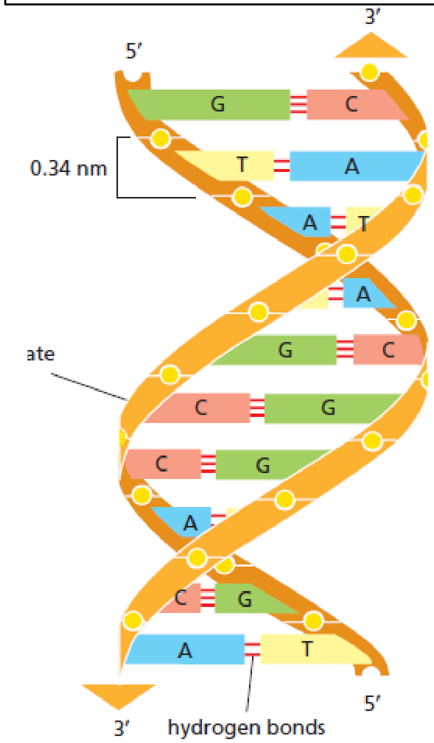
Актуальные проблемы анализа данных NGS-секвенирования

Анна Аксенова, к.б.н.

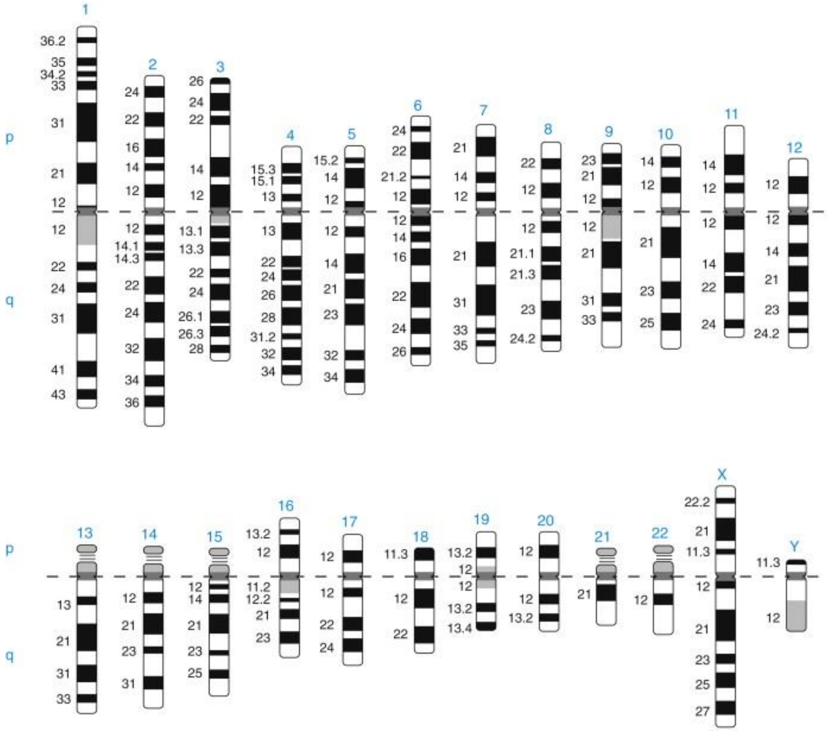
Human genome

Four-letter alphabet:
A, T, G, C
Double helix:
Forward and reverse strands
(**two directions!!!**)



3.2 billion base pairs (~ 2 meters unpacked)

46 chromosomes:
22 pairs of autosomes and X, Y



Sequencing epoch

Next generation sequencing (NGS)

First generation

Second generation

Third generation

Sanger sequencing
Maxam and Gilbert
Sanger chain termination

Infer nucleotide identity using dNTPs,
then visualize with electrophoresis

500–1,000 bp fragments

454, Solexa,
Ion Torrent,
Illumina

High throughput from the
parallelization of sequencing reactions

~50–500 bp fragments

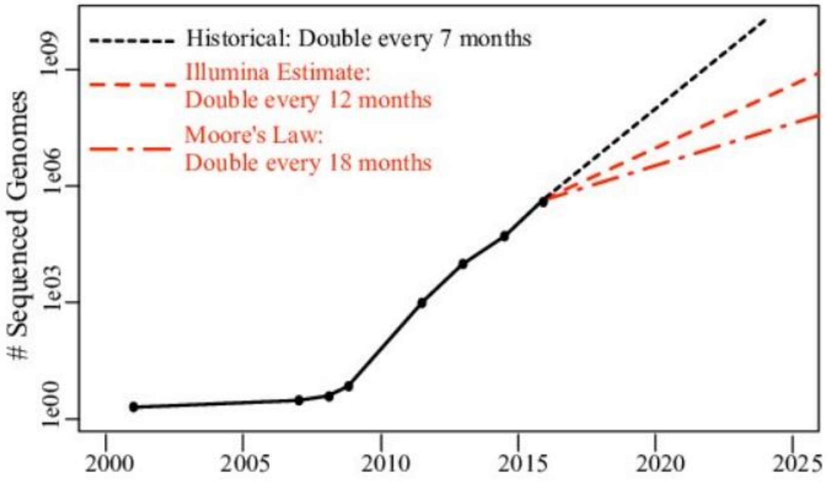
PacBio
Oxford Nanopore

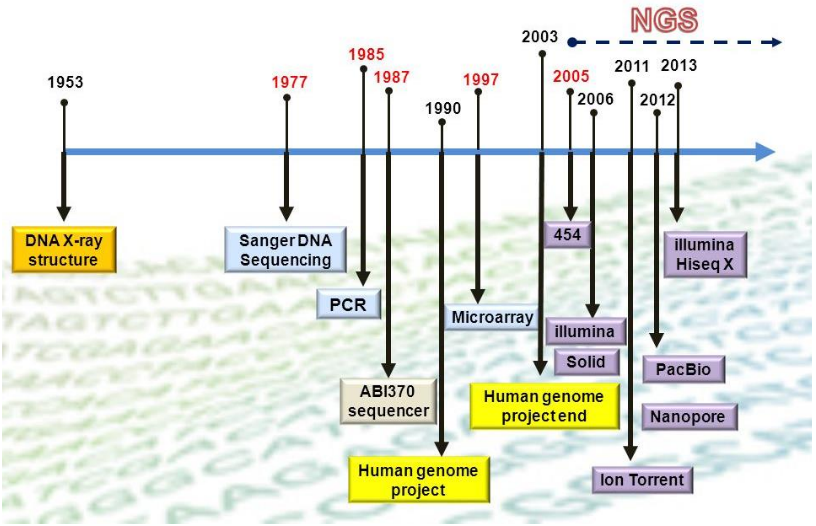
Sequence native DNA in real time
with single-molecule resolution

Tens of kb fragments, on average

Short-read sequencing

Long-read sequencing



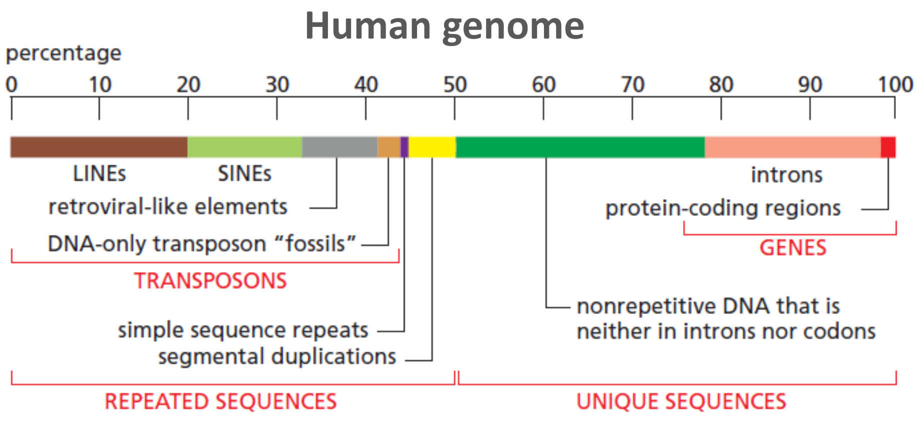


https://www.pacb.com/blog/the-evolution-of-dna-sequencing-tools/

What do we sequence?

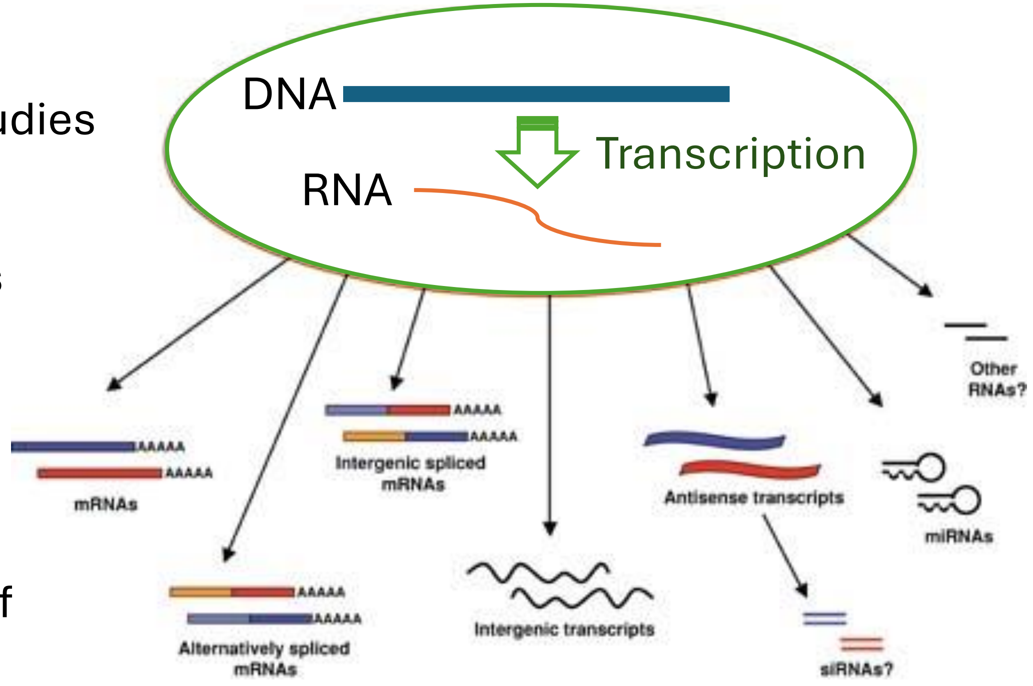
DNA

- Whole genome studies
- Whole exome studies
- Targeted panels
- Analyze epigenetic modifications
- Analyze chromatin
- Analyze 3D genome structure



RNA (cDNA)

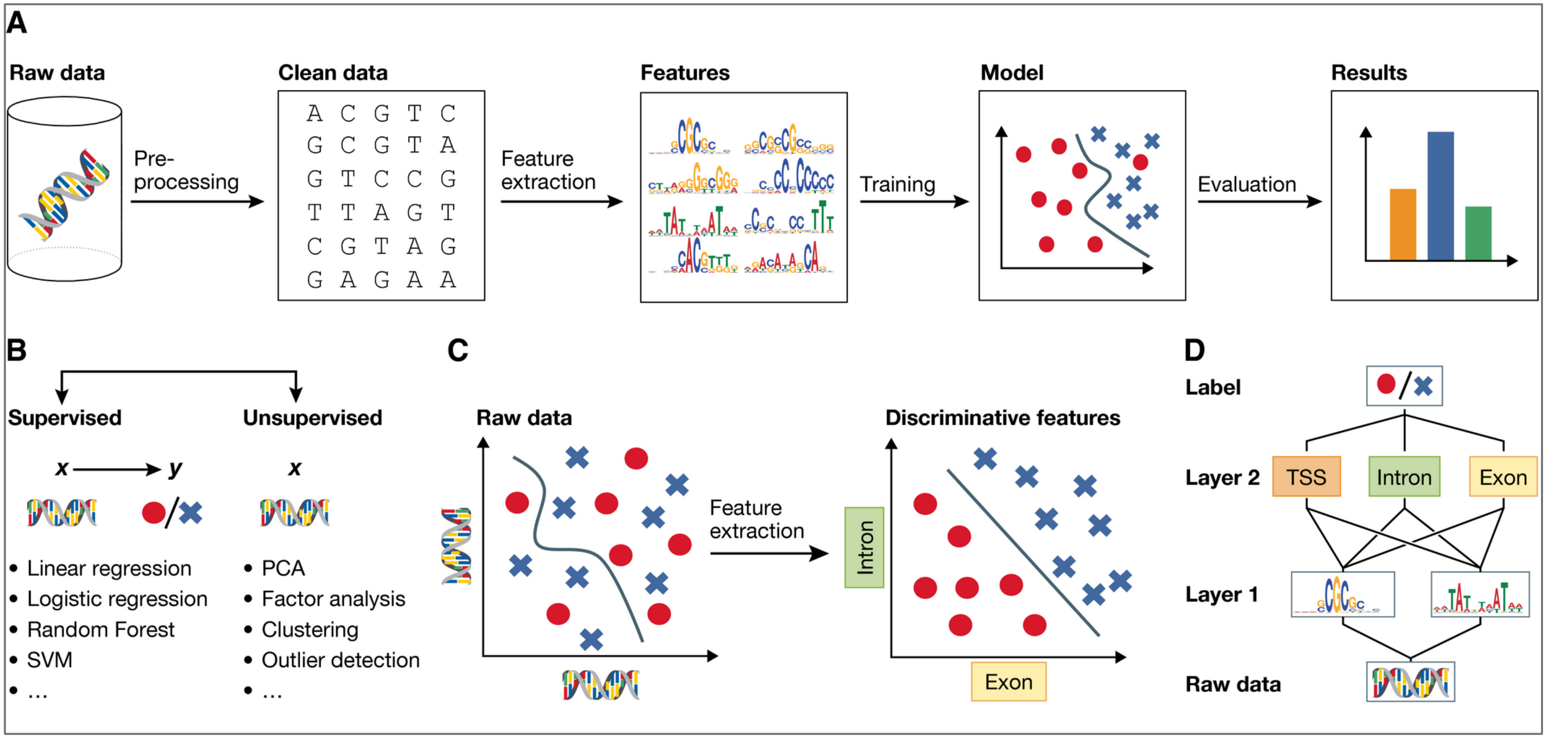
- Whole transcriptome studies
- mRNA studies
- Non-coding RNA studies
- Targeted RNA studies

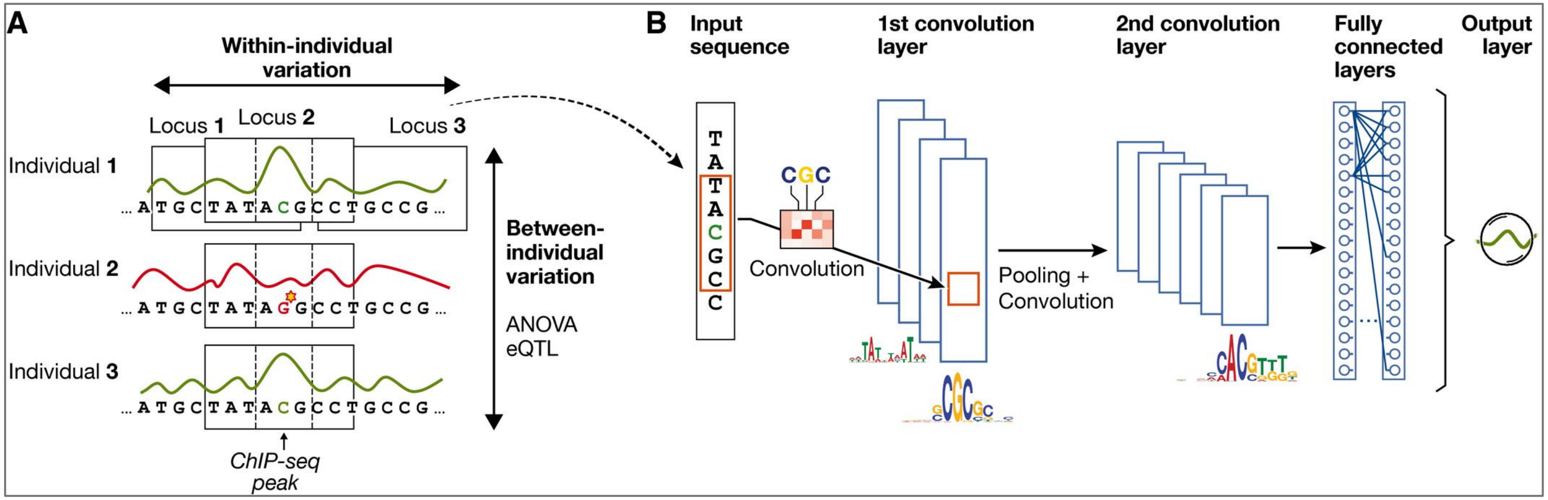


Single-cell DNA and RNA

- Various studies enabling understanding variability of individual cells

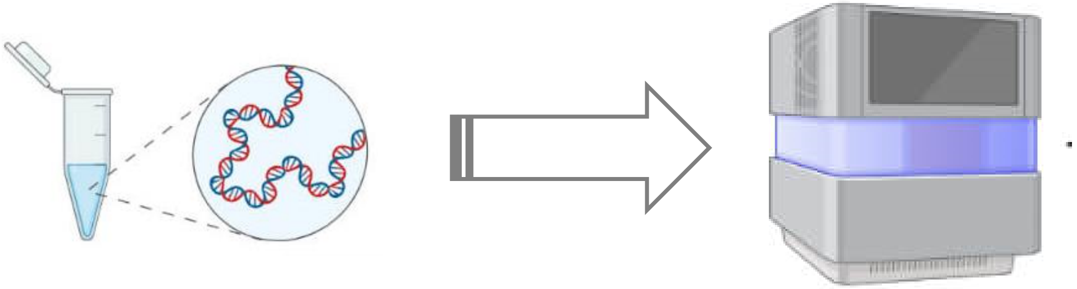
A new routine…



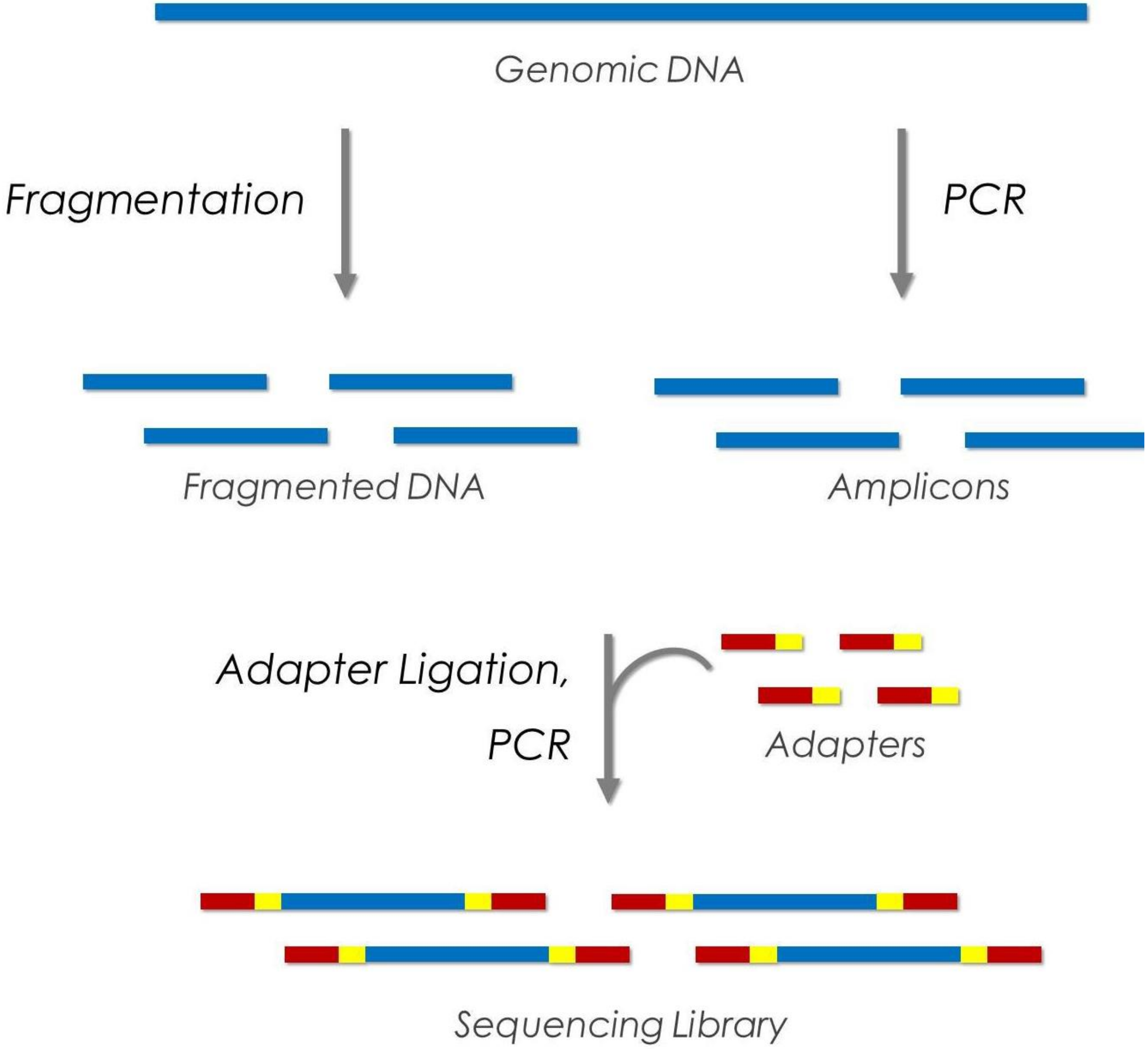


https://doi.org/10.15252/msb.20156651

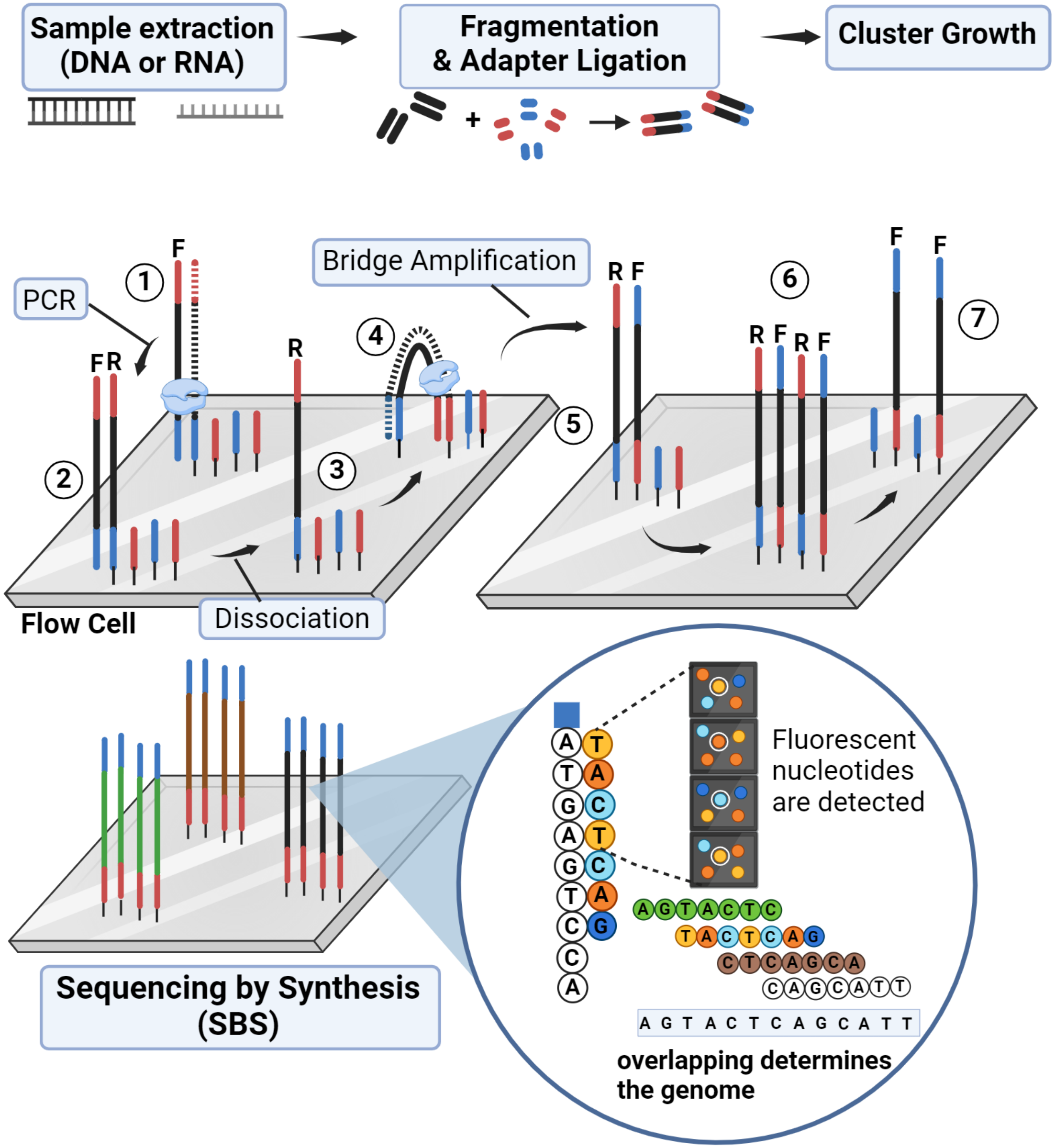
Few slides about technology…



How do we sequence DNA (technology)



