

**DTU**



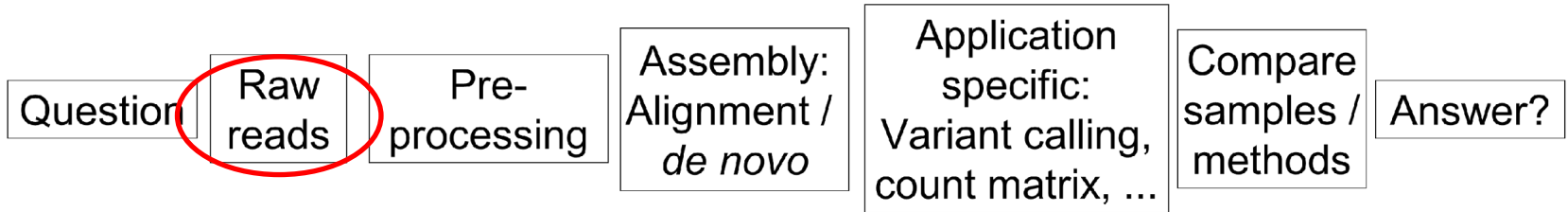
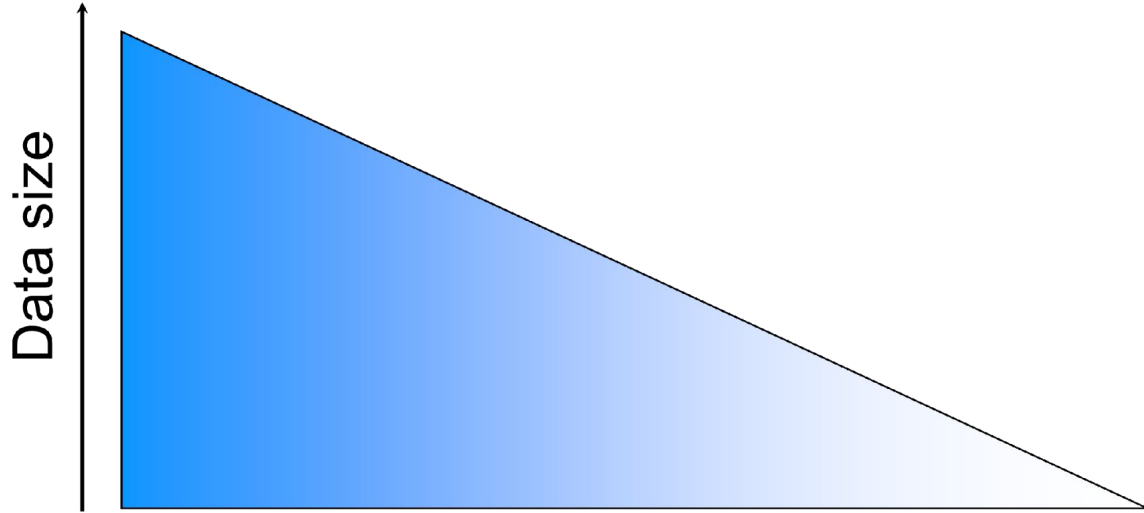


**DTU Health Technology  
Bioinformatics**

## **Data basics**

*Gabriel Renaud  
Associate Professor  
Section of Bioinformatics  
Technical University of Denmark  
[gabriel.reno@gmail.com](mailto:gabriel.reno@gmail.com)*

# Generalized NGS analysis



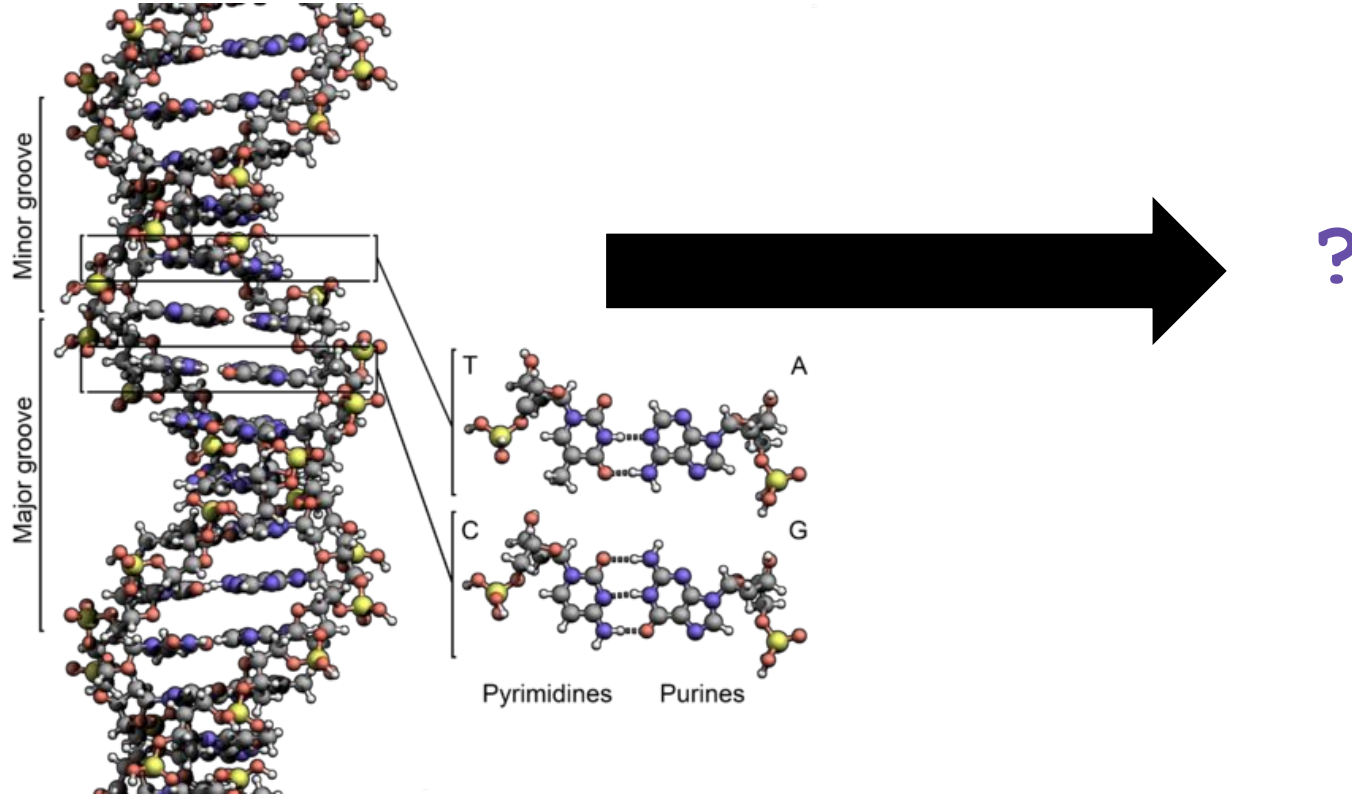
# Outline

- Starting point – what we learned yesterday
- Sequence data storage
  - Fasta format
  - Fastq format
- Quality scores
- Multiplexing/Demultiplexing (briefly mentioned)
- Sequencing read types
  - Single end
  - Paired end

# What we learned yesterday

- Sequencing technologies
- Key concepts:
  - Read length
  - Error types
  - Throughput
- 2nd generation vs 3rd generation sequencing

# The problem at hand, storing DNA information in a computer



# What is sequence data?

## Sequences are stored in fasta-files

Header →

```
>gi|218693476|ref|NC_011748.1| Escherichia coli 55989 chromosome, complete genome
```

```
GTAAGTATTTTTCAGCTTTTCATTCTGACTGCAACGGGCAATATGTCTCTGTGTGGATTAAAAAAGAGT
```

```
GTCTGATAGCAGCTTCTGAACTGGTTACCTGCCGTGAGTAAATTTAAATTTTATTGACTTAGGTCACTAA
```

