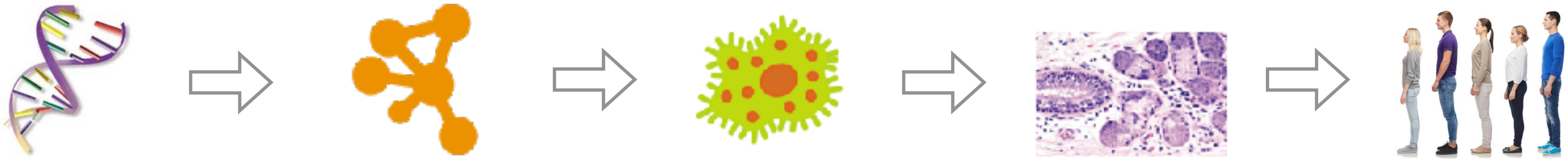
CSE 427 Computational Biology

Lecture 1: Introduction

CSE427: Computational methods for biology at different scales



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 µm)

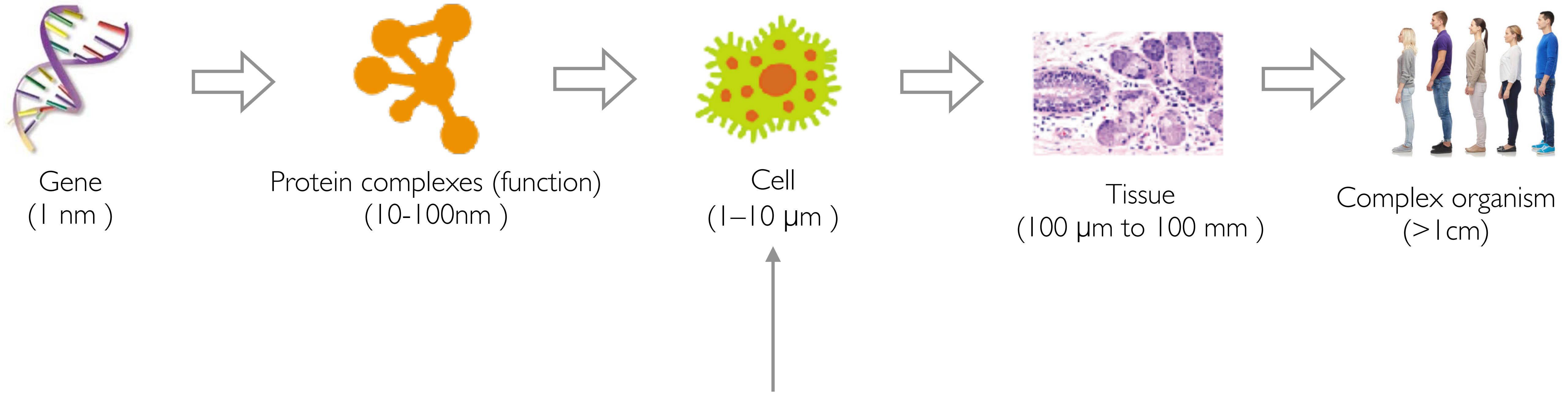
Tissue
(100 µm to 100 mm)

Complex organism
(>1cm)

A rich hierarchy of biological subsystems at multiple scales: genotypic variations in nucleotides (1 nm scale) -> proteins (1–10 nm) -> protein complexes (10–100 nm), cellular processes (100 nm) -> phenotypic behaviors of cells (1–10 µm), tissues (100 µm to 100 mm), -> complex organisms (>1 m).

source: Yu, Michael Ku, et al. "Translation of genotype to phenotype by a hierarchy of cell subsystems." *Cell systems* 2.2 (2016): 77-88.

Data structure for each scale: cell



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 μm)

Tissue
(100 μm to 100 mm)

Complex organism
(>1cm)

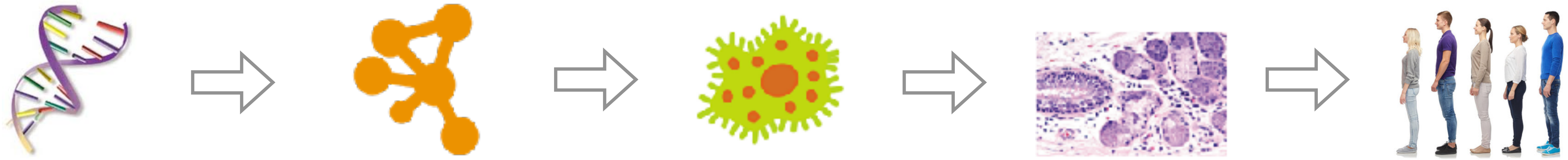
A cell by gene matrix -> vector/matrix (high-dimensional, no spatial information)
Dimensionality reduction methods (PCA, t-SNE, variety of embedding methods)

High-dimensional, noisy, large-scale

Single cell RNA sequencing (scRNA-seq)

- What is scRNA-seq?
  - A technique that can measure the gene expression vector of each cell
- What is the data structure?
  - A 2D array. Rows are cells. Columns are genes.
  - Lots of rows (millions of cells)
  - ~20k columns for human
- Analogy in other applications?
- **What is the research question here?**
  - Machine learning: dimensionality reduction, clustering, classification.

Data structure for each scale: tissue



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 µm)

Tissue
(100 µm to 100 mm)

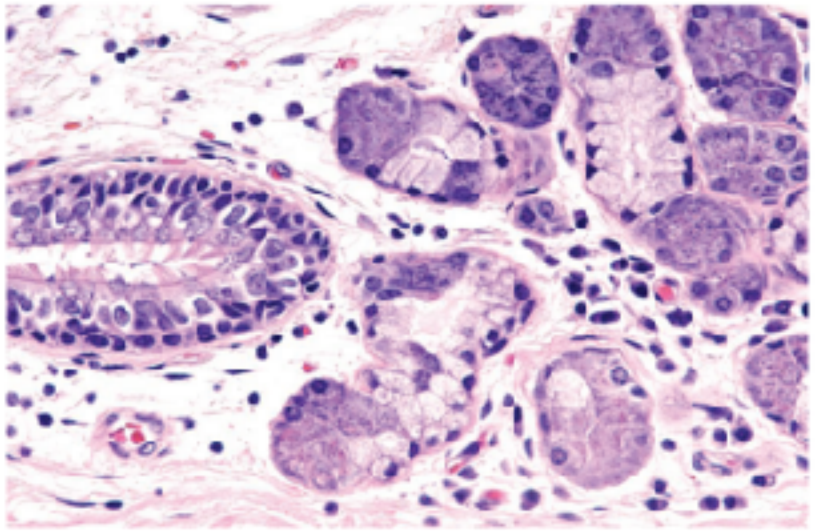
Complex organism
(>1cm)

Tissue image -> image analysis
Image analysis (segmentation, detection, CNN)

Image analysis, lack of high-quality annotations

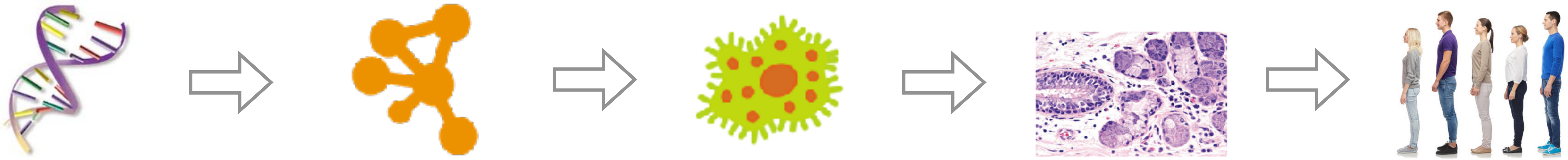
Medical imaging technology

- What is the data structure?
  - One image for a small part of the tissue
- Analogy in other applications?
  - Image analysis
- **What is the research question here?**
  - Machine learning: image segmentation (which region is tumor), image classification (tumor v.s. healthy)



Tumor tissue image

Data structure for each scale: organism



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 µm)

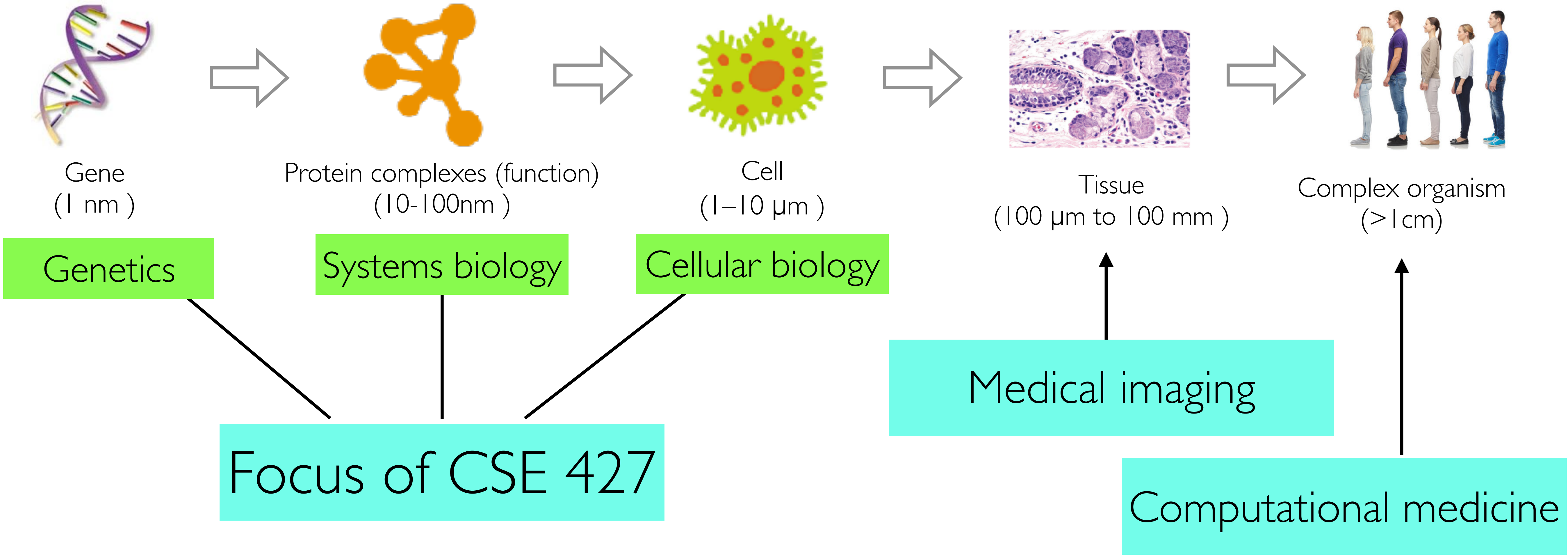
Tissue
(100 µm to 100 mm)

