

```
pos[0] = 0;
for(m = t_len - 1, nuc_pos = t_e - 1; m >= 0; --m, --nuc_pos) {
    if(nuc_pos < 0) {
        nuc_pos = template_length - 1;
    }
    D_ptr[q_len] = (0 < k) ? 0 : (W1 + (t_len - 1 - m) * U);
    Q_prev = (t_len + q_len) * (MM + U + W1);

    t_nuc = getNuc(template, nuc_pos);
    for(n = q_len - 1; n >= 0; --n) {
        E_ptr[n] = 0;

        /* update Q and P, gap openings */
        Q = D_ptr[n + 1] + W1;
        P_ptr[n] = D_prev[n] + W1;
        if(Q < P_ptr[n]) {
            D_ptr[n] = P_ptr[n];
            e = 4;
        } else {
            D_ptr[n] = Q;
            e = 2;
        }

        /* update Q and P, gap extensions */
        /* mark bit 4 and 5 as possible gap-opennings, if necessary */
        thisScore = Q_prev + U;
        if(Q < thisScore) {
            Q = thisScore;
            if(e == 2)
                D_ptr[n] |= 1;
        } else {
            E_ptr[n] |= 16;
        }
        thisScore = P_prev[n] + U;
        if(P_ptr[n] < thisScore) {
            P_ptr[n] = thisScore;
            if(D_ptr[n] < thisScore) {
                D_ptr[n] = thisScore;
                e = 5;
            }
        } else {
            E_ptr[n] |= 32;
        }

        /* Update D, match */
        thisScore = D_prev[n + 1] + d[t_nuc][query[n]];
        if(D_ptr[n] < thisScore) {
            D_ptr[n] = thisScore;
            E_ptr[n] |= 1;
        } else {
            E_ptr[n] |= e;
        }
    }
    Q_prev = Q;
}

E_ptr -= (q_len + 1);

if(k < 0 && Stat.score <= *D_ptr) {
    Stat.score = *D_ptr;
    pos[0] = m;
}

tmp = D_ptr;
D_ptr = D_prev;
D_prev = tmp;

tmp = P_ptr;
P_ptr = P_prev;
P_prev = tmp;
}
E_ptr = E;
```

# Mapping and Alignment

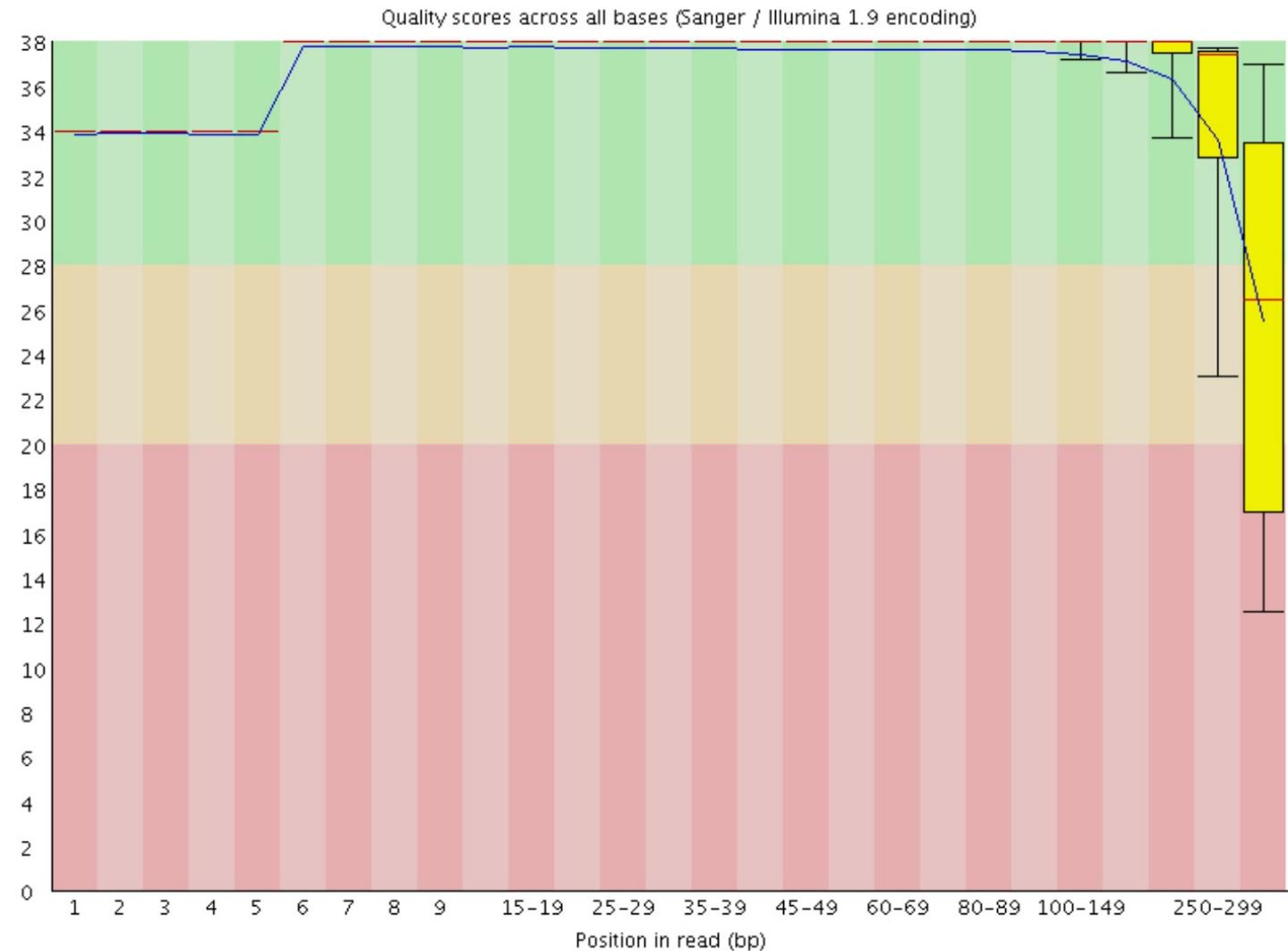
*Philip T.L.C. Clausen*

# Quick recap

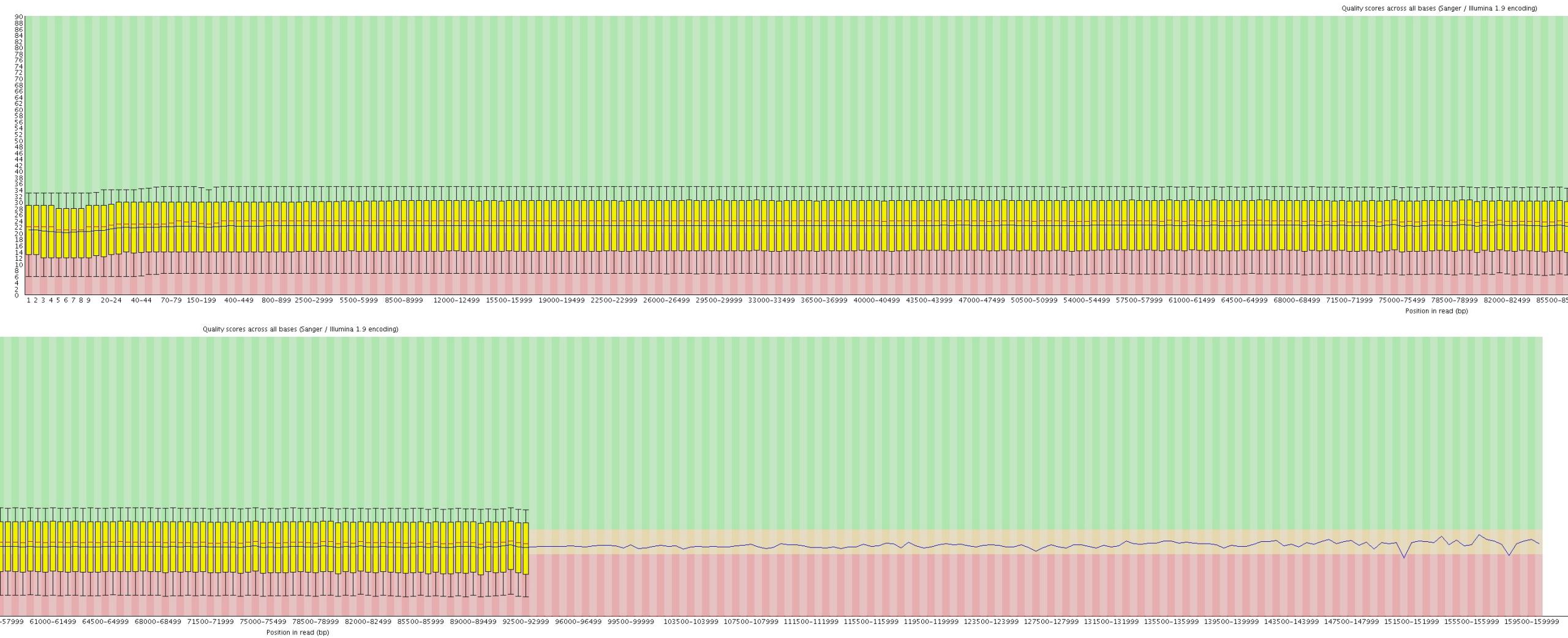
- Trimming of Illumina reads
- Trimming of ONT reads
- De Bruijn Graph assembly