Sequence →

```
ATACTTTAACCAATATAGGCATAGCGCACAGACAGATAAAAATTACAGAGTACACAACATCCATGAAACG
```

```
CATTAGCACCACCATTACCACCACCATCACCATTACCACAGGTAACGGTGCGGGCTGACGCGTACAGGAA
```

```
ACACAGAAAAAAGCCCGCACCTGACAGTGC GGCTTTTTTTTTCGACCAAAGGTAACGAGGTAACAACCAT
```

```
GCGAGTGTGAAAGTTCGGCGGTACATCAGTGGCAAATGCAGAACGTTTTCTGCGTGTGCCGATATCTG
```

```
GAAAGCAATGCCAGGCAGGGGCAGGTGGCCACCGTCCTCTCTGCCCCCGCCAAAATCACCAACCACCTGG
```

```
TGGCGATGATTGAAAAAACCATTAGCGGCCAGGATGCTTTACCCAATATCAGCGATGCCGAACGTATTTT
```

```
TGCCGAACTTTTGACGGGACTCGCCGCCGCCAGCCGGGGTTCCCGCTGGCGCAATTGAAAACTTTCGTC
```

```
GATCAGGAATTTGCCCAAATAAAACATGTCCTGCATGGCATTAGTTTGTGGGGCAGTGCCCGGATAGCA
```

**E.coli ~ 4.5 - 6 Mbases**

**Human ~ 3.2 Gbases**

# Reminder: things ~~can~~ will go wrong

template  
read

mismatch

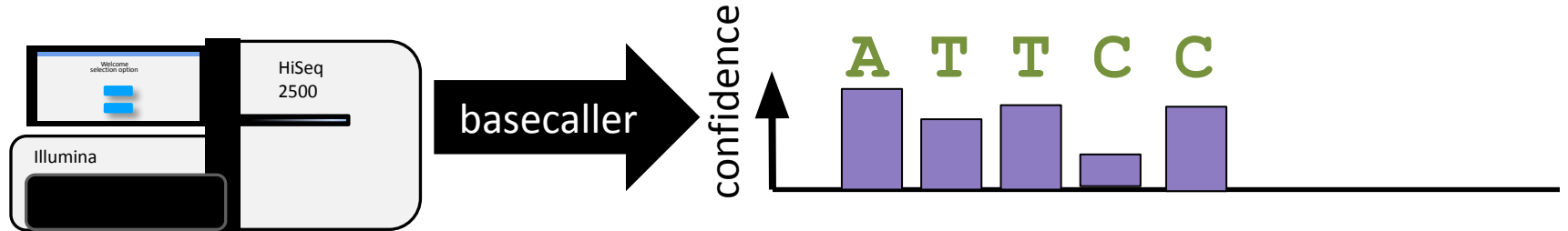
AGCAATCTCAATTACAAATATACACCAACAAA  
AGCAATCTCAATTACAGATATACACCAACAAA

AGCAATCTCAATTACA-AATATACACCAACAA  
AGCAATCTCAATTACACGATATACACCAACAA

AGCAATCTCAATTACAAATATACACCAACAAA  
AGCAATCTCAATTACA-ATATACACCAACAAA



# Basecalling

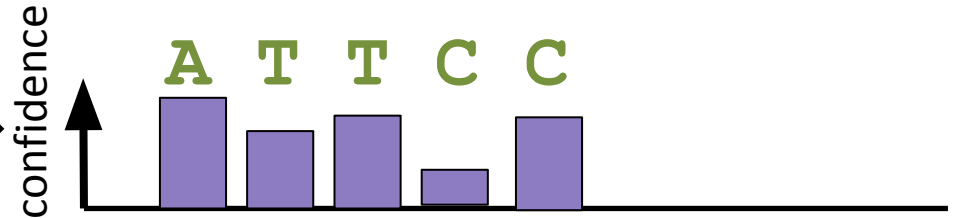
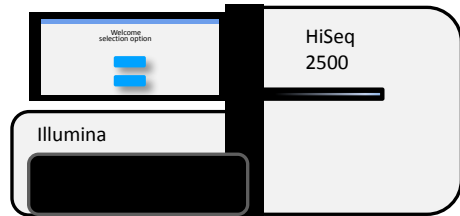


Quality predictor values of individual bases:

- Intensity profiles
- phasing
- signal to noise ratio

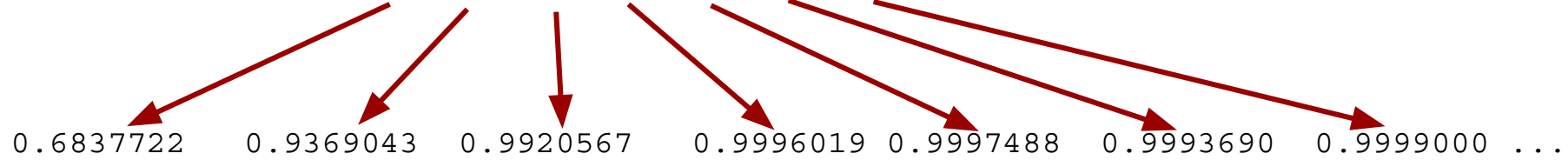
Depends on machine, chemistry, software

# Basecalling



@ILLUMINA-C90280\_0030\_FC:5:1:2675:1090#NNNNNN/1

A T T C C C G ...



# How to store the probability that a base is wrong?

Idea#1:

Store a floating-point number for each base:

```
@ILLUMINA-C90280_0030_FC:5:1:2675:1090#NNNNNN/1
```

```
ATTCCCGGCCTTTTCCAGGCCTGCCTGCTCGAGC
```

+ . . . .

```
0.6838 0.9369 0.9921 0.9996 0.9997 0.9994 0.9999
```

Not great  
(4 bytes/qual score)

'6'
'8'
'3'
'8'

# How to store the probability that a base is wrong?

Idea#2:

Store  $10 \times \log_{10}$  of a floating-point number for each base:

@ILLUMINA-C90280\_0030\_FC:5:1:2675:1090#NNNNNN/1

ATTCCCGGCCTTTTCCAGGCCTGCCTGCTCGAGC



5 12 20 34 36 32 40

better...

(2 bytes/qual score)

'0'
'5'

Phred-scale

$$Quality = -10 \times \log_{10}(P_{error})$$

$$P_{error} = 10^{\frac{-Quality}{10}}$$

Q ~ Prob

10 ~ 0.1

20 ~ 0.01

30 ~ 0.001

40 ~ 0.0001

# How to store the probability that a base is wrong?

Idea#3:

Store the ASCII character corresponding to  $10 \times \log_{10}$  of a floating-point number for each base:

@ILLUMINA-C90280\_0030\_FC:5:1:2675:1090#NNNNNN/1  
 ATTCCCGGCCTTTTTCCAGGCCTGCCTGCTCGAGC  
 ↓ ↓ . . .  
 &-5CEAI

much better...  
 (1 bytes/qual score)

'&'

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			

# Then what is NGS data?

## Fastq

Header → @ILLUMINA-C90280\_0030\_FC:5:1:2675:1090#NNNNNN/1

Sequence → ATTCCCGGCCTTTTTCCAGGCCTGCCTGCTCGAGC

+  
Qualities → BAAAGECEE<EEDFEDF3DBDBB=A+==>9>>88?  
(probability that basecall is wrong)

**Millions to billions of these...**

# Phred scale (offset 33 vs 64)

@ILLUMINA-C90280\_0030\_FC:5:1:2675:1090#NNNNNN/1

ATTCCCGGCCTTTTTCCAGGCCTGCCTGCTCGAGC

+

BAAAGECEE<EEDFEDF3DBDBB=A+==>9>>88?