NGS Sequencing technology (Illumina platform)



Sample extraction (DNA or RNA) → Fragmentation & Adapter Ligation → Cluster Growth

PCR

Bridge Amplification

Flow Cell

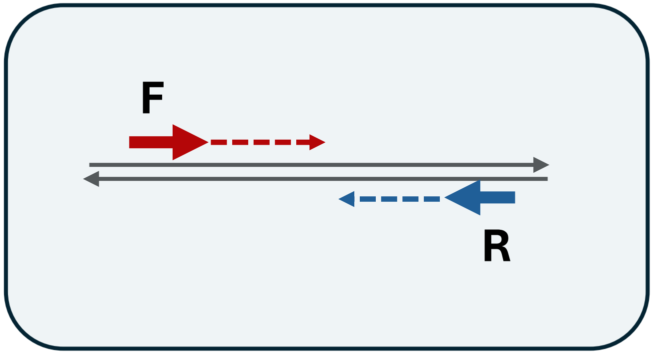
Dissociation

Sequencing by Synthesis (SBS)

Fluorescent nucleotides are detected

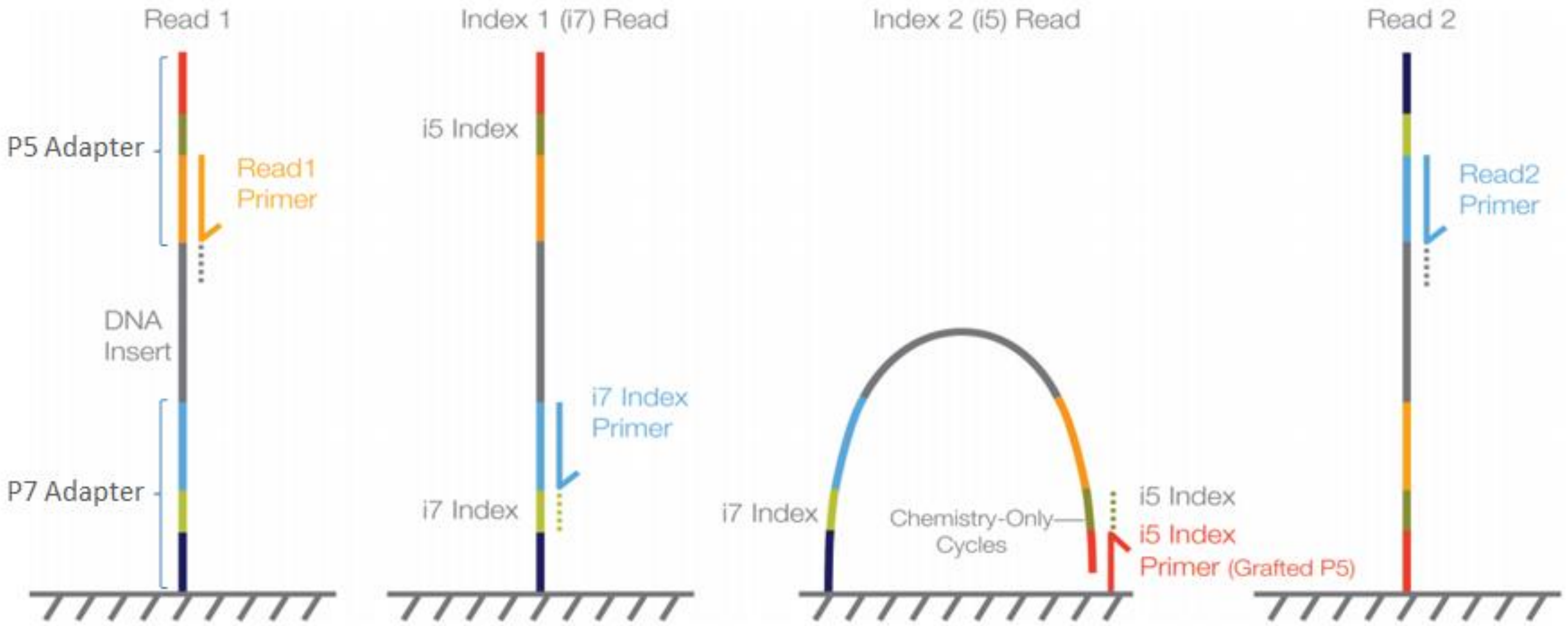
overlapping determines the genome

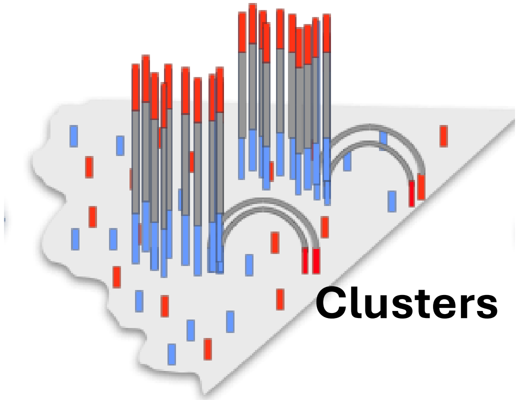
Single-end vs. Paired-end sequencing

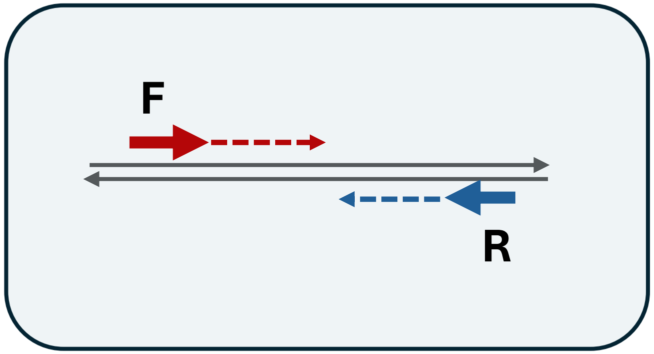


Zixin Xu/Biorender

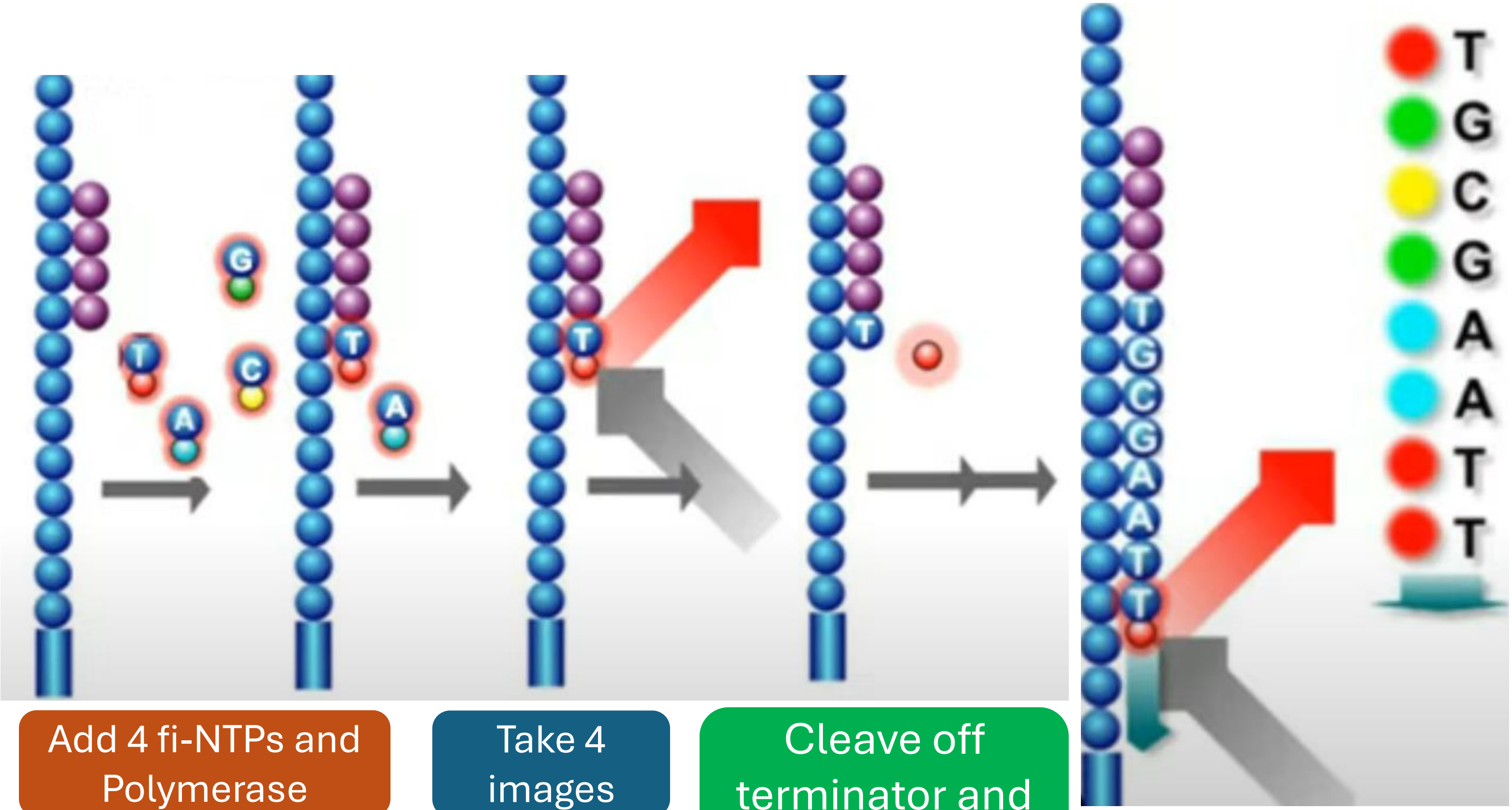
Paired-end sequencing technology (Illumina platform)







Four channel chemistry in NGS sequencing



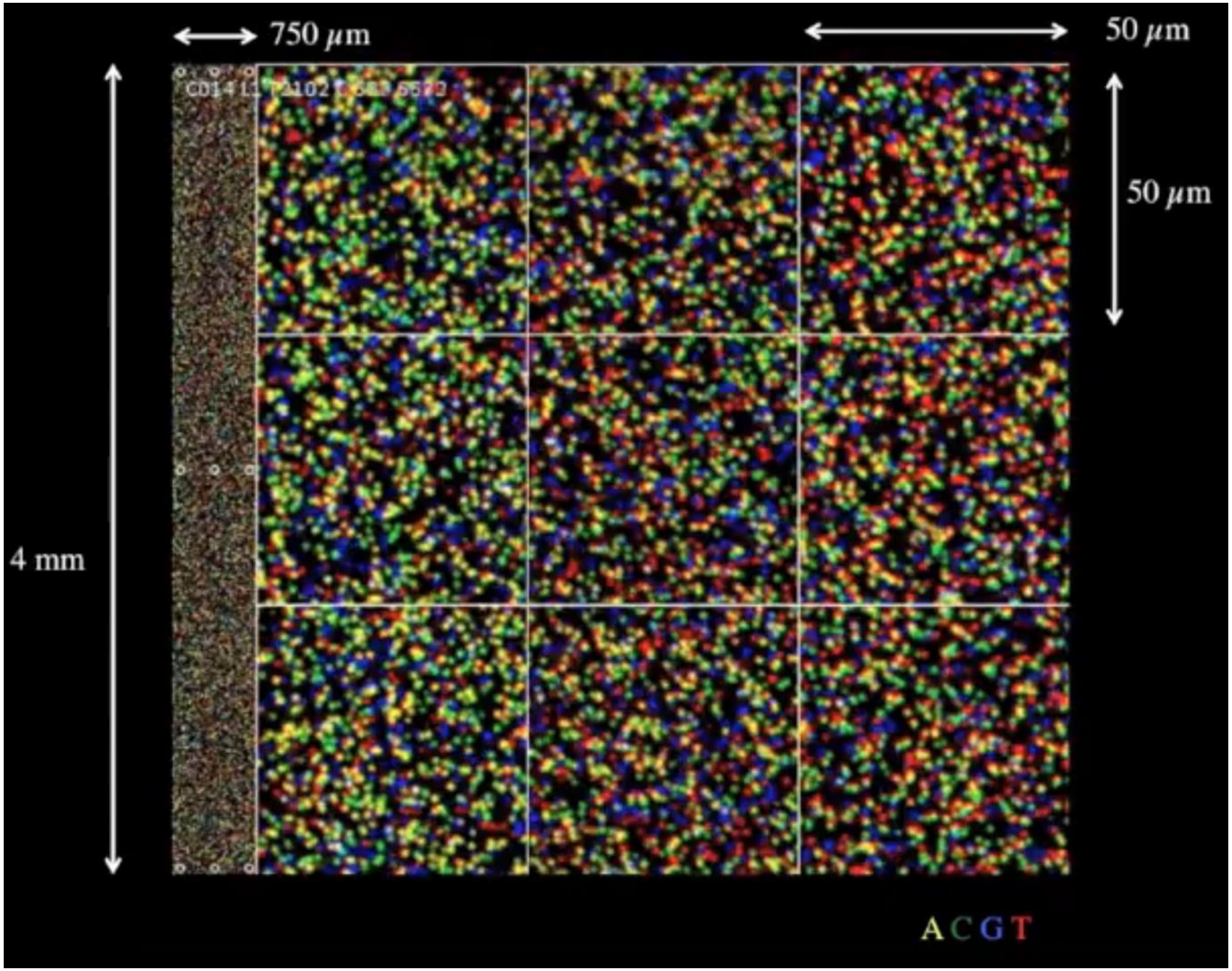
Add 4 fi-NTPs and Polymerase

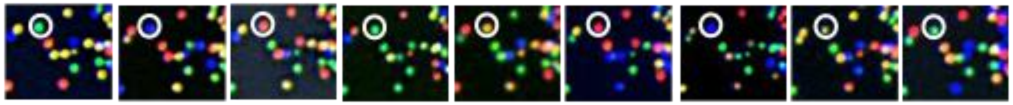
Take 4 images

Cleave off terminator and fluorescent dye

https://www.youtube.com/watch?v=oIJaA6h2bFM

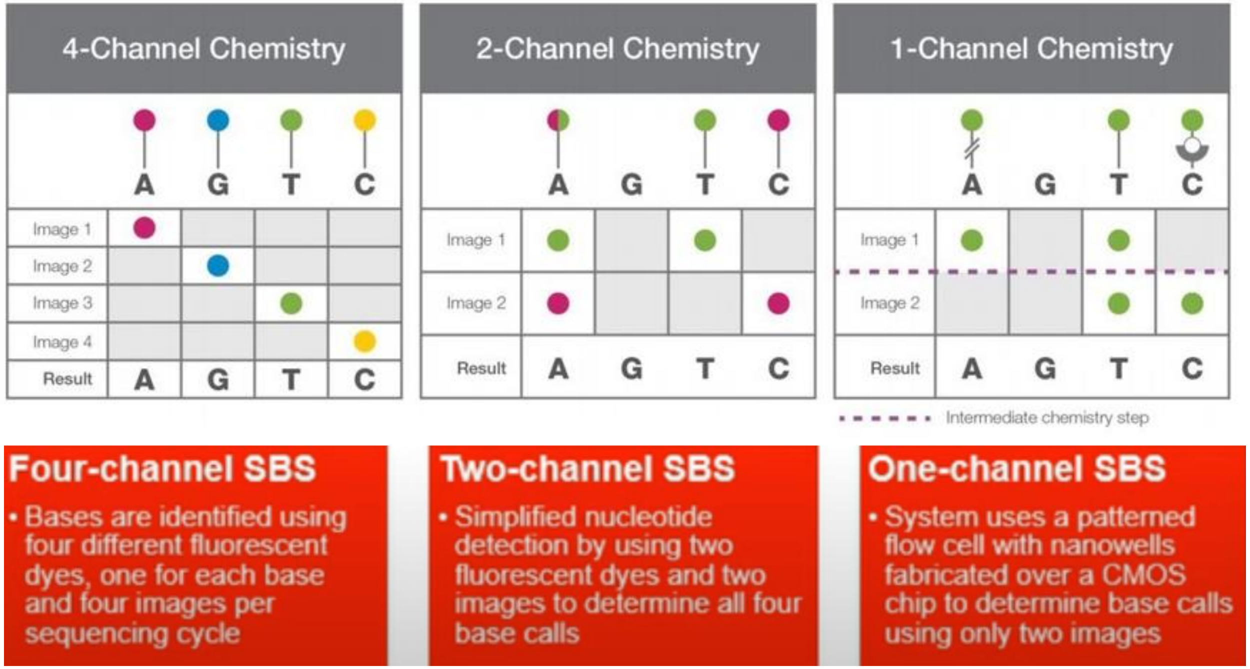
Illumina four-color sequencing by synthesis





https://www.youtube.com/watch?v=tuD-ST5B3QA

Different chemistry and different cells = different errors



4-Channel Chemistry

	A	G	T	C
Image 1	🔴			
Image 2		🔵		
Image 3			🟢	
Image 4				🟡
Result	A	G	T	C

2-Channel Chemistry

	A	G	T	C
Image 1	🟢		🟢	
Image 2	🔴			🔴
Result	A	G	T	C

1-Channel Chemistry

	A	G	T	C
Image 1	🟢		🟢	
Image 2			🟢	🟢
Result	A	G	T	C

Intermediate chemistry step

Four-channel SBS

- Bases are identified using four different fluorescent dyes, one for each base and four images per sequencing cycle

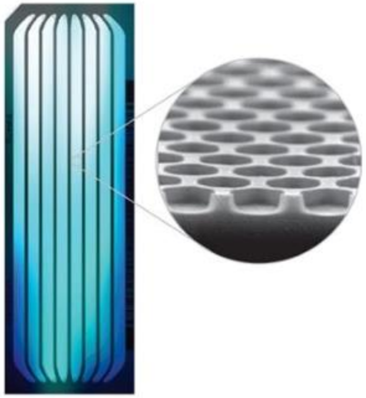
Two-channel SBS

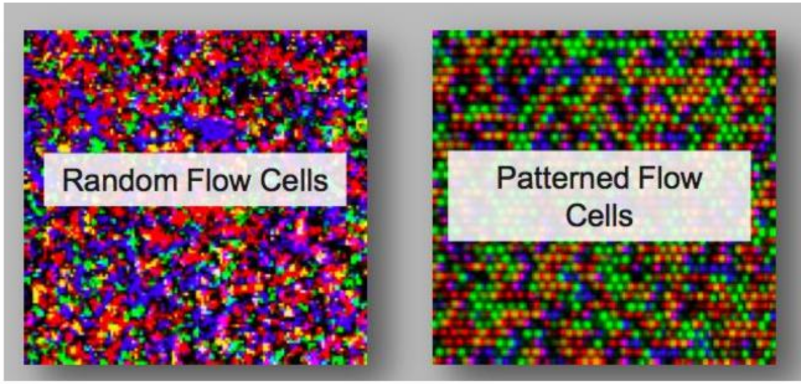
- Simplified nucleotide detection by using two fluorescent dyes and two images to determine all four base calls

One-channel SBS

- System uses a patterned flow cell with nanowells fabricated over a CMOS chip to determine base calls using only two images

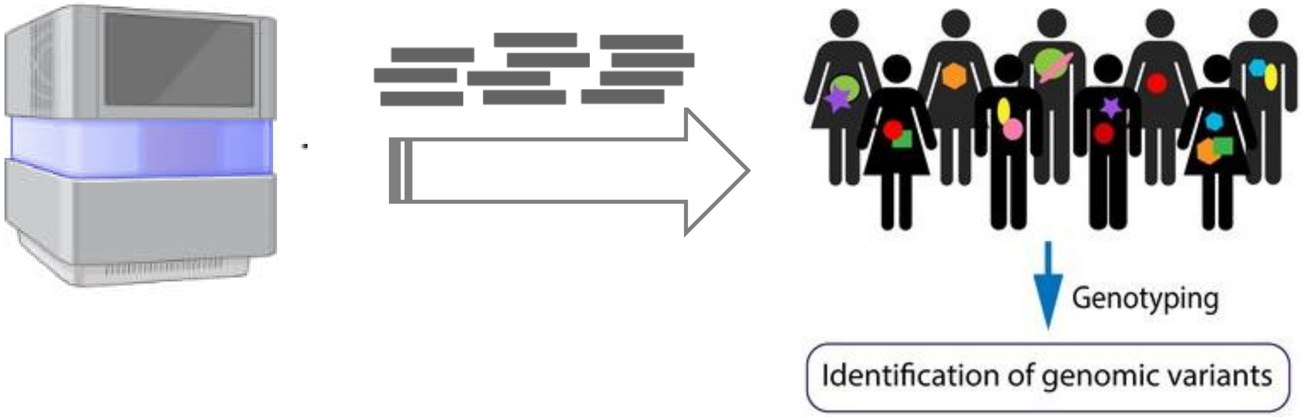
Quality?