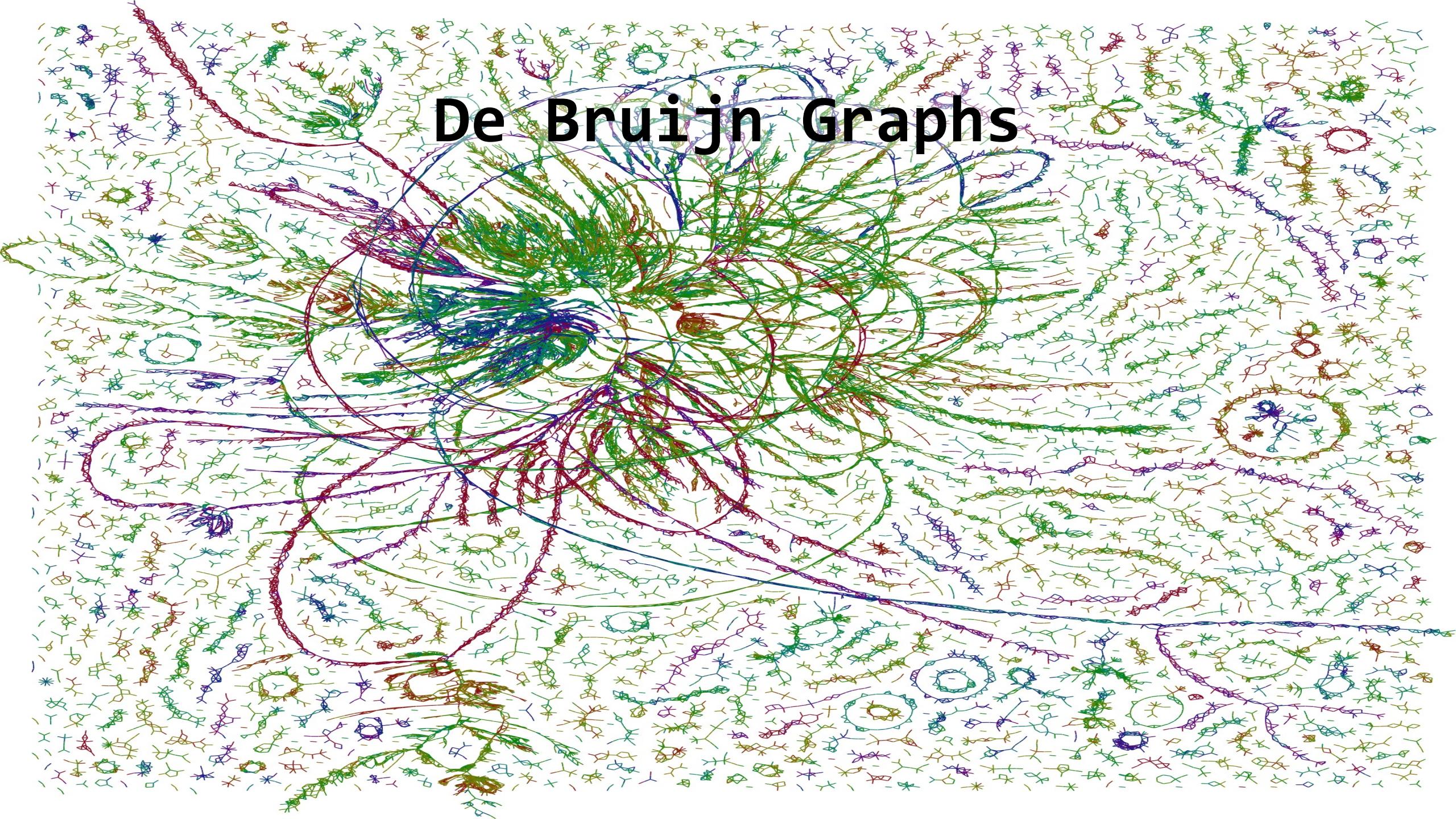


# Illumina fastqc report



# ONT fastqc report (16 threads)

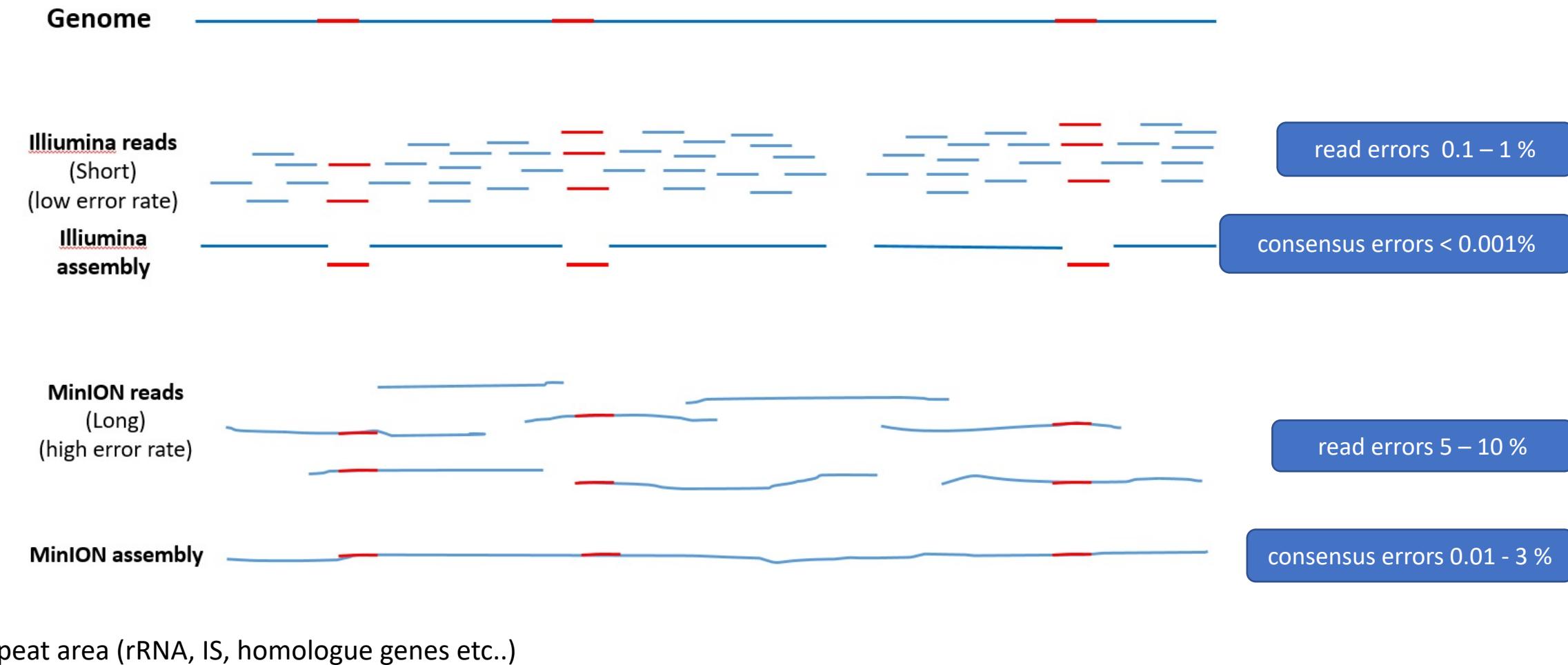




A dense, colorful De Bruijn graph visualization. The graph consists of numerous nodes represented by small, multi-colored shapes, primarily green, blue, and purple, connected by a complex web of thin, curved lines. These lines form various cycles and paths across the frame. Some nodes are highlighted with larger, more distinct shapes, such as a large green oval in the center-left and several circular nodes of different sizes scattered throughout. The overall effect is a chaotic yet structured pattern of interconnected elements.

# De Bruijn Graphs

# Technology impacts



```
pos[0] = 0;
for(m = t_len - 1, nuc_pos = t_e - 1; m >= 0; --m, --nuc_pos) {
    if(nuc_pos < 0) {
        nuc_pos = template_length - 1;
    }
    D_ptr[q_len] = (0 < k) ? 0 : (W1 + (t_len - 1 - m) * U);
    Q_prev = (t_len + q_len) * (MM + U + W1);

    t_nuc = getNuc(template, nuc_pos);
    for(n = q_len - 1; n >= 0; --n) {
        E_ptr[n] = 0;

        /* update Q and P, gap openings */
        Q = D_ptr[n + 1] + W1;
        P_ptr[n] = D_prev[n] + W1;
        if(Q < P_ptr[n]) {
            D_ptr[n] = P_ptr[n];
            e = 4;
        } else {
            D_ptr[n] = Q;
            e = 2;
        }

        /* update Q and P, gap extensions */
        /* mark bit 4 and 5 as possible gap-opennings, if necessary */
        thisScore = Q_prev + U;
        if(Q < thisScore) {
            Q = thisScore;
            if(e == 2)
                D_ptr[n] |= 1;
        }
        else {
            E_ptr[n] |= 16;
        }
        thisScore = P_prev[n] + U;
        if(P_ptr[n] < thisScore) {
            P_ptr[n] = thisScore;
            if(D_ptr[n] < thisScore) {
                D_ptr[n] = thisScore;
                e = 5;
            }
        }
        else {
            E_ptr[n] |= 32;
        }

        /* Update D, match */
        thisScore = D_prev[n + 1] + d[t_nuc][query[n]];
        if(D_ptr[n] < thisScore) {
            D_ptr[n] = thisScore;
            E_ptr[n] |= 1;
        }
        else {
            E_ptr[n] |= e;
        }
    }

    Q_prev = Q;
}

E_ptr -= (q_len + 1);

if(k < 0 && Stat.score <= *D_ptr) {
    Stat.score = *D_ptr;
    pos[0] = m;
}

tmp = D_ptr;
D_ptr = D_prev;
D_prev = tmp;

tmp = P_ptr;
P_ptr = P_prev;
P_prev = tmp;
}
E_ptr = E;
```