Q ~ Prob

10 ~ 0.1

20 ~ 0.01

30 ~ 0.001

40 ~ 0.0001

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			

ASCII_BASE=64 Old Illumina											
Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII	Q	P_error	ASCII
0	1.00000	64 @	11	0.07943	75 K	22	0.00631	86 V	33	0.00050	97 a
1	0.79433	65 A	12	0.06310	76 L	23	0.00501	87 W	34	0.00040	98 b
2	0.63096	66 B	13	0.05012	77 M	24	0.00398	88 X	35	0.00032	99 c
3	0.50119	67 C	14	0.03981	78 N	25	0.00316	89 Y	36	0.00025	100 d
4	0.39811	68 D	15	0.03162	79 O	26	0.00251	90 Z	37	0.00020	101 e
5	0.31623	69 E	16	0.02512	80 P	27	0.00200	91 [	38	0.00016	102 f
6	0.25119	70 F	17	0.01995	81 Q	28	0.00158	92 \	39	0.00013	103 g
7	0.19953	71 G	18	0.01585	82 R	29	0.00126	93 ]	40	0.00010	104 h
8	0.15849	72 H	19	0.01259	83 S	30	0.00100	94 ^	41	0.00008	105 i
9	0.12589	73 I	20	0.01000	84 T	31	0.00079	95 _	42	0.00006	106 j
10	0.10000	74 J	21	0.00794	85 U	32	0.00063	96 `			

# Read types

Fragment DNA:

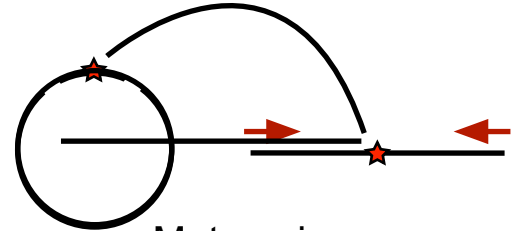


Single end



Paired end

Ins: 200-800 bp



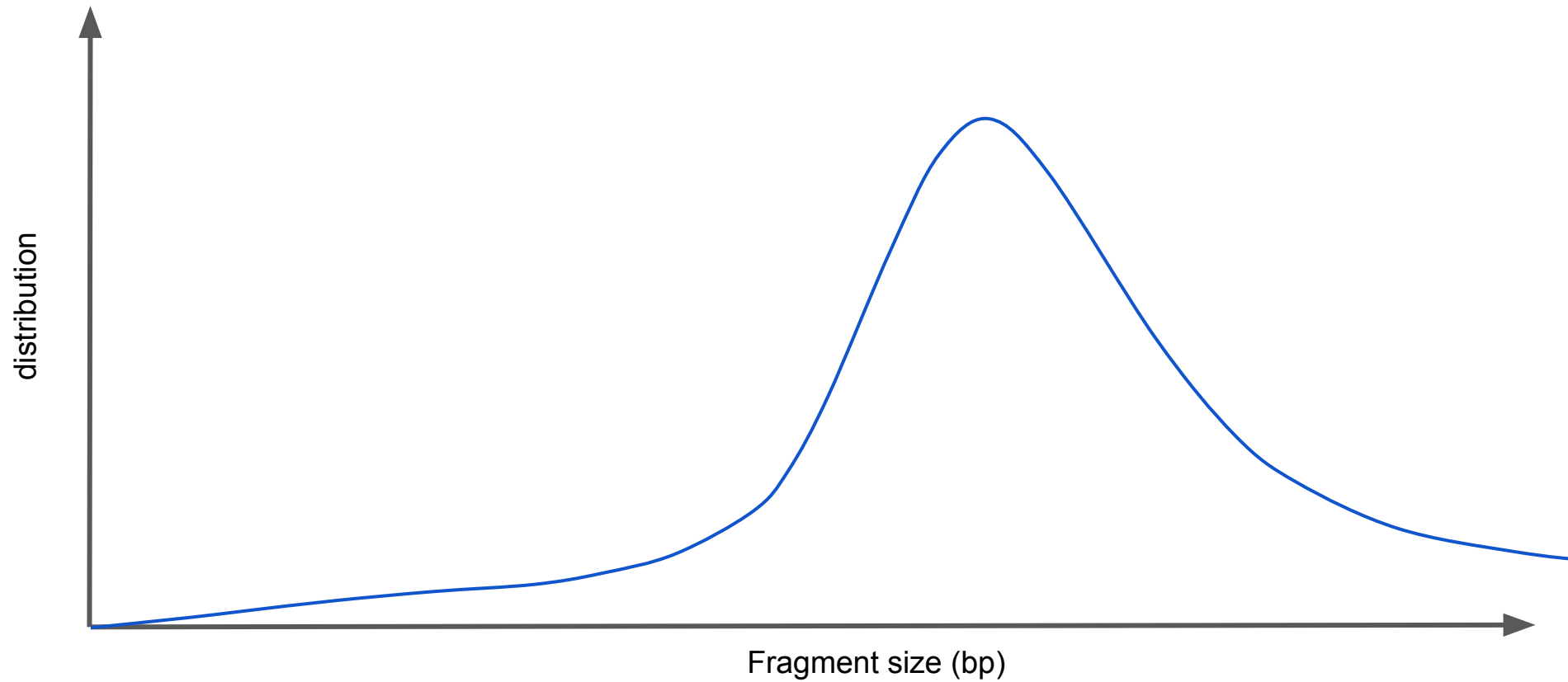
Mate pair

Ins: 2kb - 40kb (~5kb)