And there are other technologies too…

How do we make sense of the reads?

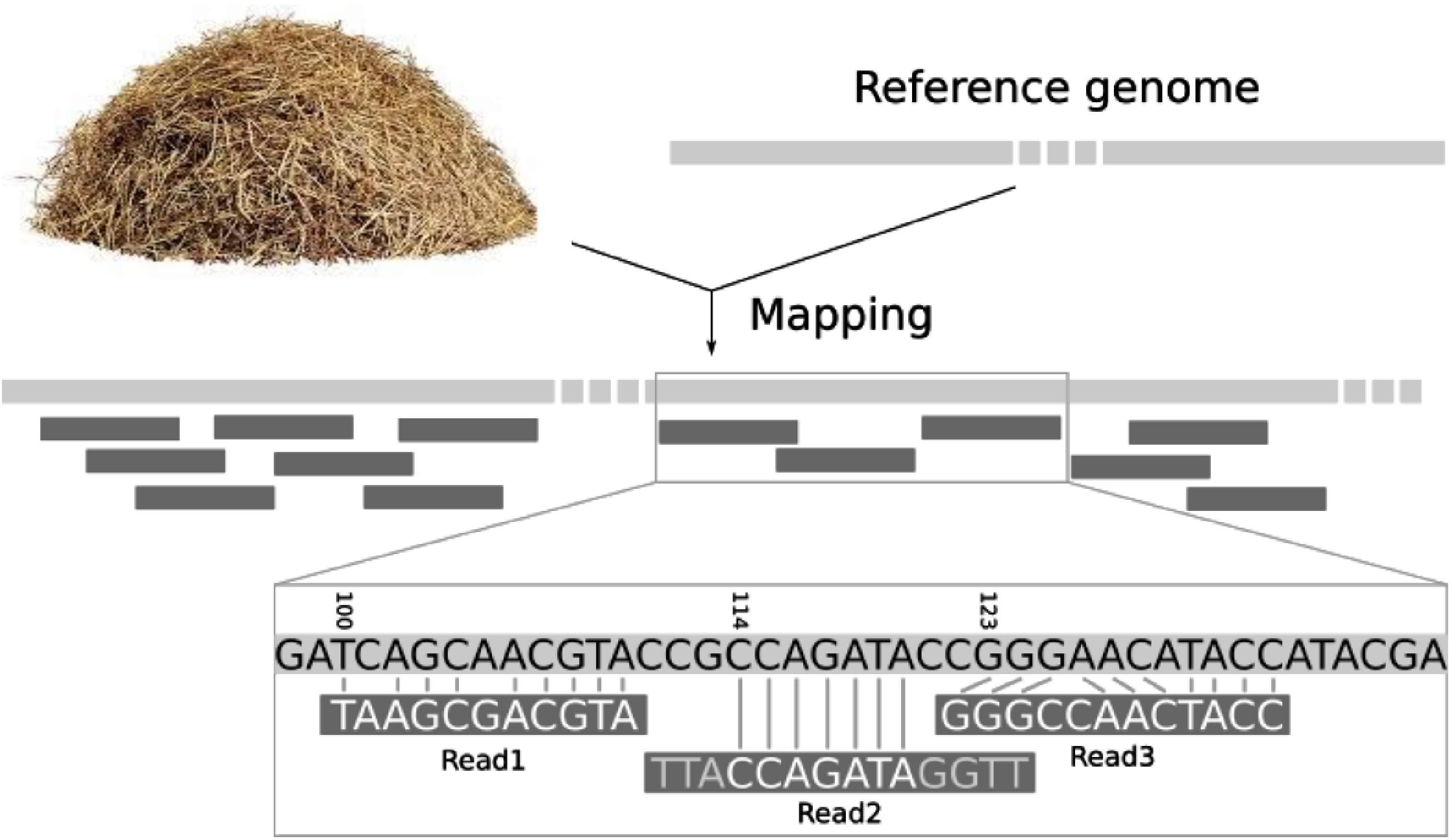


Genotyping

Identification of genomic variants

The task of mapping…

Hundreds billions of reads
(100-150 length each, raw)



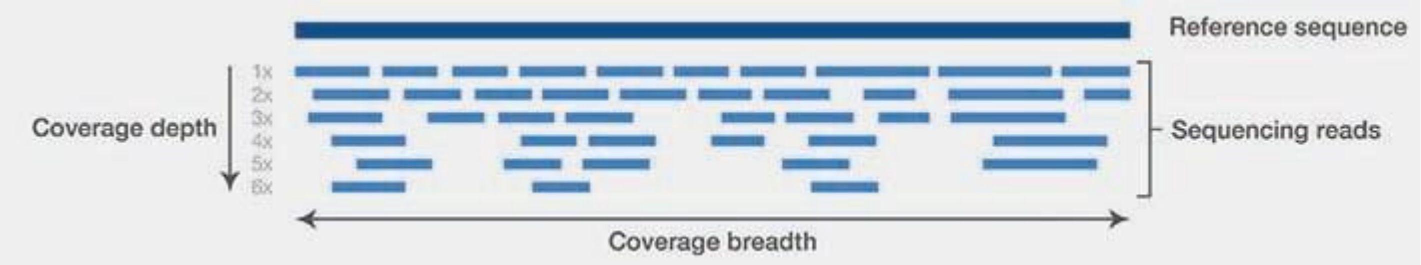
Finding the best position for every read in the reference string

Reads

Mapping
Alignment

Dynamic programming (usually)

BAM
SAM



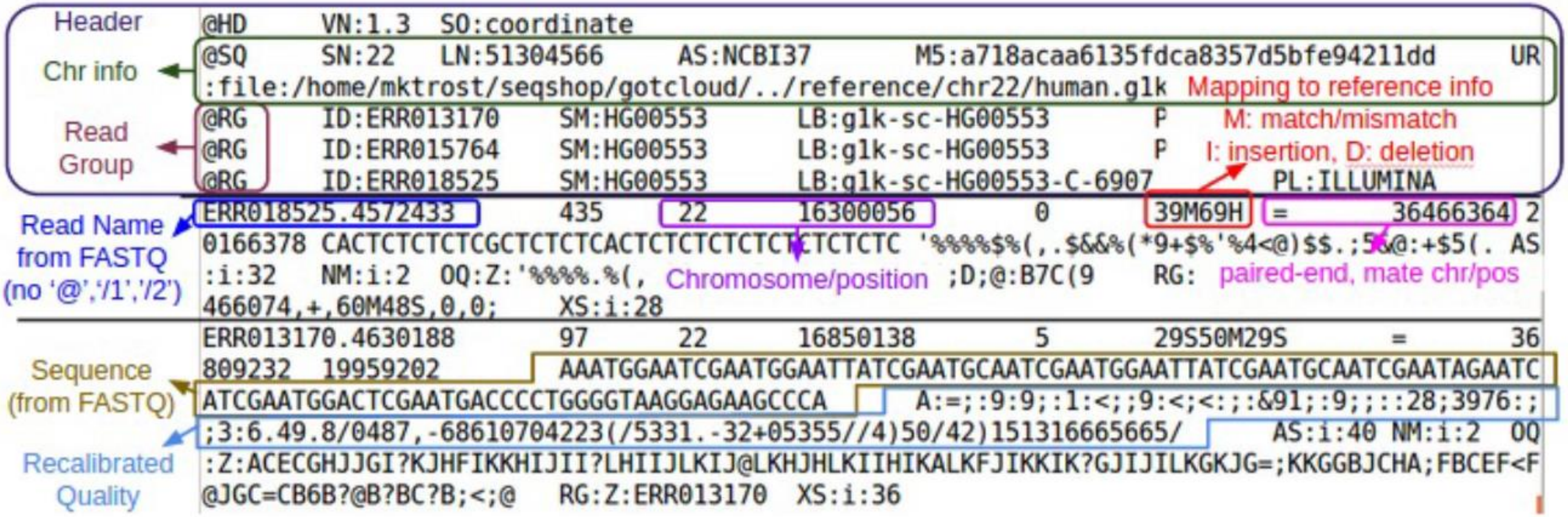
Genotyping

Interpretation

Coverage fig from: https://www.thermofisher.com/es/es/home/life-science/cloning/cloning-learning-center/invitrogen-school-of-molecular-biology/next-generation-sequencing/ngs-data-analysis-illumina.html

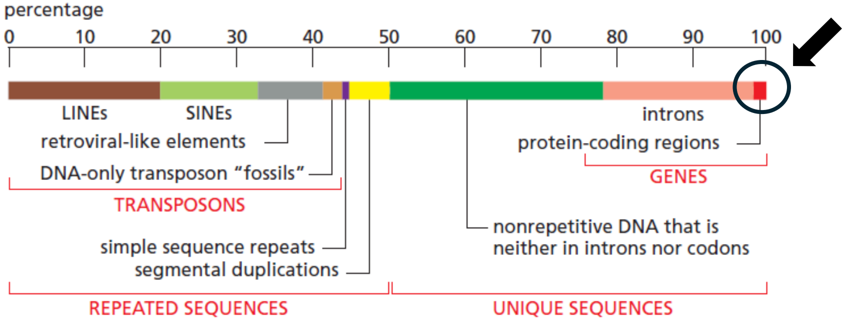
SAM file which tells us about fate of each read after alignment

Output for one read

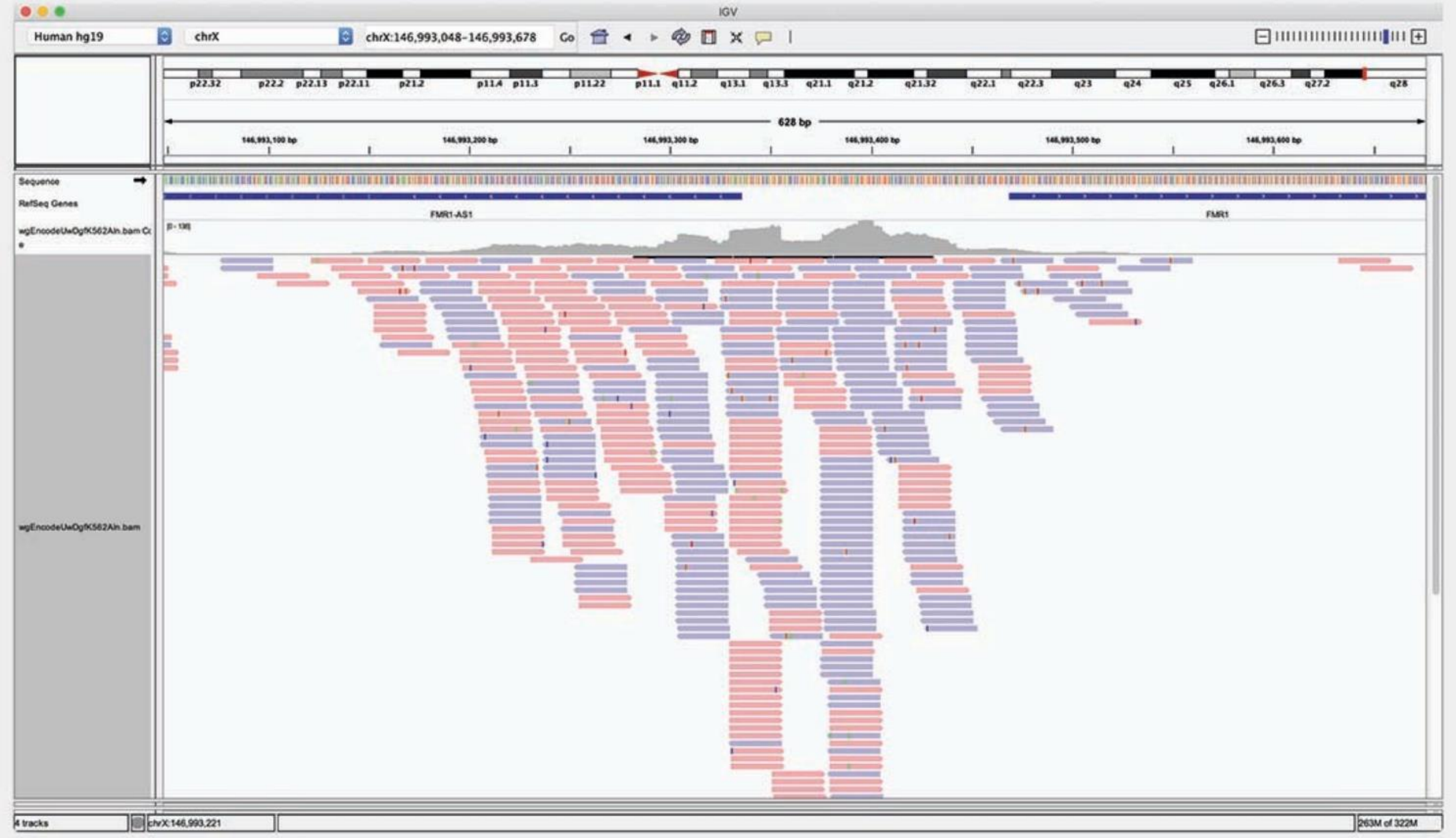


```
@HD     VN:1.3  SO:coordinate
@SQ     SN:22   LN:51304566     AS:NCBI37       M5:a718acaa6135fdca8357d5bfe94211dd     UR
:file:/home/mktrost/seqshop/gotcloud/../reference/chr22/human.g1k
@RG     ID:ERR013170    SM:HG00553      LB:g1k-sc-HG00553       P
@RG     ID:ERR015764    SM:HG00553      LB:g1k-sc-HG00553       P
@RG     ID:ERR018525    SM:HG00553      LB:g1k-sc-HG00553-C-6907        PL:ILLUMINA
ERR018525.4572433       435     22      16300056        0       39M69H  =       36466364 2
0166378 CACTCTCTCTCGCTCTCTCACTCTCTCTCTCTCTCTCTC '%%%%$%(,.$&&%(*9+$%'%4<@)$$.;5&@:+$5(. AS
:i:32   NM:i:2  OQ:Z:'%%%%.%(,      ;D;@:B7C(9      RG:
466074,+,60M48S,0,0;    XS:i:28
ERR013170.4630188       97      22      16850138        5       29S50M29S       =       36
809232  19959202        AAATGGAATCGAATGGAATTATCGAATGCAATCGAATGGAATTATCGAATGCAATCGAATAGAATC
ATCGAATGGACTCGAATGACCCCTGGGGTAAGGAGAAGCCCA      A:=;:9:9;:1:<;;9:<;<:;:&91;:9;;::28;3976:;
;3:6.49.8/0487,-68610704223(/5331.-32+05355//4)50/42)151316665665/      AS:i:40 NM:i:2  OQ
:Z:ACECGHJJGI?KJHFIKKHIJII?LHIIJLKIJ@LKHJHLKIIHIKALKFJIKKIK?GJIJILKGKJG=;KKGGBJCHA;FBCEF<F
@JGC=CB6B?@B?BC?B;<;@   RG:Z:ERR013170  XS:i:36
```

SAM files are often very big:100-500+ Gb
Even for exome

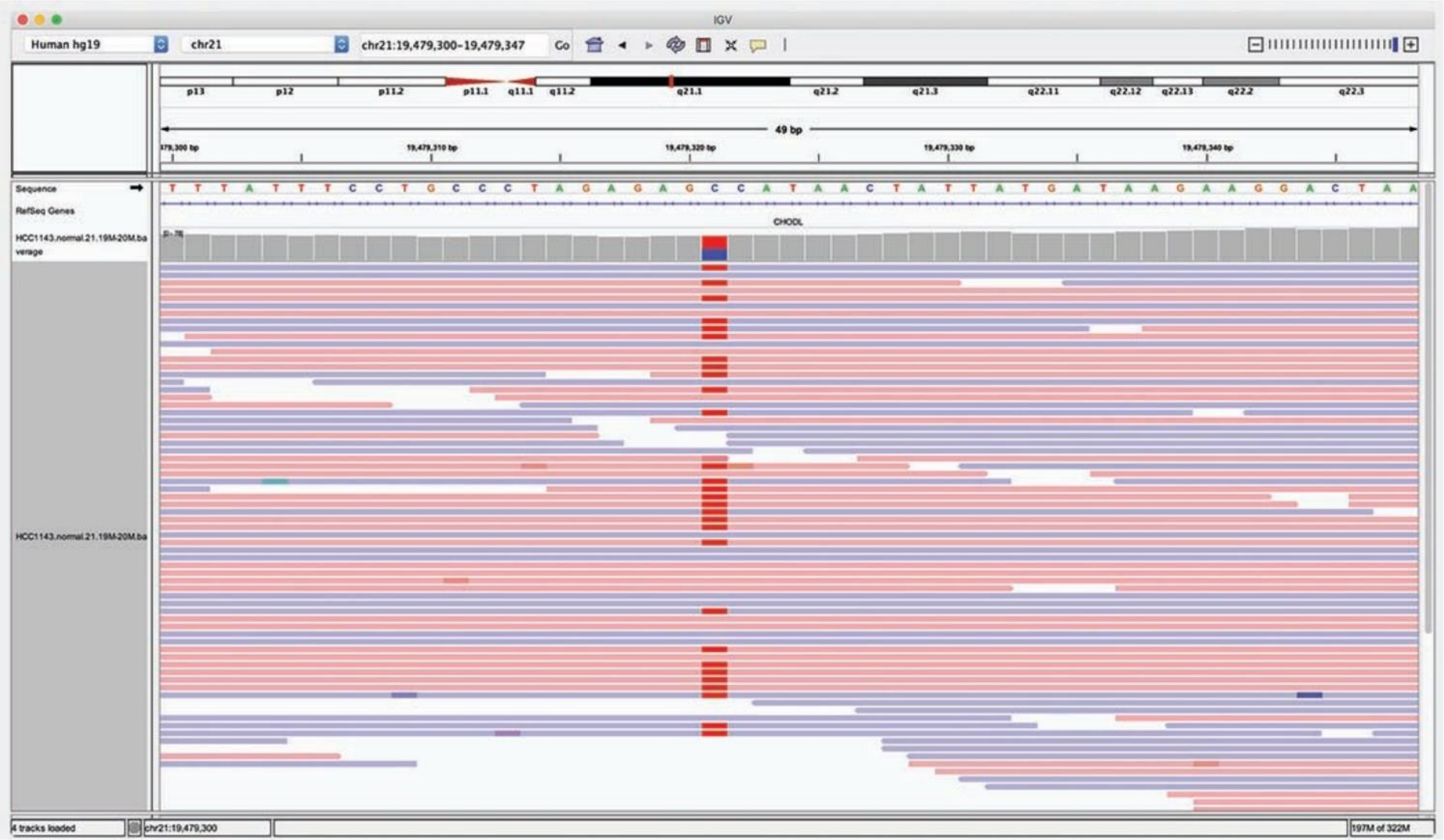


What do we get as a result of mapping?