# Mapping and Alignment

*Philip T.L.C. Clausen*

# Content:

- Mapping and Chaining
- OLC and Alignment
- Polish and ConClave

# **Goal:**

- Know what the button does
- Know which button to push
- Join the competition

# Why:

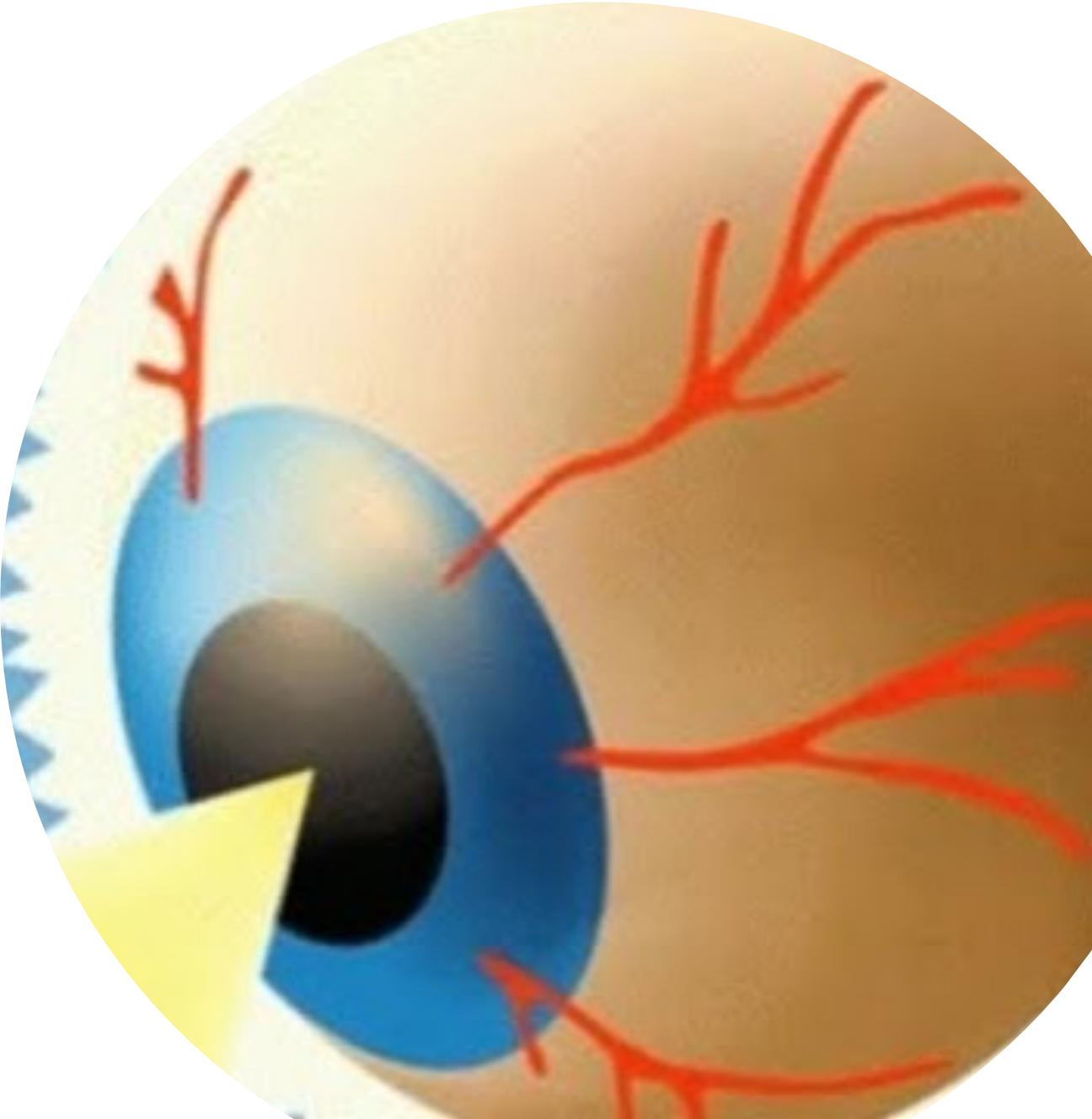
- The computer does only what we commands it to do
- We command the computer through programs
- The programs contains algorithms and methods

# Mapping and alignment

I want to compare two or more sequences,

and so what...

**1951**





**1966**

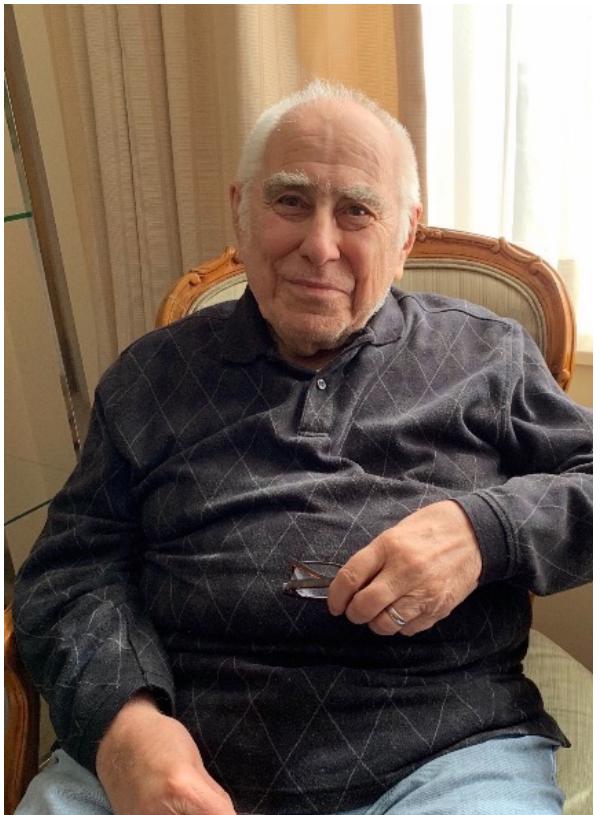
# Number of possible alignments

$$\binom{m+n}{n} = \frac{(m+n)!}{m! n!}$$



150 X 150

# 1970



Saul B. Needleman

$$D(i, j) = \max \begin{cases} D(i + 1, j + 1) + S(T_i, Q_j) \\ P(i, j) \\ Q(i, j) \end{cases}$$

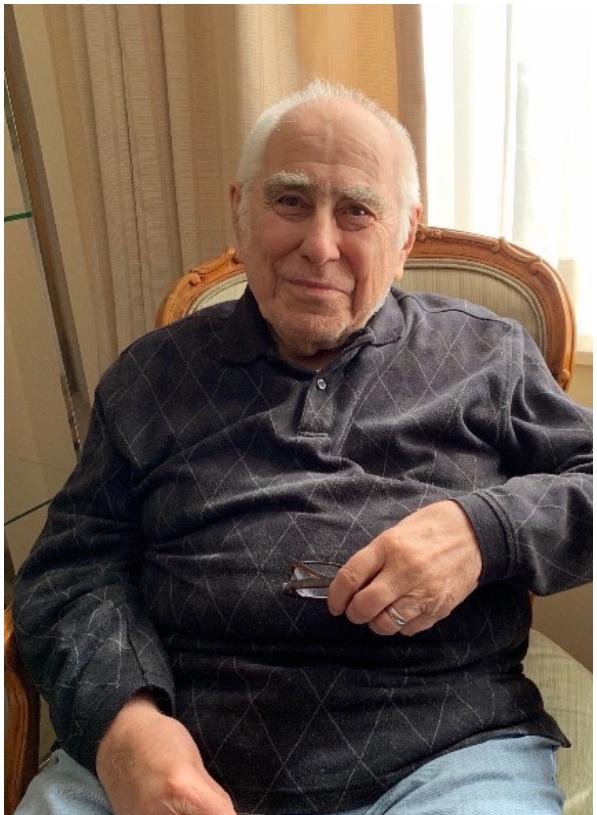
$$P(i, j) = \max \begin{cases} D(i + 1, j) + w_1 \\ P(i + 1, j) + U \end{cases}$$

$$Q(i, j) = \max \begin{cases} D(i, j + 1) + w_1 \\ Q(i, j + 1) + U \end{cases}$$



Christian D. Wunsch

# Simplified NW



Saul B. Needleman

$$D(i, j) = \max \begin{cases} D(i + 1, j + 1) + S(T_i, Q_j) \\ D(i + 1, j) + W \\ D(i, j + 1) + W \end{cases}$$



Christian D. Wunsch

# What if we made the two sequences form a matrix?

$$D(i, j) = \max \begin{cases} D(i + 1, j + 1) + S(T_i, Q_j) \\ D(i + 1, j) + W \\ D(i, j + 1) + W \end{cases}$$
$$S(x, y) = \begin{cases} M & x = y \\ P & x \neq y \end{cases}$$
$$M = 1, P = -1, W = -3$$

Q\T	A	C	G	T	-
A					
G					
T					
T					
-					0

# What if we made the two sequences form a matrix?

$$D(i, j) = \max \begin{cases} D(i + 1, j + 1) + S(T_i, Q_j) \\ D(i + 1, j) + W \\ D(i, j + 1) + W \end{cases}$$
$$S(x, y) = \begin{cases} M & x = y \\ P & x \neq y \end{cases}$$
$$M = 1, P = -1, W = -3$$

Q\T	A	C	G	T	-
A					-12
G					-9
T					-6
T					-3
-	-12	-9	-6	-3	0

# What if we made the two sequences form a matrix?

$$D(i, j) = \max \begin{cases} D(i + 1, j + 1) + S(T_i, Q_j) \\ D(i + 1, j) + W \\ D(i, j + 1) + W \end{cases}$$

$S(x, y) = \begin{cases} M & x = y \\ P & x \neq y \end{cases}$

$M = 1, P = -1, W = -3$

$$D(4, 4)$$

$$= \max \begin{cases} 0 + S(T, T) \\ -3 + (-3) \\ -3 + (-3) \end{cases}$$

$$= \max \begin{cases} 0 + 1 \\ -6 = 1 \\ -6 \end{cases}$$

Q\T	A	C	G	T	-
A					-12
G					-9
T					-6
T					-3
-	-12	-9	-6	-3	0

# What if we made the two sequences form a matrix?

$$D(i, j) = \max \begin{cases} D(i + 1, j + 1) + S(T_i, Q_j) \\ D(i + 1, j) + W \\ D(i, j + 1) + W \end{cases}$$

$S(x, y) = \begin{cases} M & x = y \\ P & x \neq y \end{cases}$

$M = 1, P = -1, W = -3$

$$D(4, 4)$$

$$= \max \begin{cases} 0 + S(T, T) \\ -3 + (-3) \\ -3 + (-3) \end{cases}$$

$$= \max \begin{cases} 0 + 1 \\ -6 \\ -6 \end{cases} = 1$$

Q\T	A	C	G	T	-
A					-12
G					-9
T					-6
T				1	-3
-	-12	-9	-6	-3	0

# What if we made the two sequences form a matrix?

$$D(i, j) = \max \begin{cases} D(i + 1, j + 1) + S(T_i, Q_j) \\ D(i + 1, j) + W \\ D(i, j + 1) + W \end{cases}$$

$$S(x, y) = \begin{cases} M & x = y \\ P & x \neq y \end{cases}$$

$$M = 1, P = -1, W = -3$$

$$D(4, 3)$$

$$= \max \begin{cases} -3 + S(T, G) \\ -6 + (-3) \\ 1 + (-3) \end{cases}$$

$$= \max \begin{cases} -3 + (-1) \\ -9 \\ -2 \end{cases} = -2$$

Q\T	A	C	G	T	-
A					-12
G					-9
T					-6
T				1	-3
-	-12	-9	-6	-3	0

# What if we made the two sequences form a matrix?

$$D(i, j) = \max \begin{cases} D(i + 1, j + 1) + S(T_i, Q_j) \\ D(i + 1, j) + W \\ D(i, j + 1) + W \end{cases}$$

$$S(x, y) = \begin{cases} M & x = y \\ P & x \neq y \end{cases}$$

$$M = 1, P = -1, W = -3$$

$$D(4, 3)$$

$$= \max \begin{cases} -3 + S(T, G) \\ -6 + (-3) \\ 1 + (-3) \end{cases}$$

$$= \max \begin{cases} -3 + (-1) \\ -9 \\ -2 \end{cases} = -2$$

Q\T	A	C	G	T	-
A					-12
G					-9
T					-6
T			-2	1	-3
-	-12	-9	-6	-3	0

# What if we made the two sequences form a matrix?

$$D(i, j) = \max \begin{cases} D(i + 1, j + 1) + S(T_i, Q_j) \\ D(i + 1, j) + W \\ D(i, j + 1) + W \end{cases}$$

$$S(x, y) = \begin{cases} M & x = y \\ P & x \neq y \end{cases}$$

$$M = 1, P = -1, W = -3$$

T: ACGT  
|\_\_|  
Q: AGTT

Q\T	A	C	G	T	-
A	0	-2	-4	-8	-12
G	-4	-1	-1	-5	-9
T	-6	-3	0	-2	-6
T	-8	-5	-2	1	-3
-	-12	-9	-6	-3	0

# And now you...

$$D_{ij} = \max \begin{cases} D_{i+1,j+1} + S(T_{i'} Q_j) \\ D_{i+1,j} + W \\ D_{i,j+1} + W \end{cases}, \quad S(x, y) = \begin{cases} M & x = y \\ P & x \neq y \end{cases}$$