Complex organism
(>1cm)

Disease mechanisms -> Multimodality
Integration of information from sequences, networks, images and matrixes

Multi-modality and heterogeneous

Computational methods for biology at different scales



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 µm)

Tissue
(100 µm to 100 mm)

Complex organism
(>1cm)

Genetics

Systems biology

Cellular biology

Medical imaging

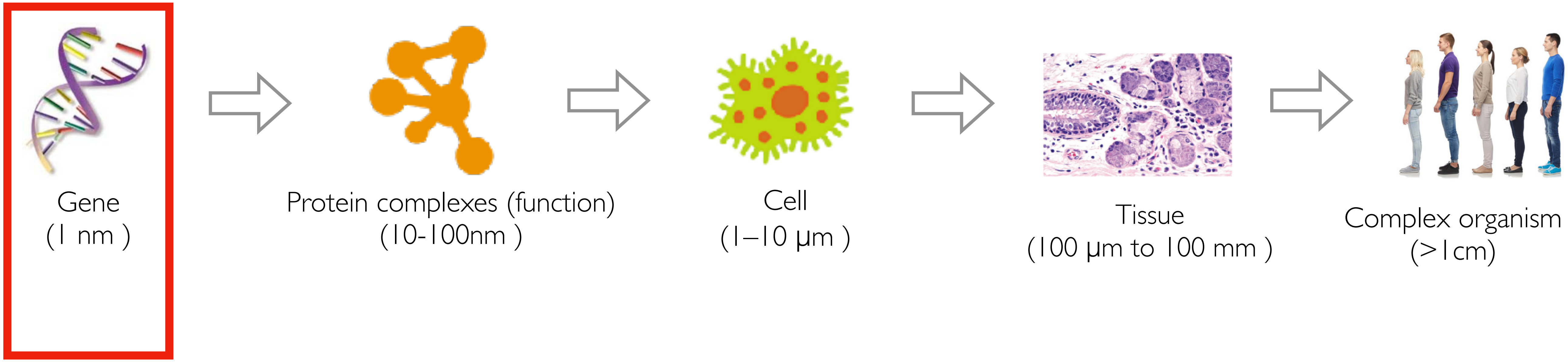
Computational medicine

Focus of CSE 427

Real world research question: how to measure the similarity between two patients

- We will have
  - DNA sequences of these two persons
  - A protein-protein interaction network
  - Gene expression matrix of cells in each person
  - Tissue image
  - Other datasets…
- Which of these data should we use?
- How should we integrate these multiple datasets?

Computational methods for biology at different scales



Gene
(1 nm)

Protein complexes (function)
(10-100nm)

Cell
(1–10 µm)

Tissue
(100 µm to 100 mm)

Complex organism
(>1cm)

A concrete example: The Cancer Genome Atlas Program

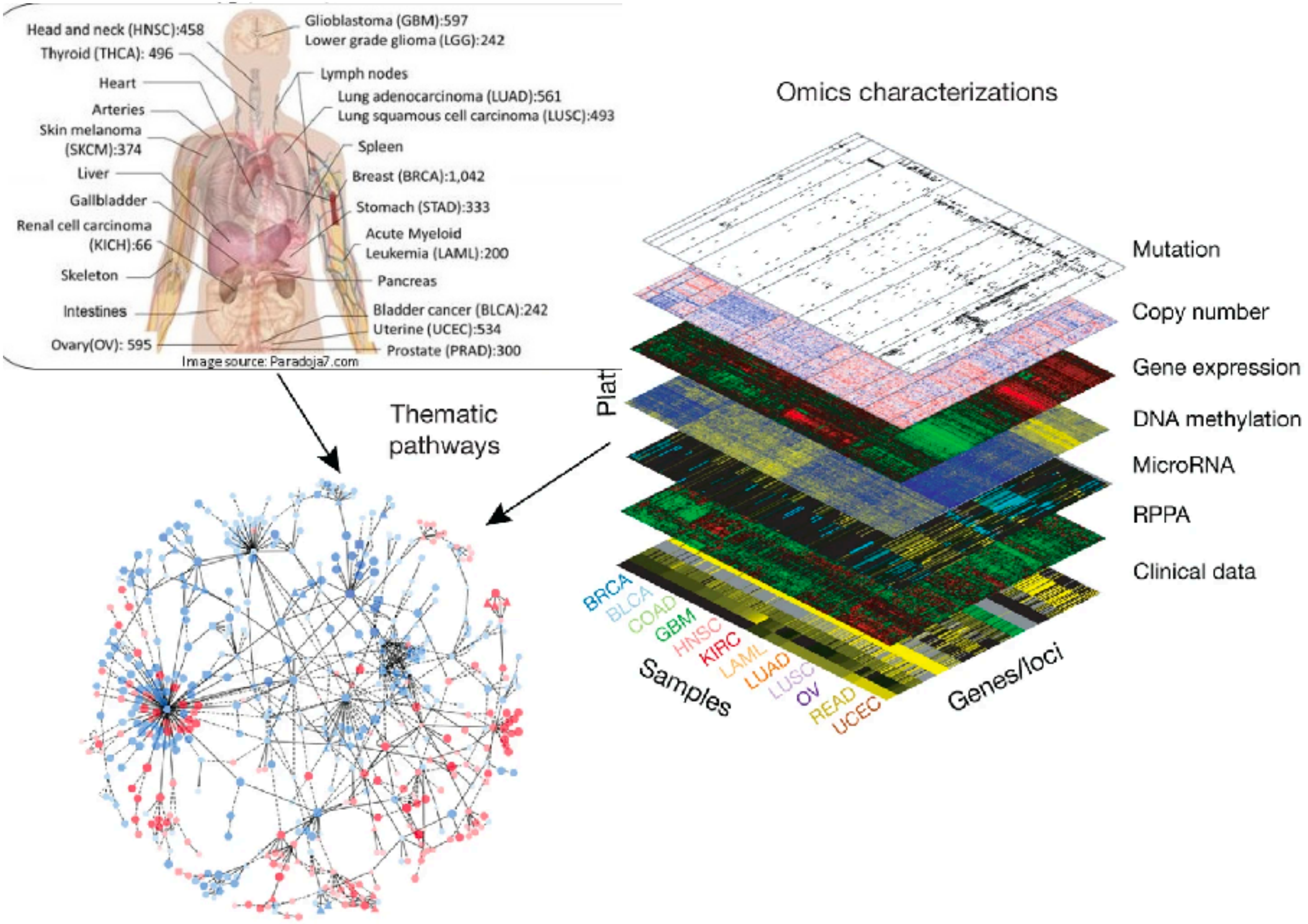
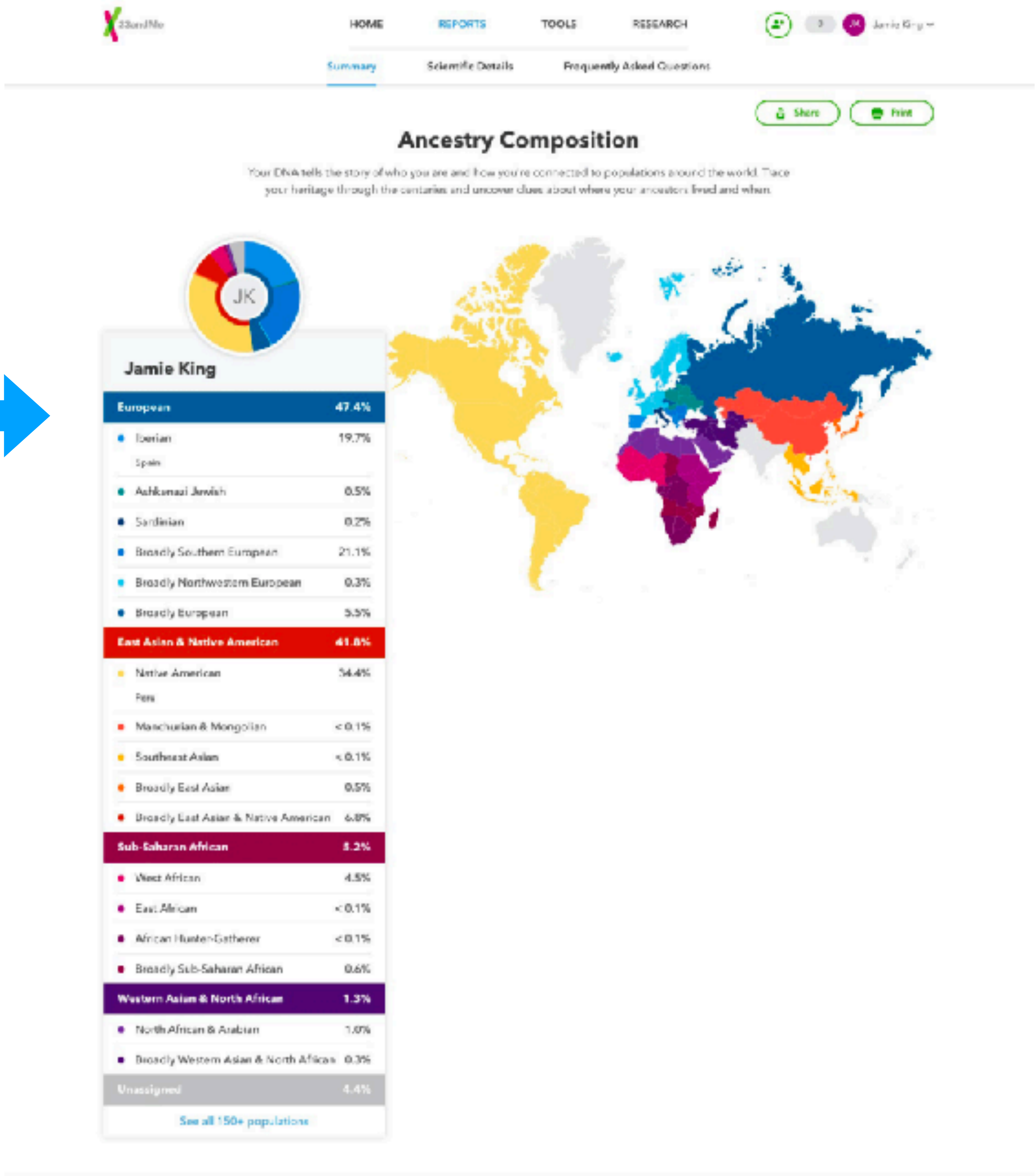


Image source: Paradoja7.com

DNA sample analysis by 23andMe

DNA sample



Ancestry Composition

Your DNA tells the story of who you are and how you're connected to populations around the world. Trace your heritage through the centuries and uncover clues about where your ancestors lived and when.

