT/1
ACNGTGTTTTTAGTTATTGTTTTGTTAAGTTGGGTTTTTTGTACCCAATAGCCAACAAGCCGCCTTTATGGCGGTT
+
_BP`ccceggcegihiiighiifhihfddgfhi^efgfhhhhhegiiiiiiiihiihihggeeccdddcccacWTT
```

Give me those FASTQ files

Quality scores

```
@FCC0CD5ACXX:1:1101:1103:2048#ACCGT/1
ACNGTGTTTTTAGTTATTGTTTTGTTAAGTTGGGTTTTTTGTACCCAATAGCCAACAAGCCGCCTTTATGGCGGTT
+
_BP`ccceggcegihiiighiifhihfddgfhi^efgfhhhhhegiiiiiiiihiihihggeeccdddcccacWTT
```

c = 99 → 99 - 64 = 35 = Q

e = 101 → 101 - 64 = 37 = Q

g = 103 → 103 - 64 = 39 = Q

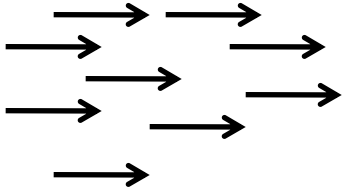
Give me those FASTQ files

$$P = 10^{\frac{-Q}{10}}$$

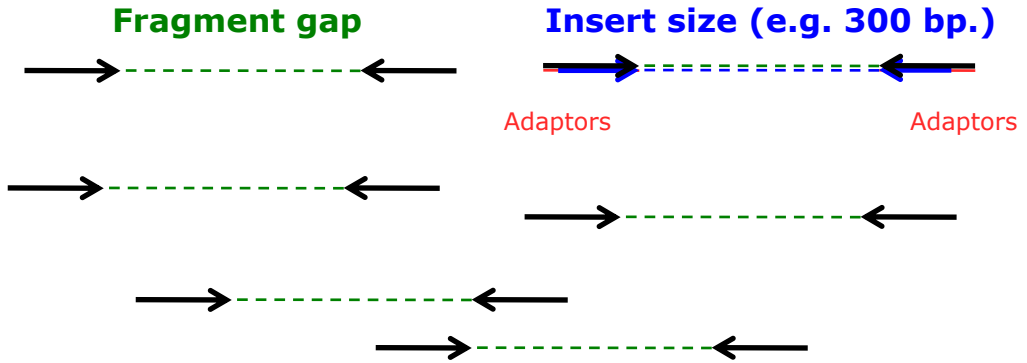
Phred Quality Score	Probability of incorrect base call	Base call accuracy
10	1 in 10	90%
20	1 in 100	99%
30	1 in 1000	99.9%
40	1 in 10,000	99.99%
50	1 in 100,000	99.999%
60	1 in 1,000,000	99.9999%

FASTQ files contain reads, but which reads?

Single-end reads



Paired-end reads



Trimming NGS data

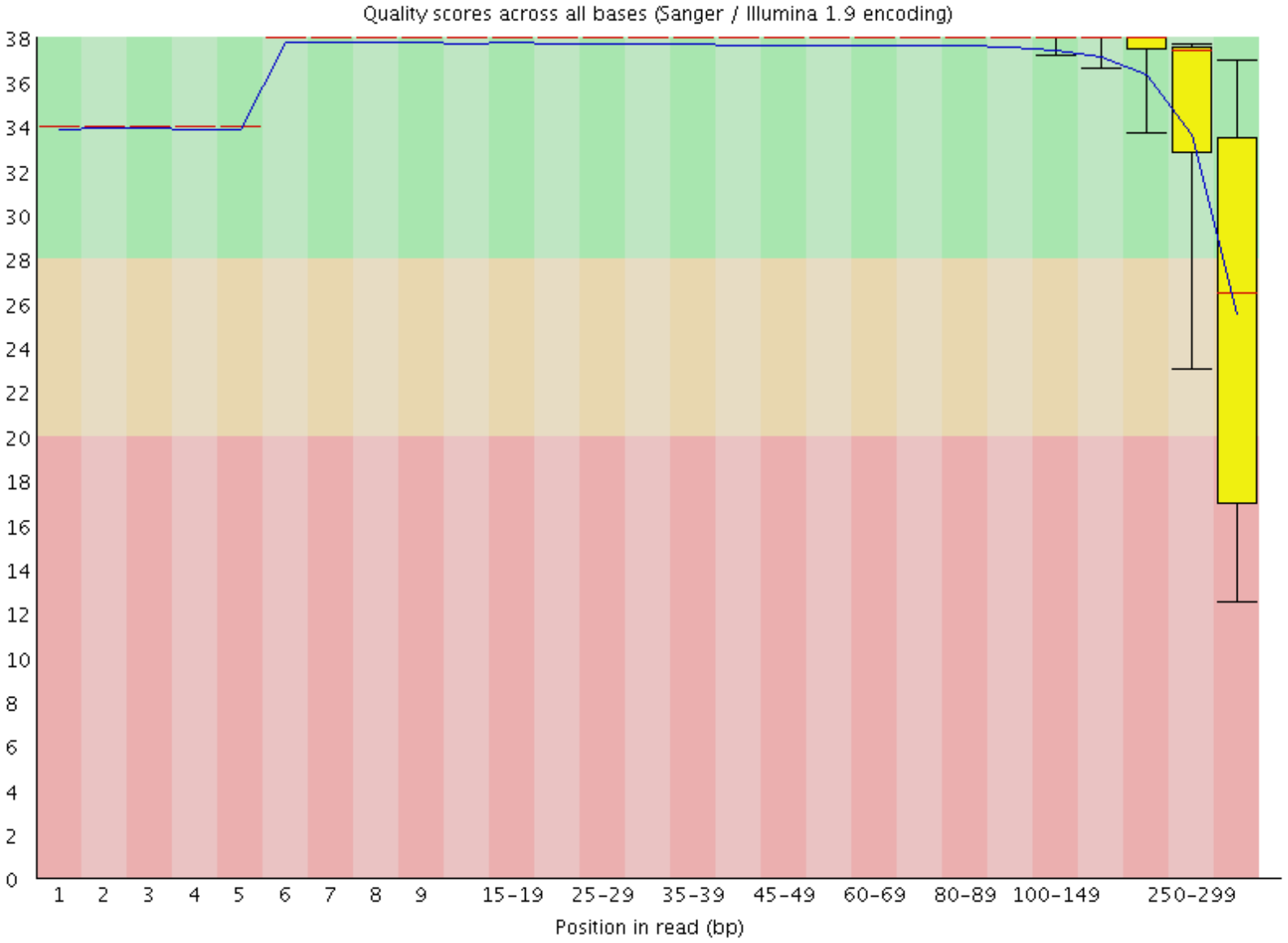
Fastqc & trimming

Why do we want to trim?