**BREAK...**

# Drastic improvement on computation speed

$$\frac{(m + n)!}{m! \ n!} \gg m \ n$$

Sequence lengths	Needleman-Wunsch	Exhaustive search	Number of e <sup>-</sup> in the universe
150 X 150	22 500	$\sim 10^{89}$	$\sim 10^{80}$

**Is a run time of “m n” good enough?**



1983

David J. Lipman

# Exact matching is fast

- Feasible for entire sequences?
- What if we divide the sequences into overlapping subsequences ( $k$ -mers).
- And make an index revealing where each subsequence belongs.

*K*-mer, where  $k \in \mathbb{N}$ , here eight

CGATTTAATATTATGAAATATATTAAAGGGGAACGTATTACTTATGAGCAGT

1 CGATTTAA

2 GATTTAAT

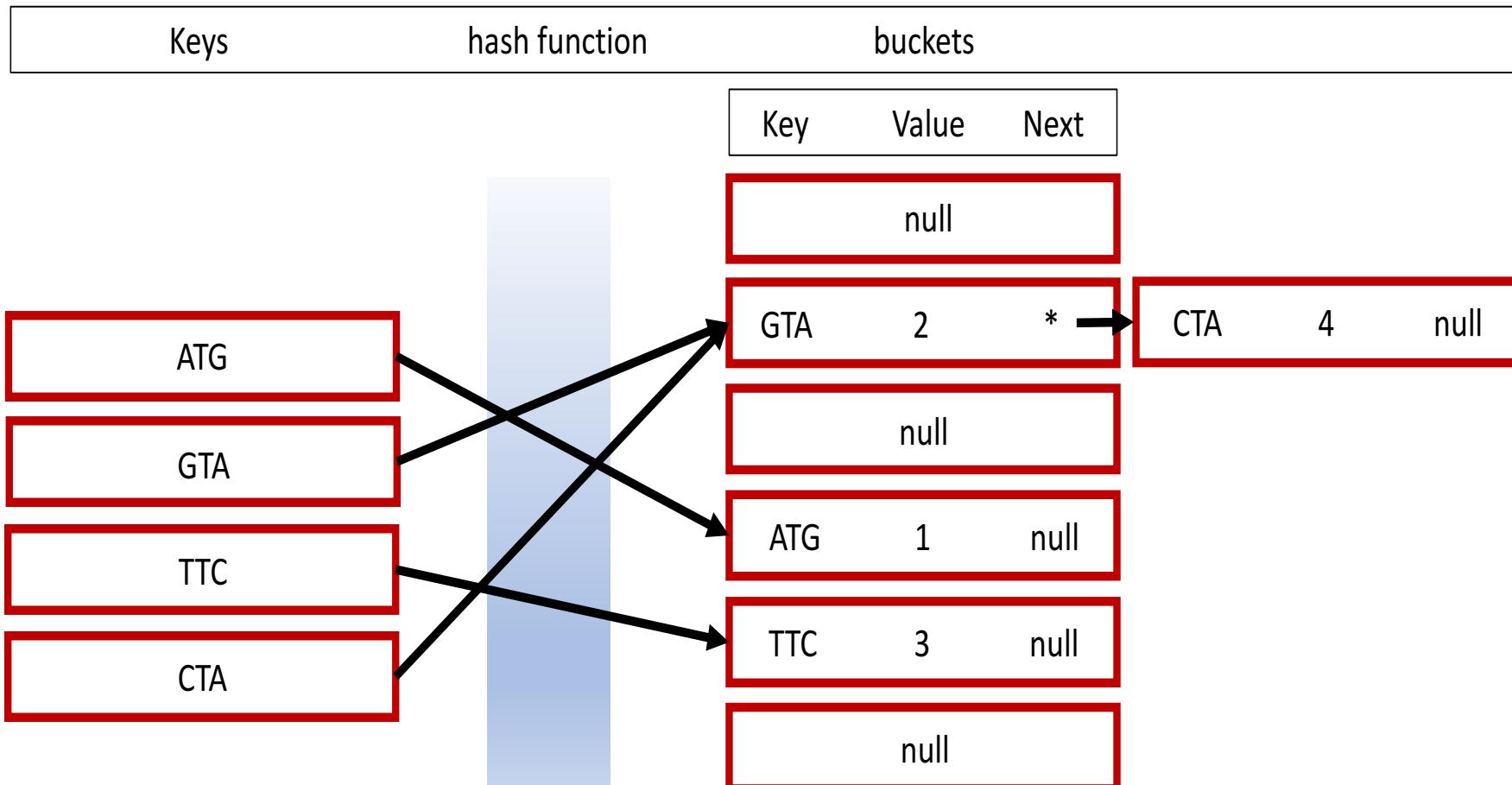
3 ATTTAATA

4 TTTAATAT

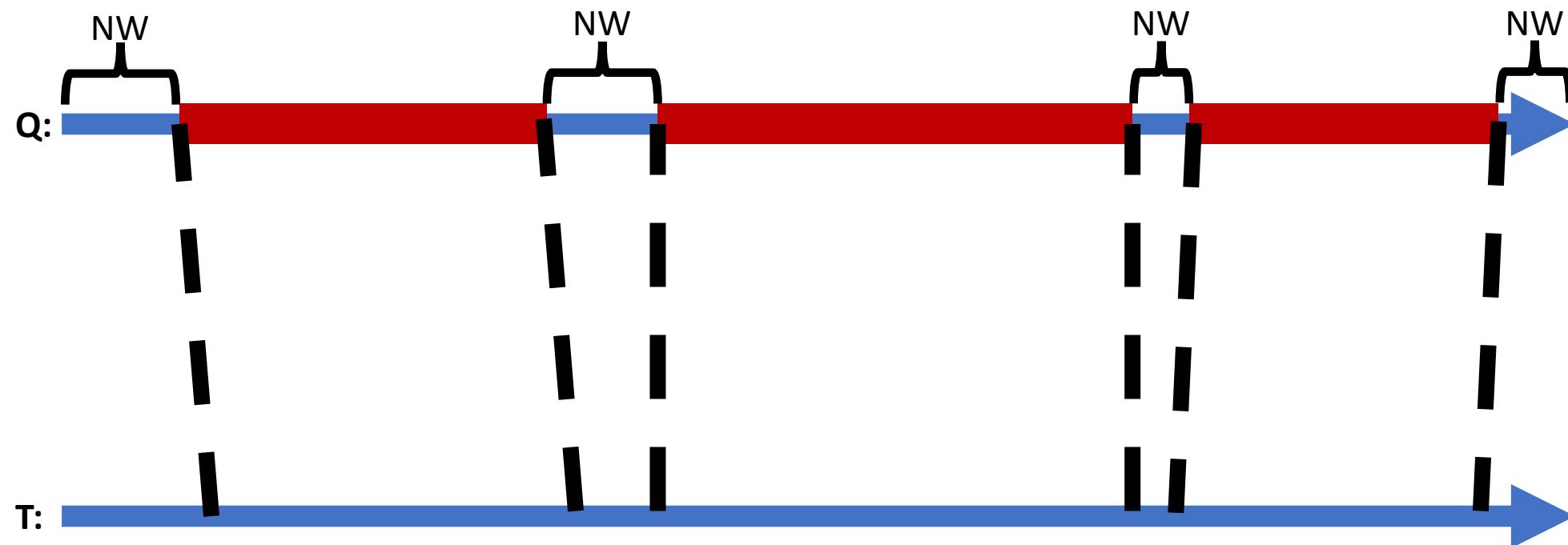
5 TTAATATT

... ... ... ... ... ... ... ...

# Store k-mers in hashmaps



# Fasta, 1985



# Seed and extend

Simple example, k = 5:

Q: AAGACTCCGACTGGGACTTTGATGTTCGAAAGA

T: GACTGGGACTTTGATG

# Seed and extend

Simple example,  $k = 5$ :

Q: AAGACTCCGACTGGGACTTTGATGTTCGAAAGA

T: GACTGGGACTTTGATG

# Seed and extend

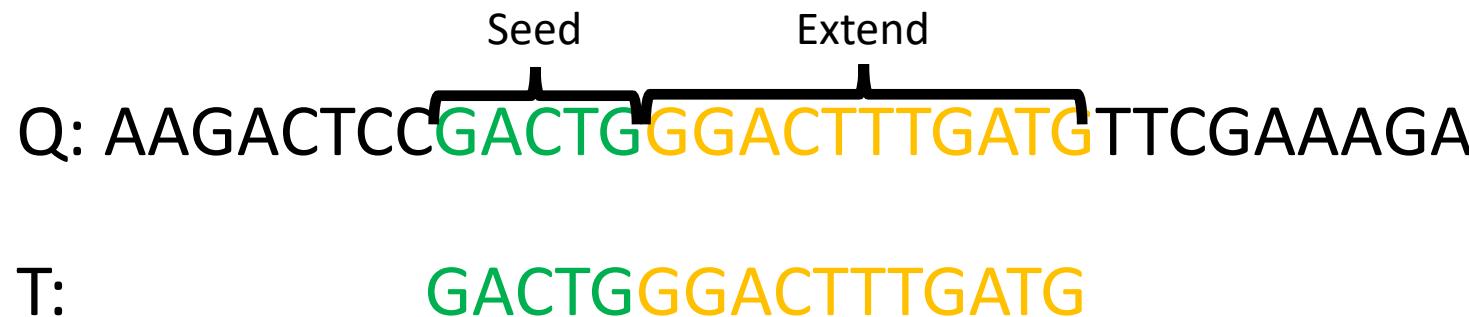
Simple example, k = 5:

Q: AAGACTCC**GACTG**GGACTTGTGATGTTCGAAAGA

T: **GACTG**GGACTTGTGATG

# Seed and extend

Simple example,  $k = 5$ :



Extended  $k$ -mer matches are also noted as Anchors, Maximum Exact Matches (MEMs), or Maximum Unique Matches (MUMs) if they are uniquely occurring in the reference.