Jamie King

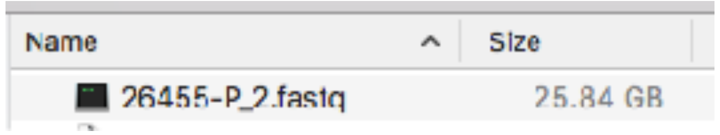
European	**47.4%**
Iberian Spain	19.7%
Ashkenazi Jewish	0.5%
Sardinian	0.2%
Broadly Southern European	21.1%
Broadly Northwestern European	0.3%
Broadly European	5.5%
East Asian & Native American	**41.8%**
Native American Peru	34.4%
Manchurian & Mongolian	< 0.1%
Southeast Asian	< 0.1%
Broadly East Asian	0.5%
Broadly East Asian & Native American	6.8%
Sub-Saharan African	**5.2%**
West African	4.5%
East African	< 0.1%
African Hunter-Gatherer	< 0.1%
Broadly Sub-Saharan African	0.6%
Western Asian & North African	**1.3%**
North African & Arabian	1.0%
Broadly Western Asian & North African	0.3%
Unassigned	**4.4%**

See all 150+ populations

How did they do this?

DNA sample

Sequencing machine
~2000 dollars

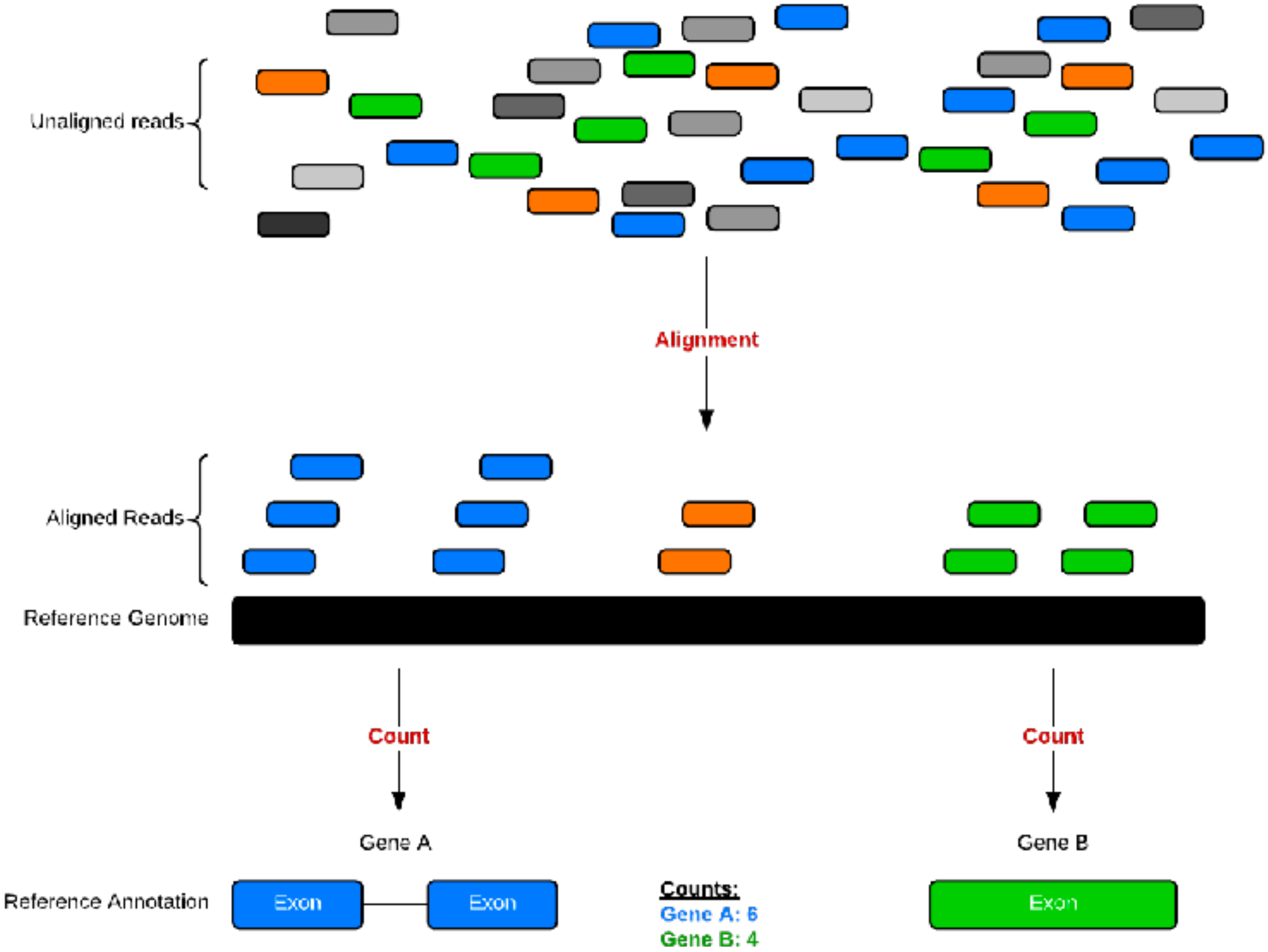


Name	Size
26455-P_2.fastq	25.84 GB

Your entire genome sequence
*.fastq file

Our job as a computer scientist: analyze *.fastq file

Process raw data using sequence alignment (dynamic programming)



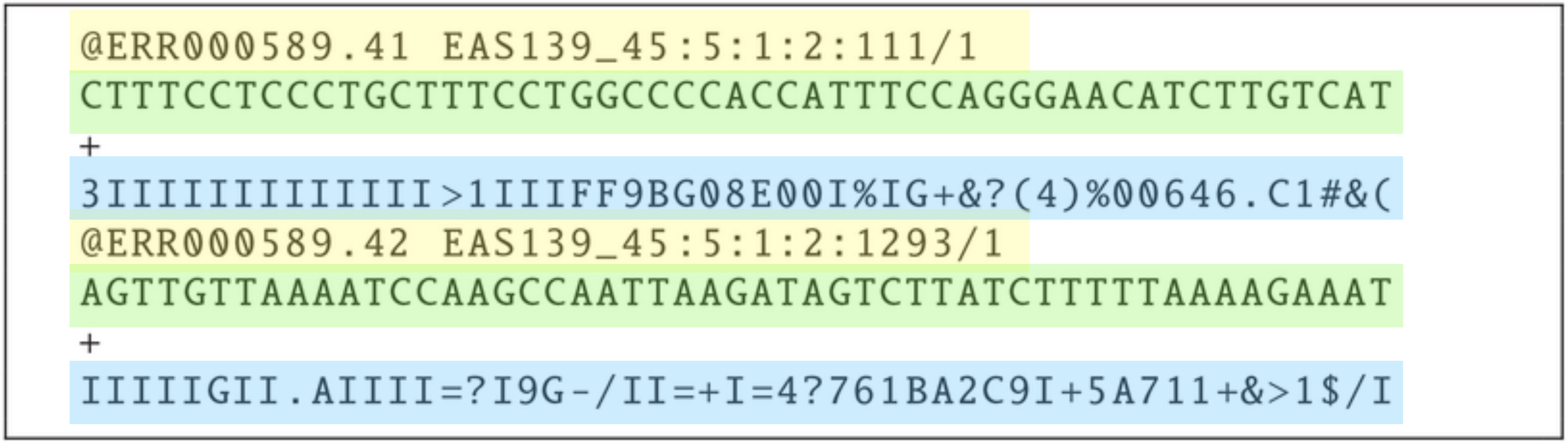
source: https://bioconnector.github.io/bims8382/r-rnaseq-airway.html

What does a fastq file look like?

Quality

Sequence

Header



```
@ERR000589.41 EAS139_45:5:1:2:111/1
CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
+
3IIIIIIIIIIIII>1IIIFF9BG08E00I%IG+&?(4)%00646.C1#&(
@ERR000589.42 EAS139_45:5:1:2:1293/1
AGTTGTTAAAATCCAAGCCAATTAAGATAGTCTTATCTTTTTAAAAGAAAT
+
IIIIIGII.AIIII=?I9G-/II=+I=4?761BA2C9I+5A711+&>1$/I
```

Very large! ~300000000 lines
Quality: ASCII chars

What should we do? Map each short sequence (we call it read) to the entire human genome

What does a fastq file look like?

Reference genome: "average" human genome.
Most widely used human genome GRCh38: derived from 13 thirteen anonymous volunteers