IGV browser helps to view alignment results

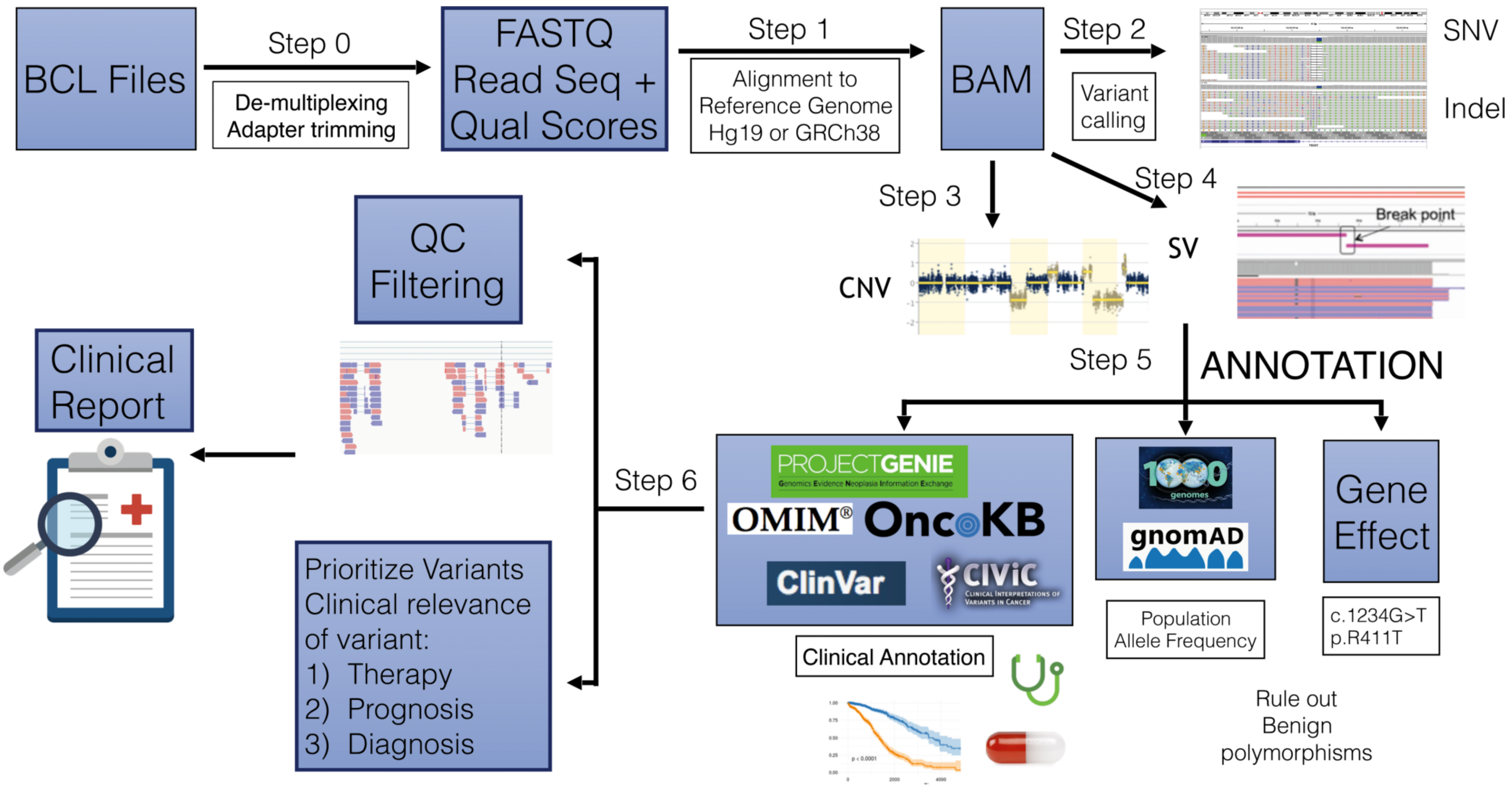
A closer look



IGV browser helps to view alignment results

Pipelines are important

Genome Analysis Toolkit (GATK)



BCL Files

Step 0

De-multiplexing
Adapter trimming

FASTQ
Read Seq +
Qual Scores

Step 1

Alignment to
Reference Genome
Hg19 or GRCh38

BAM

Step 2

Variant
calling

SNV

Indel

Step 3

CNV

Step 4

SV

Break point

Step 5

ANNOTATION

PROJECTGENIE
Genomics Evidence Neoplasia Information Exchange

OMIM®

OncoKB

ClinVar

CIViC
Clinical Interpretations of Variants in Cancer

Clinical Annotation

1000 genomes

gnomAD

Population
Allele Frequency

Gene
Effect

c.1234G>T
p.R411T

Rule out
Benign
polymorphisms

Step 6

QC
Filtering

Prioritize Variants
Clinical relevance
of variant:

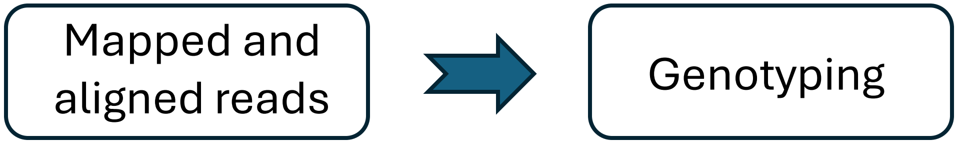
1) Therapy
2) Prognosis
3) Diagnosis

Clinical
Report

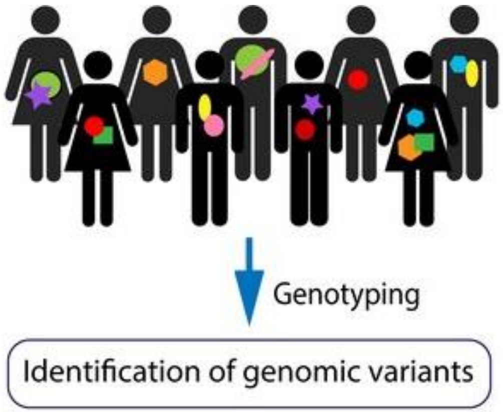
Arch Pathol Lab Med. 2020;144(9):1118-1130. doi:10.5858/arpa.2019-0476-RA

Genotyping results

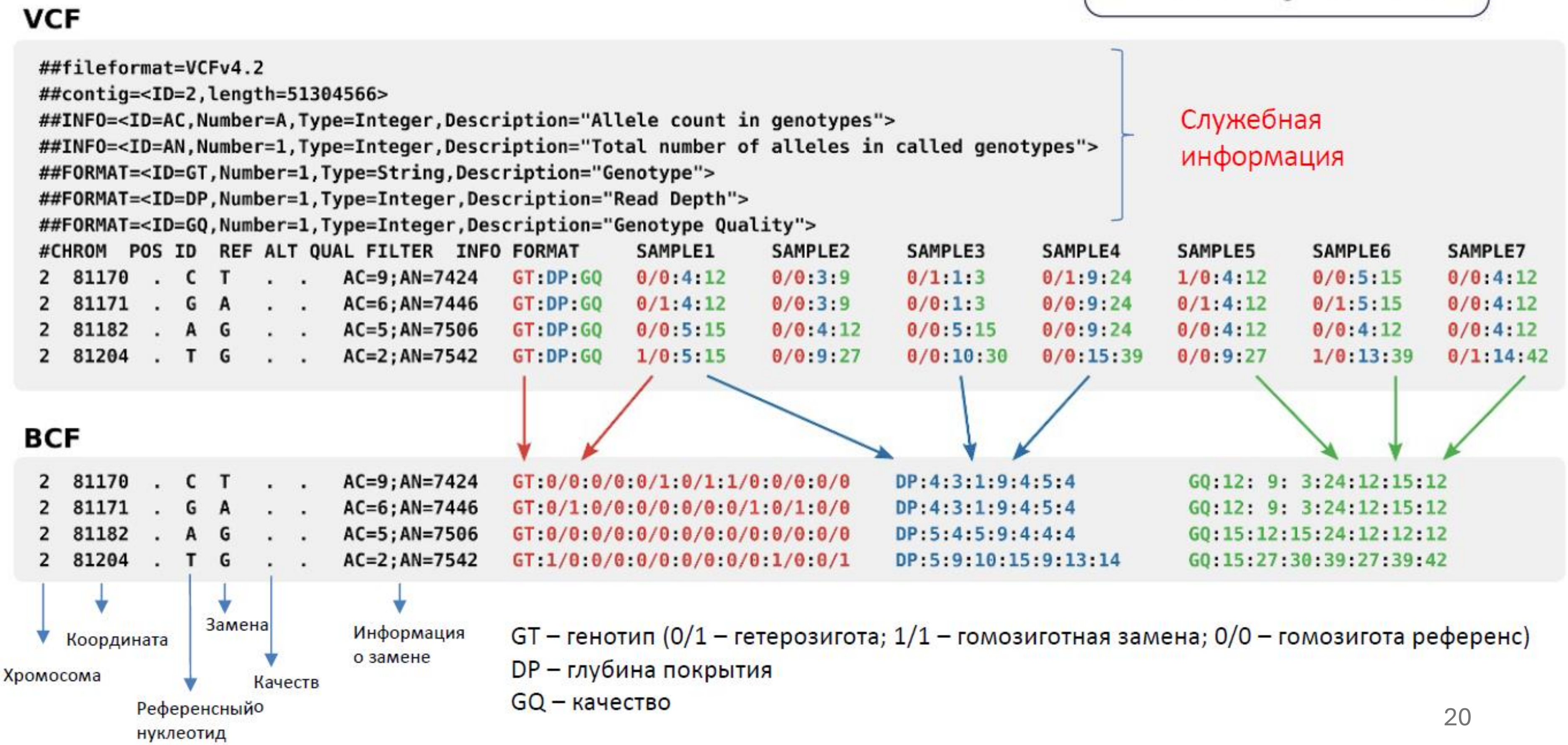
We need to map and align reads to learn about genotype


Mapped and aligned reads

Genotyping


Genotyping

Identification of genomic variants


VCF

```
##fileformat=VCFv4.2
##contig=<ID=2,length=51304566>
##INFO=<ID=AC,Number=A,Type=Integer,Description="Allele count in genotypes">
##INFO=<ID=AN,Number=1,Type=Integer,Description="Total number of alleles in called genotypes">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##FORMAT=<ID=DP,Number=1,Type=Integer,Description="Read Depth">
##FORMAT=<ID=GQ,Number=1,Type=Integer,Description="Genotype Quality">
#CHROM  POS ID  REF ALT QUAL FILTER  INFO FORMAT     SAMPLE1     SAMPLE2     SAMPLE3     SAMPLE4     SAMPLE5     SAMPLE6     SAMPLE7
2  81170  .  C  T   .  .   AC=9;AN=7424  GT:DP:GQ   0/0:4:12    0/0:3:9     0/1:1:3     0/1:9:24    1/0:4:12    0/0:5:15    0/0:4:12
2  81171  .  G  A   .  .   AC=6;AN=7446  GT:DP:GQ   0/1:4:12    0/0:3:9     0/0:1:3     0/0:9:24    0/1:4:12    0/1:5:15    0/0:4:12
2  81182  .  A  G   .  .   AC=5;AN=7506  GT:DP:GQ   0/0:5:15    0/0:4:12    0/0:5:15    0/0:9:24    0/0:4:12    0/0:4:12    0/0:4:12
2  81204  .  T  G   .  .   AC=2;AN=7542  GT:DP:GQ   1/0:5:15    0/0:9:27    0/0:10:30   0/0:15:39   0/0:9:27    1/0:13:39   0/1:14:42
```

Служебная информация

BCF

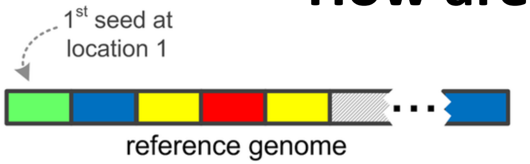
```
2  81170  .  C  T   .  .   AC=9;AN=7424  GT:0/0:0/0:0/1:0/1:1/0:0/0:0/0   DP:4:3:1:9:4:5:4      GQ:12: 9: 3:24:12:15:12
2  81171  .  G  A   .  .   AC=6;AN=7446  GT:0/1:0/0:0/0:0/0:0/1:0/1:0/0   DP:4:3:1:9:4:5:4      GQ:12: 9: 3:24:12:15:12
2  81182  .  A  G   .  .   AC=5;AN=7506  GT:0/0:0/0:0/0:0/0:0/0:0/0:0/0   DP:5:4:5:9:4:4:4      GQ:15:12:15:24:12:12:12
2  81204  .  T  G   .  .   AC=2;AN=7542  GT:1/0:0/0:0/0:0/0:0/0:1/0:0/1   DP:5:9:10:15:9:13:14  GQ:15:27:30:39:27:39:42
```

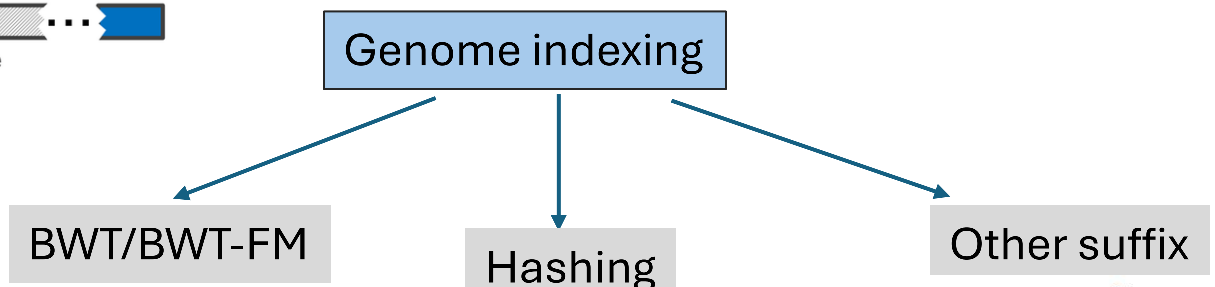
GT – генотип (0/1 – гетерозигота; 1/1 – гомозиготная замена; 0/0 – гомозигота референс)

DP – глубина покрытия

GQ – качество

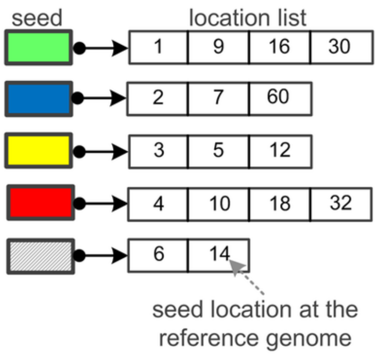
How are the reads mapped to the genome?

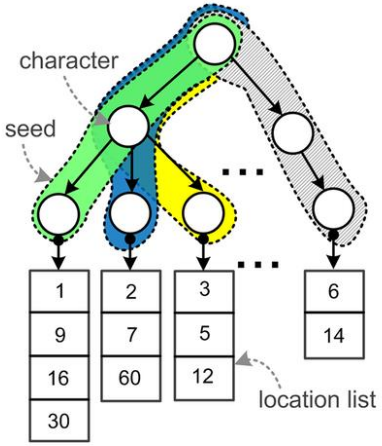


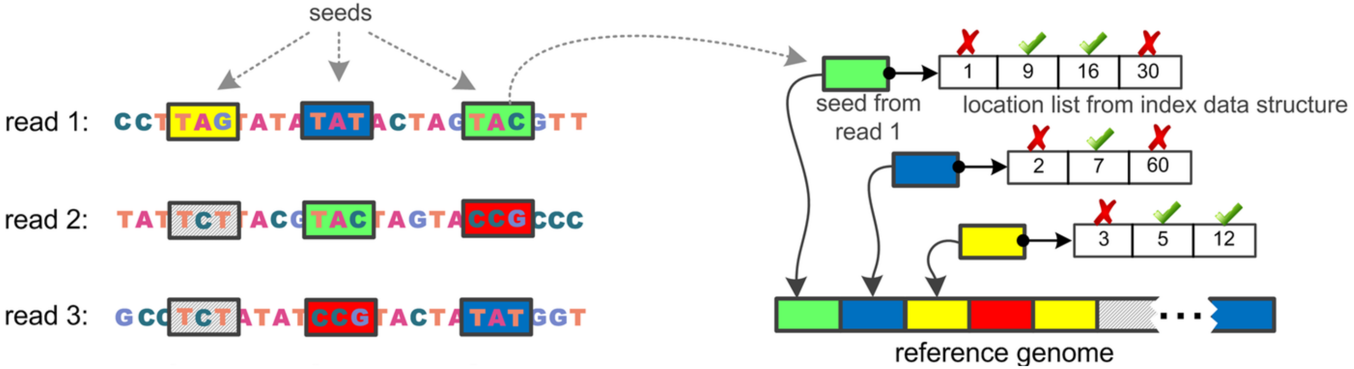


Compressed Suffix Arrays (CSA) (Grossi and Vitter)
FM-index (Ferragina and Manzini)

i	SA[i]	BWT[i]	O[a,i]				
			$	a	c	g	t
0	12	t	0	0	0	0	1
1	2	t	0	0	0	0	2
2	3	a	0	1	0	0	2
3	10	t	0	1	0	0	3
4	0	$	1	1	0	0	3
5	4	a	1	2	0	0	3
6	6	g	1	2	0	1	3
7	5	c	1	2	1	1	3
8	7	c	1	2	2	1	3
9	11	a	1	3	2	1	3
10	1	a	1	4	2	1	3
11	9	t	1	4	2	1	4
12	8	g	1	4	2	2	4
		C[a]	0	1	5	7	9






seeds
read 1: CCTTAGTATATATACTAGTACGTT
read 2: TATTCTTACGTACTAGTACCGCCC
read 3: GCCTCTATATCCGTACTATATGGT
seed from read 1
location list from index data structure
1 9 16 30
2 7 60
3 5 12
reference genome

Modified from: https://doi.org/10.1186/s13059-021-02443-7

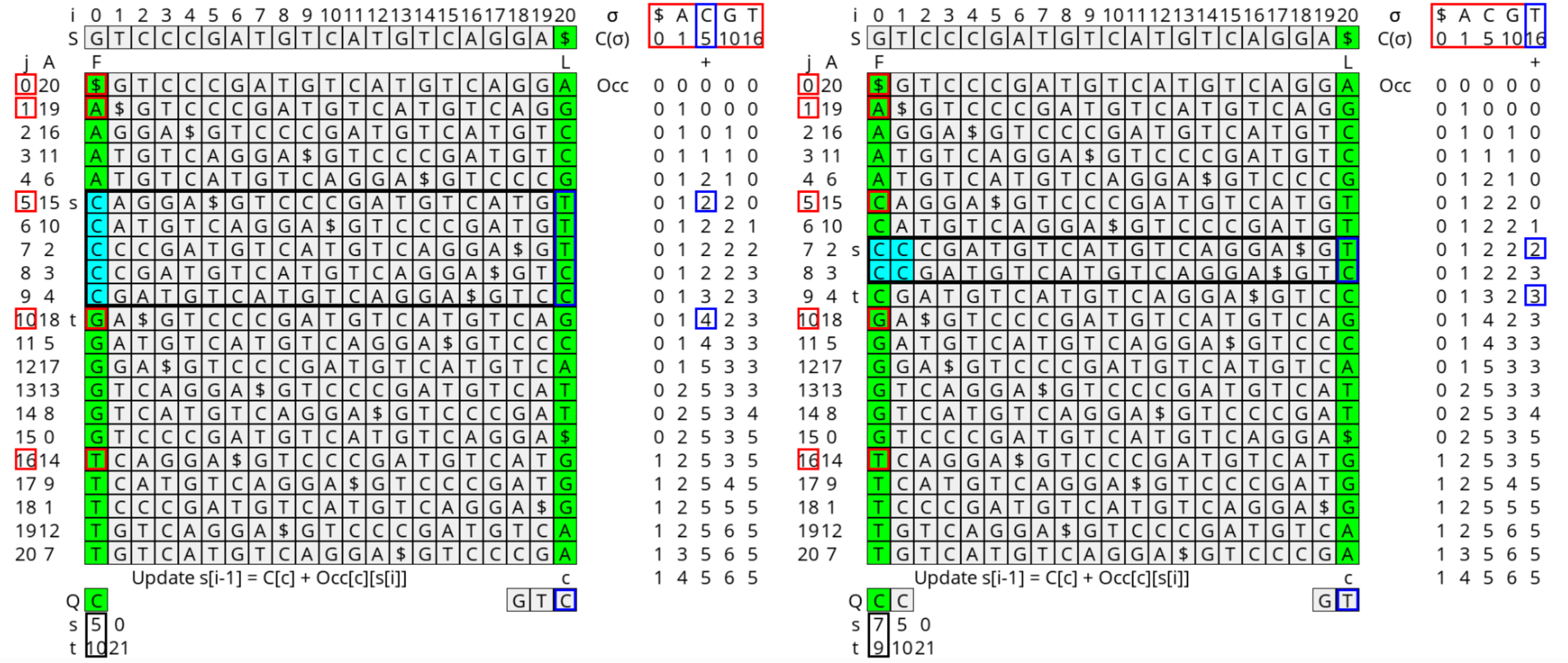
How are the seeds found in the reference?

The algorithm behind the calculation of seeds in *BWA-MEM* depends on the FM index, a data structure introduced by Ferragina and Manzini.