# Seed and extend, impact on computation

$$m \cdot n \gg m + n$$

# And now you...

3-piece handout:

Reads, Reference and Index

Index is alphabetically sorted  $k$ -mers of all 10-mers in the reference prefixed with ‘T’.

Find identify the  $k$ -mer matches between reads and reference using the index,  
and extend on both sides of the matches.

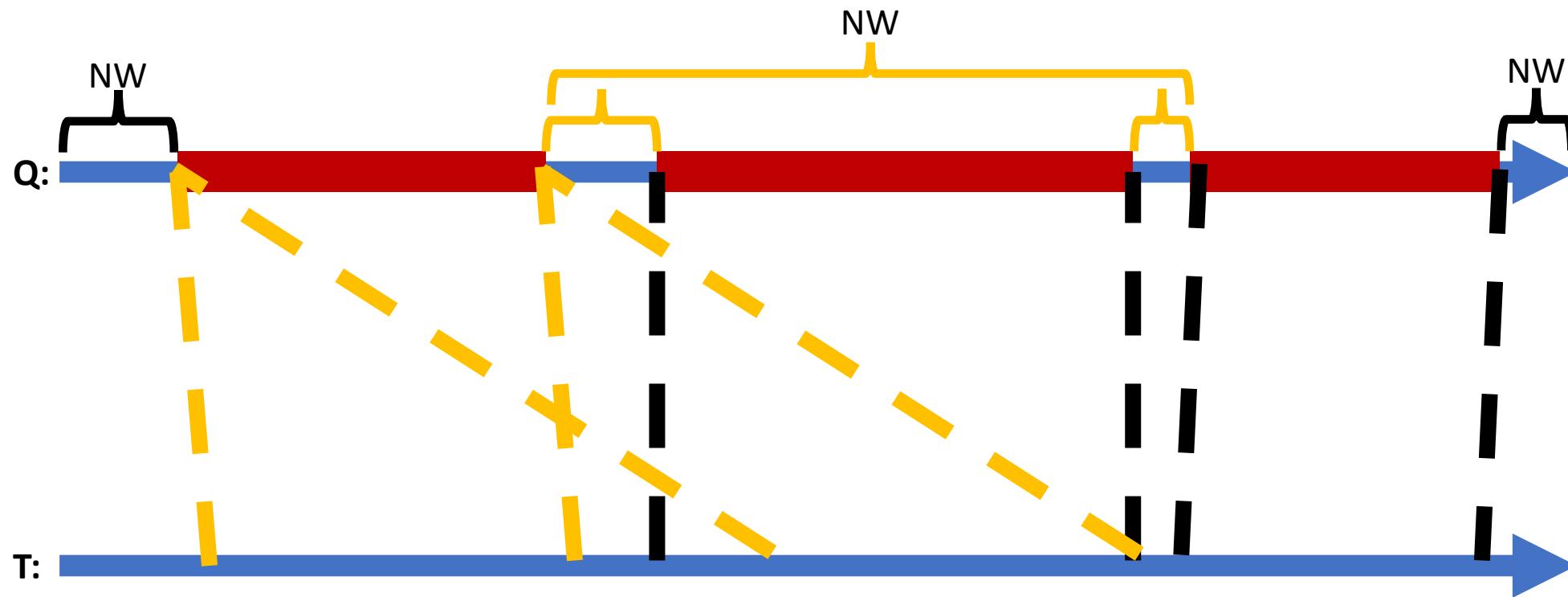
When the  $k$ -mer cannot be extended further, find the next  $k$ -mer match.

$$K = 10 \quad , \quad \text{prefix : } T$$

TAAAAGATGC	91	TCTAGCTCC	593	TGTTTTGCT	59
TAAGTTCTG	209	TCTAACAGC	143	TTAAAGTTCT	208
TAAGTTGCA	635	TCTCAGAATG	285	TTAACTGGCG	573
AAAATCTGA	692	TCTCGCGGTA	717	TTAACGATTG	847
TAACACTGCG	386	TCTGACAACG	407	TTAATAGACT	612
TAACCATGAG	373	TCTGCGCTG	653	TTACATCGAA	128
TAACCGCTT	436	TCTGCTATGT	215	TTACGGATGG	331
TAACTCGCCT	469	TCTGGAGCCG	696	TTACTCTAGC	589
TAACTGGCGA	574	TCTTACGGAT	329	TTACTTCTGA	402
TAAGAGAATT	349	TGAAAGTAAA	85	TTATCCCCTG	234
TAAGATCCTT	155	TGAACGAAAT	806	TTATCTACAC	772
TAAGCATTGG	848	TGAAGATCAG	101	TTATGCAGTG	357
TAAGCCCTCC	752	TGAAGCCATA	503	TTATTCCCTT	28
TAATAGACTG	613	TGAATGAAGC	499	TTATTGCTGA	682
TACACGACGG	777	TGACAACGAT	409	TTCAACATT	7
TACACTATT	277	TGACACCACG	527	TTCCAATGAT	190
TACATCGAAC	129	TGACAGTAAG	343	TTCCCGGCAA	599
TACCAAACGA	511	TGACGCCGGG	245	TTCCCTTTT	31
TACGGATGGC	332	TGACTTGGTT	293	TTCCGGCTGG	667
TACTCACCA	306	TGAGAGTTT	164	TTCCGTGTG	15
TACTCTAGCT	590	TGAGATAGGT	827	TTCCCTGTTT	55
TACTTACTCT	586	TGAGCACTT	199	TTCGCCCCGA	172
TACTTCTGAC	403	TGAGCGTGGG	707	TTCTCAGAAT	284
TAGACAGATC	815	TGAGTACTCA	302	TTCTGACAAC	406
TAGACTGGAT	616	TGAGTATTCA	1	TTCTGCGCTC	652
TAGCTTCCCG	595	TGAGTGATAA	379	TTCTGCTATG	214
TAGGTGCTC	832	TGATAAATCT	689	TTGACGCCGG	244
TAGTTATCTA	769	TGATAACACT	383	TTGAGAGTTT	163
TATCATTGCA	725	TGATCGTTG	479	TTGAGTACTC	301
TATCCCGTGT	235	TGATGAGCAC	196	TTGATCGTTG	478
TATCGTAGTT	764	TGATTAAGCA	844	TTGCACAACA	447
TATCTACACG	773	TGCACAACT	448	TTGCAGCACT	730
TATCGAGTGC	358	TGCACGAGTG	116	TTGCAGGACC	640
TATGGATGAA	800	TGCAGCAATG	542	TTGCCTTCCT	50
TATGTGGCGC	220	TGCAGCACTG	731	TTGCGCAAAC	561
TATTAACCTG	571	TGCAGGACCA	641	TTGCGGCATT	40
TATTATCCCG	232	TGCAGTGCTG	360	TTGCTCACCC	64
TATTCACAT	5	TGCCATAACC	368	TTGCTGATAA	685
TATTCCTTT	29	TGCCTCACTG	836	TTGGGAACCG	485
TATTCTCAGA	282	TGCCTGCAGC	538	TTGGGTGAC	111
TATTGCTGAT	683	TGCCTTCCTG	51	TTGGTTGAGT	297
TCAACAGCGG	145	TGCGCAAAC	562	TTAAAGTTC	207
TCAACATTTC	8	TGCGCTCGGC	655	TTTATTGCTG	681
TCACAGAAAA	316	TGCGGCATTT	41	TTTCCAATGA	189
TCACCACTCA	309	TGCGGCCAAC	392	TTTCCGTGTC	14
TCACCCAGAA	68	TGCTATGTGG	217	TTTCGCCCCG	171
TCACTGATTA	840	TGCTCACCCA	65	TTTGCACAAC	446
TCAGAATGAC	287	TGCTGAAGAT	98	TTTGCCTTCC	49

**BREAK...**

# There is some more to it...



# Seed-Extend issues

- MEMs might be placed several places in the reference.
- Some MEMs might be erroneous mappings (frequent with smaller  $k$ -mers).
- Depending on the mapping method, some MEMs might have been missed (i.e. too large  $k$ -mer).