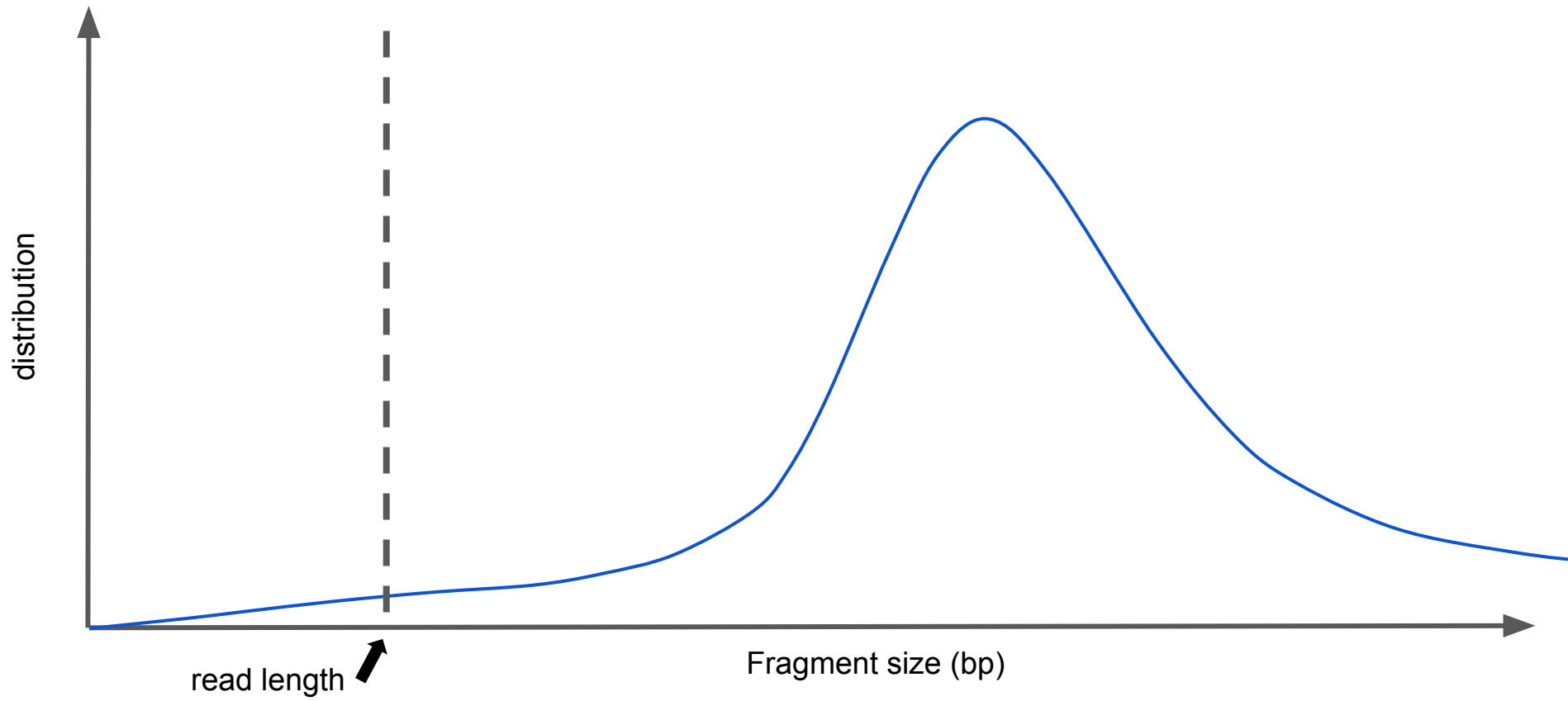
Protocol/technology dependent

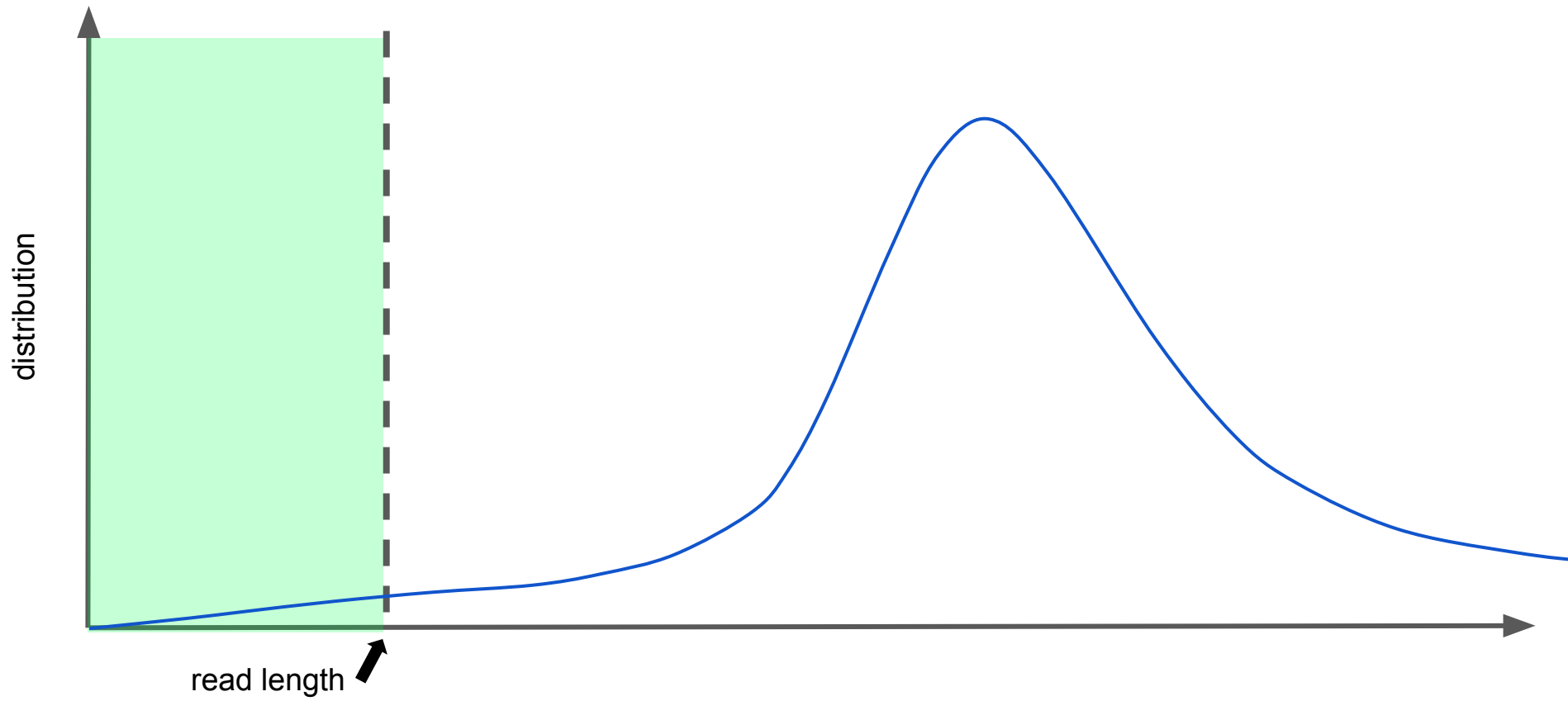
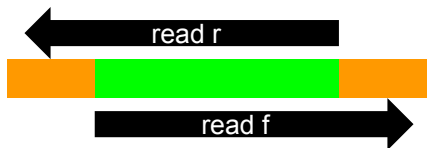


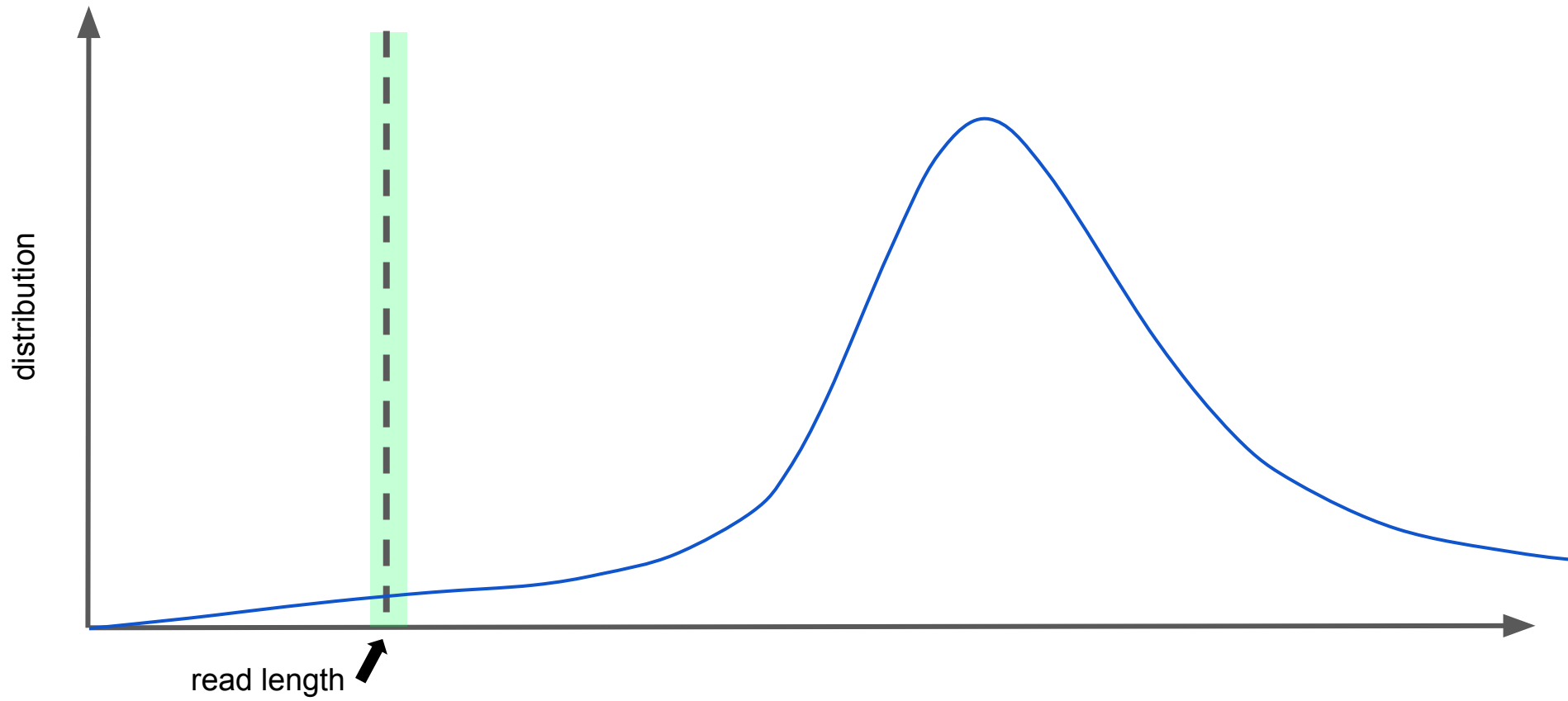
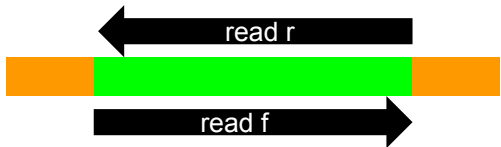
# Insert size distribution