- Errors in called nucleotides
  - Incorrect Single Nucleotide Polymorphisms (SNPs)
  - Bad assemblies once reads are assembled to contigs
  - Variations in genes (alleles) that are wrong

- Regions that are not from the original source DNA
  - adaptors

Per base sequence quality



Trimming

What should a trimmer do

- Trim regions that include adaptor regions
- Trim regions that have low phred scores
- Completely remove reads that become too short

Name	year	Citations	url
cutadapt	2011	12944	
Trimmomatic	2014	25976	https://doi.org/10.1093/bioinformatics/btu170
Bbduk.sh	Not published		https://jgi.doe.gov/data-and-tools/bbtools/bb-tools-user-guide/
fastp	2018	1922	10.1093/bioinformatics/bty560

Trimming with bbduk

- K-trim
  - Adaptor sequences
- Q-trim
  - Removing nucleotides with low quality scores (Q)

Q-trimming

trimq=20, qtrim=r, minlen=50

ASCII_BASE=33 Illumina, Ion Torrent, PacBio and Sanger

Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	33 !	11	0.07943	44 ,	22	0.00631	55 7	33	0.00050	66 B
1	0.79433	34 "	12	0.06310	45 -	23	0.00501	56 8	34	0.00040	67 C
2	0.63096	35 #	13	0.05012	46 .	24	0.00398	57 9	35	0.00032	68 D
3	0.50119	36 $	14	0.03981	47 /	25	0.00316	58 :	36	0.00025	69 E
4	0.39811	37 %	15	0.03162	48 0	26	0.00251	59 ;	37	0.00020	70 F
5	0.31623	38 &	16	0.02512	49 1	27	0.00200	60 <	38	0.00016	71 G
6	0.25119	39 '	17	0.01995	50 2	28	0.00158	61 =	39	0.00013	72 H
7	0.19953	40 (	18	0.01585	51 3	29	0.00126	62 >	40	0.00010	73 I
8	0.15849	41)	19	0.01259	52 4	30	0.00100	63 ?	41	0.00008	74 J
9	0.12589	42 *	20	0.01000	53 5	31	0.00079	64 @	42	0.00006	75 K
10	0.10000	43 +	21	0.00794	54 6	32	0.00063	65 A			

Output: Q(F)=37, Q(8)=23, Q(-)=12

@A00197:28:H5N3GDMXX:1:1101:11541:1000 1:N:0:GAGATTCC+GTACTGAC

NGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGA

AGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGAAGACAAAAAAGATCAGAA

\+

#8FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF-FFFF

FFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFF-F88FFF--F-FF8-FF-FF--88-

Q-trimming with bbduk

Bbduk.sh uses the phred algorithm to perform Quality trimming

- **Discarded region has an average quality < Q-threshold**
- **Retained region has an average quality >= Q-threshold**
- **Retained region cannot be extended without adding a subregion with average quality < Q-threshold**
- **Retained region cannot be reduced without removing a subregion with average quality >= Q-threshold**