# Chaining, 90's

$$f(i) = \max \left\{ \max_{i > j \geq 1} \{ f(j) + \alpha(j, i) - \beta(j, i) \}, w_i \right\}$$

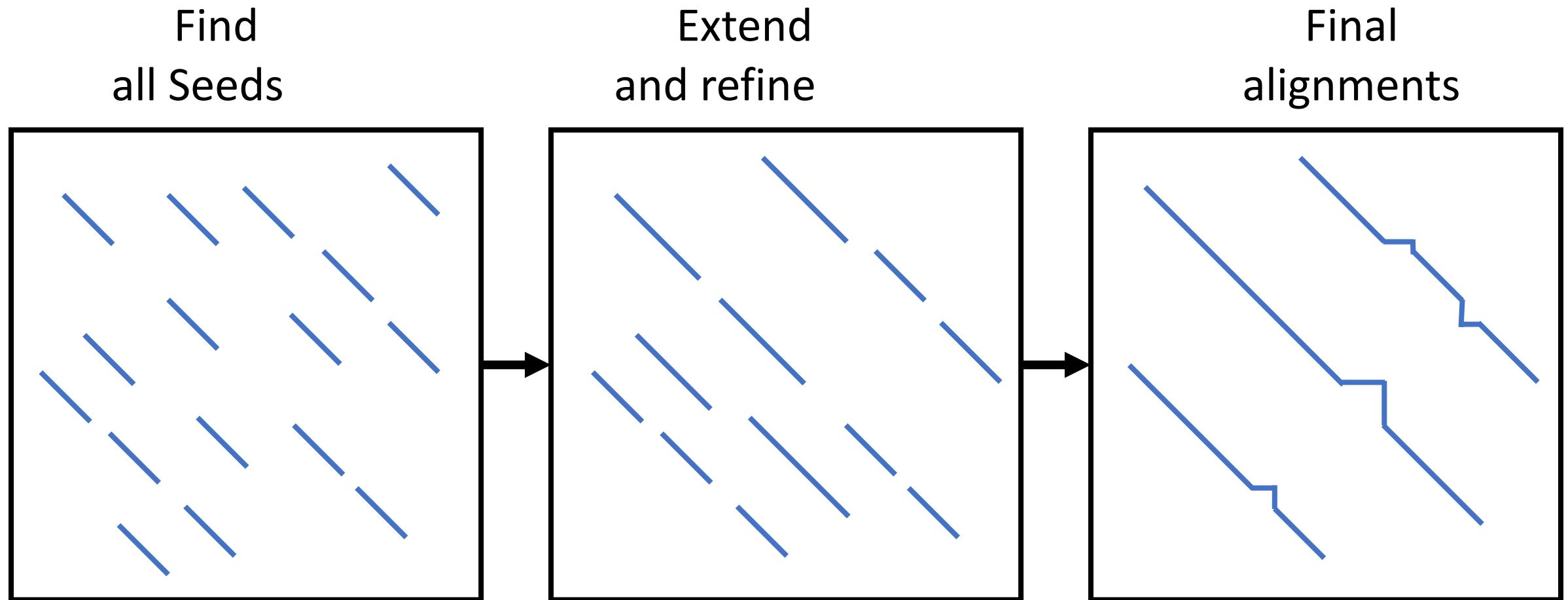
$$\alpha(j, i) = \min \{ \min \{ y_i - y_j, x_i - x_j \}, w_i \}$$

$$\beta(j, i) = \gamma_c ((y_i - y_j) - (x_i - x_j))$$

# The Idea of Chaining

- Co-linear anchors (MEMs) are likely to belong together and produce high quality alignments, and can be chained together.
- Distantly placed (horizontally or vertically) placed anchors are likely to be erroneous mappings, and will be unlikely to produce high quality overlaps if chained together.

# Mapping and Alignment



# Methods these algorithms cover

- BLAST
  - $k$ -mer mapping, chaining, alignment
- BWA-MEM / Bowtie2
  - FM-mapping, chaining, alignment
- Minimap2
  - minimizer mapping, chaining, alignment
- GraphMap
  - $k$ -mer mapping, chaining, alignment

# Methods these algorithms cover

- **BLAST**  
*k*-mer mapping, chaining, alignment
- BWA-MEM / Bowtie2  
FM-mapping, chaining, alignment
- **Minimap2**  
minimizer mapping, chaining, alignment
- **GraphMap**  
*k*-mer mapping, chaining, alignment

**Why does Minimap(2) sort anchors  
w.r.t. to query positions  
prior to chaining?**

**Does BLAST use the same strategy...**

`-max_target_seqs`

# **Misunderstood parameter of NCBI BLAST impacts the correctness of bioinformatics workflows**

*Nidhi Shah, Michael G Nute, Tandy Warnow, Mihai Pop*

# OLC, Combined with last week

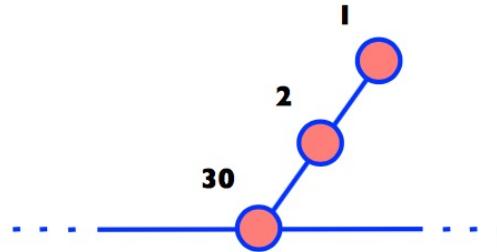
- Overlap Layout Consensus (OLC) assemblies
- Identify overlaps between reads with Minimap2 (mapping and chaining).
- Form unitigs with Miniasm.

# Miniasm

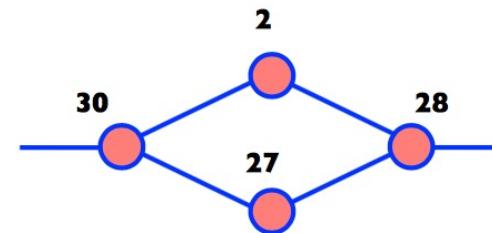
- Trim low depth regions (and tips).
- Drop contained reads.
- Remove transitive edges.
- Pop short bubbles, favor largest overlap when forming the consensus.

# Basic Graph Trimming

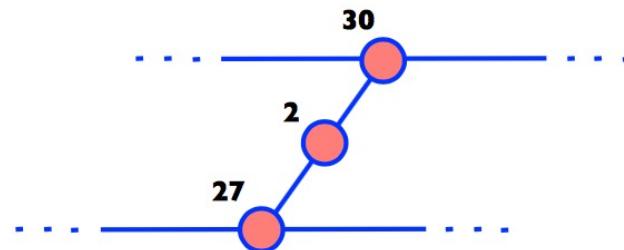
Clip tips  
(seq err, end)



Pinch bubbles  
(seq err, middle, SNP)



Remove low cov. links



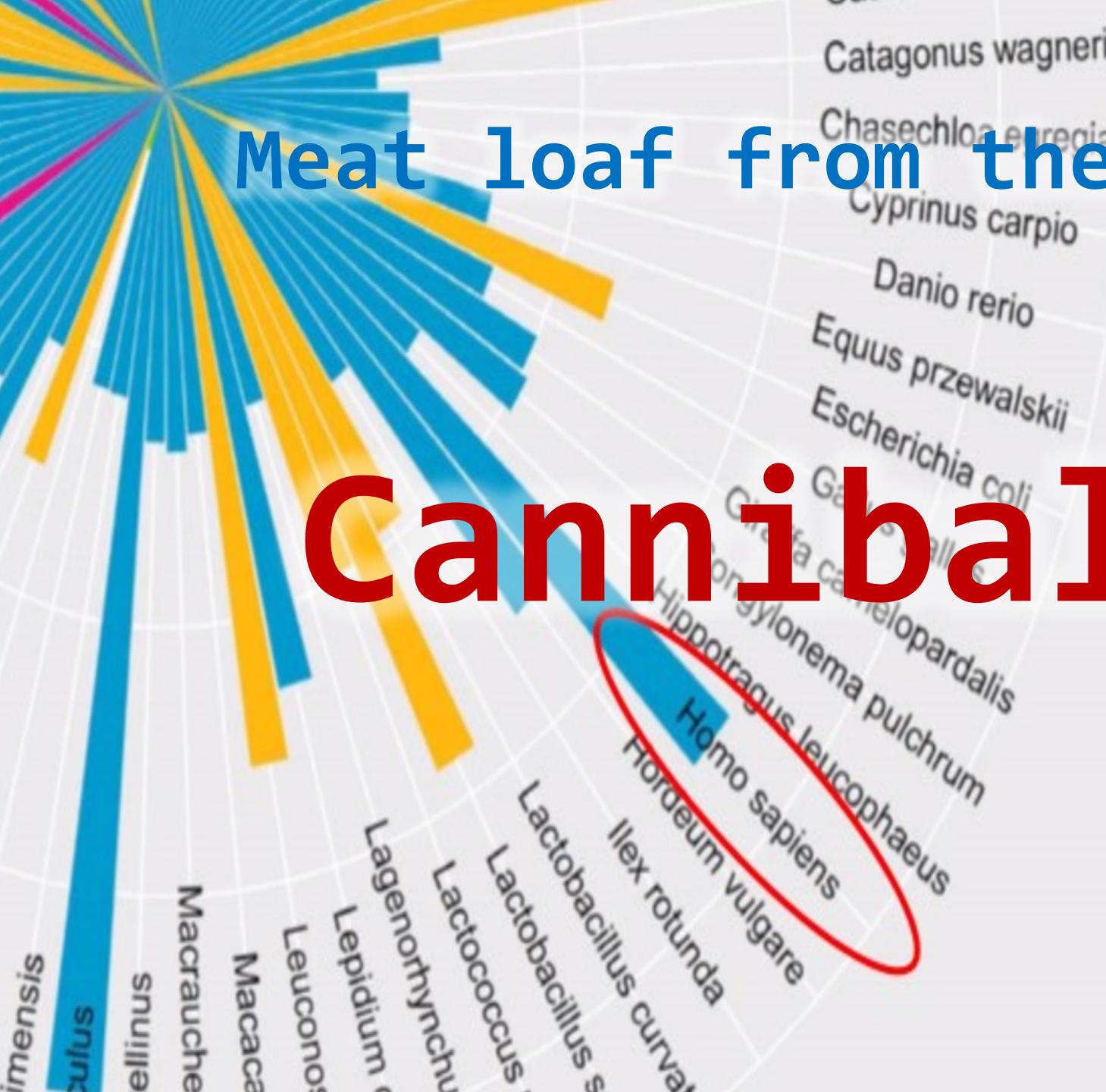
# Miniasm, pitfalls

- Quick and dirty solution:
  - Gives the overall synteny of the assembled sequences.
  - Usually followed by realigning the reads to assembly iteratively.
    - E.g. Minimap2 or KMA
  - “Polishers” exist that can correct for some of the common ONT errors.
    - E.g. Medaka or Pilon

**BREAK...**

# The redundant sequences

- Use the mapping and alignment methods just introduced.
- Something else?



# Meat loaf from the FLI canteen

# Cannibalism...

- 183 blaTEM variants

- Max distance of 38 SNPs

- 150 bp subsequence from the middle of blaTEM-1A:

GTCCTGCAACTTATCCGCCTCCATCCAGTCTATTAAATTGTTGCCGGGAAGCTAGAGTAAGTAGTTGCCAGTTA  
ATAGTTGCGAACGTTGCCATTGCTGCAGGCATCGTGGTGTACCGCTCGTCTTGGTATGGCTTCATTCA

- 131 perfect matches.