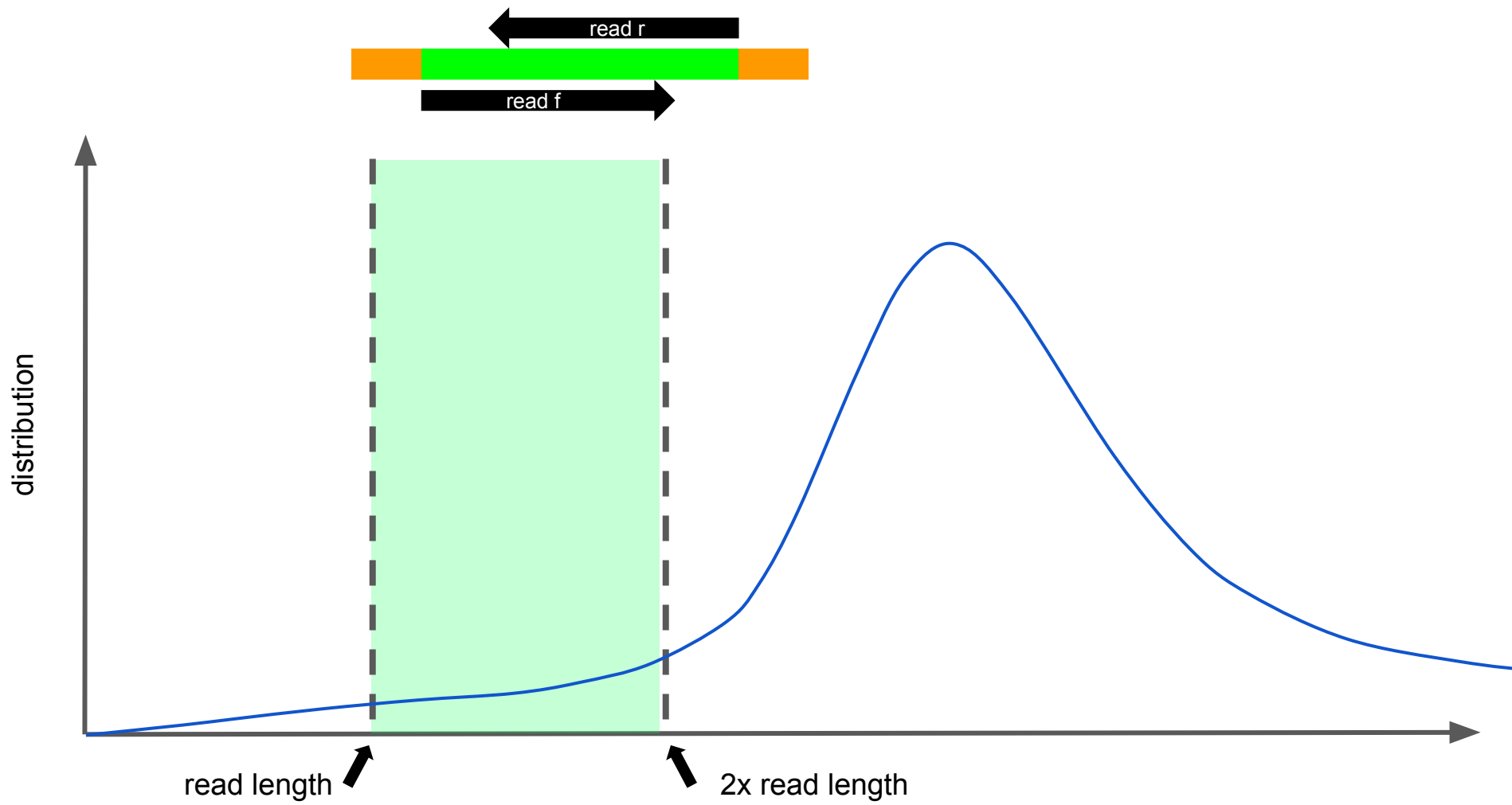


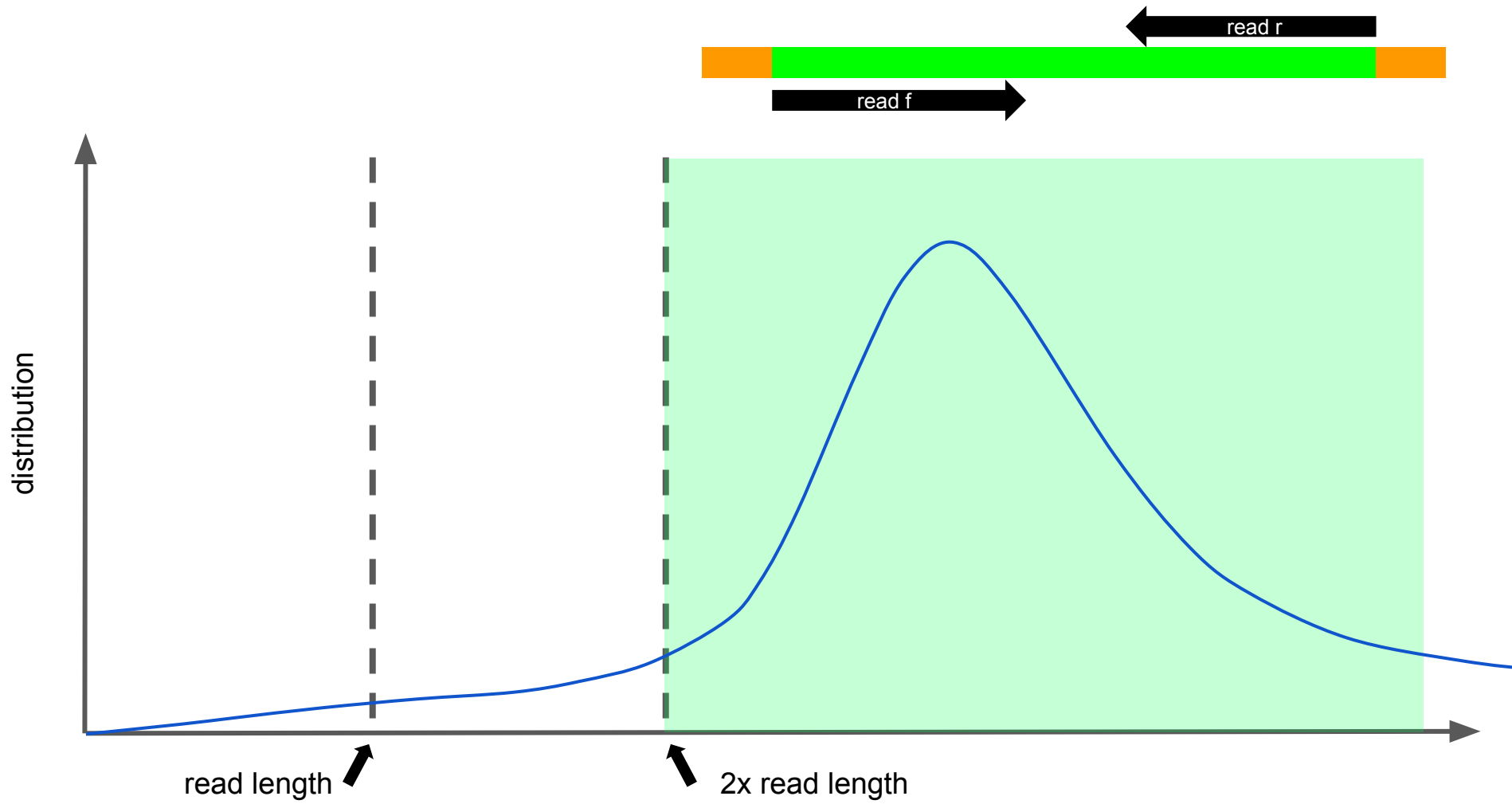
# Insert size distribution

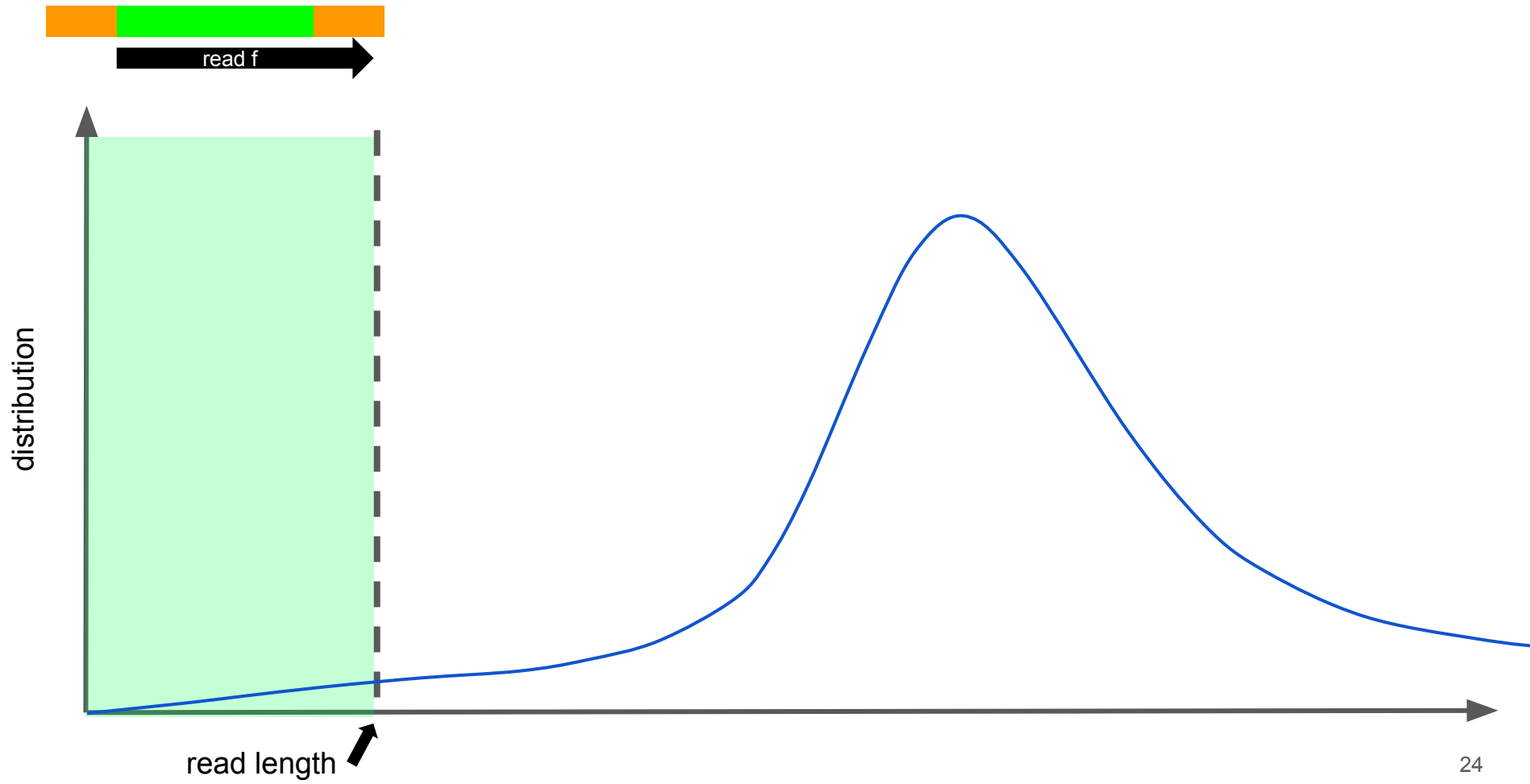






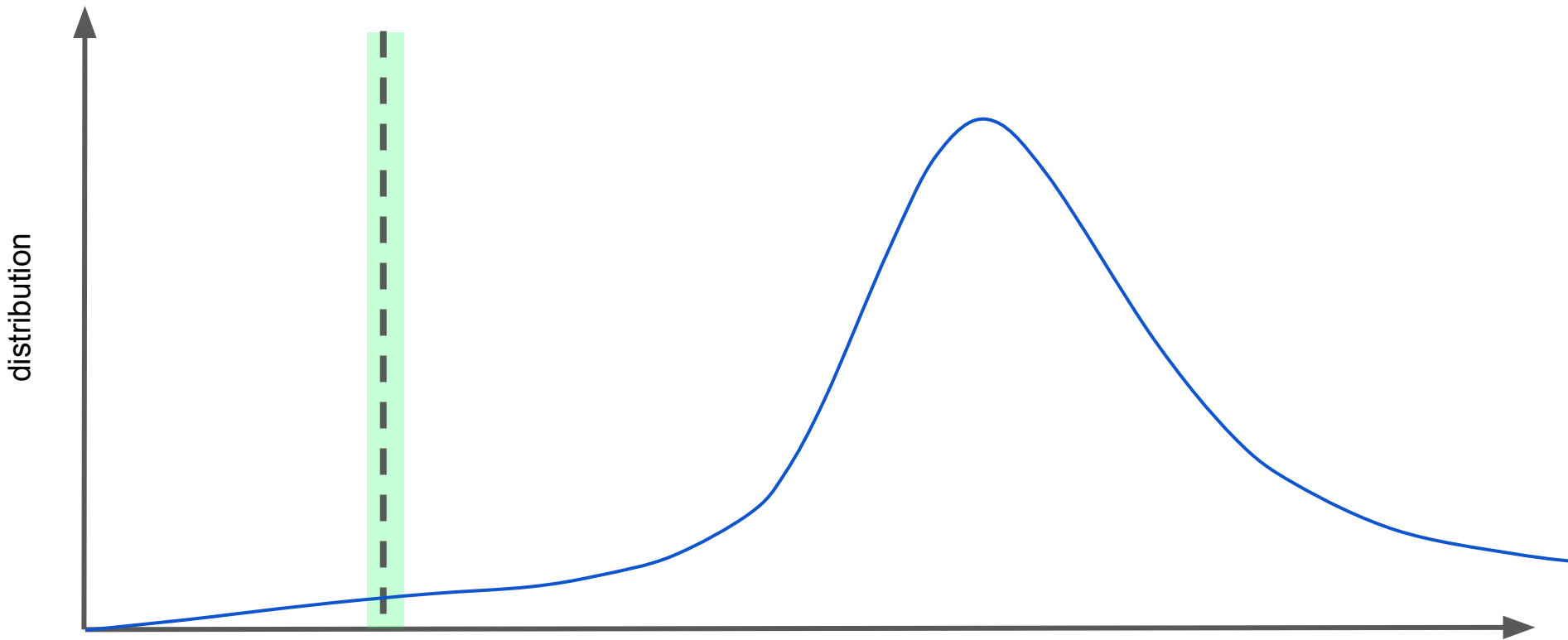






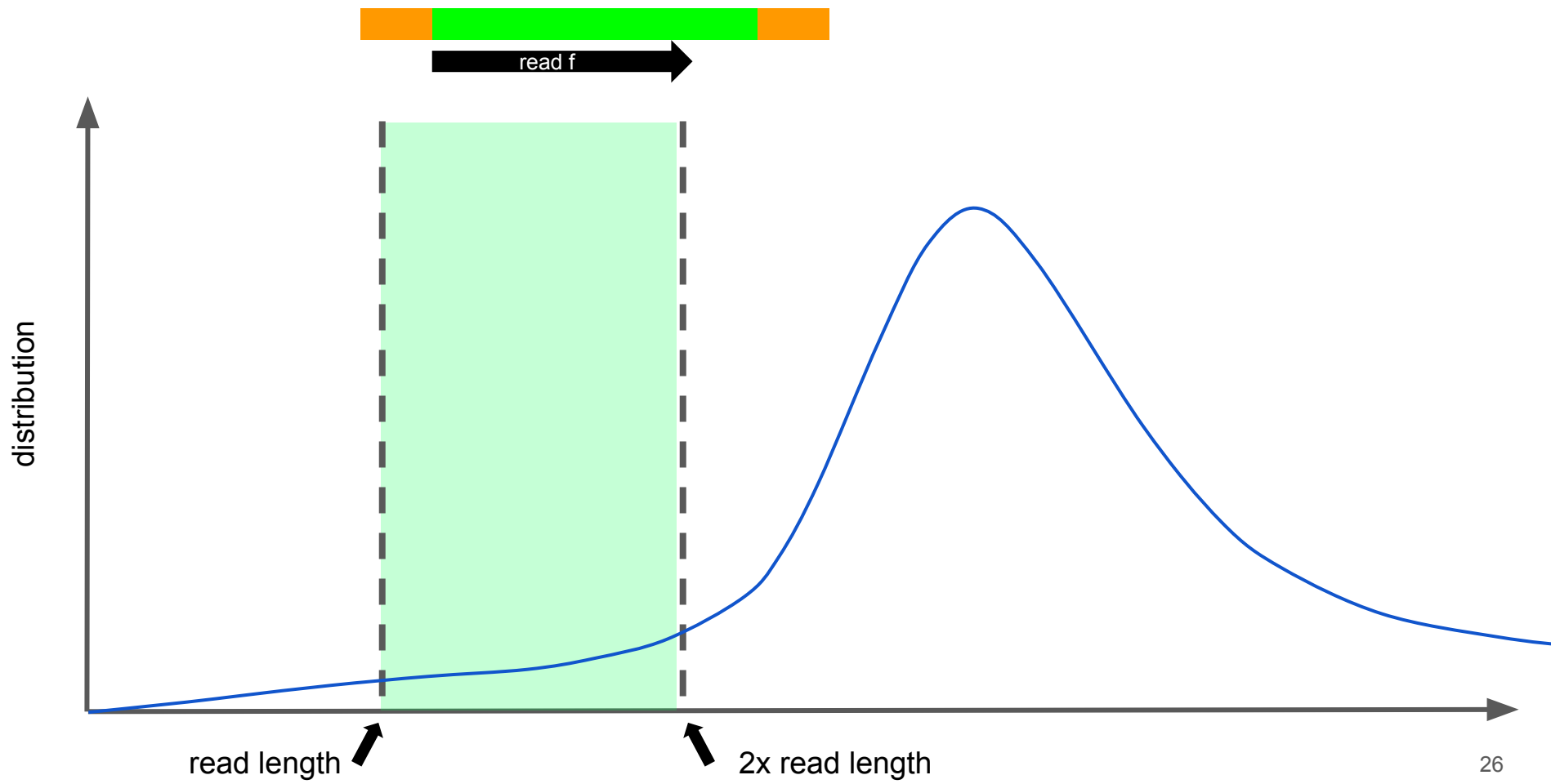


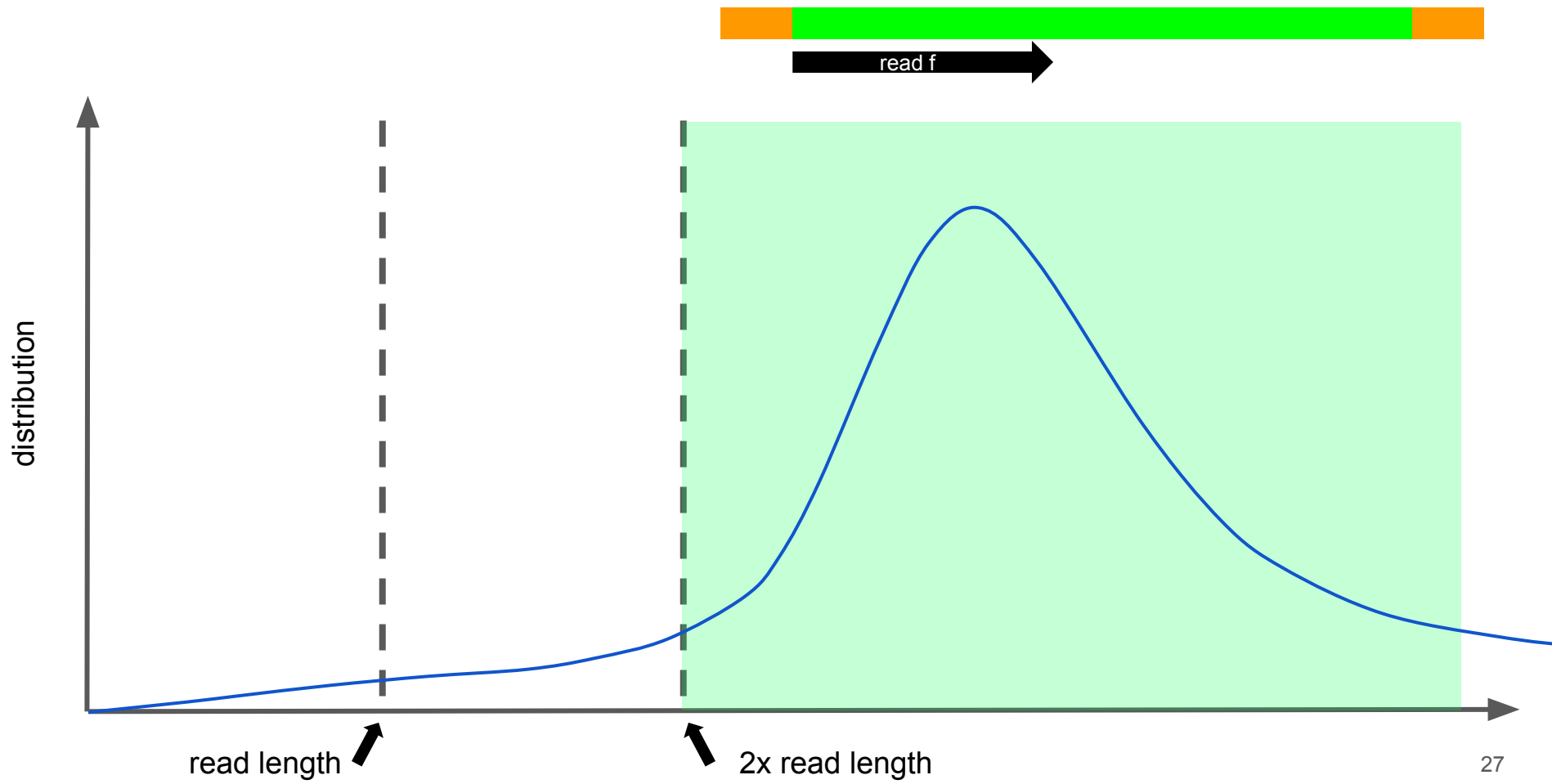
read f



read length