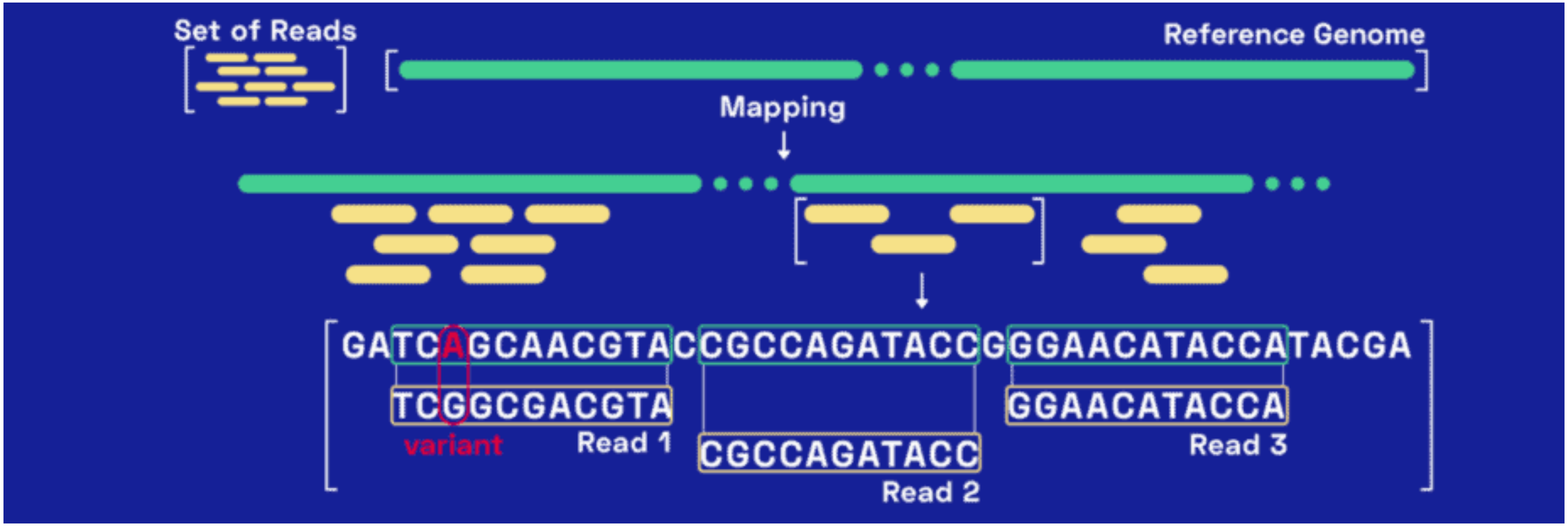
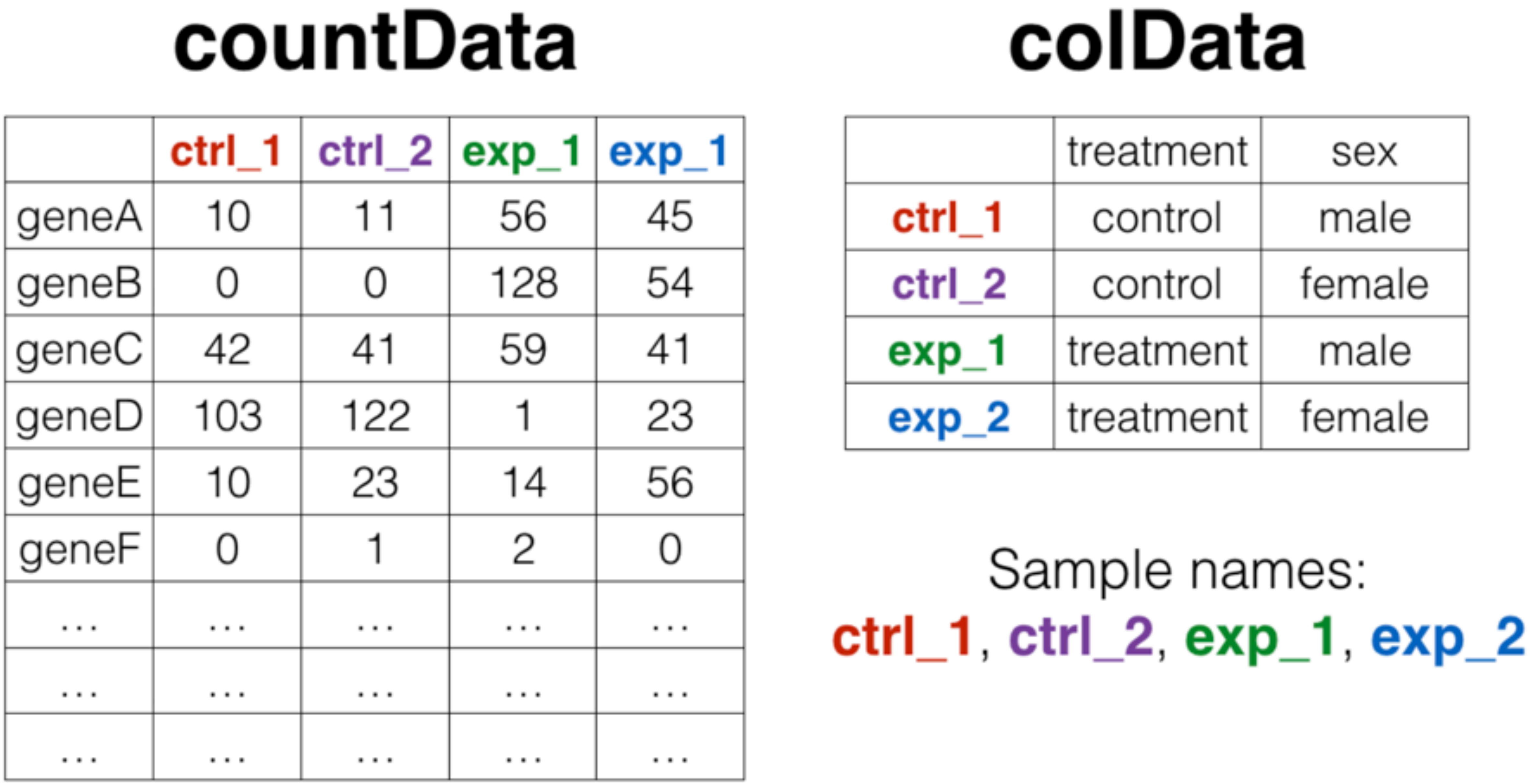


Image from Nebula Genomics

Processed data



countData

	ctrl_1	ctrl_2	exp_1	exp_1
geneA	10	11	56	45
geneB	0	0	128	54
geneC	42	41	59	41
geneD	103	122	1	23
geneE	10	23	14	56
geneF	0	1	2	0
…	…	…	…	…
…	…	…	…	…
…	…	…	…	…

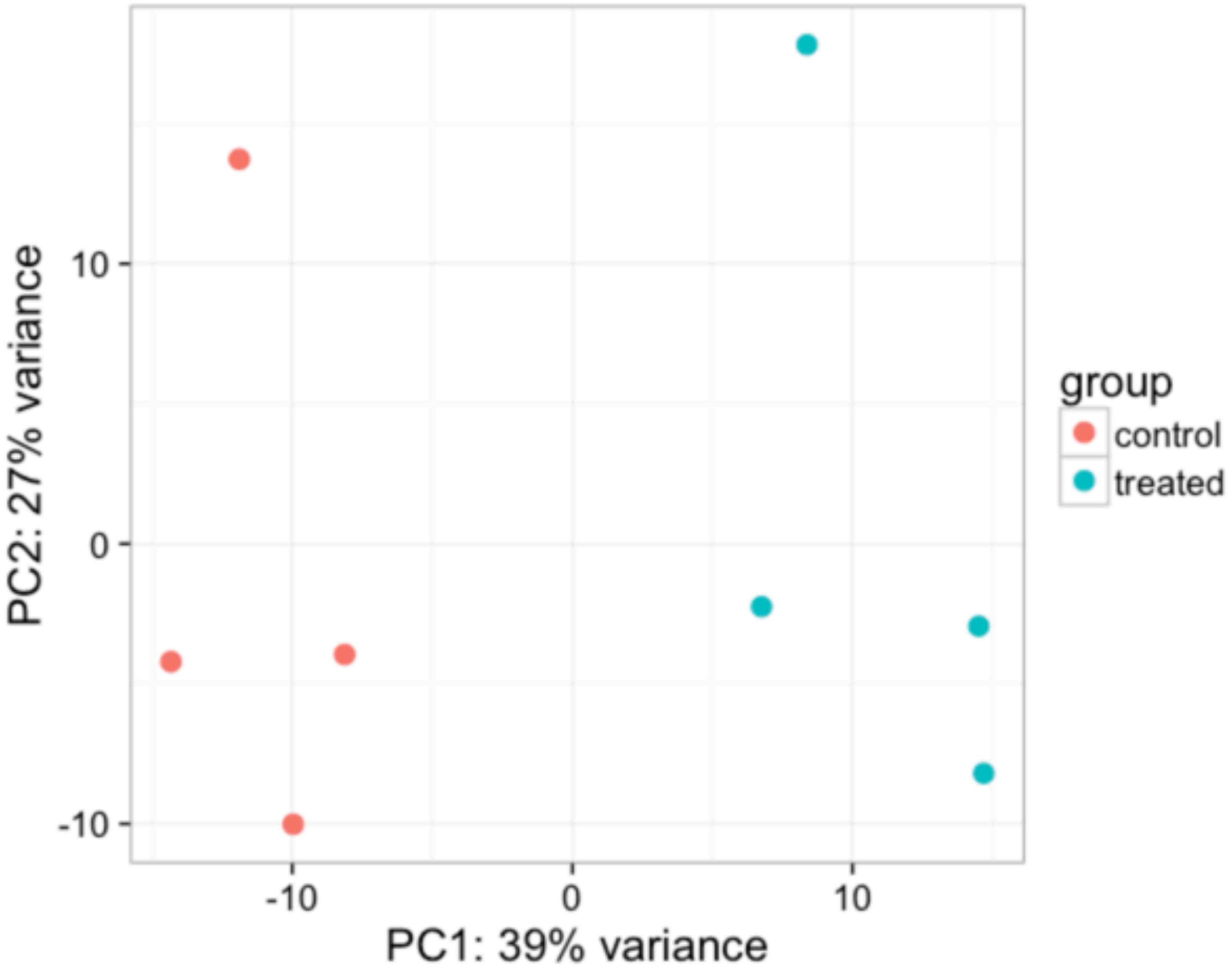
colData

	treatment	sex
ctrl_1	control	male
ctrl_2	control	female
exp_1	treatment	male
exp_2	treatment	female

Sample names:
ctrl_1, **ctrl_2**, **exp_1**, **exp_2**

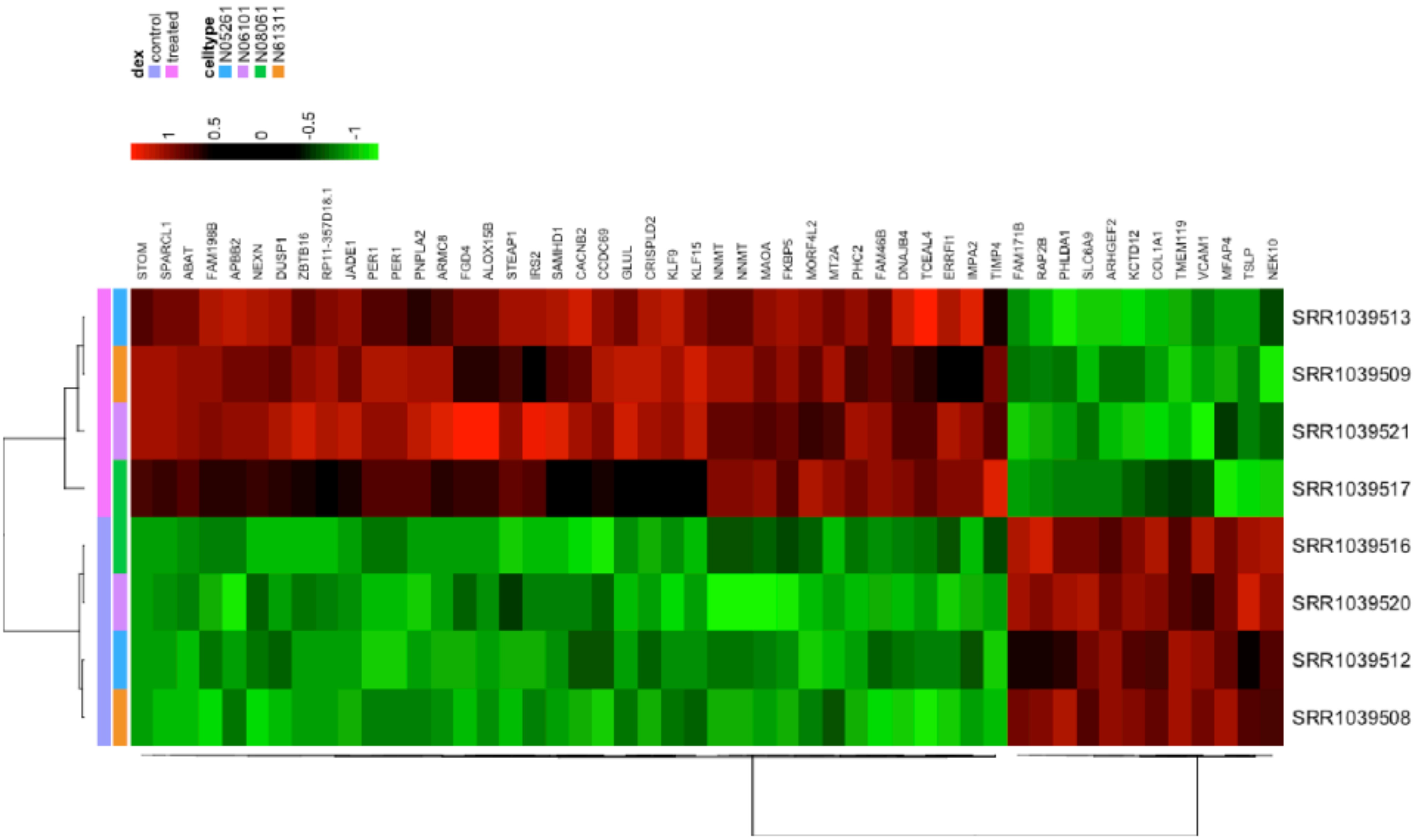
source: https://bioconnector.github.io/bims8382/r-rnaseq-airway.html