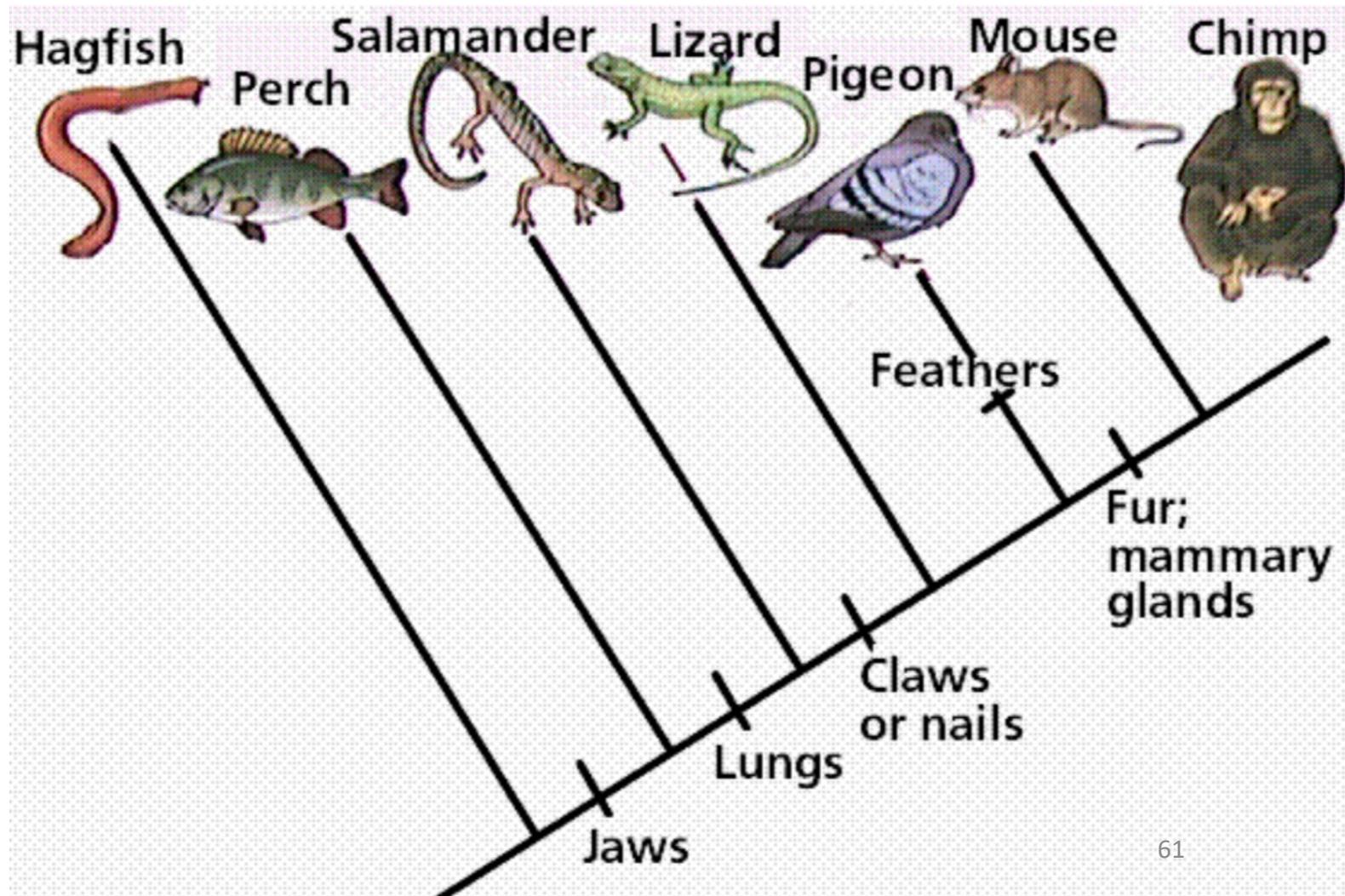
# ConClave, 2018



# Find best candidates

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \}$$

- Feathers
  - $T_m \in \{\text{Pigeon}\}$
- Jaws
  - $T_m \in \{\text{Perch, Salamander, Lizard, Pigeon, Mouse, Chimp}\}$
- Lungs
  - $T_m \in \{\text{Salamander, Lizard, Pigeon, Mouse, Chimp}\}$



# Calculate the total score of all reads against the best candidates

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases}$$

- Feathers
  - $T_m \in \{\text{Pigeon}\}$
- Jaws
  - $T_m \in \{\text{Perch, Salamander, Lizard, Pigeon, Mouse, Chimp}\}$
- Lungs
  - $T_m \in \{\text{Salamander, Lizard, Pigeon, Mouse, Chimp}\}$

t	C(t)
Hagfish	0
Perch	1
Salamander	2
Lizard	2
Pigeon	3
Mouse	2
Chimp	2

# Calculate the total score of all reads against the best candidates

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \}$$

- Feathers
  - $T_m \in \{\text{Pigeon}\}$
  - $S_k \in \{\text{Pigeon}\}$
- Jaws
  - $T_m \in \{\text{Perch, Salamander, Lizard, Pigeon, Mouse, Chimp}\}$
  - $S_k \in \{\text{Pigeon}\}$
- Lungs
  - $T_m \in \{\text{Salamander, Lizard, Pigeon, Mouse, Chimp}\}$
  - $S_k \in \{\text{Pigeon}\}$

t	C(t)
Hagfish	0
Perch	1
Salamander	2
Lizard	2
Pigeon	3
Mouse	2
Chimp	2

# And now you...

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \} \quad (1)$$

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases} \quad (2)$$

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \} \quad (3)$$

Query sequence

		Score	Templates
GCGCGGTATTATCCGTGTTGACGCCGGCAAGAGTAACTCGGTCGCCGCATAACTATTCTCAGAATGACTTG	64	2,12	
GAATGACTTGCTTGAGTACTCACCAAGTCACAGAAAAGCATCTTACGGATGGCATGACAGTAAGAGAATTATGCA	63	1,2,3,7,8,9	
TAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTGCACAACATTG	72	1,2,3,4,5,6,8,9,11,12	
CCACTTCTGCGCTCGGCCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCG	66	2,5	
TCCTTGAGAGTTTCGCCCCGAAGAACGTTTCCAATGATGAGCACTTTAAAGTTCTGCTATGTGGCGCGGT	74	2,5,12	
GGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTGCACAACATTGGGAATCATG	63	1,2,3,4,5,6,8,9,11,12	
ACTCTAGCTTCCGGCAACAATTAAATAGACTGGATGGAGGGGATTAAGTTGCAGGACCACTCTGCCTCGGC	39	1,2,3,4,6,7,8,9,10,11	
CCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCGCGGTATTCAGCAC	71	2,5,12	

# Templates:

01 blaTEM-1A\_4\_HM749966

ConClave Scores:

\_\_\_\_\_

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \} \quad (1)$$

02 blaTEM-2\_1\_X54606

\_\_\_\_\_

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases} \quad (2)$$

03 blaTEM-2\_3\_AJ251946

\_\_\_\_\_

04 blaTEM-3\_1\_X64523

\_\_\_\_\_

05 blaTEM-6\_1\_X57972

\_\_\_\_\_

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \} \quad (3)$$

06 blaTEM-8\_1\_X65252

\_\_\_\_\_

07 blaTEM-10\_1\_AF093512

\_\_\_\_\_

08 blaTEM-11\_1\_AY874537

\_\_\_\_\_

09 blaTEM-12\_1\_M88143

\_\_\_\_\_

10 blaTEM-15\_1\_AM849805

\_\_\_\_\_

11 blaTEM-16\_1\_X65254

\_\_\_\_\_

12 blaTEM-17\_1\_Y14574

\_\_\_\_\_

Predicted Template: \_\_\_\_\_

Query sequence	Score	Templates
GCGCGGTATTATCCCGTGGTACGCCGGCAAGAGTAACCTGGTCGCCGCATAACACTATTCTCAGAATGACTTG	64	2,12
GAATGACTTGTGCTTGAGTACTCACCAAGTACACAGAAAAGCATCTAACGGATGGCATGACAGTAAGAGAAATTATGCA	63	1,2,3,7,8,9
TAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTTGACAAACATTG	72	1,2,3,4,5,6,8,9,11,12
CCACTTCTCGCTCGGCCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCG	66	2,5
TCCTTGAGAGTTTCGCCCGAAGAACGTTTCCAATGATGAGCACTTTAAAGTTCTGCTATGTGGCGCGGT	74	2,5,12
GGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTTGACAAACATTGGGAATCATG	63	1,2,3,4,5,6,8,9,11,12
ACTCTAGCTTCCGGCAACAATTAAATAGACTGGATGGAGGGGATTAAGTTGCAGGACCACTCTGCCTCGGC	39	1,2,3,4,6,7,8,9,10,11
CCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCGCGGTATTCAGCAC	71	2,5,12

# Templates:

01 blaTEM-1A\_4\_HM749966

02 blaTEM-2\_1\_X54606

03 blaTEM-2\_3\_AJ251946

04 blaTEM-3\_1\_X64523

05 blaTEM-6\_1\_X57972

06 blaTEM-8\_1\_X65252

07 blaTEM-10\_1\_AF093512

08 blaTEM-11\_1\_AY874537

09 blaTEM-12\_1\_M88143

10 blaTEM-15\_1\_AM849805

11 blaTEM-16\_1\_X65254

12 blaTEM-17\_1\_Y14574

ConClave Scores:

\_\_\_\_\_

64

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

64

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \} \quad (1)$$

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases} \quad (2)$$

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \} \quad (3)$$

Predicted Template: \_\_\_\_\_

Query sequence		Score	Templates
GCGCGGTATTATCCCGTGGT GACGCCGGGCAAGAGTAAC TCGGTGCCGCATA CACTATTCTCAGAATGACTTG	64	2, 12	
GAATGACTTGCTTGAGTACTCACCA GTCACAGAAAAGCATCTTACGGATGGCATGACAGTAAGAGAATTATGCA	63	1, 2, 3, 7, 8, 9	
TAAACACTGCGGCCAAC TACTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTTGACAAACATTTG	72	1, 2, 3, 4, 5, 6, 8, 9, 11, 12	
CCACTTCTCGCTCGGCCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCG	66	2, 5	
TCCTTGAGAGTTTCGCCCGAAGAACGTTTCCAATGATGAGCACTTTAAAGTTCTGCTATGTGGCGCGGT A	74	2, 5, 12	
GGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTTGACAAACATTGGGAATCATG	63	1, 2, 3, 4, 5, 6, 8, 9, 11, 12	
ACTCTAGCTTCCCGCAACAATTAA TAGACTGGATGGAGGGGATTAAGTTGCAGGACCACTCTGC GCTCGGC	39	1, 2, 3, 4, 6, 7, 8, 9, 10, 11	
CCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCGCGGTATTCAGCAC	71	2, 5, 12	