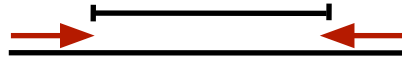
# Read orientation

Single end



Forward

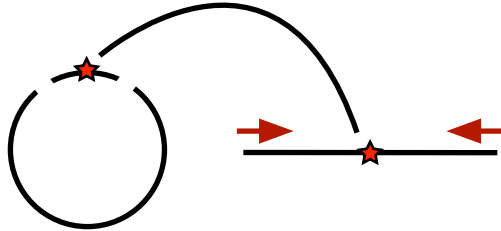
Paired end



Illumina: Forward - Reverse



Mate pair



Illumina: Reverse - Forward



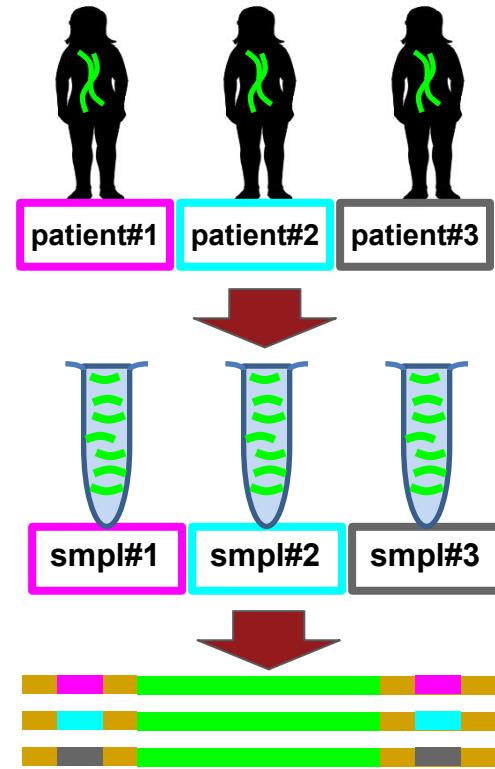
Different for other technologies!



# Multiplexing/Demultiplexing

example: 100M reads per patient is sufficient