Clustering analysis using dimensionality reduction



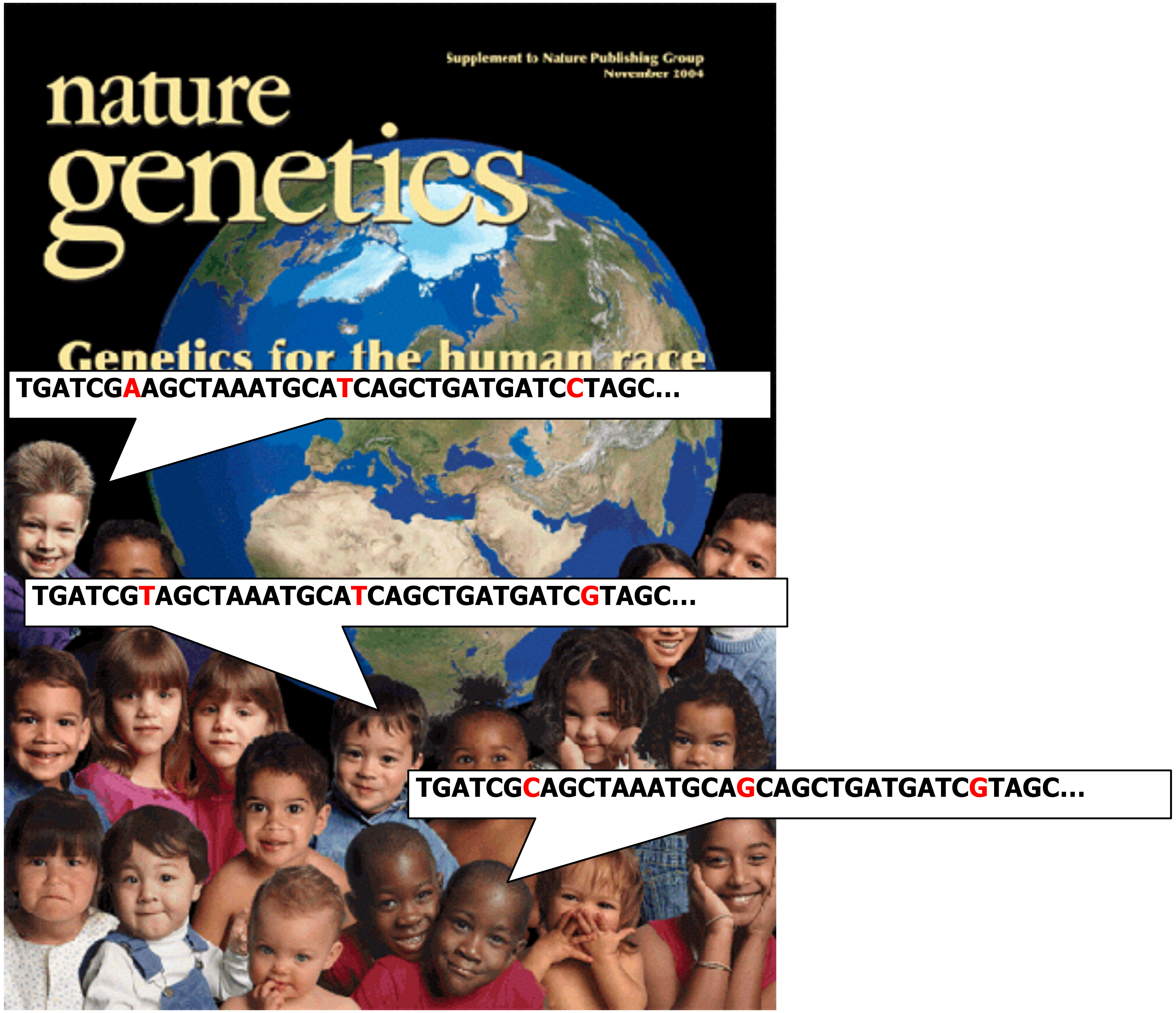
source: https://bioconnector.github.io/bims8382/r-rnaseq-airway.html

Heatmap for visualization



source: https://bioconnector.github.io/bims8382/r-rnaseq-airway.html

Each individual has a slightly different version of the DNA sequence

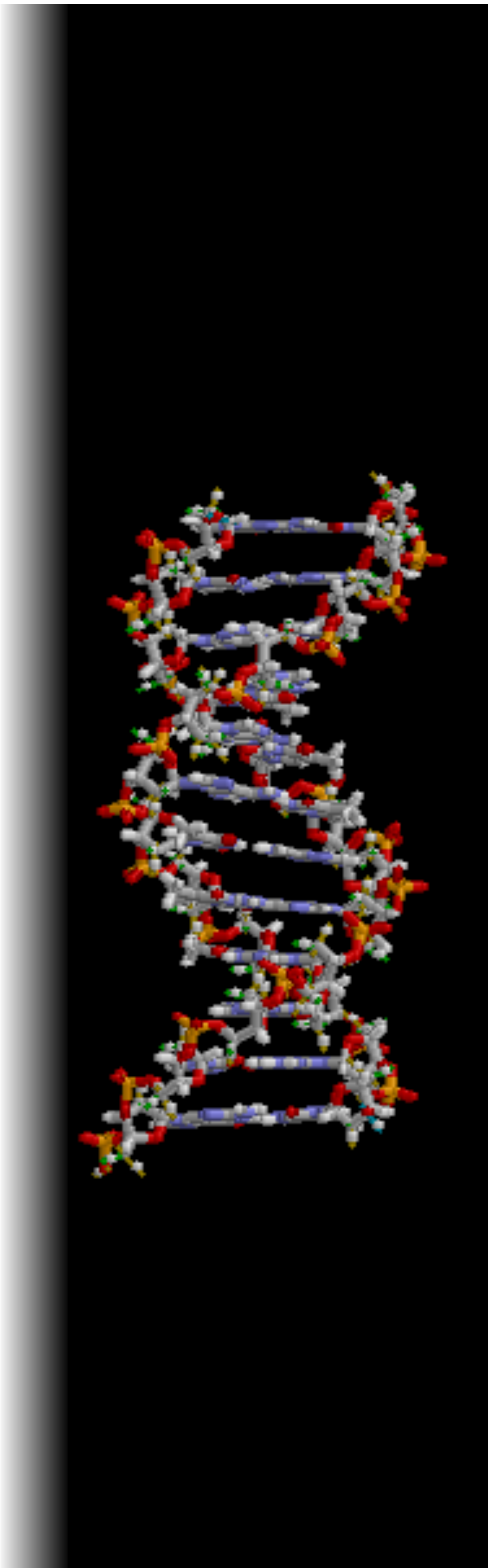

Supplement to Nature Publishing Group
November 2004
nature
genetics
Genetics for the human race
TGATCGAAGCTAAATGCATCAGCTGATGATCCTAGC...
TGATCGTAGCTAAATGCATCAGCTGATGATCGTAGC...
TGATCGCAGCTAAATGCAGCAGCTGATGATCGTAGC...

DNA: "Blueprints" for a cell

- Genetic information encoded in long strings of double-stranded DNA (Deoxyribo Nucleic Acid)