- *Average quality is calculated by **averaging error-rates P as Q scores are on log scale** and can NOT be used*

- *BBDuk actually does have an option for window-based trimming ("qtrim=window,5" for a 5-bp window) which is easier to understand but doesn't give optimal results.*
- *We use Q-threshold Q=20 => P(20)=0.01*

K-trimming

Adaptor removal using kmers

- A **k**-mer is a string of characters where **k** is the length of the **k**-mer

```
ATGCATG (length 7bp) => 5 kmers, k=3
ATG
 TGC
  GCA
   CAT
    ATG
```

- Very fast lookup (hash table/map)

K-trimming

Adaptor removal using kmers

/home/projects/co_23260/data/adaptors/adaptors.fa

```
>Illumina_Single_End_Apapter_1
ACACTCTTTCCCTACACGACGCTGTTCCATCT
>Illumina_Single_End_Apapter_2
CAAGCAGAAGACGGCATACGAGCTCTTCCGATCT
>Illumina_Single_End_PCR_Primer_1
AATGATACGGCGACCACCGAGATCTACACTCTTTCCCTACACGACGCTCTTCCGATCT
>Illumina_Single_End_PCR_Primer_2
CAAGCAGAAGACGGCATACGAGCTCTTCCGATCT
>Illumina_Single_End_Sequencing_Primer
ACACTCTTTCCCTACACGACGCTCTTCCGATCT
>Illumina_Paired_End_Adapter_1
ACACTCTTTCCCTACACGACGCTCTTCCGATCT
```

K-trimming

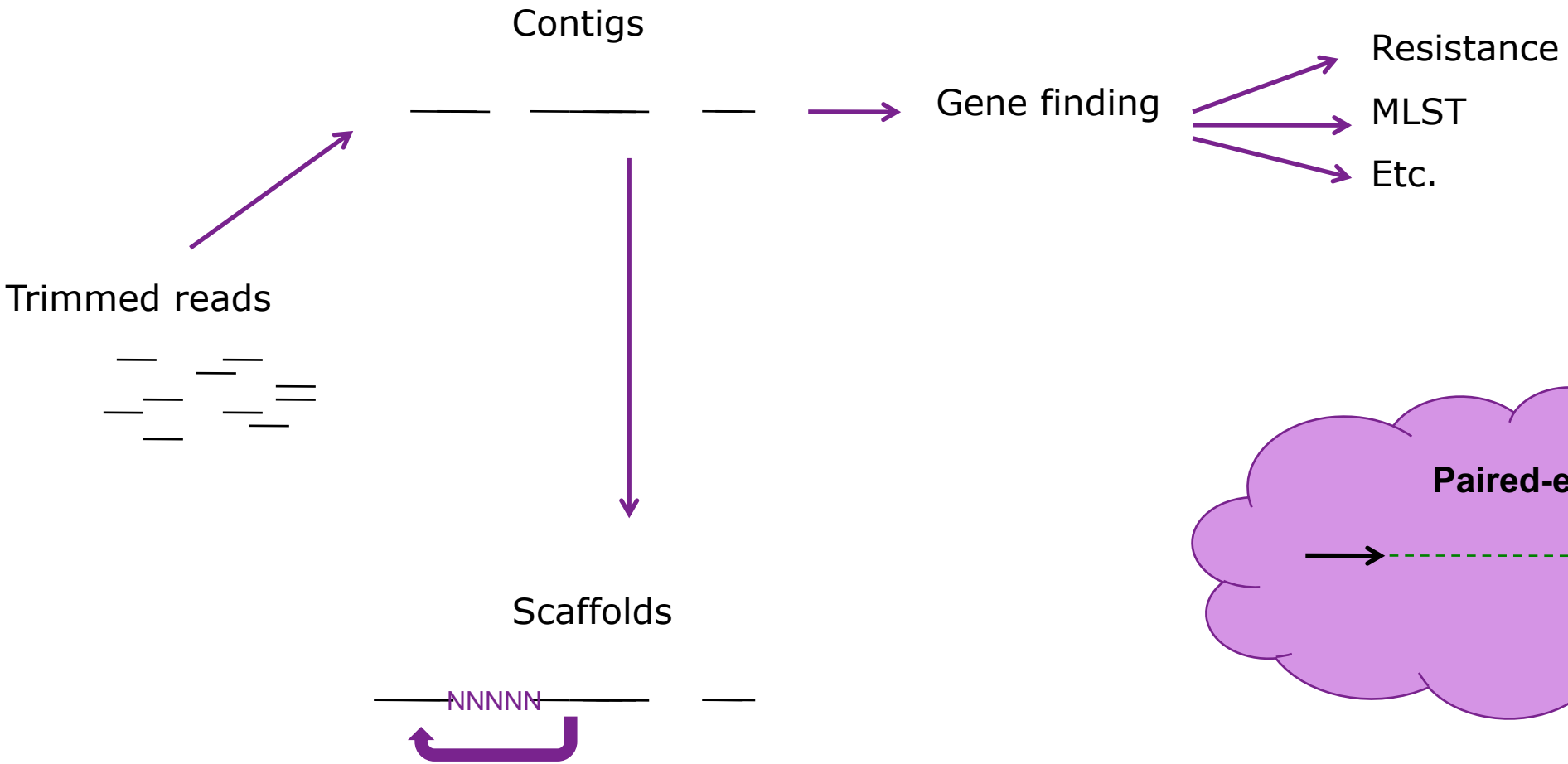
Adaptor removal using k-mers

```
>Illumina_Single_End_Apapter_1 (Found in adaptor file)
ACACTCTTTCCCTACACGACGCTGTTCCATCT
```

- Settings in bbduk (our thresholds)
  - k=19 (ACACTCTTTCCCTACACGA)
  - Mink=11 (ACACTCTTTCC)

De Novo Assembly

De novo assembly output



De Bruijn Graphs

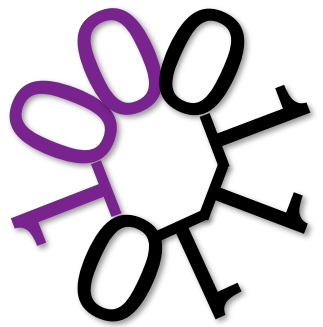
Nicolaas de Bruijn

That's de Braun!

```
000, 001,
010, 011,
100, 101,
110, 111
```

The Superstring problem

Find a shortest circular 'superstring' that contains all possible 'substrings' of length k (k-mers) over a given alphabet A containing n symbols.



00011101

De Bruijn Graphs

Example

K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000

De Bruijn Graphs

Example

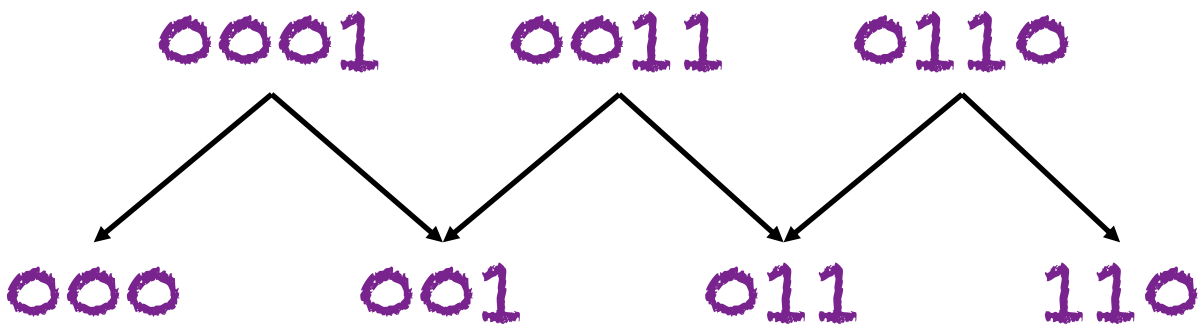
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



...and so on.

Construct a graph B for which every possible (k – 1)-mer is assigned to a node.

De Bruijn Graphs

Example

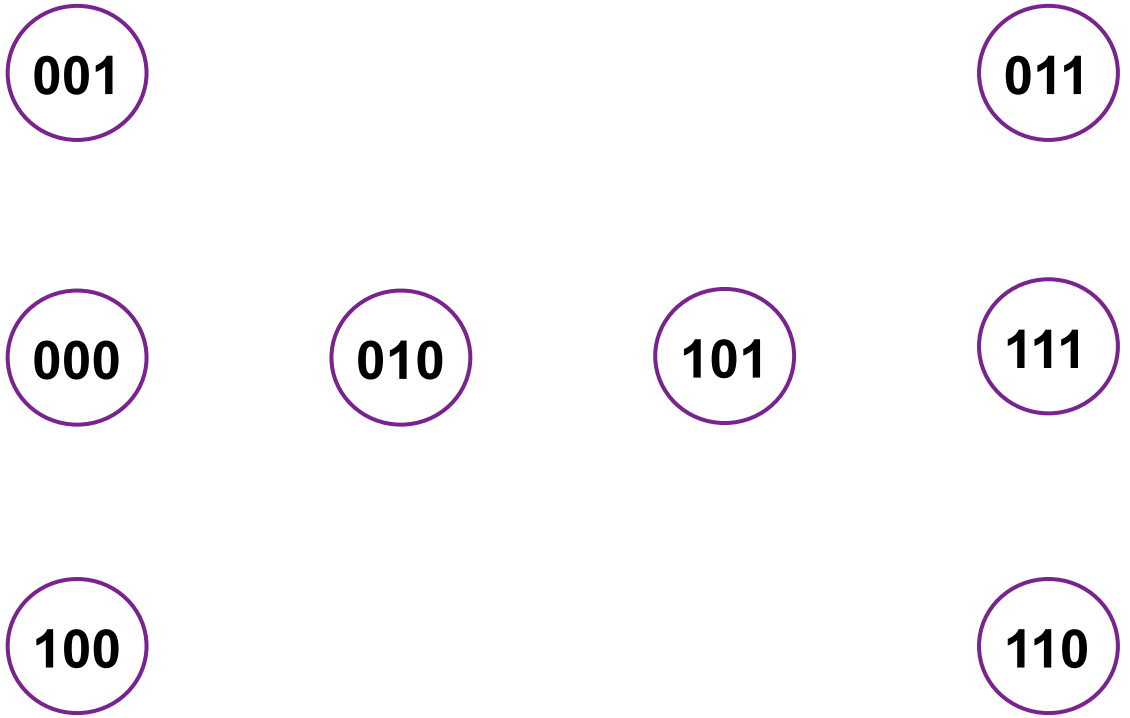
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



Construct a graph B for which every possible (k – 1)-mer is assigned to a node. Connect one (k – 1)-mer by a directed edge to a second (k – 1)-mer if there is some k-mer whose prefix is the former and whose suffix is the latter.