idea: pool samples together to save costs

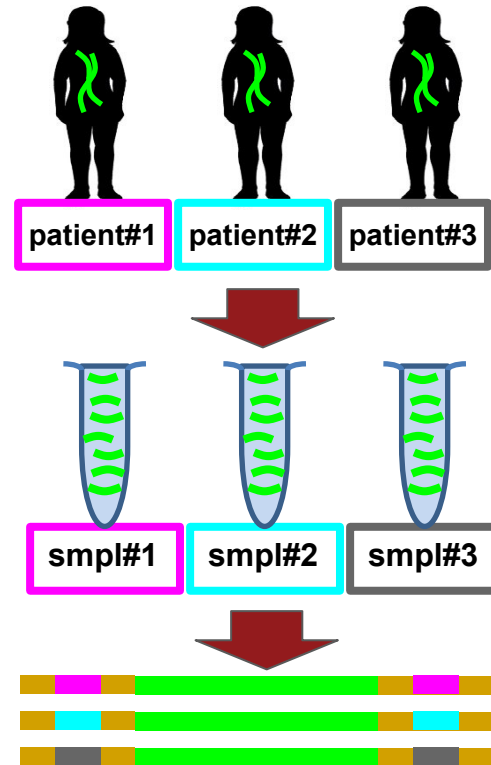


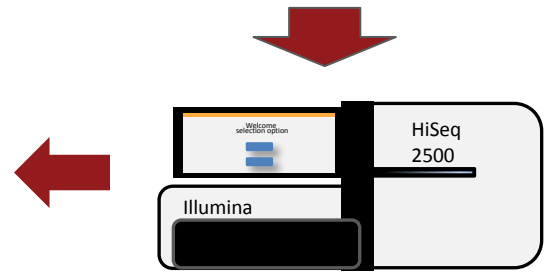
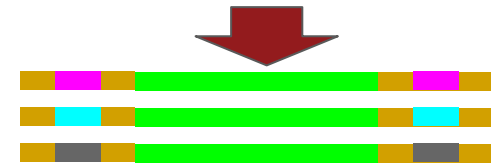
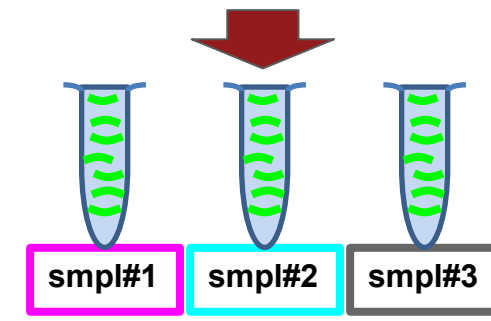
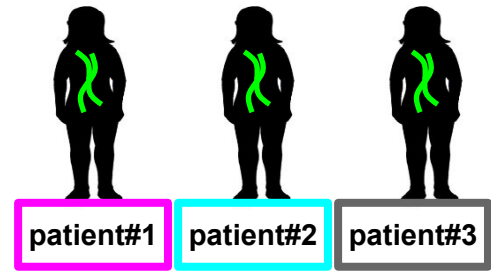
indices



TTCAGACTAGTTGGAAGTTTGTCT AGATCTC AGATTTTGTCACTAGAGGACGCACGCT  