FM-index allows searching for any given pattern P in a collection of text in O(|P | log n + occ log2 n) and occupy O(n) bits

BWT/FM index utilizes the underlying properties of the *Burrows Wheeler Transform* introduced by Burrows and Wheeler and Last-to-first mapping

Table 1: *A step of the pattern matching algorithm. On the left, 2 characters (cc) have been processed to give the range from s_2=7 and t_2=9. The next character is c=s_1=T. There are C[T]=16 characters <T, 2 T's before the start of the interval, and 3 T's before the end of the interval. Thus, the new interval (shown on the right) is from s_1=16+2=18 to t_1=16+3=19.*



i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
S	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$

σ	$	A	C	G	T
C(σ)	0	1	5	10	16
			+		

j	A		F																				L	Occ				
0	20		$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	0	0	0	0	0
1	19		A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	0	1	0	0	0
2	16		A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	0	1	0	1	0
3	11		A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	0	1	1	1	0
4	6		A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	0	1	2	1	0
5	15	s	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	0	1	2	2	0
6	10		C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	0	1	2	2	1
7	2		C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	0	1	2	2	2
8	3		C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	0	1	2	2	3
9	4		C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	0	1	3	2	3
10	18	t	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	0	1	4	2	3
11	5		G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	0	1	4	3	3
12	17		G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	0	1	5	3	3
13	13		G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	0	2	5	3	3
14	8		G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	0	2	5	3	4
15	0		G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	0	2	5	3	5
16	14		T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	1	2	5	3	5
17	9		T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	1	2	5	4	5
18	1		T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	1	2	5	5	5
19	12		T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	1	2	5	6	5
20	7		T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	1	3	5	6	5
																								1	4	5	6	5

Update s[i-1] = C[c] + Occ[c][s[i]]   c

Q C   G T C

s 5 0

t 10 21

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
S	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$

σ	$	A	C	G	T
C(σ)	0	1	5	10	16
					+

j	A		F																				L	Occ				
0	20		$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	0	0	0	0	0
1	19		A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	0	1	0	0	0
2	16		A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	0	1	0	1	0
3	11		A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	0	1	1	1	0
4	6		A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	0	1	2	1	0
5	15		C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	0	1	2	2	0
6	10		C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	0	1	2	2	1
7	2	s	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	0	1	2	2	2
8	3		C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	0	1	2	2	3
9	4	t	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	0	1	3	2	3
10	18		G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	0	1	4	2	3
11	5		G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	0	1	4	3	3
12	17		G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	0	1	5	3	3
13	13		G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	0	2	5	3	3
14	8		G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	0	2	5	3	4
15	0		G	T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	0	2	5	3	5
16	14		T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	T	G	1	2	5	3	5
17	9		T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	1	2	5	4	5
18	1		T	C	C	C	G	A	T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	1	2	5	5	5
19	12		T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	T	G	T	C	A	1	2	5	6	5
20	7		T	G	T	C	A	T	G	T	C	A	G	G	A	$	G	T	C	C	C	G	A	1	3	5	6	5
																								1	4	5	6	5

Update s[i-1] = C[c] + Occ[c][s[i]]   c

Q C C   G T

s 7 5 0

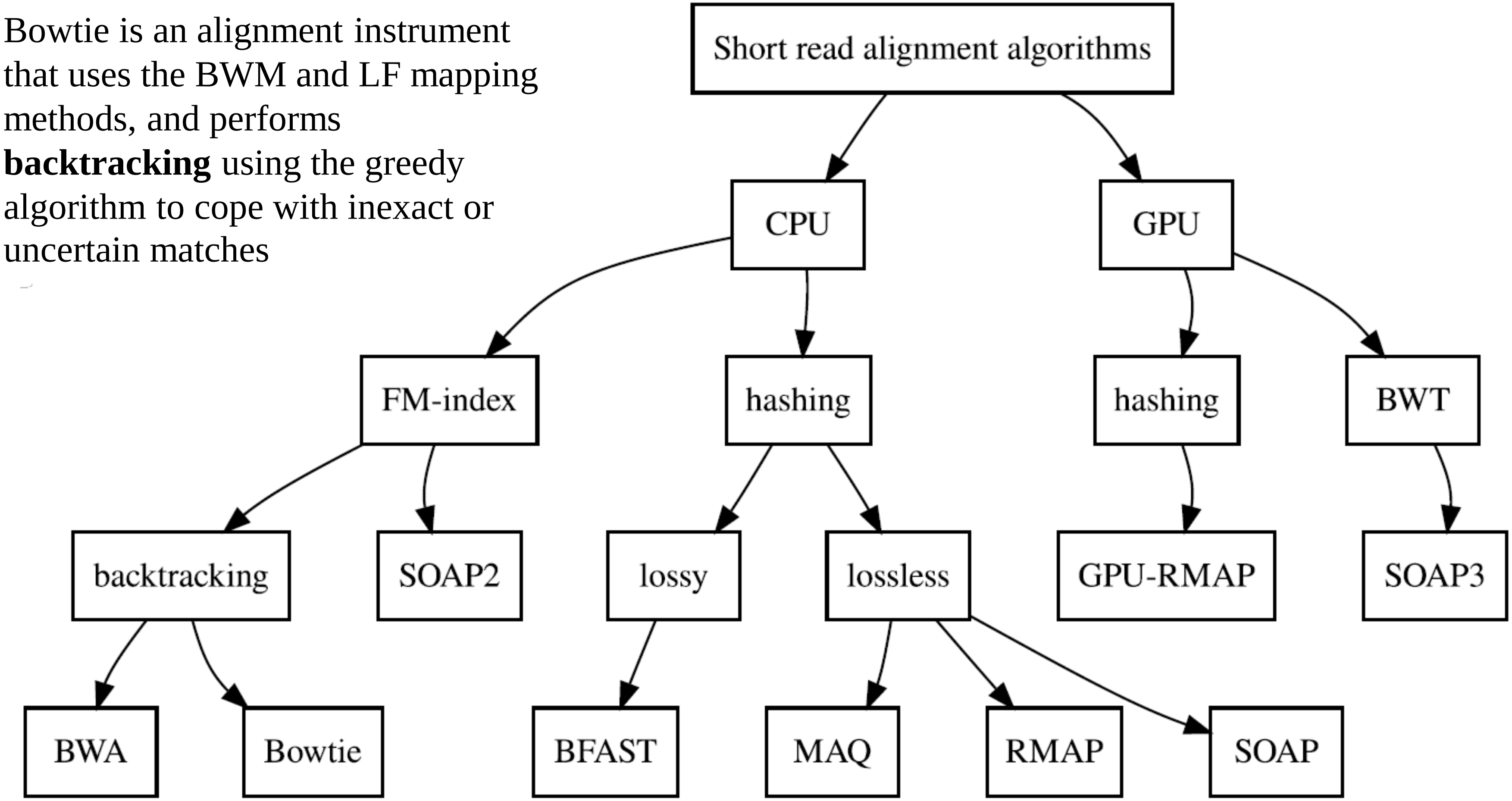
t 9 10 21

https://curiouscoding.nl/posts/bwt/

DOI: 10.13140/RG.2.2.33045.19687

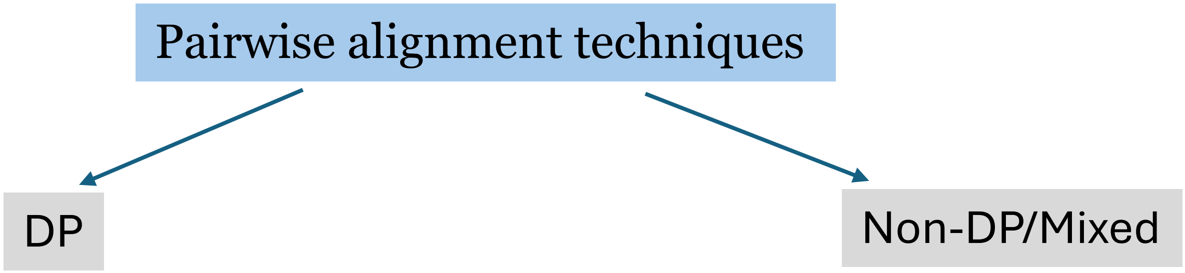
Different implementations on CPU and GPU

Bowtie is an alignment instrument that uses the BWM and LF mapping methods, and performs **backtracking** using the greedy algorithm to cope with inexact or uncertain matches



10.1109/ACCESS.2020.3031159

Seed extension and alignment



Pairwise alignment techniques

DP

Non-DP/Mixed

Smith-Waterman 28.3%

Needleman-Wunsch 16.2%

+ Hirschberg's algorithm (space-efficient version of the Needleman–Wunsch)

+ Landau-Vishkin

Hamming distance 19.2%

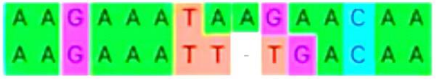
Heuristic 13.1%

Multiple Methods 9.1%

Edit distance:
Levenshtein Distance vs. Hamming Distance

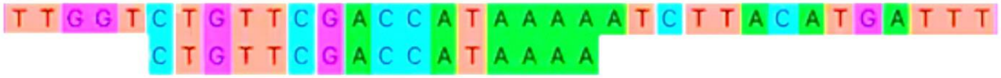
Types of paired alignment:

Pair global



Needleman-Wunsch

Pair local


T T G G T C T G T T C G A C C A T A A A A A T C T T A C A T G A T T T
C T G T T C G A C C A T A A A A

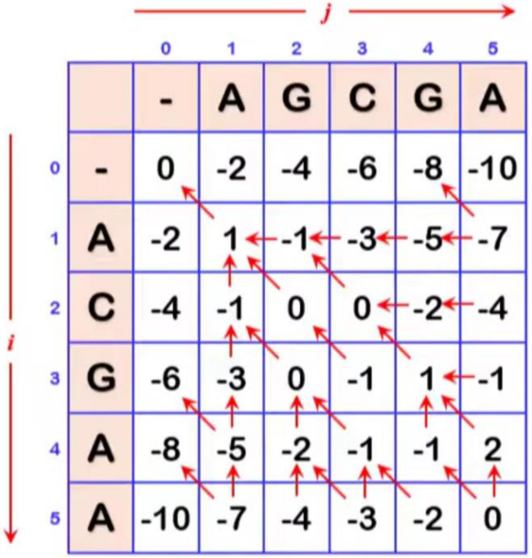
Smith-Waterman

Numbers are from: https://doi.org/10.1186/s13059-021-02443-7

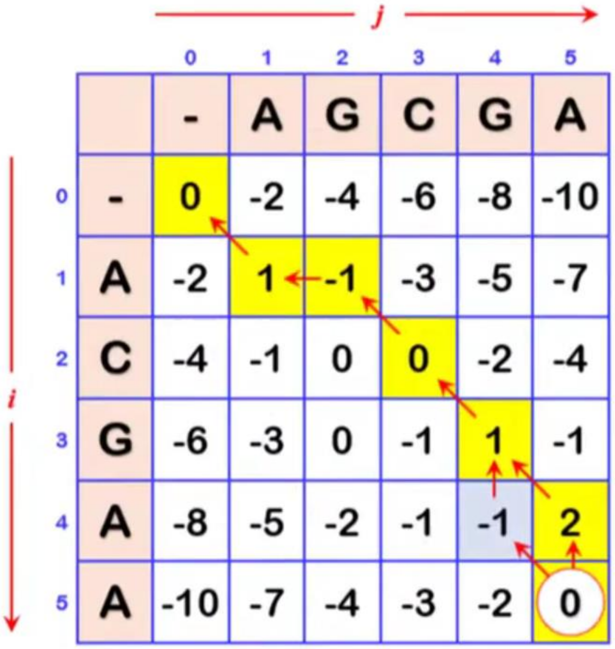
Needleman-Wunsch algorithm

The algorithm was developed by Saul B. Needleman and Christian D. Wunsch and published in 1970. Time complexity is *O(mn)* for sequences of m and n length.

$$M_{i,j} = max \begin{cases} \nwarrow \\ M_{i-1,j-1} + s(a_i, b_j) & s(a_i, b_j) = +1,\ if\ a_i = b_j\ (Match) \\ \leftarrow & s(a_i, b_j) = -1,\ if\ a_i \neq b_j\ (Mismatch) \\ M_{i,j-1} + s(a_i, -) & s(a_i, -) = -2,\ if\ b_j = -\ (Insertion) \\ \uparrow & s(-, b_j) = -2,\ if\ a_i = -\ (Deletion) \\ M_{i-1,j} + s(-, b_j) \end{cases}$$



		0	1	2	3	4	5
		-	A	G	C	G	A
0	-	0	-2	-4	-6	-8	-10
1	A	-2	1	-1	-3	-5	-7
2	C	-4	-1	0	0	-2	-4
3	G	-6	-3	0	-1	1	-1
4	A	-8	-5	-2	-1	-1	2
5	A	-10	-7	-4	-3	-2	0



		0	1	2	3	4	5
		-	A	G	C	G	A
0	-	0	-2	-4	-6	-8	-10
1	A	-2	1	-1	-3	-5	-7
2	C	-4	-1	0	0	-2	-4
3	G	-6	-3	0	-1	1	-1
4	A	-8	-5	-2	-1	-1	2
5	A	-10	-7	-4	-3	-2	0

Alignment 1:

```
A-CGAA
|  |||
AGCGA-

+1-2+1+1+1-2 = 0
```

Alignment 2:

```
A-CGAA
|  || |
AGCG-A

+1-2+1+1-2+1 = 0
```

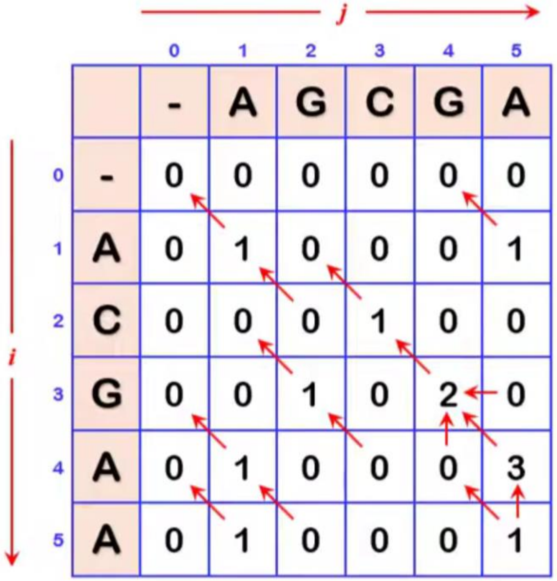
$$Score_{Total} = \sum Score_{Match} + \sum Score_{Mismatch} + \sum Score_{Insertion} + \sum Score_{Deletion}$$

Smith-Waterman algorithm

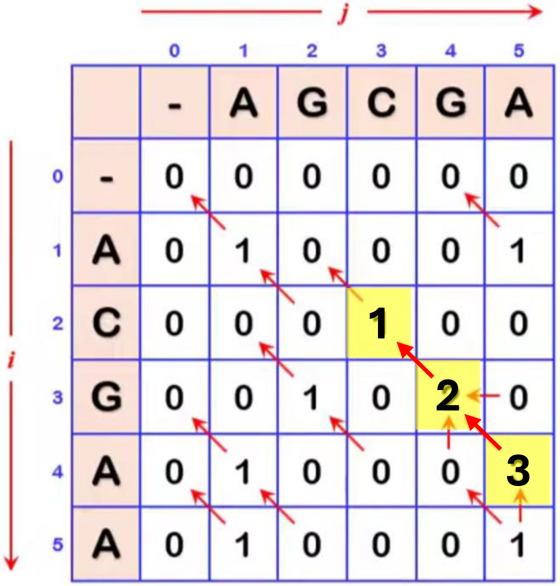
The algorithm was first proposed by Temple F. Smith and Michael S. Waterman in 1981.
The main difference to the Needleman–Wunsch algorithm is that negative scoring matrix cells are set to zero.
Can be optimized to *O(mn)* complexity for sequences of m and n length.

$$M_{i,j} = \max \begin{cases} M_{i-1,j-1} + s(a_i, b_j) & \nwarrow \\ M_{i,j-1} + s(a_i, -) & \leftarrow \\ M_{i-1,j} + s(-, b_j) & \uparrow \\ 0 \end{cases}$$

$$s(a_i, b_j) = +1,\ if\ a_i = b_j\ (Match)$$
$$s(a_i, b_j) = -1,\ if\ a_i \neq b_j\ (Mismatch)$$
$$s(-, b_j) = -2,\ if\ a_i = -\ (Insertion)$$
$$s(a_i, -) = -2,\ if\ b_j = -\ (Deletion)$$



		0	1	2	3	4	5
		-	A	G	C	G	A
0	-	0	0	0	0	0	0
1	A	0	1	0	0	0	1
2	C	0	0	0	1	0	0
3	G	0	0	1	0	2	0
4	A	0	1	0	0	0	3
5	A	0	1	0	0	0	1



		0	1	2	3	4	5
		-	A	G	C	G	A
0	-	0	0	0	0	0	0
1	A	0	1	0	0	0	1
2	C	0	0	0	1	0	0
3	G	0	0	1	0	2	0
4	A	0	1	0	0	0	3
5	A	0	1	0	0	0	1

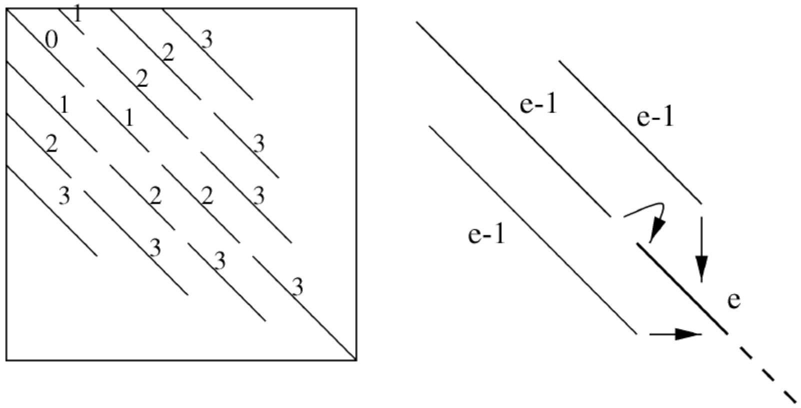
Alignment

```
CGA
|||
CGA
```

$$Score_{Total} = \sum Score_{Match} + \sum Score_{Mismatch} + \sum Score_{Insertion} + \sum Score_{Deletion}$$

Landau-Vishkin algorithm for Approximate String Matching

The parallel algorithm requires *O(logm+k)* using n processors
The serial algorithm runs in *O(nk)* time for an alphabet whose size is fixed.



Ukkonen *O*(*k*2) algorithm:
Computes the edit distance.
The way to compute the strokes in diagonal transition algorithms.
The solid bold line is guaranteed to be part of the new stroke of *e* errors, while the dashed part continues as long as both strings match.

Landau-Vishkin algorithm:
Dynamic programming matrix is computed diagonal-wise (i.e. stroke by stroke) instead of column-wise.

A recurrence on diagonals (*d*) and number of errors (*e*), instead of rows (*i*) and columns (*j*), is set up in the following way:

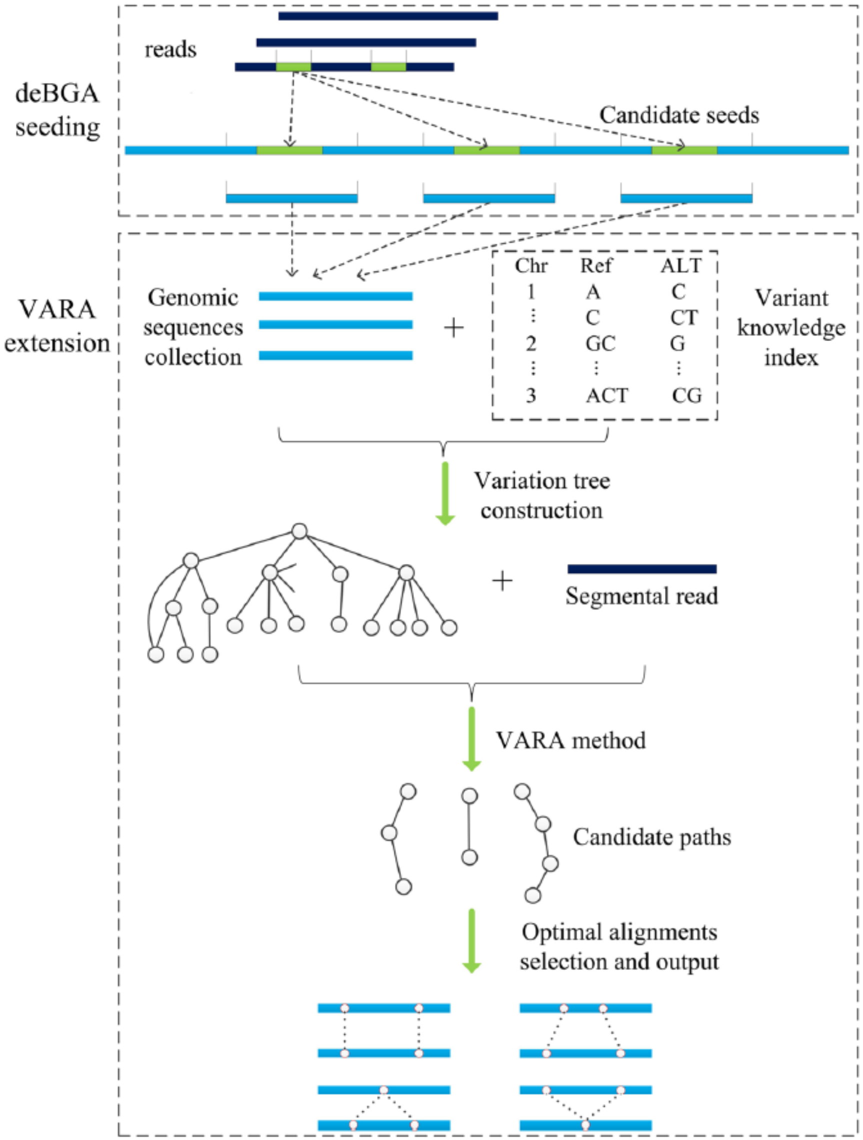
	-3	-2	-1	0	1	2	3	4	5	6	7
0		0	3	0	0	0	0	0	0	0	0
1	1	1	4	5	3	1	1	1	1	1	1
2	2	5	6	6	6	3	2	3	2	2	2

The diagonal transition matrix to search "survey" in the text "surgery" with two errors. Bold entries indicate matching diagonals. The rows are *e* values and the columns are the *d* values.

$$
\begin{aligned}
L_{d,-1} &= L_{n+1,e} \;=\; -1, \text{ for all } e, d \\
L_{d,|d|-2} &= |d| - 2, \text{ for } -(k+1) \leq d \leq -1 \\
L_{d,|d|-1} &= |d| - 1, \text{ for } -(k+1) \leq d \leq -1 \\
L_{d,e} &= i + \max_{\ell}(P_{i+1..i+\ell} = T_{d+i+1..d+i+\ell}) \\
\text{where } i &= \max(L_{d,e-1} + 1, \\
&\qquad L_{d-1,e-1}, L_{d+1,e-1} + 1)
\end{aligned}
$$

ACM Computing Surveys, Vol. 33, No. 1, March 2001, pp. 31–88
JOURNAL OF ALGORITHMS 10,157-169 (1989)

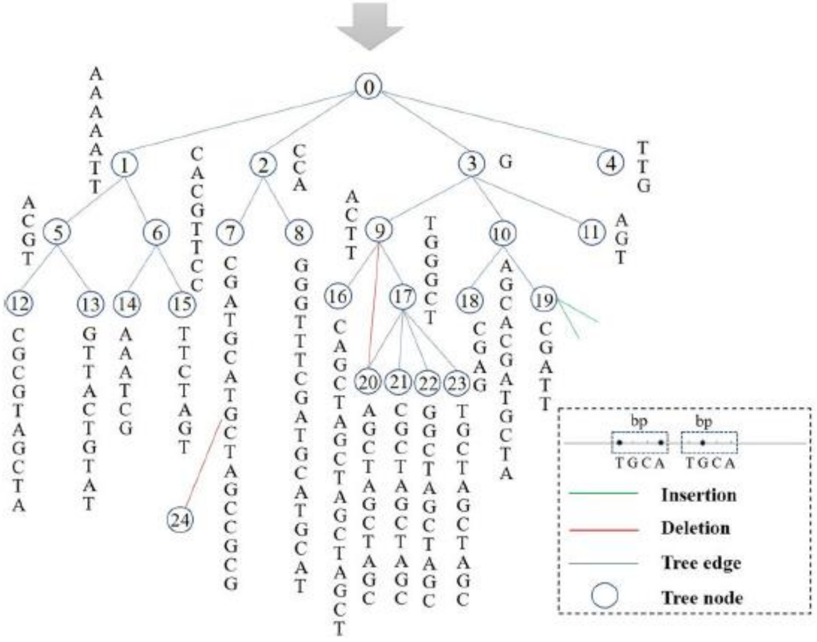
Variation-aware read alignments with Landau-Vishkin algorithm