De Bruijn Graphs

Example

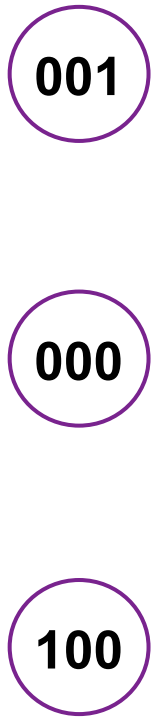
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



100

Construct a graph B for which every possible (k – 1)-mer is assigned to a node. Connect one (k – 1)-mer by a directed edge to a second (k – 1)-mer if there is some k-mer whose prefix is the former and whose suffix is the latter.

De Bruijn Graphs

Example

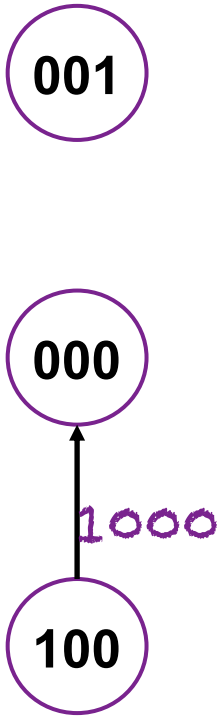
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



1000

Construct a graph B for which every possible (k – 1)-mer is assigned to a node. Connect one (k – 1)-mer by a directed edge to a second (k – 1)-mer if there is some k-mer whose prefix is the former and whose suffix is the latter.

De Bruijn Graphs

Example

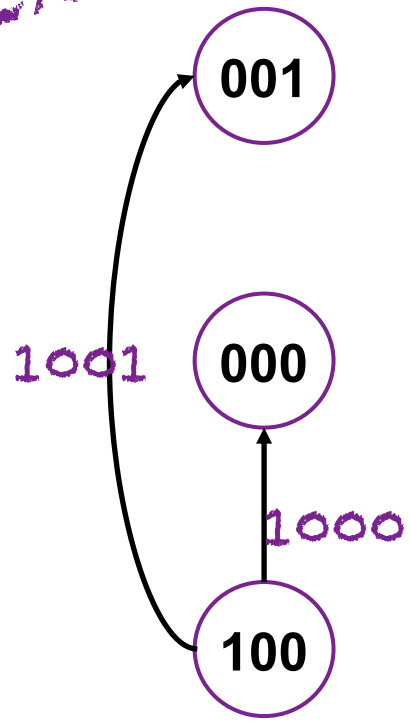
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



1001

Construct a graph B for which every possible (k – 1)-mer is assigned to a node. Connect one (k – 1)-mer by a directed edge to a second (k – 1)-mer if there is some k-mer whose prefix is the former and whose suffix is the latter.

De Bruijn Graphs

Example

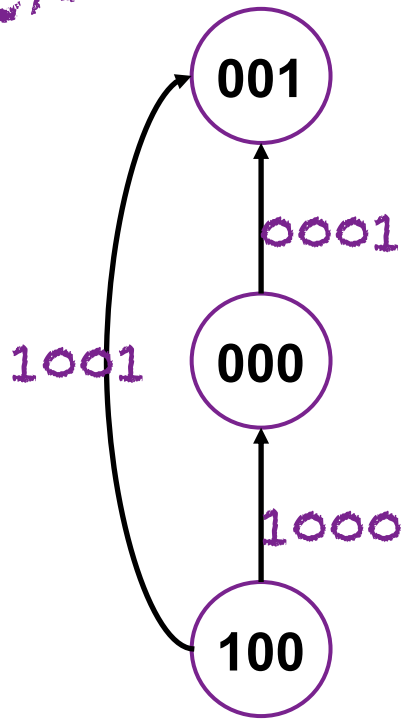
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



0001

Construct a graph B for which every possible (k – 1)-mer is assigned to a node. Connect one (k – 1)-mer by a directed edge to a second (k – 1)-mer if there is some k-mer whose prefix is the former and whose suffix is the latter.

De Bruijn Graphs

Example

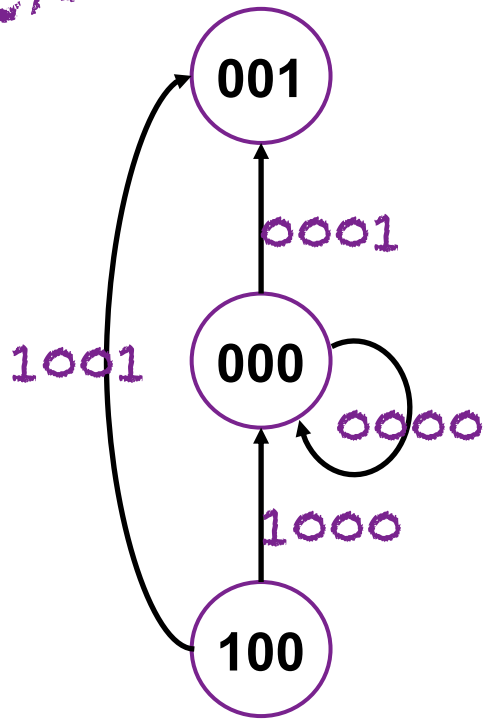
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



0000

Construct a graph B for which every possible (k – 1)-mer is assigned to a node. Connect one (k – 1)-mer by a directed edge to a second (k – 1)-mer if there is some k-mer whose prefix is the former and whose suffix is the latter.