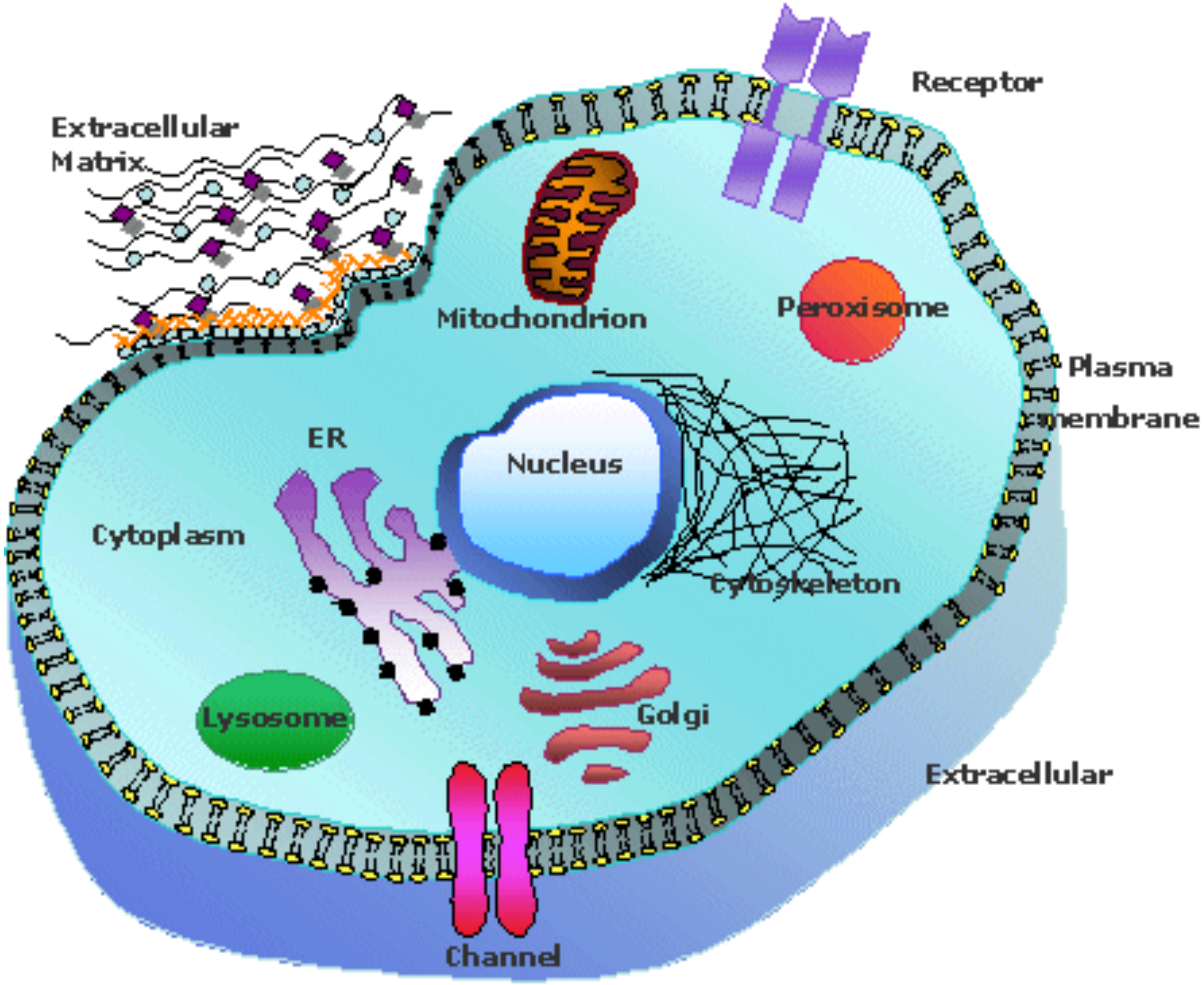
- DNA comes in only four flavors: Adenine, Cytosine, Guanine, Thymine
  - In human, DNA is a 3 billion-long string of As, Cs, Gs and Ts

- DNA acts as the "brain" of the cell, telling the cell how to properly grow and work



Cell

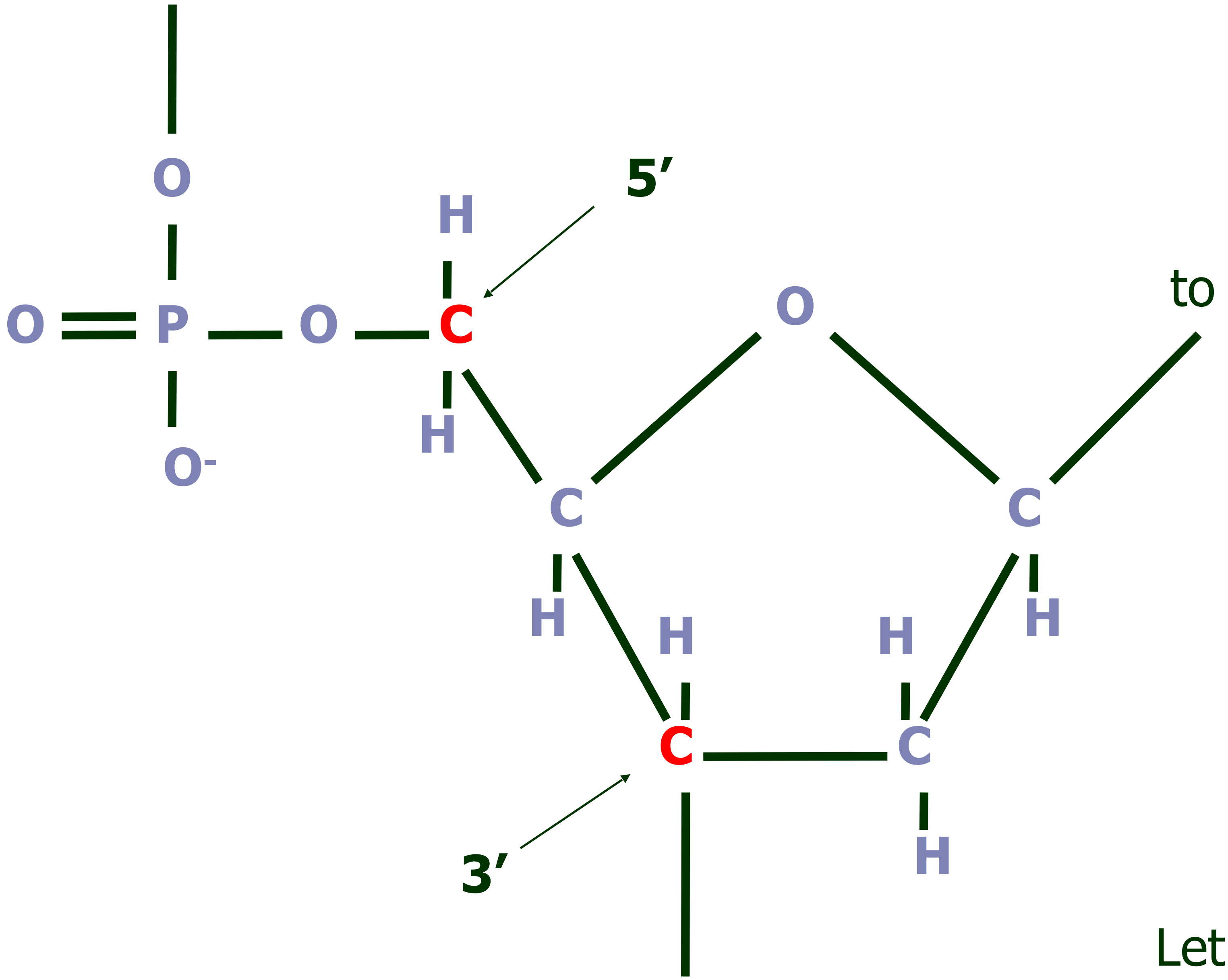
Cell, nucleus, cytoplasm, mitochondrion



Nucleotide

Nucleotide, base, A, C, G, T, 3’, 5’

to previous nucleotide

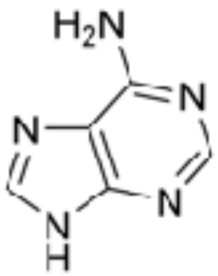


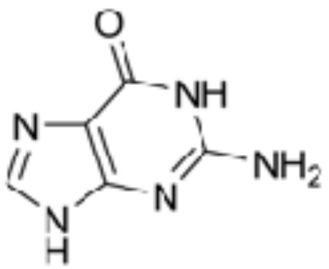
5′

to base

3′

to next nucleotide



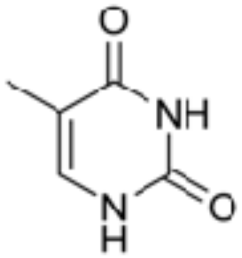


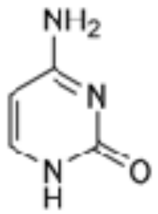
Adenine (A)

Guanine (G)

Thymine (T)

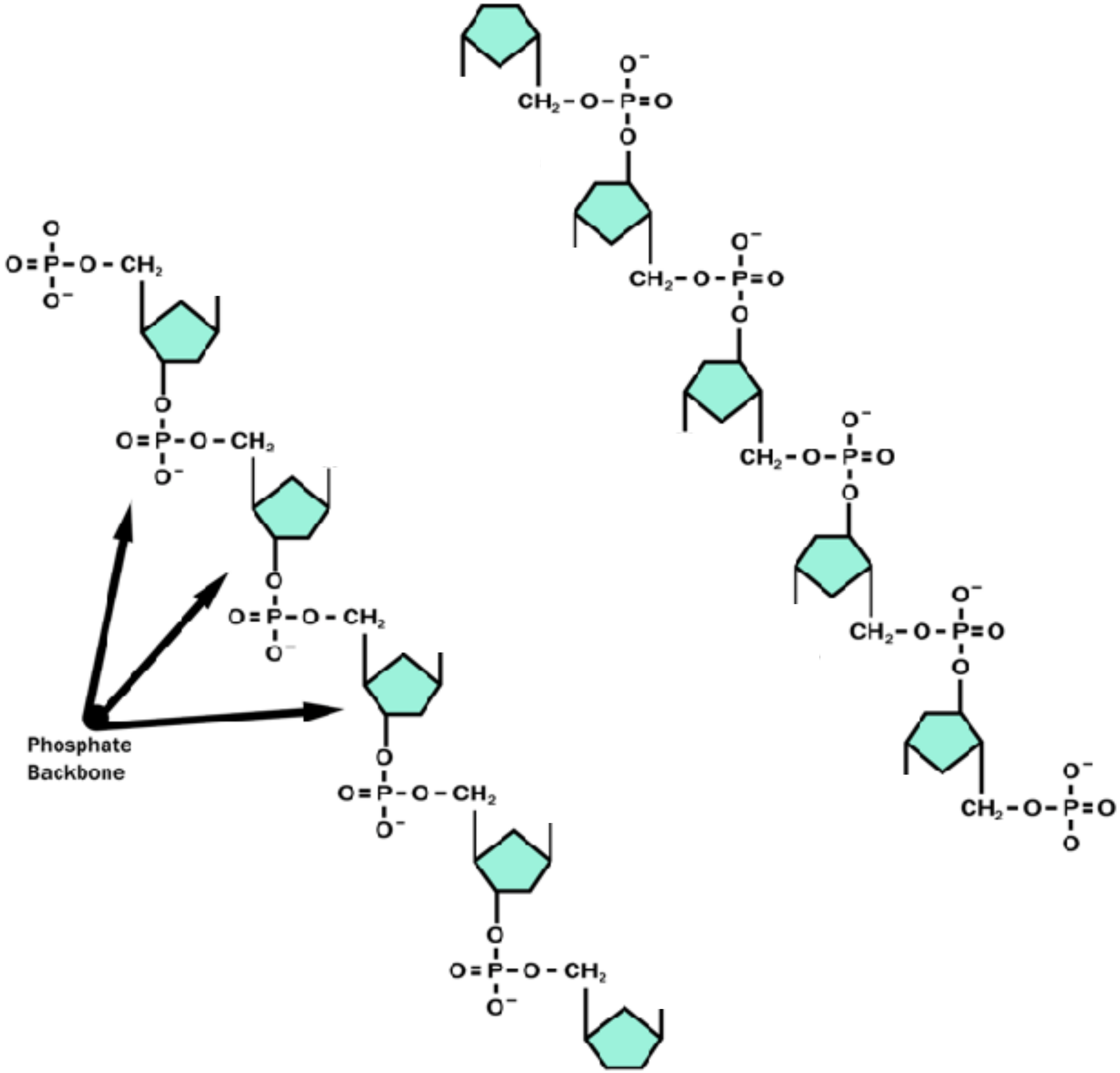
Cytosine (C)