Chr	Ref	ALT
1	A	C
⋮	C	CT
2	GC	G
⋮	⋮	⋮
3	ACT	CG

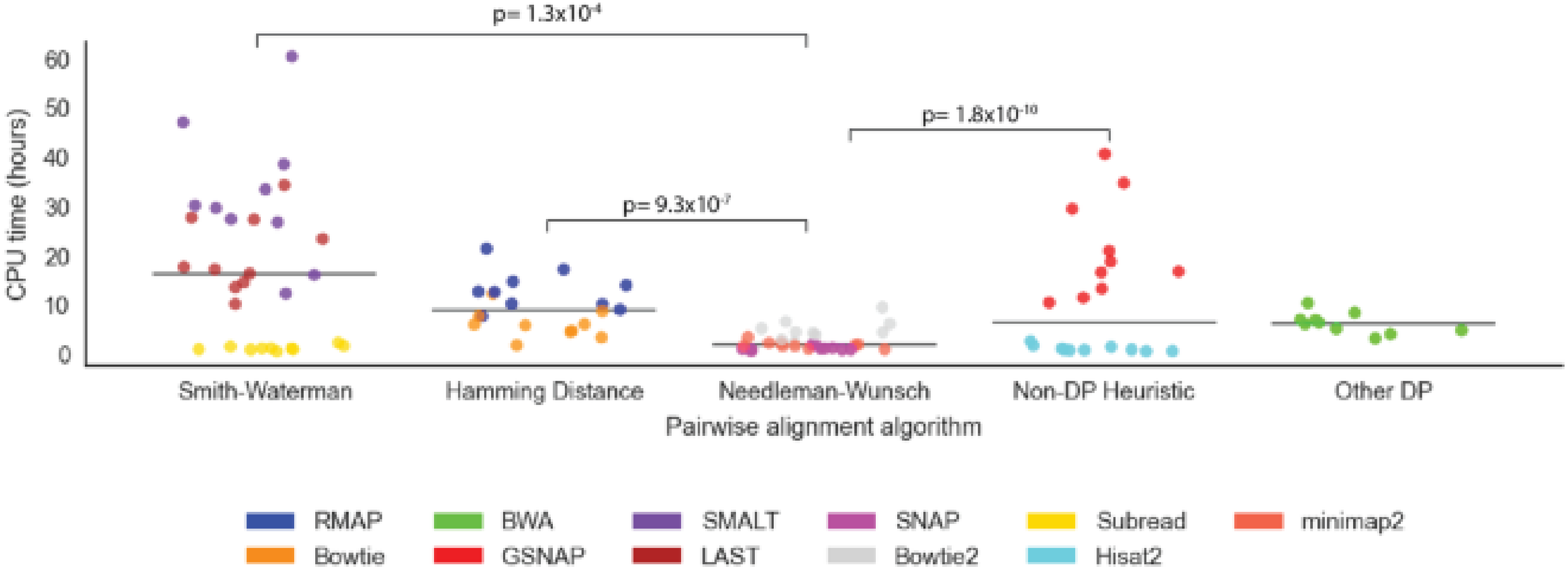
Reference Sequence (22 bp)

AAAAATTACGTCGCGTAGCTAA
AAAAATTACGTGTTACTGTATC
AAAAATTCACCTTCCAAATCGT
AAAAATTCACCTTCCTTCTAGT
CCACGATGCATGCTAGCCGCGA
CCAGGGTTTCGATGCATGCATG
GACTTCAGCTAGCTAGCTAGCT
GACTTTGGGCTAGCTAGCTAGC
GACTTTGGGCTCGCTAGCTAGC
GACTTTGGGCTGGCTAGCTAGC
GACTTTGGGCTTGCTAGCTAGC
GAGCACGATGCTAGCTACGAGA
GAGCACGATGCTAGCTACGATT
GAGTCGTAGCTAGCTACGATCG
TTGTTGCTAGCTCGTCGTACGT



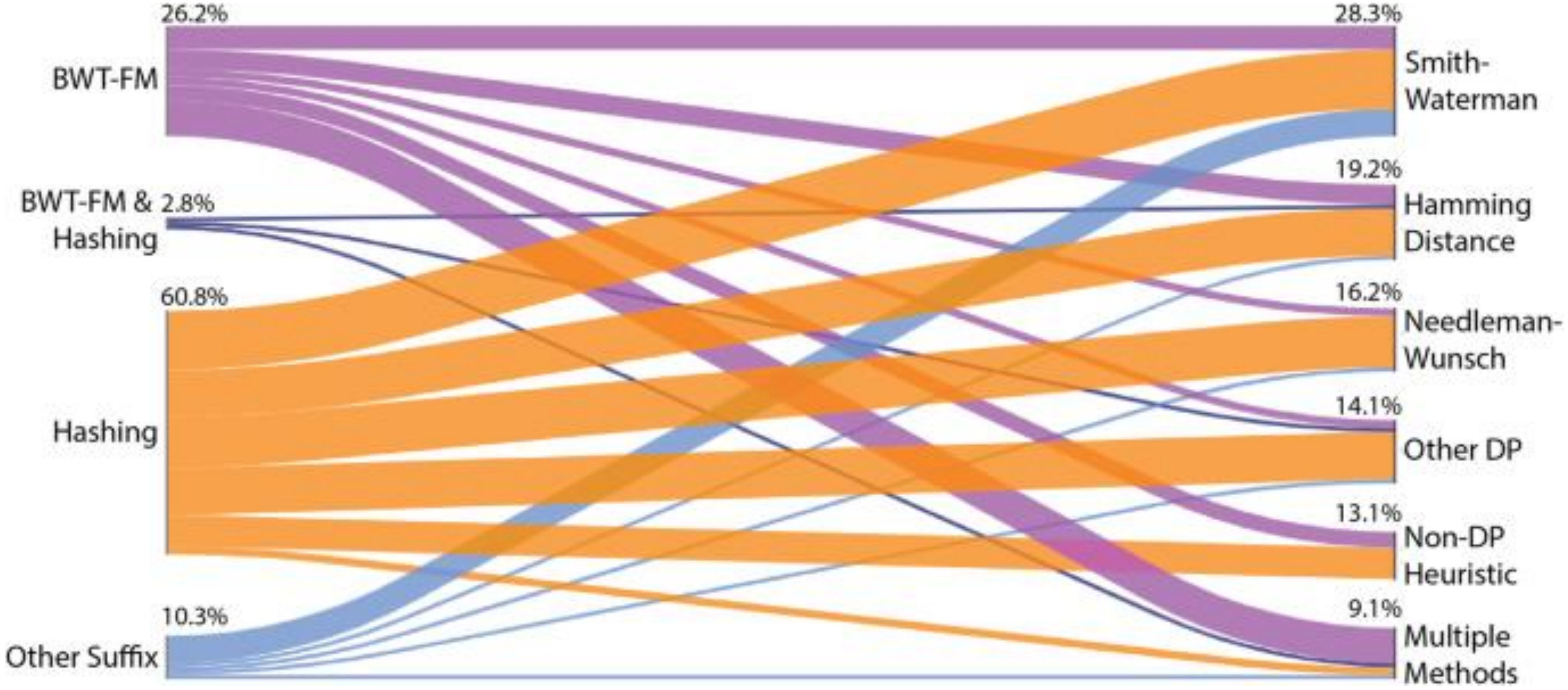
https://doi.org/10.1186/s12911-019-0960-3

Performance of different alignment algorithms on CPU



From: https://doi.org/10.1186/s13059-021-02443-7

Combination of algorithms utilized by read alignment tools



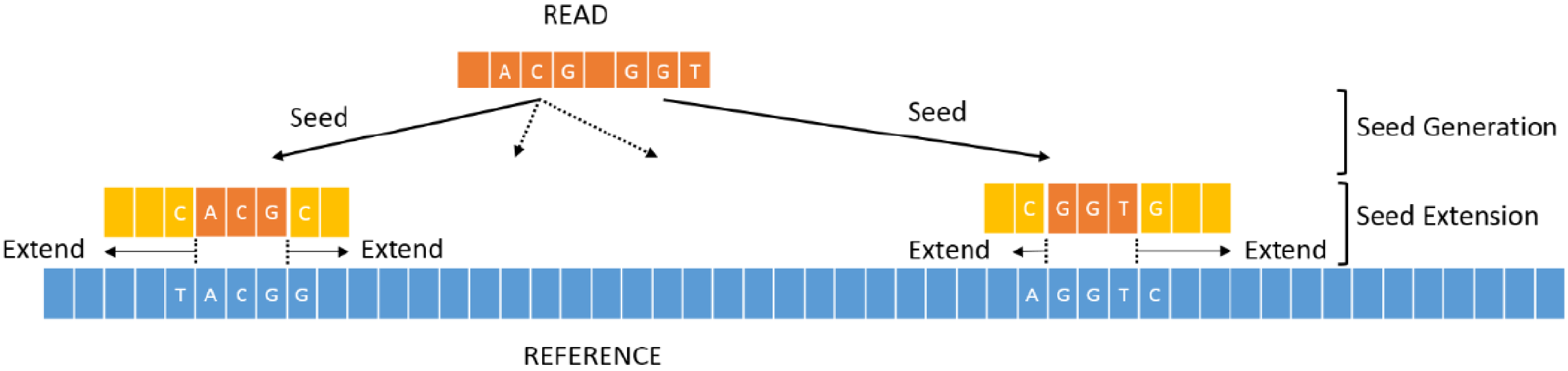
Based on studies of 107 read alignment tools that were designed for the short- and long-read sequencing technologies and were published from 1988 to 2020

https://doi.org/10.1186/s13059-021-02443-7

BWA-MEM Aligner

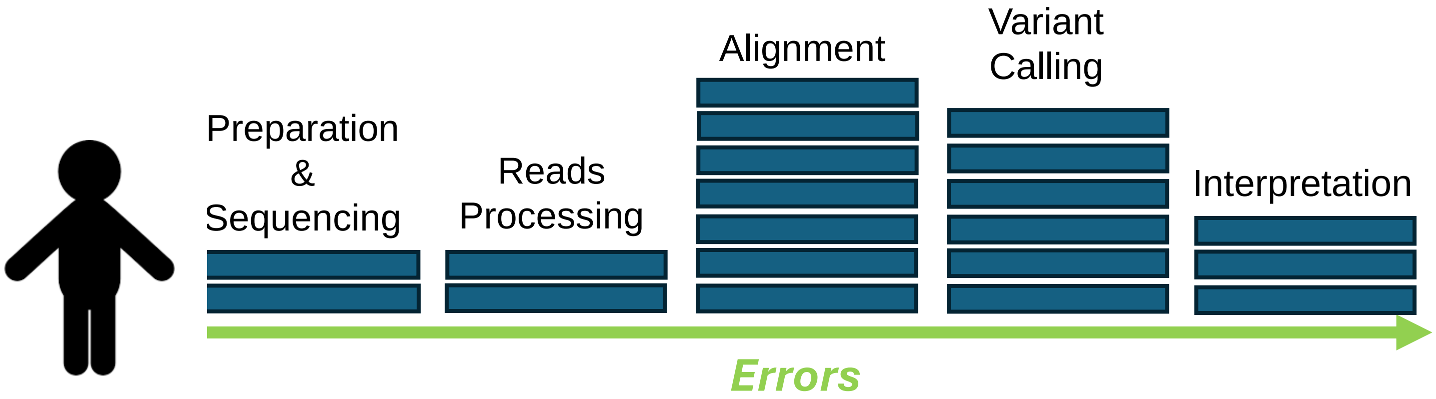
The conception of seeded alignment:

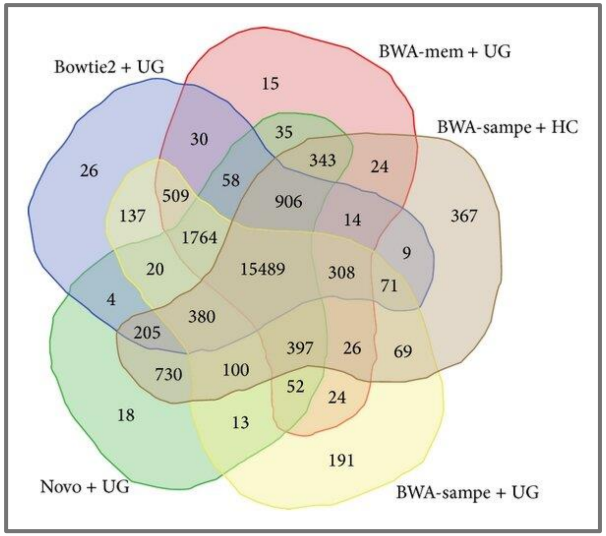
- Uses FM-index
- The seeds are *maximal exact matches* (*MEMs)*.
- *MEMs* cannot be extended either forward or backward without creating a mismatch
- *MEM* can represent a *super-maximal exact match (SMEM)* if it is not contained in any other *MEMs* on the query sequence.
- The extension of SMEMs is performed using the *Smith-Waterman* dynamic programming algorithm.



DOI: 10.13140/RG.2.2.33045.19687

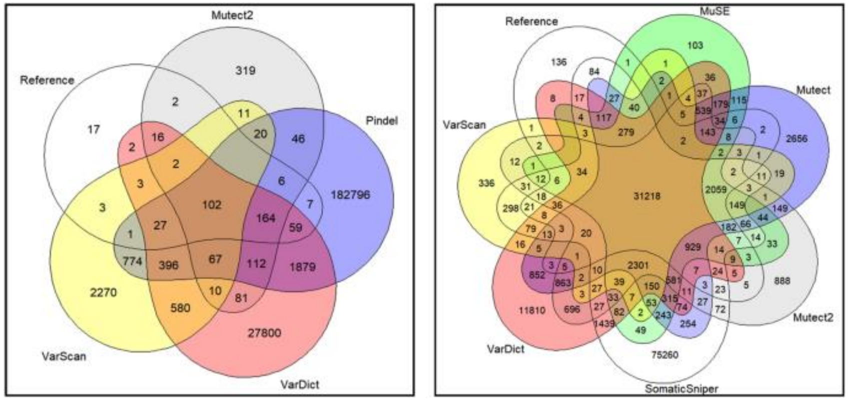
Influence of Different Alignment Tools on the Results





https://doi.org/10.1155/2015/456479

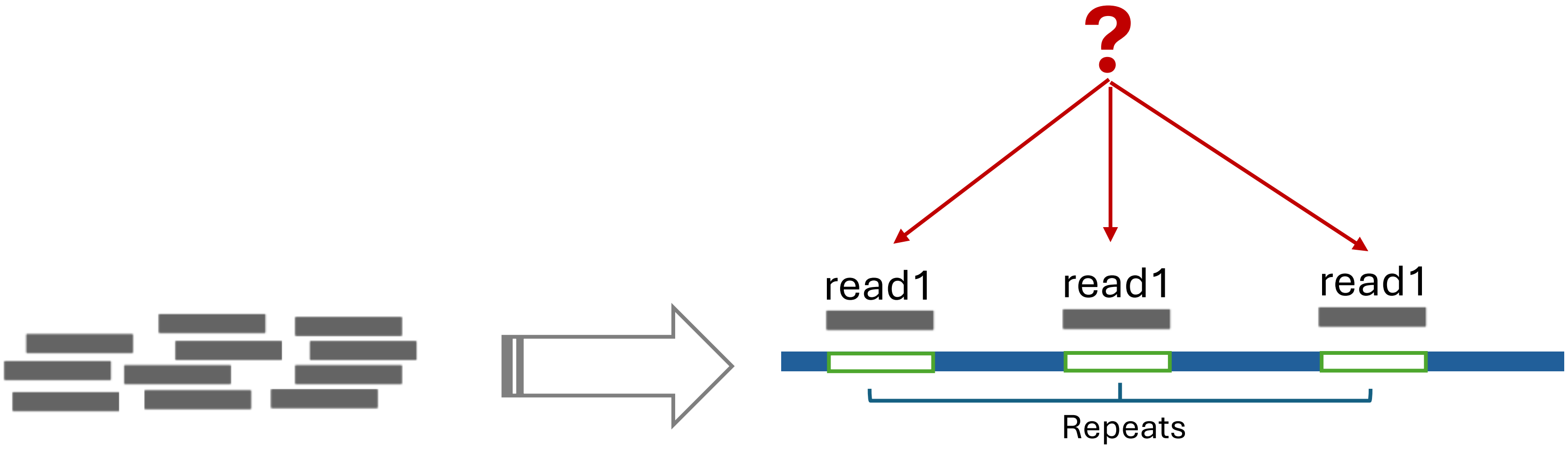
Somatic mutations

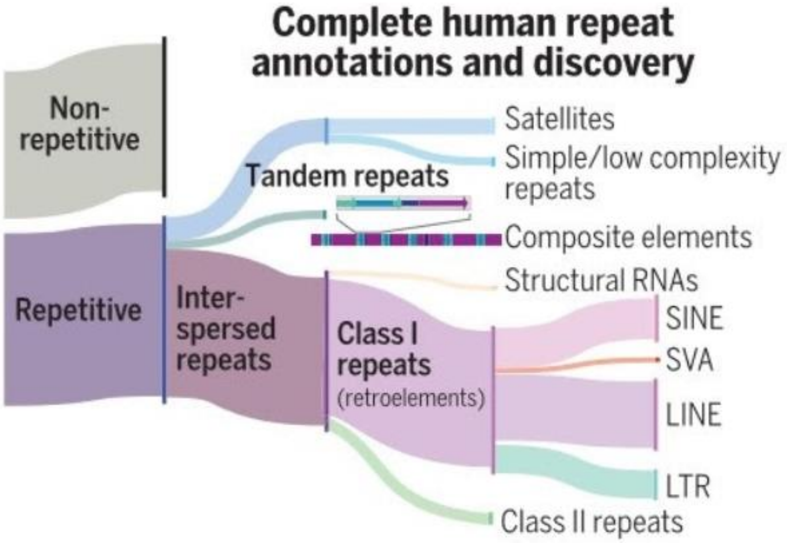


https://doi.org/10.1038/s41598-023-34925-y

Several ambiguity problems…

1. Ambiguity of reference

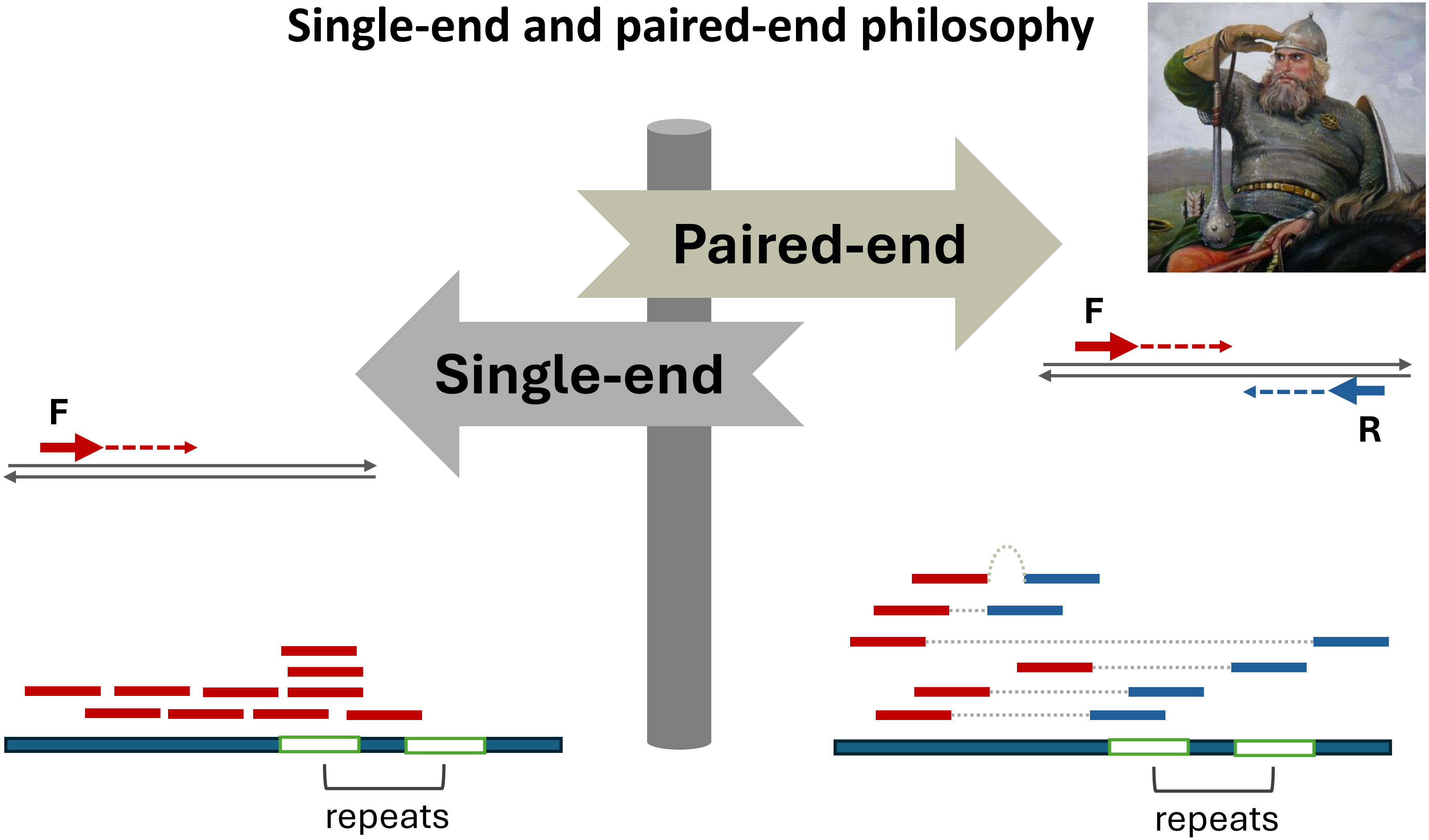




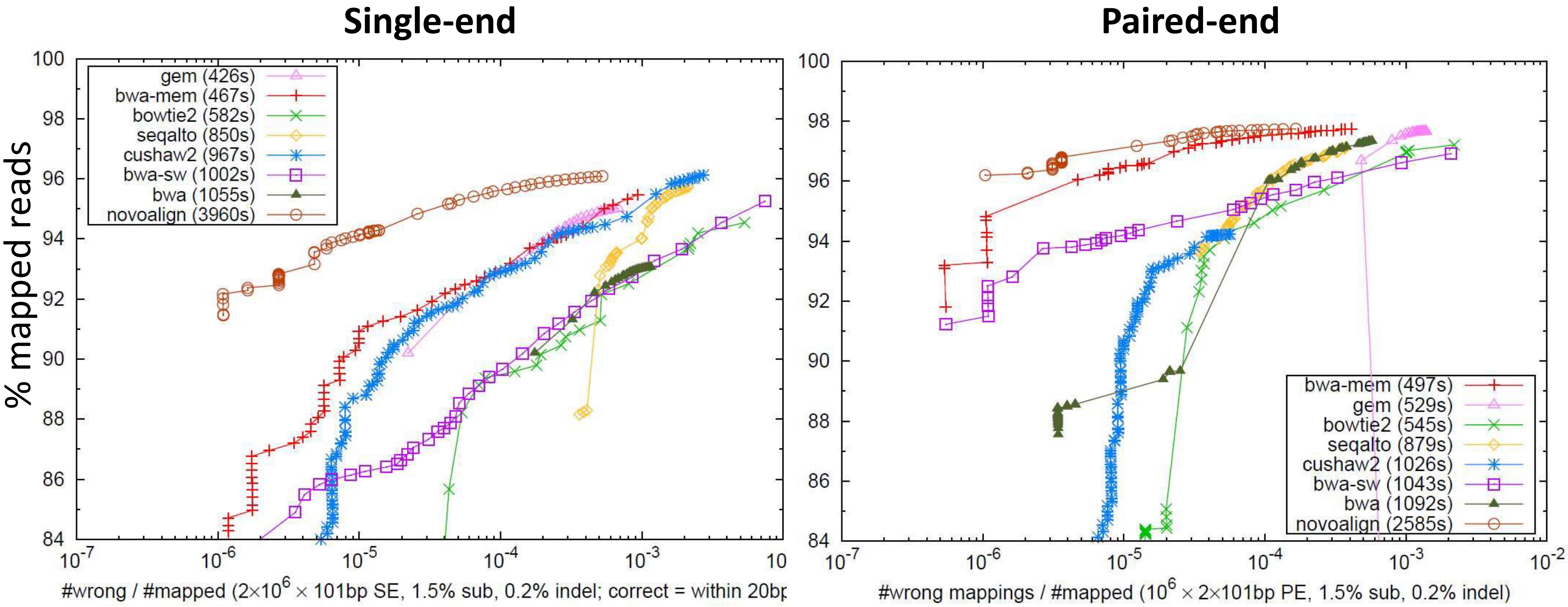
- SINEs 12.8%
- Retrotransposon 0.15%
- LINEs 20.7%
- LTRs 8.8%
- DNA transposons 3.6%
- Tandem Simple repeats 8%

TOTAL ~54%.