De Bruijn Graphs

Example

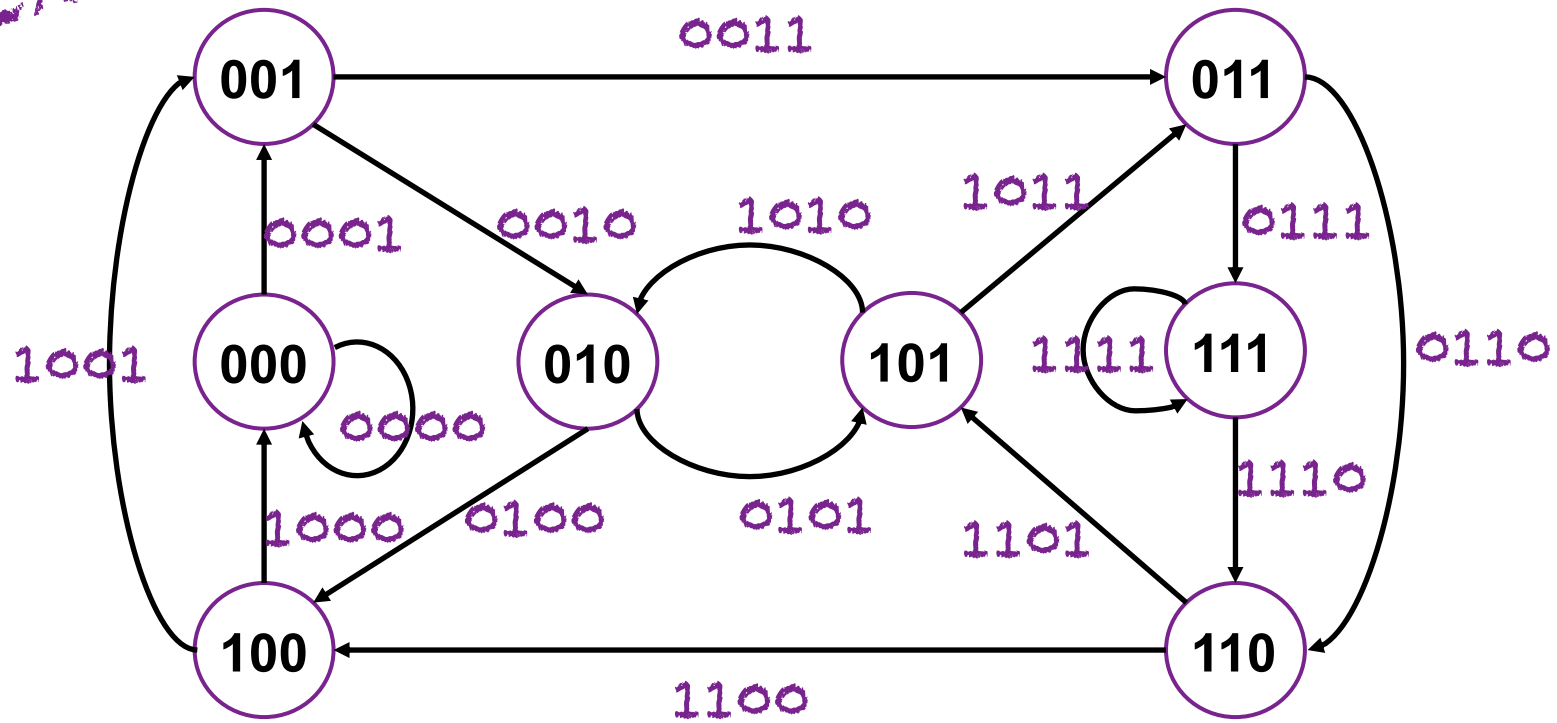
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



Construct a graph B for which every possible (k – 1)-mer is assigned to a node. Connect one (k – 1)-mer by a directed edge to a second (k – 1)-mer if there is some k-mer whose prefix is the former and whose suffix is the latter.

De Bruijn Graphs

000011001

Example

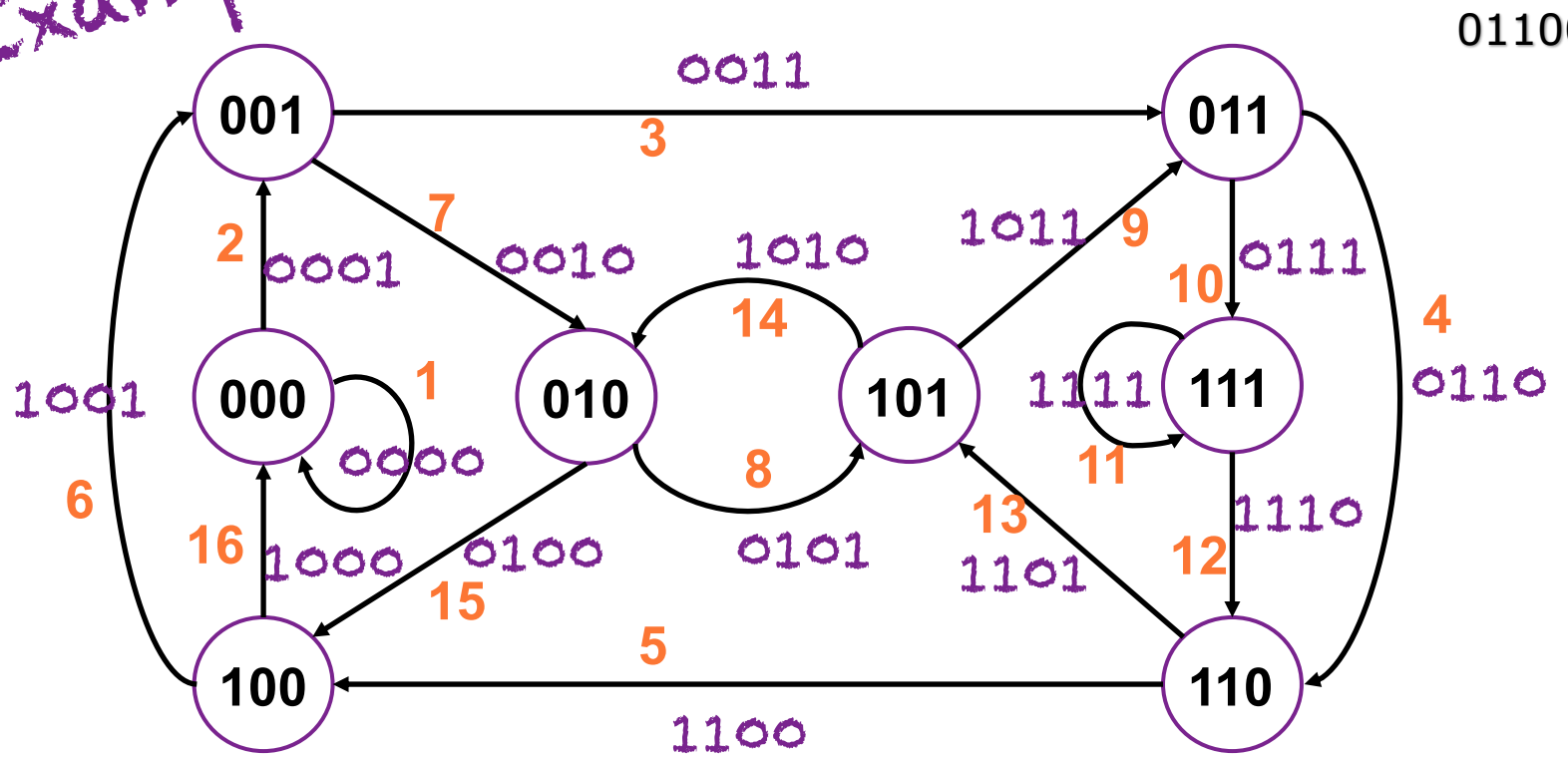
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



Construct a graph B for which every possible (k – 1)-mer is assigned to a node. Connect one (k – 1)-mer by a directed edge to a second (k – 1)-mer if there is some k-mer whose prefix is the former and whose suffix is the latter.