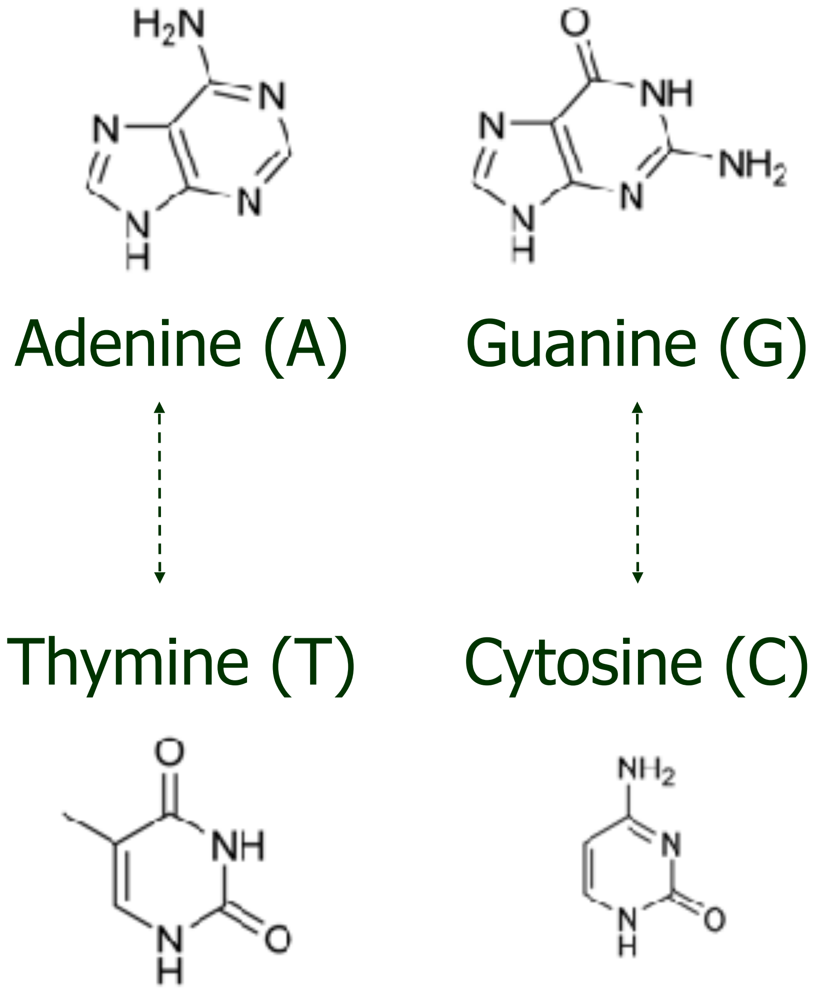


Let’s write “AGACC”!

“AGACC” (backbone)



“AGACC” (DNA)

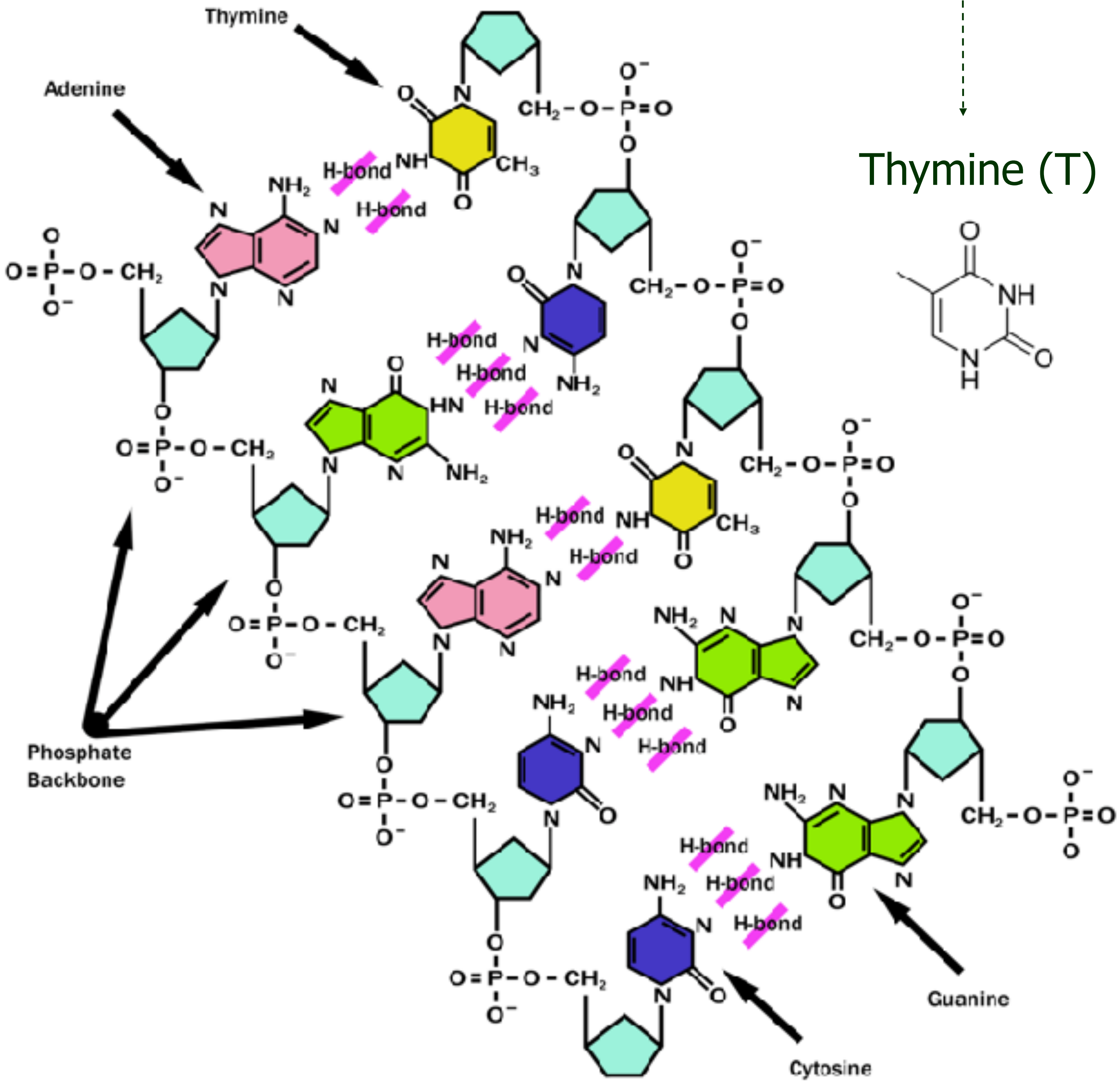


Adenine (A)

Guanine (G)

Thymine (T)

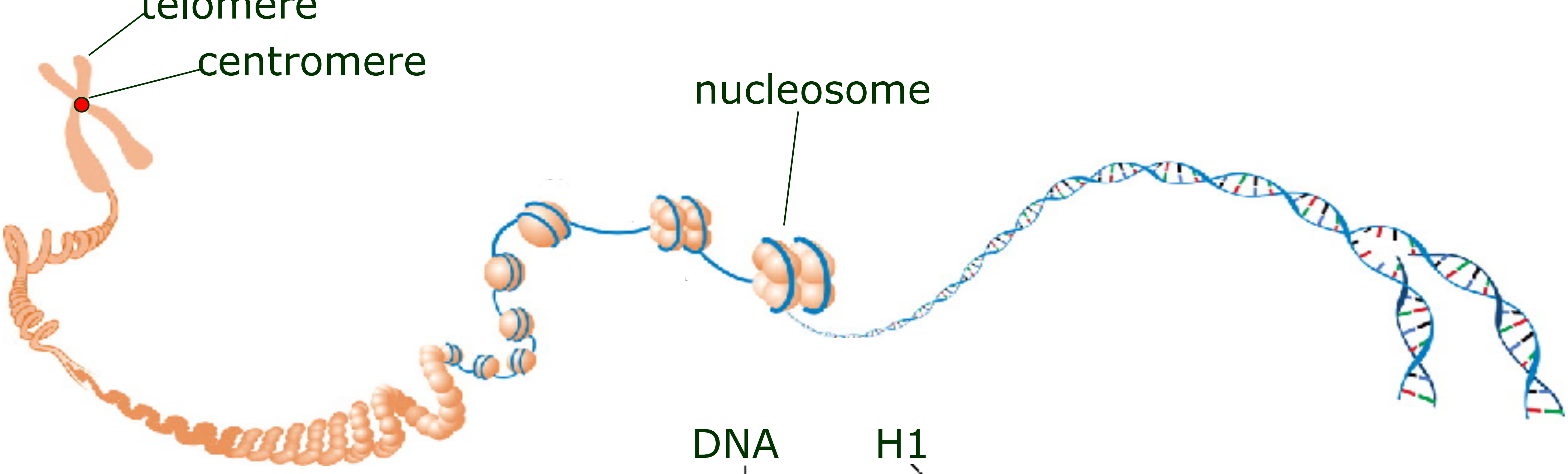
Cytosine (C)

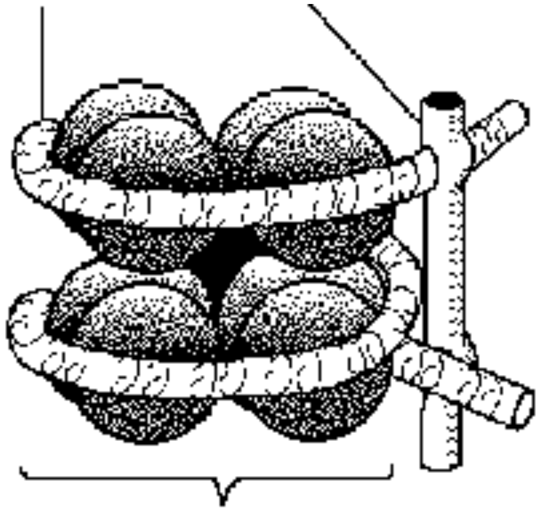


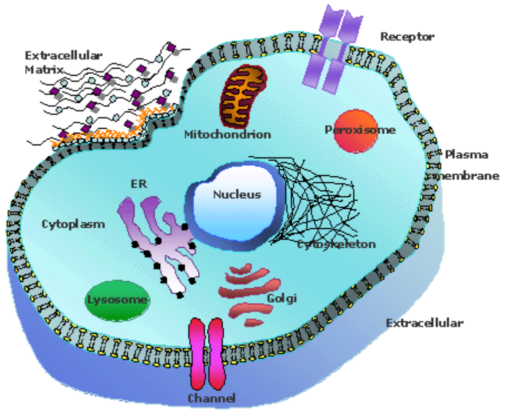
DNA packaging (DNA is 6 feet long!)