Single-end and paired-end philosophy



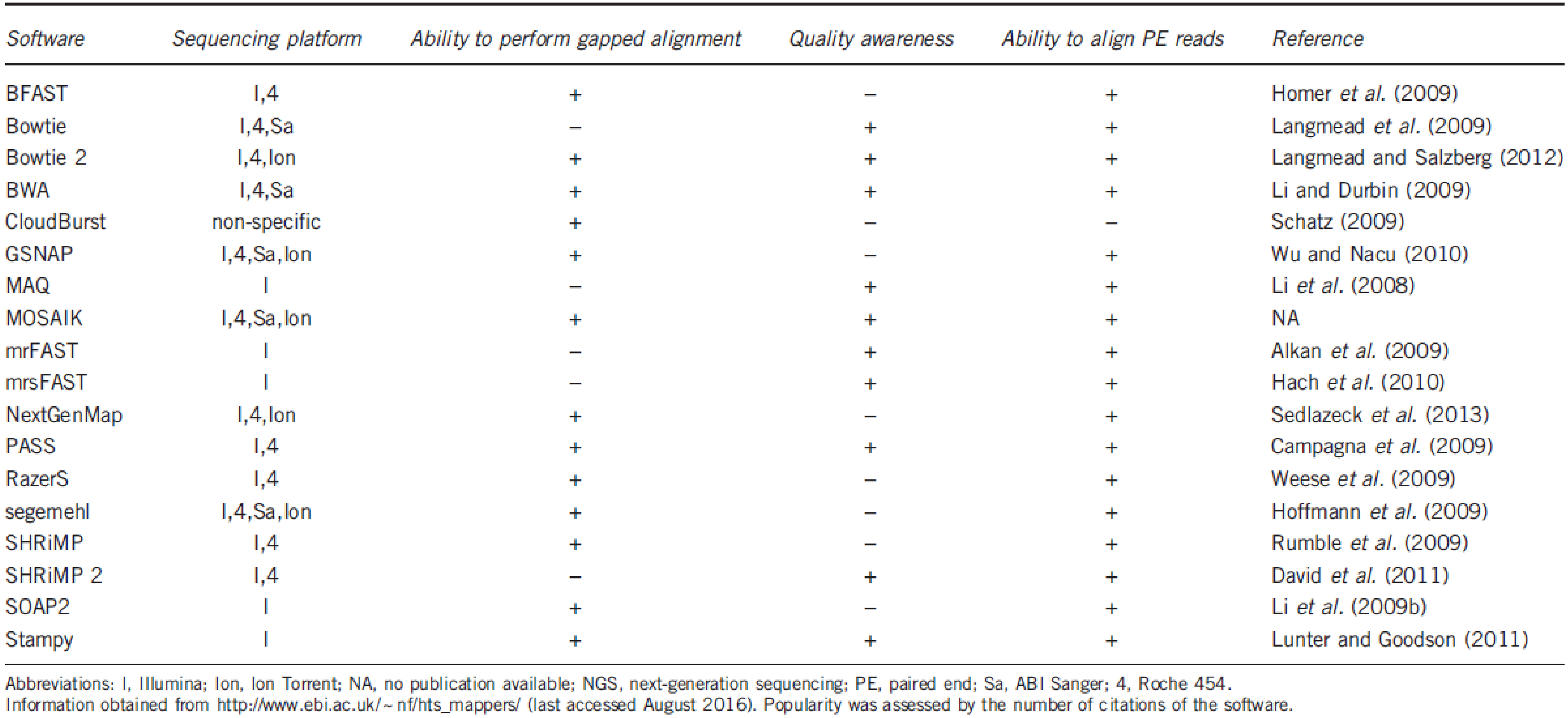
Performance of various aligners on simulated short reads from human genome


Single-end

Paired-end

From: Li (Broad Institute), http://arxiv.org/pdf/1303.3997v2.pdf

Most popular DNA aligners do perform paired-end



Software	*Sequencing platform*	*Ability to perform gapped alignment*	*Quality awareness*	*Ability to align PE reads*	*Reference*
BFAST	I,4	+	–	+	Homer *et al.* (2009)
Bowtie	I,4,Sa	–	+	+	Langmead *et al.* (2009)
Bowtie 2	I,4,Ion	+	+	+	Langmead and Salzberg (2012)
BWA	I,4,Sa	+	+	+	Li and Durbin (2009)
CloudBurst	non-specific	+	–	–	Schatz (2009)
GSNAP	I,4,Sa,Ion	+	–	+	Wu and Nacu (2010)
MAQ	I	–	+	+	Li *et al.* (2008)
MOSAIK	I,4,Sa,Ion	+	+	+	NA
mrFAST	I	–	+	+	Alkan *et al.* (2009)
mrsFAST	I	–	+	+	Hach *et al.* (2010)
NextGenMap	I,4,Ion	+	–	+	Sedlazeck *et al.* (2013)
PASS	I,4	+	+	+	Campagna *et al.* (2009)
RazerS	I,4	+	–	+	Weese *et al.* (2009)
segemehl	I,4,Sa,Ion	+	–	+	Hoffmann *et al.* (2009)
SHRiMP	I,4	+	–	+	Rumble *et al.* (2009)
SHRiMP 2	I,4	–	+	+	David *et al.* (2011)
SOAP2	I	+	–	+	Li *et al.* (2009b)
Stampy	I	+	+	+	Lunter and Goodson (2011)

Abbreviations: I, Illumina; Ion, Ion Torrent; NA, no publication available; NGS, next-generation sequencing; PE, paired end; Sa, ABI Sanger; 4, Roche 454.
Information obtained from http://www.ebi.ac.uk/~nf/hts_mappers/ (last accessed August 2016). Popularity was assessed by the number of citations of the software.

doi:10.1038/hdy.2016.102

2. Ambiguity of alignment

```
Reference  CTTTAGTTTCTTTT----CTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC

           CTTTAGTTTCTTTT----GCCGCTTTCTTTCTTTCTT
           CTTTAGTTTCTTTT----GCCGCTTTCTTTCTTTCTT
Reads      CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC
           CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC
           CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC
           CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC
```

For these reads, aligner preferred to make a few SNPs rather than insertion

For these reads, insertion was a better choice

Aligner, like BWA, works on one read (fragment) at a time, does not see a bigger picture…)

But we can try to shift things around a bit:

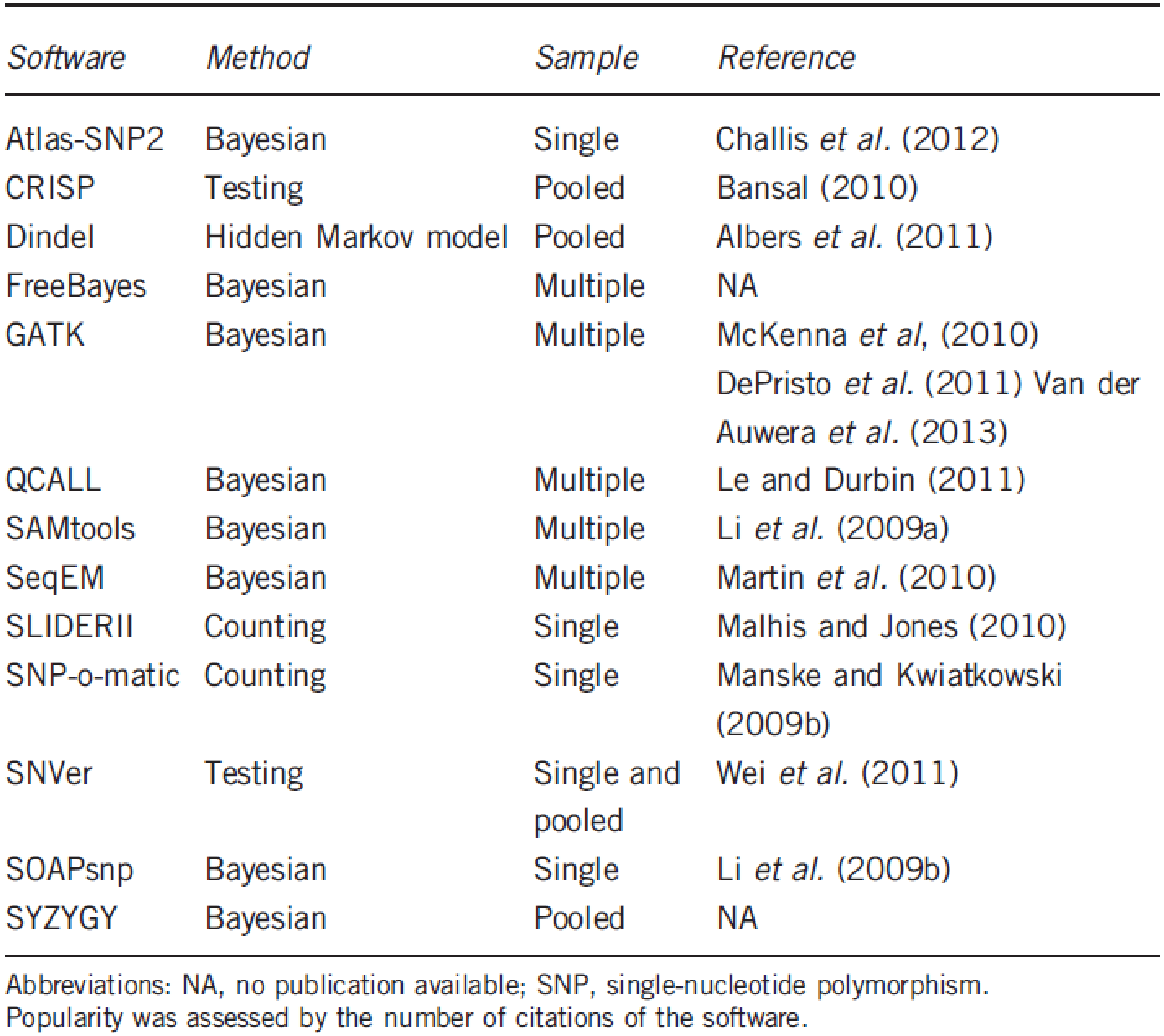
```
Reference  CTTTAGTTTCTTTT----CTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC

           CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTT
           CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTT
Reads      CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC
           CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC
           CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC
           CTTTAGTTTCTTTTGCCGCTTTCTTTCTTTCTTTTTTTTTAAGTCTCCCTC
```

This looks better !

Only seen after aligning all (at least some) reads!

SNP callers can reevaluate alignment



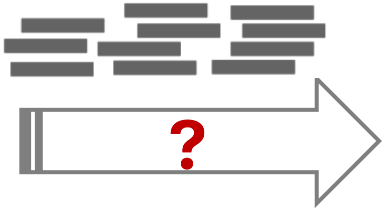
Software	*Method*	*Sample*	*Reference*
Atlas-SNP2	Bayesian	Single	Challis *et al.* (2012)
CRISP	Testing	Pooled	Bansal (2010)
Dindel	Hidden Markov model	Pooled	Albers *et al.* (2011)
FreeBayes	Bayesian	Multiple	NA
GATK	Bayesian	Multiple	McKenna *et al*, (2010) DePristo *et al.* (2011) Van der Auwera *et al.* (2013)
QCALL	Bayesian	Multiple	Le and Durbin (2011)
SAMtools	Bayesian	Multiple	Li *et al.* (2009a)
SeqEM	Bayesian	Multiple	Martin *et al.* (2010)
SLIDERII	Counting	Single	Malhis and Jones (2010)
SNP-o-matic	Counting	Single	Manske and Kwiatkowski (2009b)
SNVer	Testing	Single and pooled	Wei *et al.* (2011)
SOAPsnp	Bayesian	Single	Li *et al.* (2009b)
SYZYGY	Bayesian	Pooled	NA

Abbreviations: NA, no publication available; SNP, single-nucleotide polymorphism.
Popularity was assessed by the number of citations of the software.

https://doi.org/10.1038/hdy.2016.102

3. Ambiguity of reads

Source of reads



Reference genome

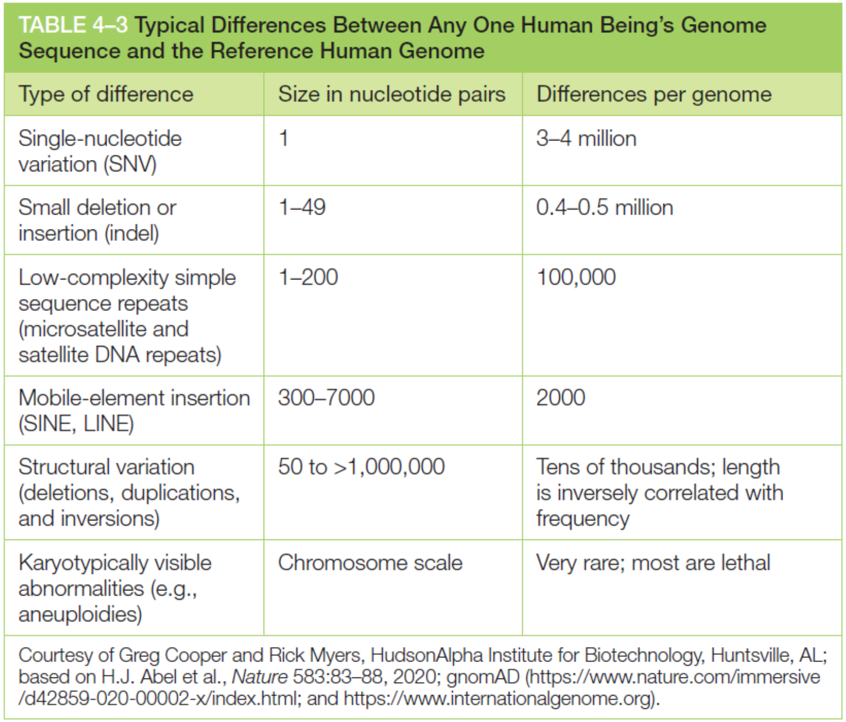
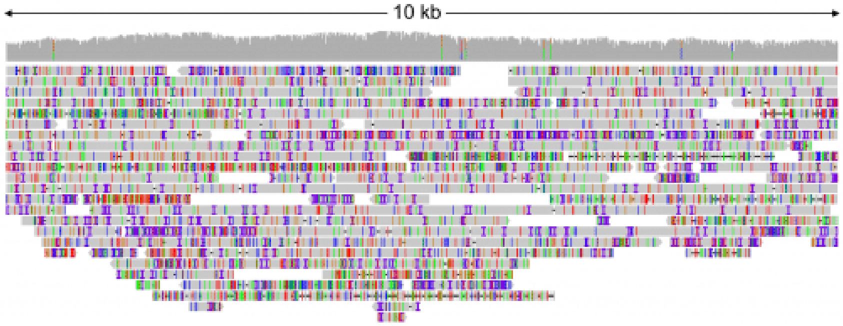


TABLE 4–3 Typical Differences Between Any One Human Being's Genome Sequence and the Reference Human Genome

Type of difference	Size in nucleotide pairs	Differences per genome
Single-nucleotide variation (SNV)	1	3–4 million
Small deletion or insertion (indel)	1–49	0.4–0.5 million
Low-complexity simple sequence repeats (microsatellite and satellite DNA repeats)	1–200	100,000
Mobile-element insertion (SINE, LINE)	300–7000	2000
Structural variation (deletions, duplications, and inversions)	50 to >1,000,000	Tens of thousands; length is inversely correlated with frequency
Karyotypically visible abnormalities (e.g., aneuploidies)	Chromosome scale	Very rare; most are lethal

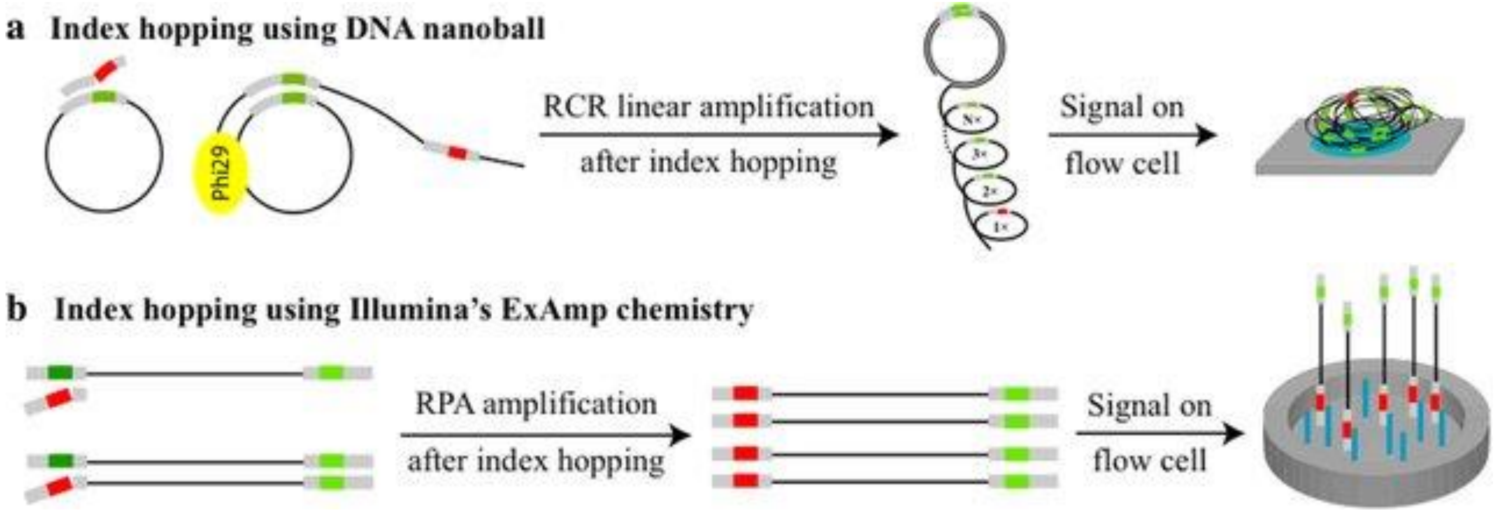
Courtesy of Greg Cooper and Rick Myers, HudsonAlpha Institute for Biotechnology, Huntsville, AL; based on H.J. Abel et al., *Nature* 583:83–88, 2020; gnomAD (https://www.nature.com/immersive/d42859-020-00002-x/index.html; and https://www.internationalgenome.org).