De Bruijn Graphs

0000110010111101

Example

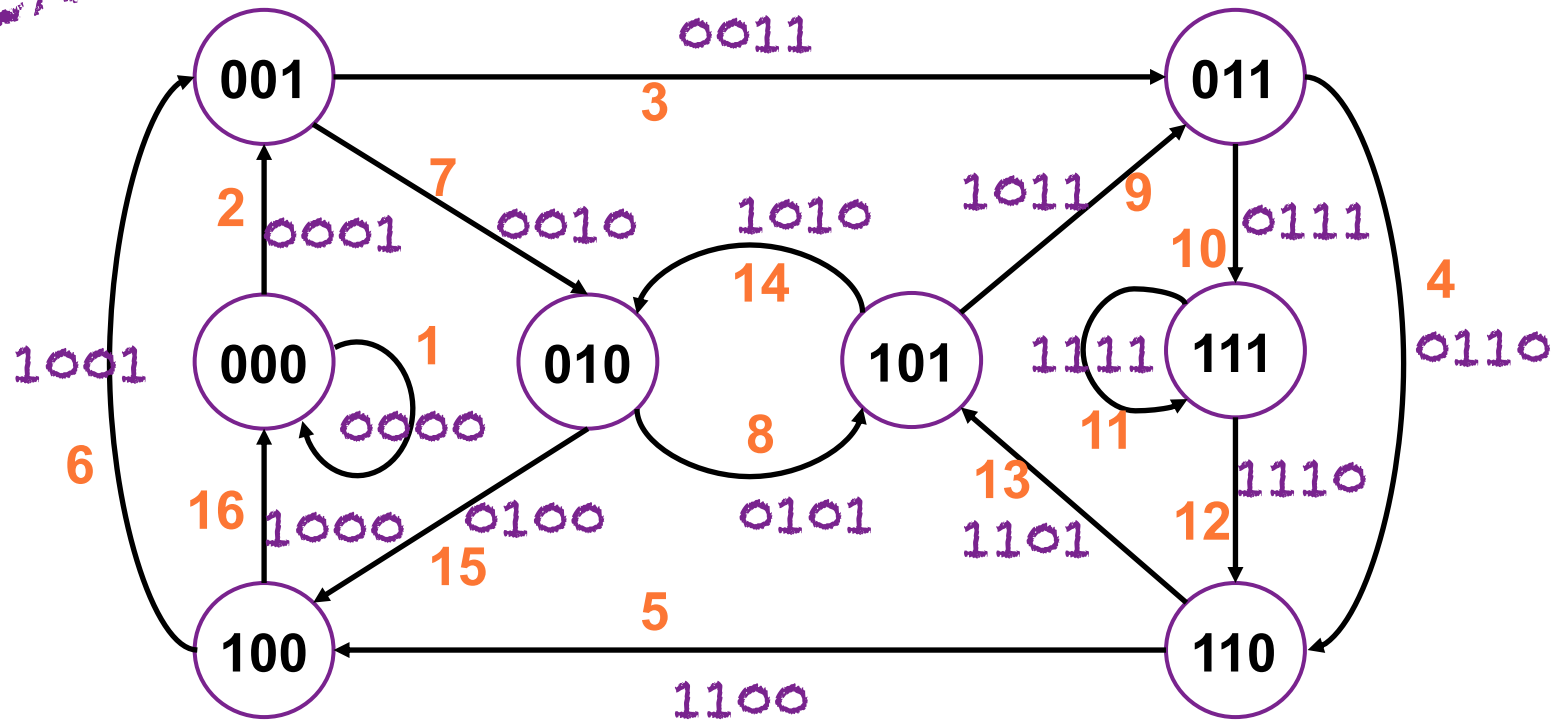
K = 4

n = 2

A = { 0, 1 }

Possible 4-mers: $2^4 = 16$

0000, 0001,
0011, 0110,
1100, 1001,
0010, 0101,
1011, 0111,
1111, 1110,
1101, 1010,
0100, 1000



Construct a graph B for which every possible (k – 1)-mer is assigned to a node. Connect one (k – 1)-mer by a directed edge to a second (k – 1)-mer if there is some k-mer whose prefix is the former and whose suffix is the latter.