# Templates:

01 blaTEM-1A\_4\_HM749966

ConClave Scores:

63

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \} \quad (1)$$

02 blaTEM-2\_1\_X54606

64+63

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases} \quad (2)$$

03 blaTEM-2\_3\_AJ251946

63

04 blaTEM-3\_1\_X64523

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \} \quad (3)$$

05 blaTEM-6\_1\_X57972

06 blaTEM-8\_1\_X65252

07 blaTEM-10\_1\_AF093512

63

08 blaTEM-11\_1\_AY874537

63

09 blaTEM-12\_1\_M88143

63

10 blaTEM-15\_1\_AM849805

11 blaTEM-16\_1\_X65254

12 blaTEM-17\_1\_Y14574

64

Predicted Template: \_\_\_\_\_

Query sequence	Score	Templates
GCGCGGTATTATCCCGTGGT GACGCCGGCAAGAGTAAC TCGGTCGCCGCATA CACTATTCTCAGAATGACTTG	64	2,12
GAATGACTTGTGAGTACTCACCAAGTCACAGAAAAGCATCTTACGGATGGCATGACAGTAAGAGAAATTATGCA	63	1 2 3 7 8 9
TAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTTGACAAACATTG	72	1,2,3,4,5,6,8,9,11,12
CCACCTCTGCGCTCGGCCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCG	66	2,5
TCCTTGAGAGTTTCGCCCGAAGAACGTTTCCAATGATGAGCACTTTAAAGTTCTGCTATGTGGCGCGGT	74	2,5,12
GGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTTGACAAACATTGGGAATCATG	63	1,2,3,4,5,6,8,9,11,12
ACTCTAGCTTCCCGCAACAAATTAAATAGACTGGATGGAGGGGATTAAGTTGCAGGACCACTCTGCGCTCGGC	39	1,2,3,4,6,7,8,9,10,11
CCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCGCGGTATTCAGCAC	71	2,5,12

# Templates:

01 blaTEM-1A\_4\_HM749966

02 blaTEM-2\_1\_X54606

03 blaTEM-2\_3\_AJ251946

04 blaTEM-3\_1\_X64523

05 blaTEM-6\_1\_X57972

06 blaTEM-8\_1\_X65252

07 blaTEM-10\_1\_AF093512

08 blaTEM-11\_1\_AY874537

09 blaTEM-12\_1\_M88143

10 blaTEM-15\_1\_AM849805

11 blaTEM-16\_1\_X65254

12 blaTEM-17\_1\_Y14574

ConClave Scores:

63+72

64+63+72

63+72

72

72

72

63

63+72

63+72

72

64+72

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \} \quad (1)$$

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases} \quad (2)$$

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \} \quad (3)$$

Predicted Template: \_\_\_\_\_

Query sequence	Score	Templates
GCGCGGTATTATCCCGTGTGACGCCGGCAAGAGTAACCTGGTCGCCGCATAACTATTCTCAGAATGACTTG	64	2,12
GAATGACTTGCTTGAGTACTCACCAAGTCACAGAAAAGCATCTTACGGATGGCATGACAGTAAGAGAATTATGCA	63	1,2,3,7,8,9
TAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGAGCTAACCGTTTTGCACAAACATTG	72	1 2 3 4 5 6 8 9 11 12
<b>CCACTTCTCGCTCGGCCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCG</b>	<b>66</b>	<b>2,5</b>
TCCCTGAGAGTTTCGCCCCGAAGAACGTTTCCAATGATGAGCACTTTAAAGTTCTGCTATGTGGCGCGGTA	74	2,5,12
GGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTTGACAAACATTGGGAATCATG	63	1,2,3,4,5,6,8,9,11,12
ACTCTAGCTTCCCGCAACAATTAAATAGACTGGATGGAGGGGATTAAGTTGCAGGACCACTTCTCGCCTCGGC	39	1,2,3,4,6,7,8,9,10,11
CCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCGCGGTATTCAGCAC	71	2,5,12

# Templates:

01 blaTEM-1A\_4\_HM749966

02 blaTEM-2\_1\_X54606

03 blaTEM-2\_3\_AJ251946

04 blaTEM-3\_1\_X64523

05 blaTEM-6\_1\_X57972

06 blaTEM-8\_1\_X65252

07 blaTEM-10\_1\_AF093512

08 blaTEM-11\_1\_AY874537

09 blaTEM-12\_1\_M88143

10 blaTEM-15\_1\_AM849805

11 blaTEM-16\_1\_X65254

12 blaTEM-17\_1\_Y14574

ConClave Scores:

63+72

64+63+72+66

63+72

72

72+66

72

63

63+72

63+72

72

64+72

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \} \quad (1)$$

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases} \quad (2)$$

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \} \quad (3)$$

Predicted Template: \_\_\_\_\_

Query sequence	Score	Templates
GCGCGGTATTATCCGTGTTGACGCCGGCAAGAGTAACTCGGTCGCCGCATAACTATTCTCAGAATGACTTG	64	2,12
GAATGACTTGCTTGAGTACTCACCAAGTCACAGAAAAGCATCTTACGGATGGCATGACAGTAAGAGAATTATGCA	63	1,2,3,7,8,9
TAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTTGACAAACATTG	72	1,2,3,4,5,6,8,9,11,12
CCACTTCTGCGCTCGGCCCCATCCGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCG	66	2,5
TCCTTGAGAGTTTCGCCCGAAGAACGTTTCCAATGATGAGCACTTTAAAGTTCTGCTATGTGGCGCGGT	74	2,5,12
GGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTTGACAAACATTGGGAATCATG	63	1,2,3,4,5,6,8,9,11,12
ACTCTAGCTTCCGGCAACAATTAAATAGACTGGATGGAGGGGGATTAAGTTGCAGGACCACTCTGCGCTCGGC	39	1,2,3,4,6,7,8,9,10,11
CCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCGCGGTATTCAGCAC	71	2,5,12

# Templates:

01 blaTEM-1A\_4\_HM749966  
 02 blaTEM-2\_1\_X54606  
 03 blaTEM-2\_3\_AJ251946  
 04 blaTEM-3\_1\_X64523  
 05 blaTEM-6\_1\_X57972  
 06 blaTEM-8\_1\_X65252  
 07 blaTEM-10\_1\_AF093512  
 08 blaTEM-11\_1\_AY874537  
 09 blaTEM-12\_1\_M88143  
 10 blaTEM-15\_1\_AM849805  
 11 blaTEM-16\_1\_X65254  
 12 blaTEM-17\_1\_Y14574

ConClave Scores:

63+72+63+39  
64+63+72+66+74+63+39+71  
63+72+63+39  
72+63+39  
72+66+74+63+71  
72+63+39  
63+39  
63+72+63+39  
63+72+63+39  
39  
72+63+39  
64+72+74+63+71

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \} \quad (1)$$

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases} \quad (2)$$

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \} \quad (3)$$

Predicted Template: \_\_\_\_\_

Query sequence	Score	Templates
GCGCGGTATTATCCGTGTTGACGCCGGCAAGAGTAACTCGGTCGCCGCATAACTATTCTCAGAATGACTTG	64	2,12
GAATGACTTGCTTGAGTACTCACCAAGTCACAGAAAAGCATCTTACGGATGGCATGACAGTAAGAGAATTATGCA	63	1,2,3,7,8,9
TAACACTGCGGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTGCACAACATTG	72	1,2,3,4,5,6,8,9,11,12
CCACTTCTCGCTCGGCCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCG	66	2,5
TCCTTGAGAGTTTCGCCCGAAGAACGTTTCCAATGATGAGCACTTTAAAGTTCTGCTATGTGGCGCGGT	74	2,5,12
GGCCAACTTACTTCTGACAACGATCGGAGGACCGAAGGGAGCTAACCGCTTTGCACAACATTGGGAATCATG	63	1,2,3,4,5,6,8,9,11,12
ACTCTAGCTTCCGGCAACAATTAAATAGACTGGATGGAGGGGATTAAGTTGCAGGACCACTCTGCCTCGGC	39	1,2,3,4,6,7,8,9,10,11
CCATCCGGCTGGCTGGTTATTGCTGATAAAATCTGGAGCCGGTGAGCGTGGGTCTCGCGGTATTCAGCAC	71	2,5,12

# Templates:

01 blaTEM-1A\_4 HM749966

02 blaTEM-2\_1\_X54606

03 blaTEM-2\_3\_AJ251946

04 blaTEM-3\_1\_X64523

05 blaTEM-6\_1\_X57972

06 blaTEM-8\_1\_X65252

07 blaTEM-10\_1\_AF093512

08 blaTEM-11\_1\_AY874537

09 blaTEM-12\_1\_M88143

10 blaTEM-15\_1\_AM849805

11 blaTEM-16\_1\_X65254

12 blaTEM-17\_1\_Y14574

ConClave Scores:

237

512

237

174

346

174

102

237

237

39

174

344

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \} \quad (1)$$

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases} \quad (2)$$

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \} \quad (3)$$

Predicted Template: blaTEM-2\_1\_X54606

**But is it Relevant for ONT...**

# Final definitions

- Global Alignment
- Local Alignment
- Minimizer

# And KMA

$$T_m \in \operatorname{argmax}_{i \in T} \{ f(t_i) \} \quad (1)$$

$$C(t) = \sum_{k \in K} \max \begin{cases} f(t_k) & \tau \leq f(t_k) \\ 0 & \text{else} \end{cases} \quad (2)$$

$$S_k \in \operatorname{argmax}_{i \in T_m} \{ C(t_i) \} \quad (3)$$

# KMA nano-branch

$$T_m(q) \in \operatorname{argmax}_{t \in T} \{ f(q, t) \} \quad (1)$$

$$C(t) \in \sum_{q \in Q} \begin{cases} f(q, t) & \tau \leq f(t_k) \wedge t \in T_m(q) \\ 0 & \text{else} \end{cases}. \quad (2)$$

$$S_q \in \operatorname{argmax}_{t \in T_m(q)} \{ C(t) \} \quad (3)$$

$$T_m(q) \in \operatorname{argmax}_{t \in T} \left\{ \min \left\{ \frac{\max_{r \in T} \{ f(q, r) \}}{f(q, t)} \right\} \varepsilon \in [0; 1] \right\} \quad (4)$$

# ONT Assembly Challenge

- Built the best ONT assembly pipeline
- The same criteria as last week, but ONT instead of Illumina.
- Same deadline and prices too.