The problem get worse for the long reads where special tools were also developed



4. Ambiguity of letters

Errors arising during library preparation



No error amplification

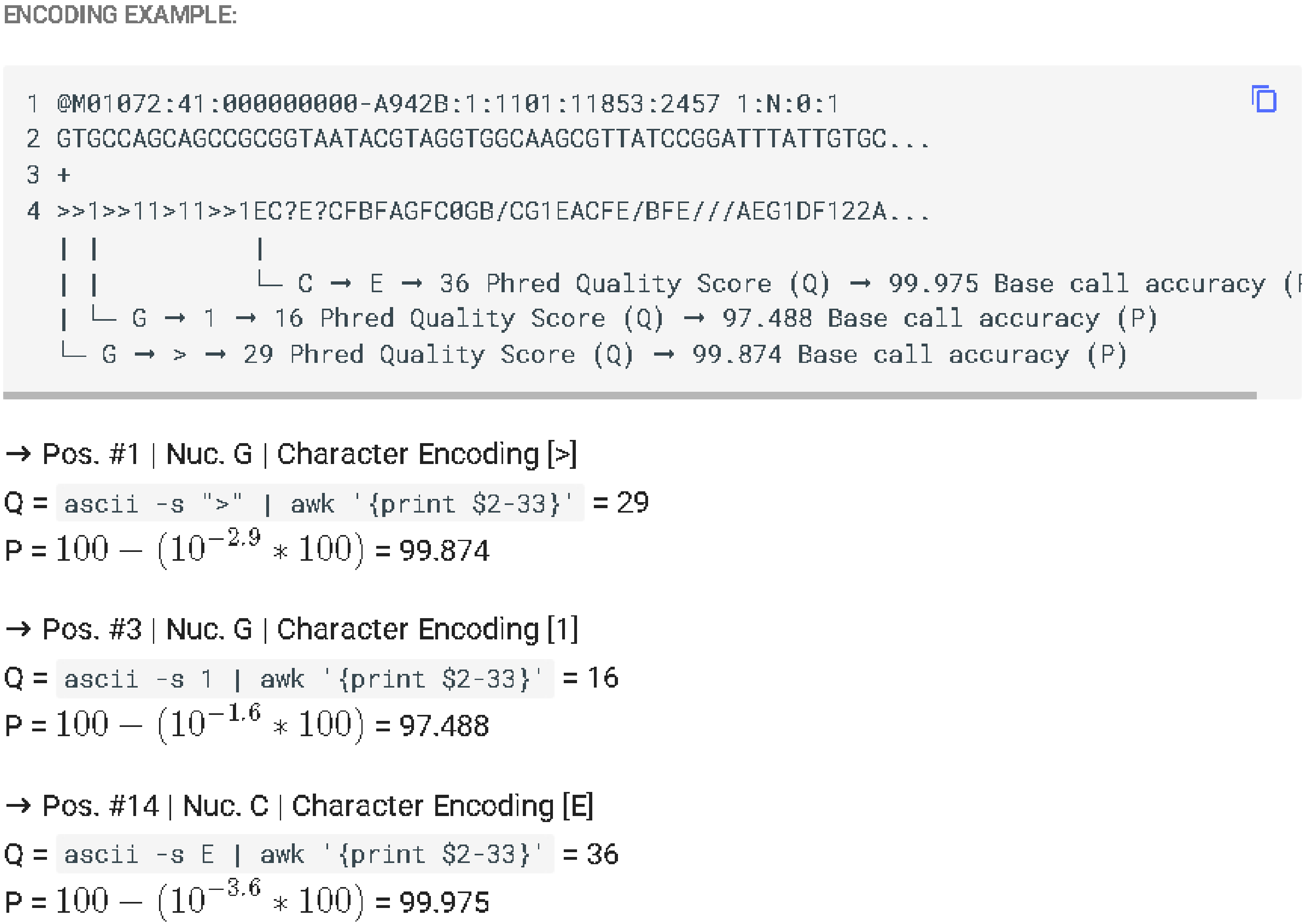
Exponential error amplification

+ Sequencing errors

Reads may contain errors!!!

https://doi.org/10.1186/s12864-019-5569-5

What do we know about read quality of reads?



ENCODING EXAMPLE:

```
@M01072:41:000000000-A942B:1:1101:11853:2457 1:N:0:1
GTGCCAGCAGCCGCGGTAATACGTAGGTGGCAAGCGTTATCCGGATTTATTGTGC...
+
>>1>>11>11>>1EC?E?CFBFAGFC0GB/CG1EACFE/BFE///AEG1DF122A...
| |          |
| |          └ C → E → 36 Phred Quality Score (Q) → 99.975 Base call accuracy (
| └ G → 1 → 16 Phred Quality Score (Q) → 97.488 Base call accuracy (P)
└ G → > → 29 Phred Quality Score (Q) → 99.874 Base call accuracy (P)
```

→ Pos. #1 | Nuc. G | Character Encoding [>]

Q = `ascii -s ">" | awk '{print $2-33}'` = 29

P = $100 - (10^{-2.9} * 100)$ = 99.874

→ Pos. #3 | Nuc. G | Character Encoding [1]

Q = `ascii -s 1 | awk '{print $2-33}'` = 16

P = $100 - (10^{-1.6} * 100)$ = 97.488

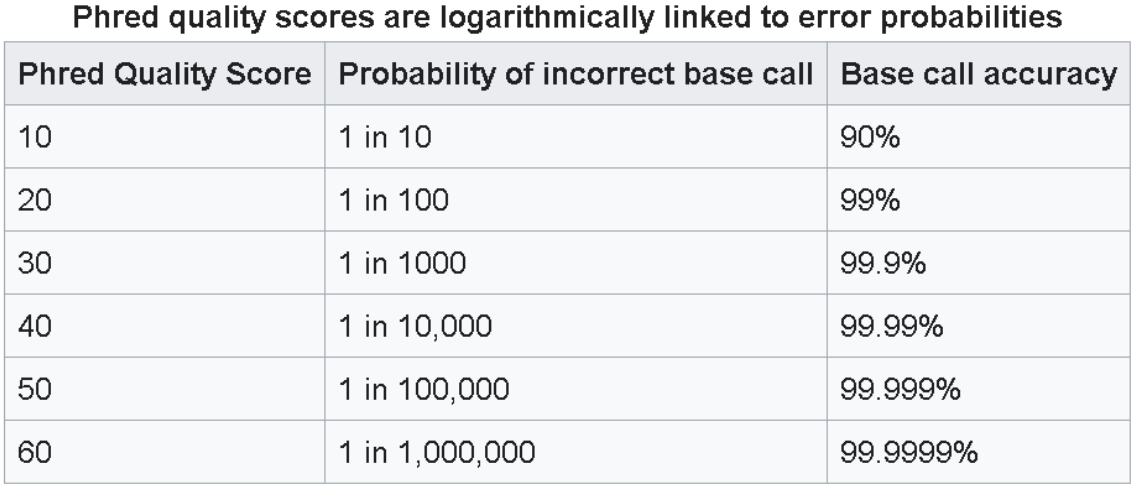
→ Pos. #14 | Nuc. C | Character Encoding [E]

Q = `ascii -s E | awk '{print $2-33}'` = 36

P = $100 - (10^{-3.6} * 100)$ = 99.975

Sequence quality: Phred quality scores, Q

$$Q = -10 \log_{10} P$$

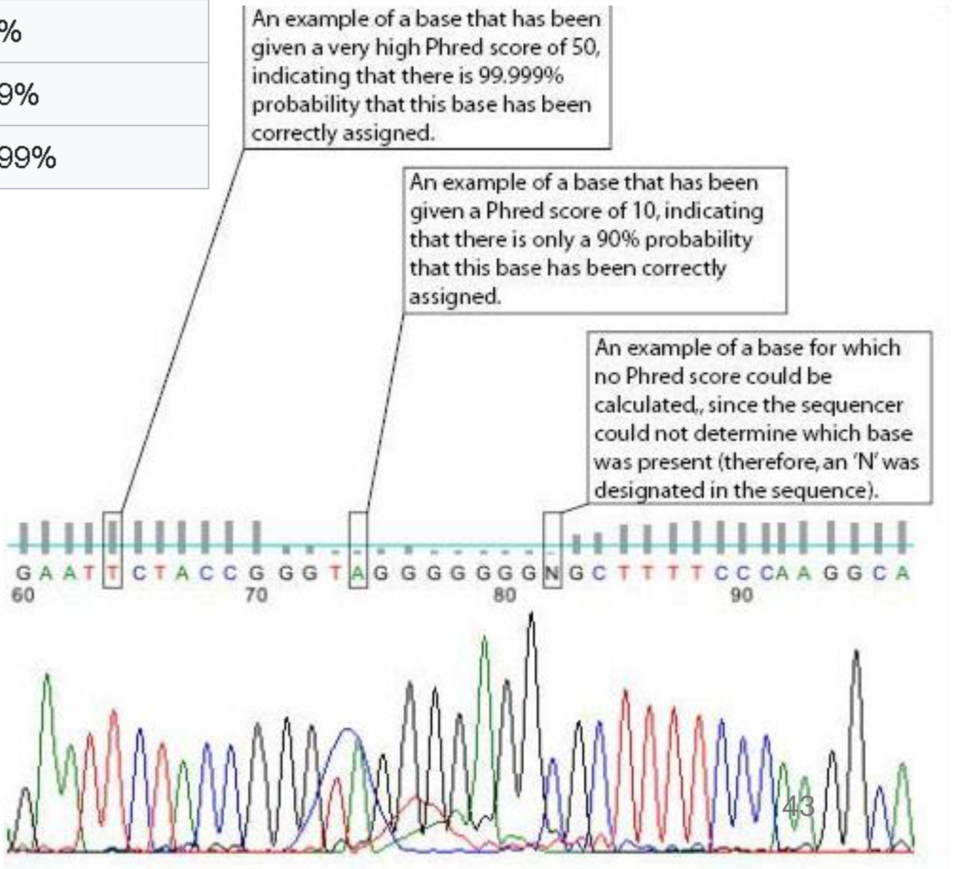


Phred quality scores are logarithmically linked to error probabilities

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%

$$\boxed{P = 10^{-Q/10}}$$

Phred score 20 →



An example of a base that has been given a very high Phred score of 50, indicating that there is 99.999% probability that this base has been correctly assigned.

An example of a base that has been given a Phred score of 10, indicating that there is only a 90% probability that this base has been correctly assigned.

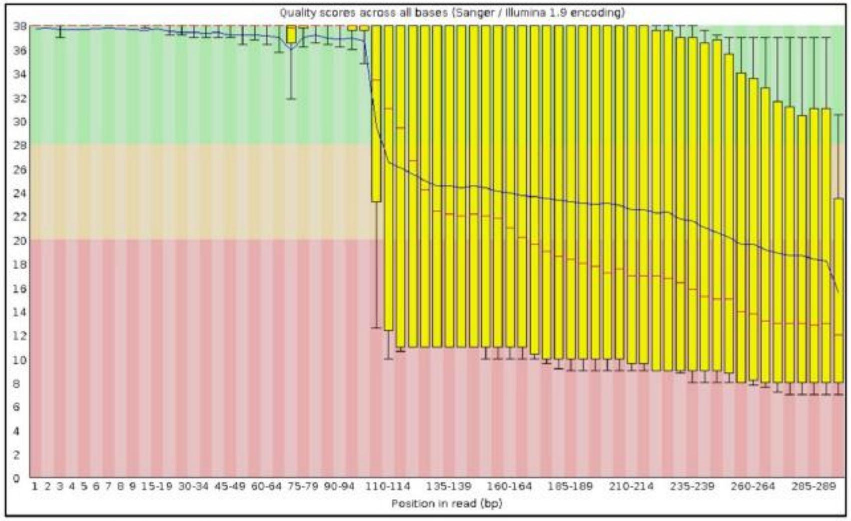
An example of a base for which no Phred score could be calculated,, since the sequencer could not determine which base was present (therefore, an 'N' was designated in the sequence).

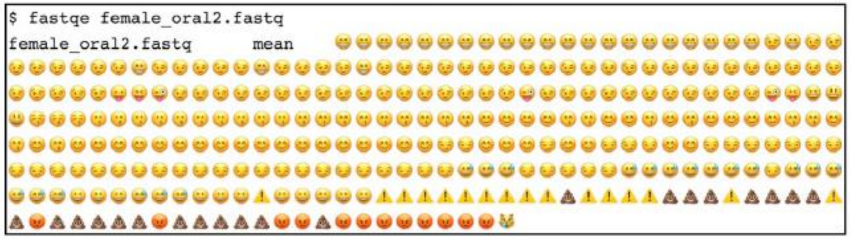
G A A T T C T A C C G G G T A G G G G G G G N G C T T T T C C C A A G G C A
60
70
80
90

The good, the bad and the ugly reads

Symbol	Phred	Error
!	0	1.000
"	1	0.794
#	2	0.631
$	3	0.501
%	4	0.398
&	5	0.316
'	6	0.251
(	7	0.199
)	8	0.158
*	9	0.126
+	10	0.100
,	11	0.079
-	12	0.063
.	13	0.050
/	14	0.040
0	15	0.032
1	16	0.025
2	17	0.020
3	18	0.016
4	19	0.013
5	20	0.010
6	21	0.008
7	22	0.006
8	23	0.005
9	24	0.004
:	25	0.003
;	26	0.002
<	27	0.002
=	28	0.001
>	29	0.001
?	30	0.001
@	31	0.0008
A	32	0.0006
B	33	0.0005
C	34	0.0004
D	35	0.0003
E	36	0.0002
F	37	0.0002
G	38	0.0002
H	39	0.0001
I	40	0.0001

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%





```
$ fastqe female_oral2.fastq
female_oral2.fastq      mean
```

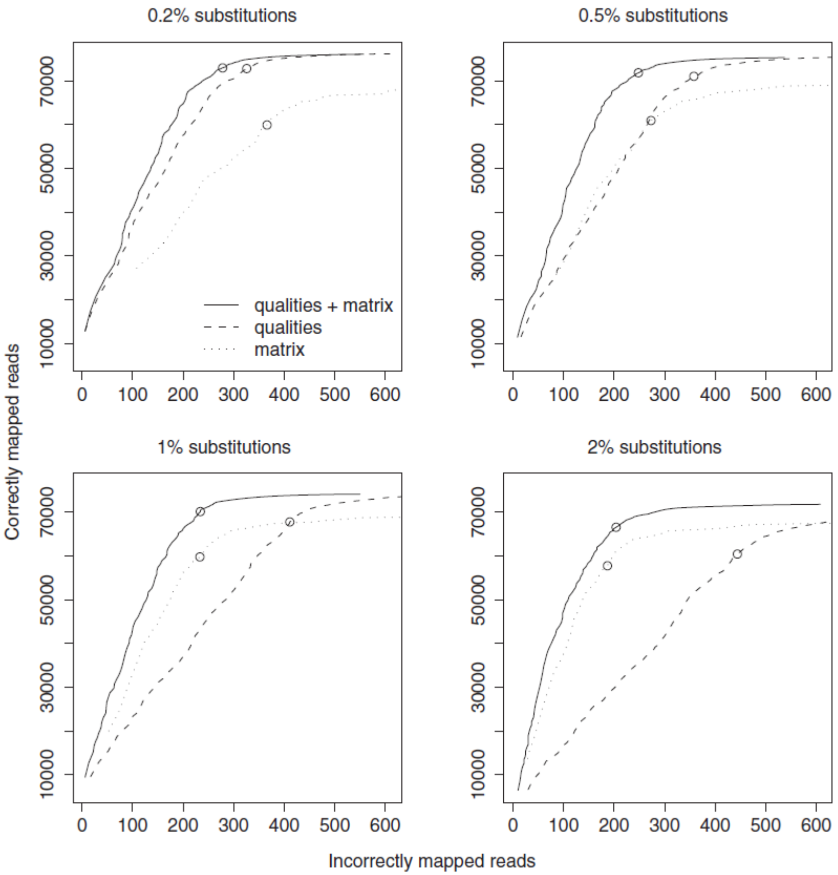
D. Phred score: Emoji scale

0	!	🚫	21	6	😀
1	"	❌	22	7	😅
2	#	👺	23	8	😏
3	$	💔	24	9	😊
4	%	🙅	25	:	😙
5	&	👾	26	;	😗
6	'	👿	27	<	😚
7	(	💀	28	=	😃
8	)	👻	29	>	😘
9	*	🙈	30	?	😆
10	+	🙉	31	@	😄
11	,	🙊	32	A	😋
12	-	🐵	33	B	😄
13	.	😿	34	C	😝
14	/	😾	35	D	😛
15	0	🙀	36	E	😜
16	1	💣	37	F	😉
17	2	🔥	38	G	😁
18	3	😡	39	H	😄
19	4	💩	40	I	😎
20	5	⚠	41	J	😍

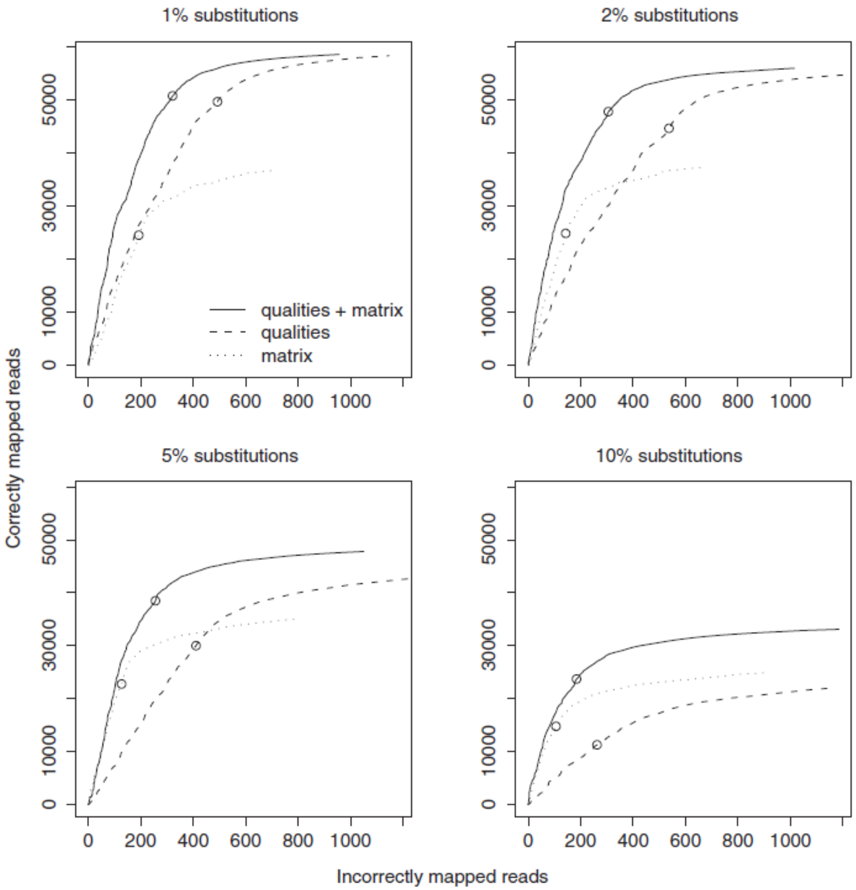
DOI: 10.24918/cs.2021.17

Incorporating sequence quality data into alignment

Mapping accuracy for 100 000 simulated 36-nt reads



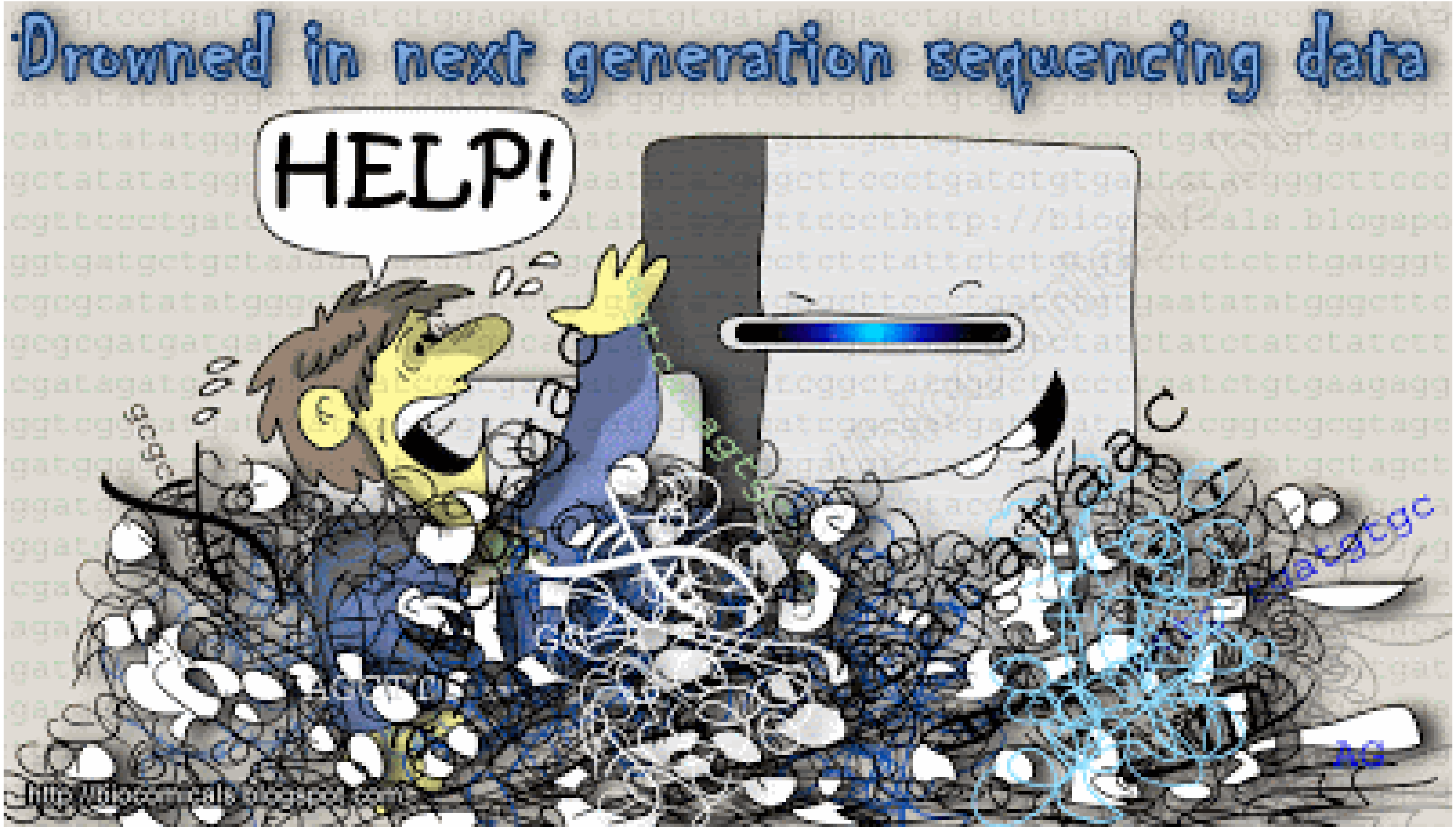
Mapping accuracy for 100 000 simulated 51-nt reads



The reads differ from the genome by a certain rate of ‘real’ substitutions (0.2, 0.5, 1 or 2%) plus sequencer errors. Circles indicate a score cutoff of 150/or 180. Dotted lines show the accuracy when we model the substitutions but not the sequencer errors. Dashed lines show the accuracy when we model the sequencer errors but not the substitutions. Solid lines show the accuracy for both.

doi:10.1093/nar/gkq010

Thank you!


Drowned in next generation sequencing data
HELP!
http://biocomicals.blogspot.com
AG