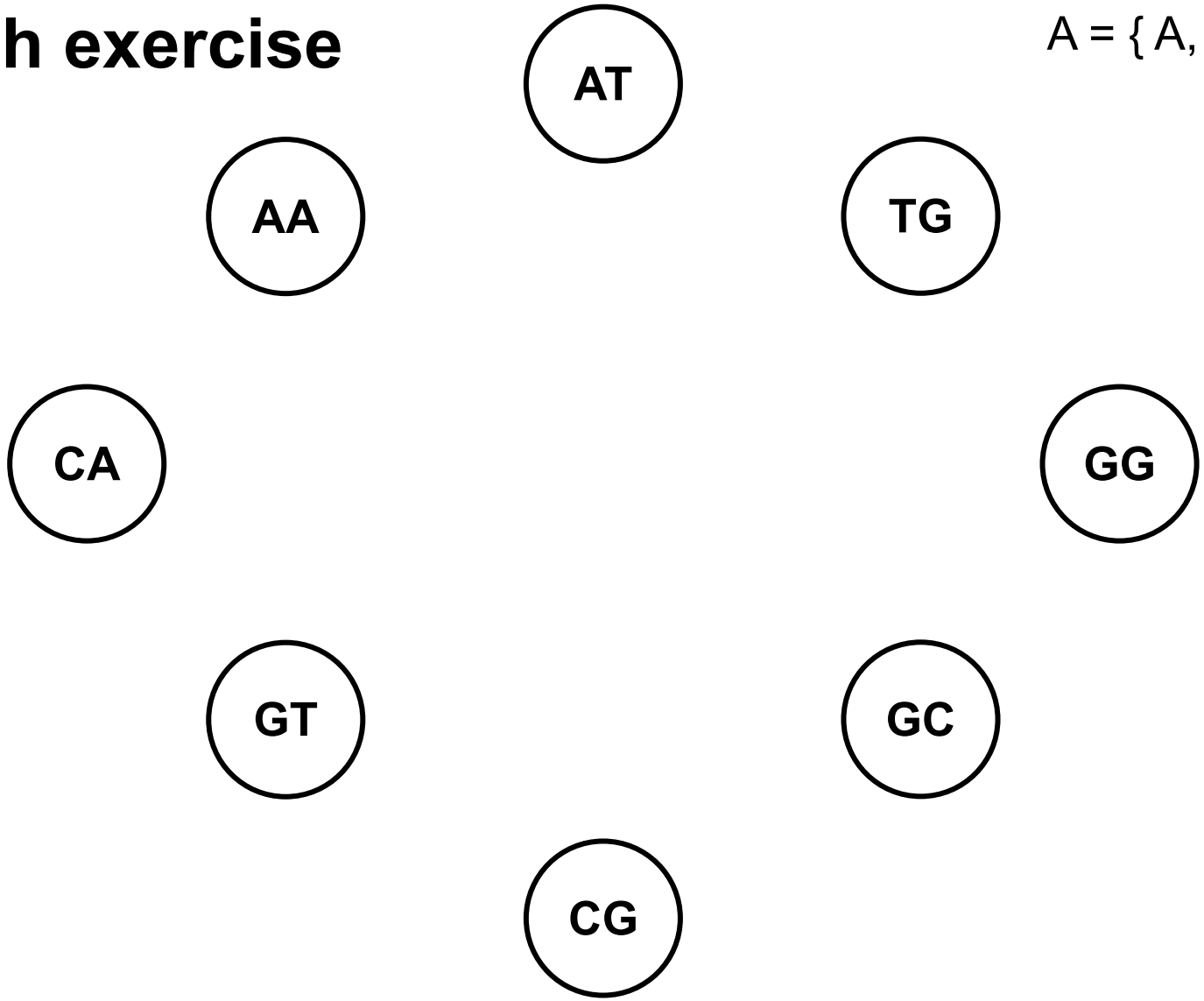
De Bruijn Graph exercise

K = 3

n = 4

A = { A, T, G, C }

ATG	GCA
TGG	CAA
TGC	AAT
GCG	GTG
CGT	GGC



De Bruijn Graph – The assumptions we made

- We can generate all k-mers present in the genome
- All k-mers are error free
- Each k-mer appears at most once
- Single circular chromosome

Larger k results in longer contigs in high coverage areas, but will tend to break contigs in areas with low coverage.

De novo assembly

K-mer based assemblers

- Everything is defined in “Kmer-space”
  - Nucleotide length = Kmer_length + K-1
  - Kmer_coverage = Nucleotide_coverage * (Read_length-K+1)/Read_length

De novo assembly

Velvet assembly

Example

>NODE_1_length_91928_cov_23.136574

AGTTCATTGATAAATCTTTTTTGATTATCATCAACGAGTGCCCACACAGATTGATTGGTT

TATATTGTTAAAGAGCTTTTCCTATCGAAATCGCTTTTAAGCTCAATTCGCTAGGGCTGC

GTATATTACGCTTATTCAGTTGAGTGTCAAACGTTATTTTCTA...

K = 83

Kmer_length + K-1 = Nucleotide length

91928 + 83 - 1 = 92010

Kmer_coverage = Nucleotide_coverage * (Read_length-K+1)/Read_length

$$\frac{23.136574}{(300 - 83 + 1) / 300} = 31.84$$

De novo assembly output

Fasta example:

Header/ID

Sequence

```
>gi|218693476|ref|NC_011748.1| Escherichia coli 55989 chromosome, complete genome
GTAAGTATTTTTCAGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAAGAGT
GTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTAAAATTTTATTGACTTAGGTCACTAA
ATACTTTAACCAATATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCCATGAAACG
CATTAGCACCACCATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTACAGGAA
ACACAGAAAAAAGCCCGCACCTGACAGTGCGGGCTTTTTTTTCGACCAAAGGTAACGAGGTAACAACCAT
GCGAGTGTTGAAGTTCGGCGGTACATCAGTGGCAAATGCAGAACGTTTTCTGCGTGTTGCCGATATTCTG
GAAAGCAATGCCAGGCAGGGGCAGGTGGCCACCGTCCTCTCTGCCCCCGCCAAAATCACCAACCACCTGG
TGGCGATGATTGAAAAAACCATTAGCGGCCAGGATGCTTTACCCAATATCAGCGATGCCGAACGTATTTT
TGCCGAACTTTTGACGGGACTCGCCGCCGCCCAGCCGGGGTTCCCGCTGGCGCAATTGAAAACTTTCGTC
GATCAGGAATTTGCCCAAATAAAACATGTCCTGCATGGCATTAGTTTGTTGGGGCAGTGCCCGGATAGCA
```

Quality Control

More on data quality… quantity…

Coverage: The number of times the genome is covered by the data.

$$C = N \cdot \frac{L}{G}$$

- N: Number of reads
- L: Read length
- G: Genome size
  (target **or** assembly)

Example:
N = 5 mill
L = 100 bp
G = 5 Mbp
C = 5*100/5 = 100X

On average, 100 reads covers each position in the genome.

Breadth-of-coverage:

$$C = \frac{\text{assembly size}}{\text{target size}}$$

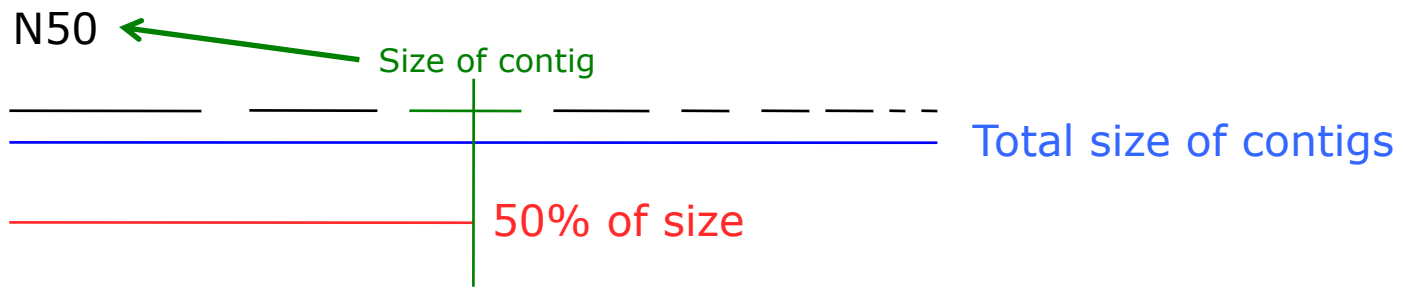
Example:
assembly = 4.9 mill
target = 5 mill
c = 4.9/5 = 0.98

Depth: Number reads that covers a particular nucleotide in each position in the genome.