C  
TTCAGACTAGTTGGAAGTTTGTCT TTCACTA AGATTTTGTCACTAGAGGACGCACGCT  
C  
TTCAGACTAGTTGGAAGTTTGTCT GACTAGT AGATTTTGTCACTAGAGGACGCACGCT  
C

Multiplexing: in the wet lab

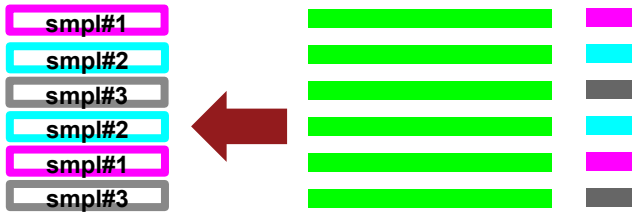
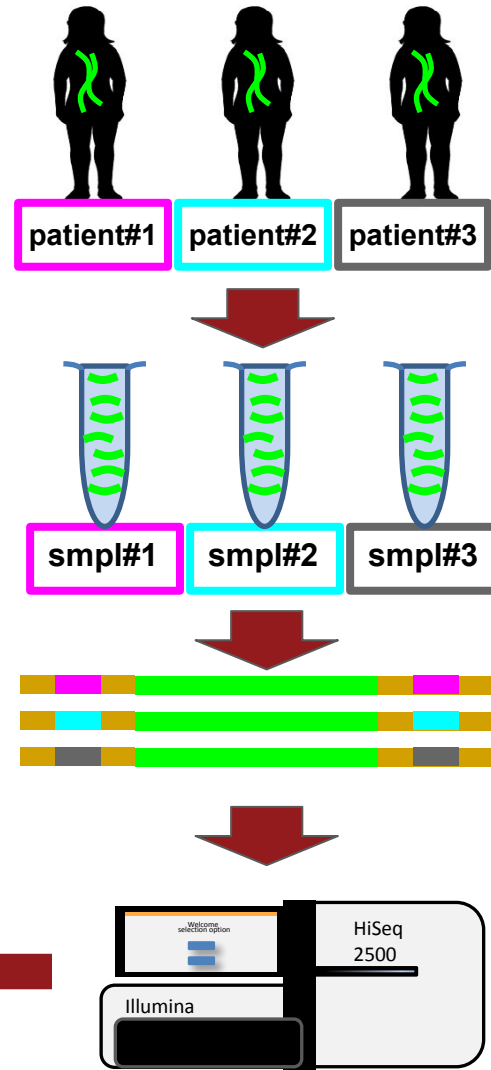




Basecalling



# Demultiplexing: computational



**Exercise time!**