$$\frac{\text{reads}}{\text{site}} = \text{depth}$$

De novo assembly quality

- Number of contigs
- Size of largest contig
- Assembly size
- N50



What quality parameters to look for

- Data output
  - How much?
- Assembly is a good indicator
  - N50
  - Contigs >500 bp
  - Assembly size

The good, the bad, and the ugly data

- Data output
  - How much?

Coverage of at least 20X

- Assembly is a good indicator
  - N50
  - Contigs >500 bp
  - Assembly size

N50 >= 30 000

No more than 500 contigs

The good, the bad, and the ugly data

Coverage E. coli & Klebsiella:
Bases / 5 = 20
Bases > 100

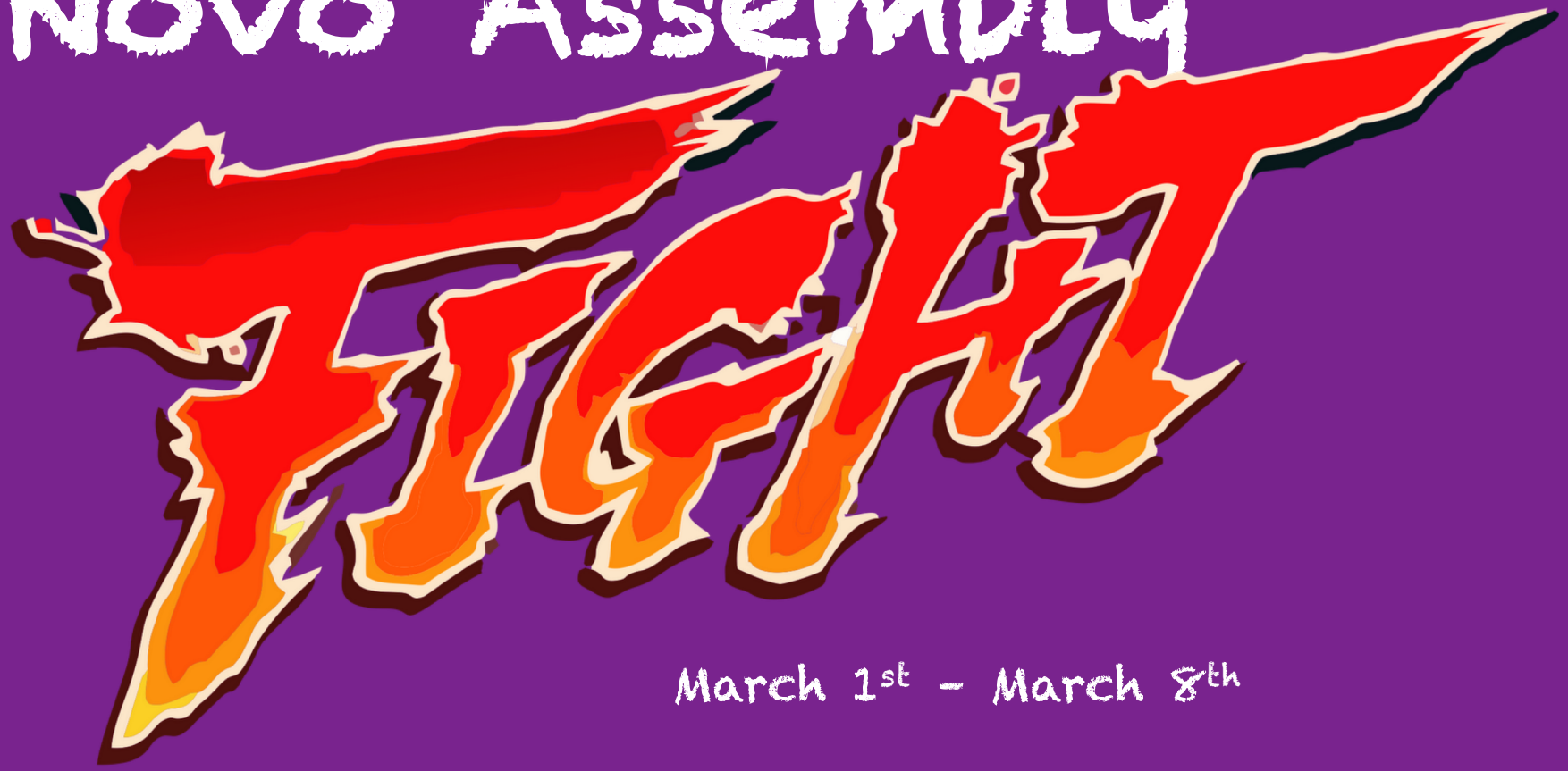
S. aureus: Bases > 60

Example

Isolate ID	Expected species	Output from Quality Control pipline					
		Bases (MB)	Reads	N50	Number of contigs	Longest	Total BPs
1	Escherichia coli	405	2 746 488	254 172	101	399 093	5 302 962
2	Escherichia coli	385	2 595 090	173 253	196	386 355	**10 511 210**
3	Escherichia coli	1 532	11 072180	354 248	57	629 640	5 060 120
4	Escherichia coli	816	5 875 512	188 239	66	556 756	4 779 513
5	Klebsiella pneumoniae	**40**	289 816	**2 920**	**2 346**	29 790	4 994 958
6	Staphylococcus aureus	134	921 174	116 189	51	232 965	2 809 977
7	Staphylococcus aureus	151	1 053 706	135 887	117	342 404	**5 038 817**
8	Staphylococcus aureus	560	3 935 968	125 616	43	504 236	2 746 951
9	Staphylococcus aureus	446	3 124 582	539 934	19	992 722	2 709 295
10	Staphylococcus aureus	**14**	101 884	**1 102**	**1 501**	20 858	**1 545 320**


DTU

De Novo Assembly



March 1st - March 8th

K-trimming

Adaptor removal using kmers

- Hamming distance (hdist) is the number of edits to make 2 sequences identical

- Sequence (k=3, hdist=0): ATGC
- K-mers: ATG TGC

- Hdist=1 (expanding k-mer=ATG)
- ATG TTG CTG GTG
- ATG AAG ACG AGG
- ATG ATA ATC ATG

Number of k-mers: $4k^{hdist}$

hdist > 0 => more memory requirements to store all new kmers