Histone, nucleosome, chromatin, chromosome, centromere, telomere

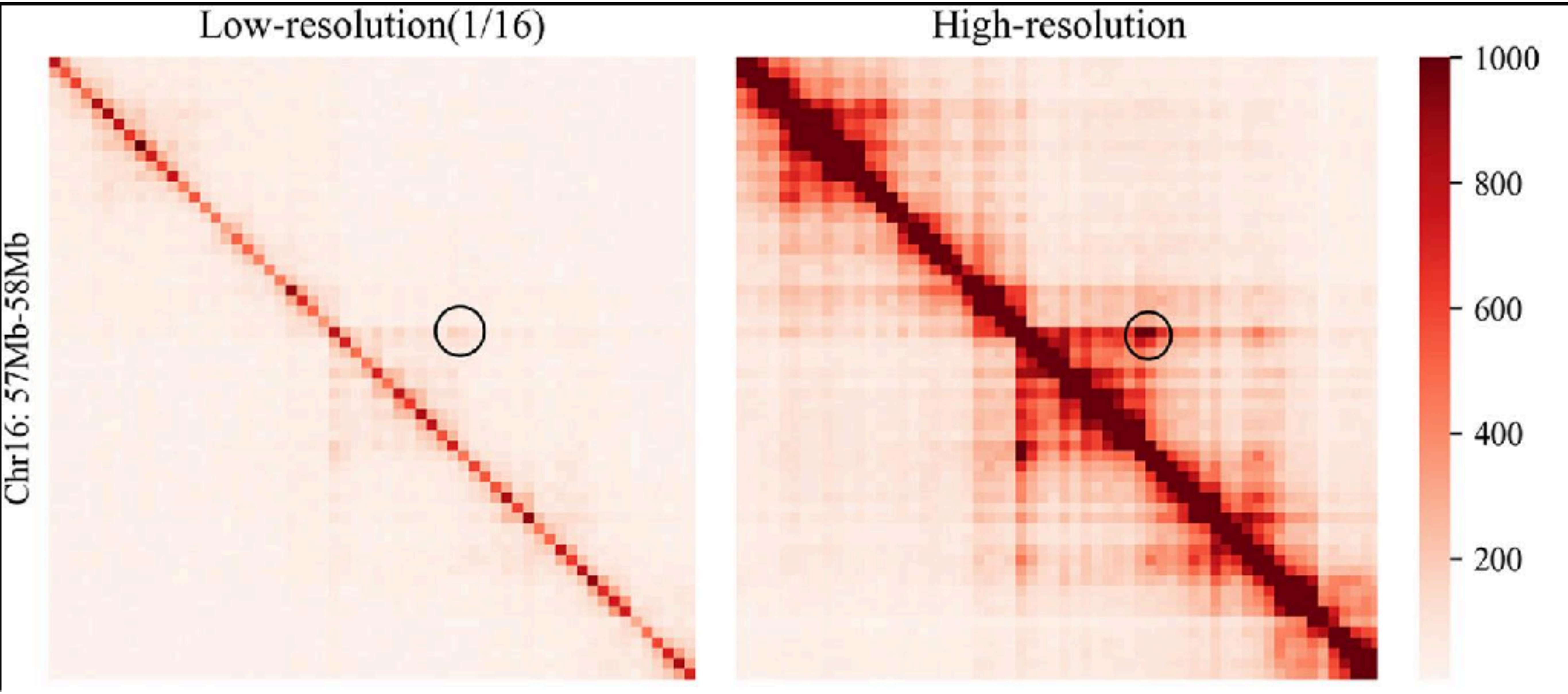
http://www.youtube.com/watch?v=9kQpYdCnU14





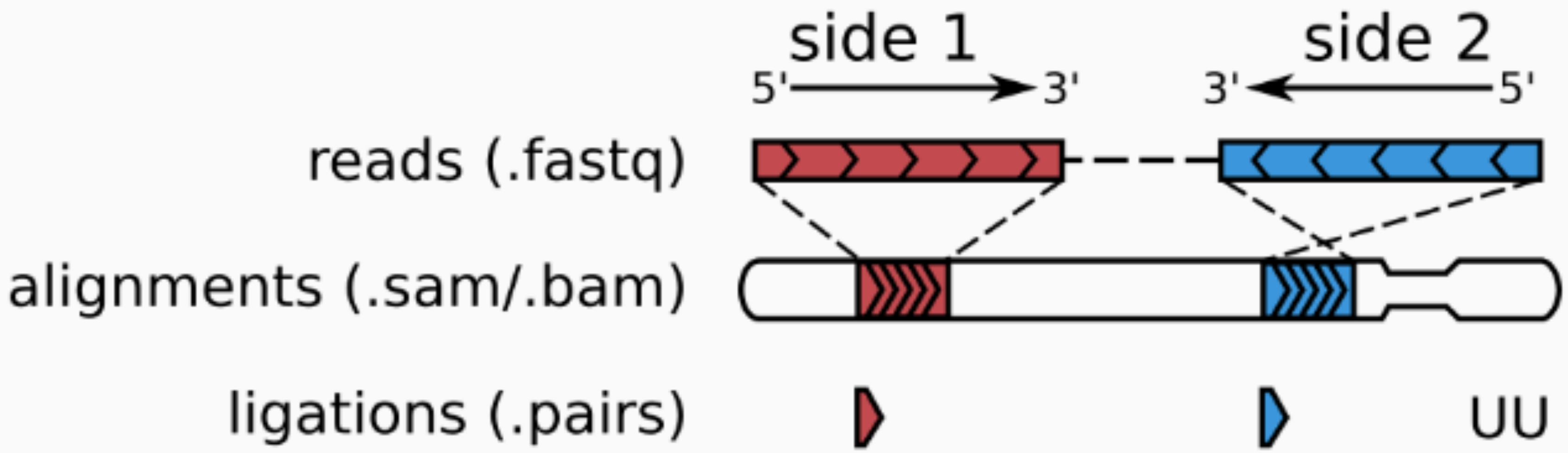


Data structure and computational problem



source: SRHiC: A Deep Learning Model to Enhance the Resolution of Hi-C Data

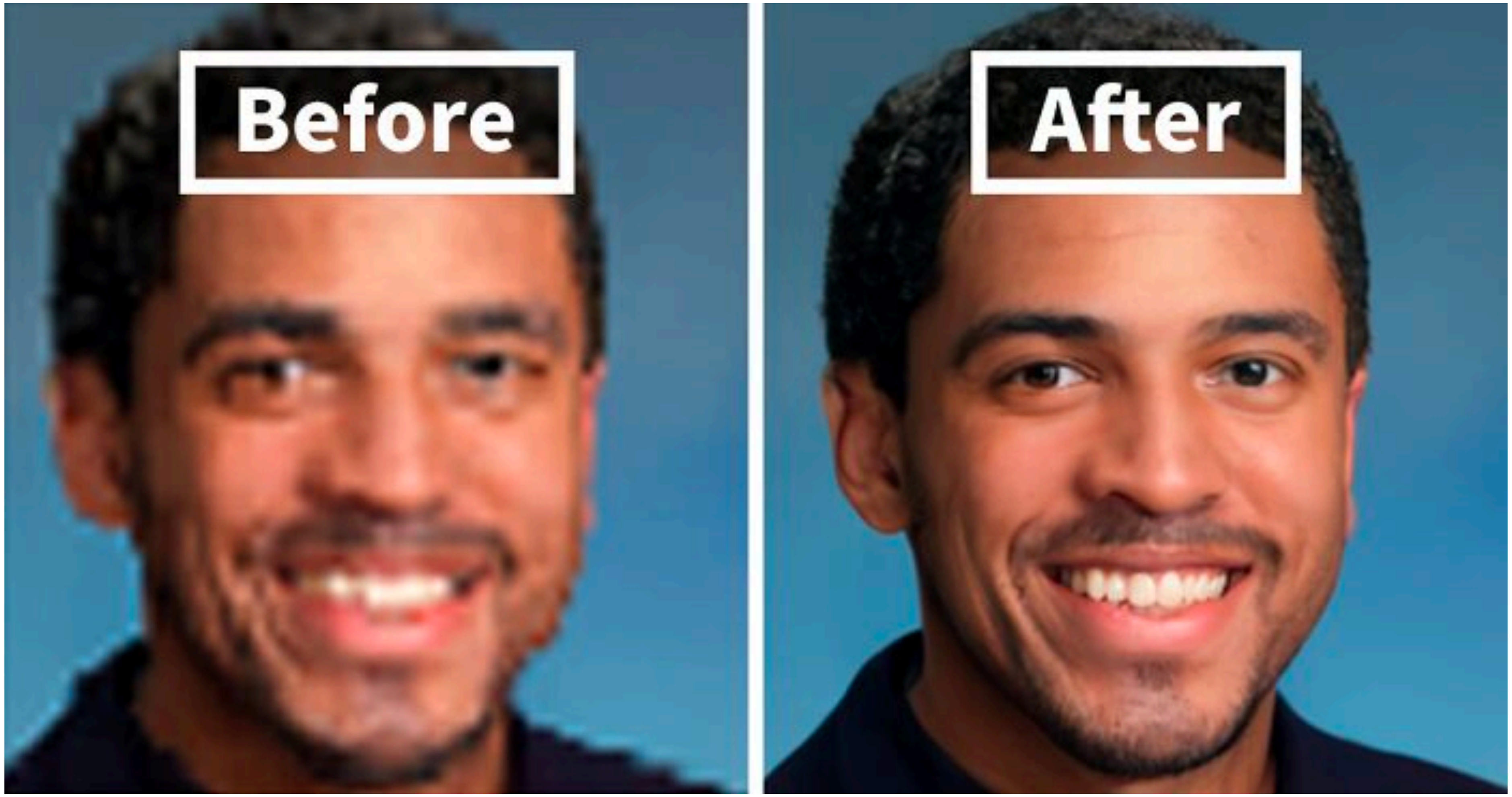
What will the data look like?
Two .fastq files. Lines correspond to each other



DNA sequences (reads) are aligned to the reference genome and converted into ligation events

```
bowtie2 -p 20 -x hg38index -U hicExp1_R1_fastq.trimmed > hicExp1_R1.hg38.sam
bowtie2 -p 20 -x hg38index -U hicExp1_R2_fastq.trimmed > hicExp1_R2.hg38.sam
```

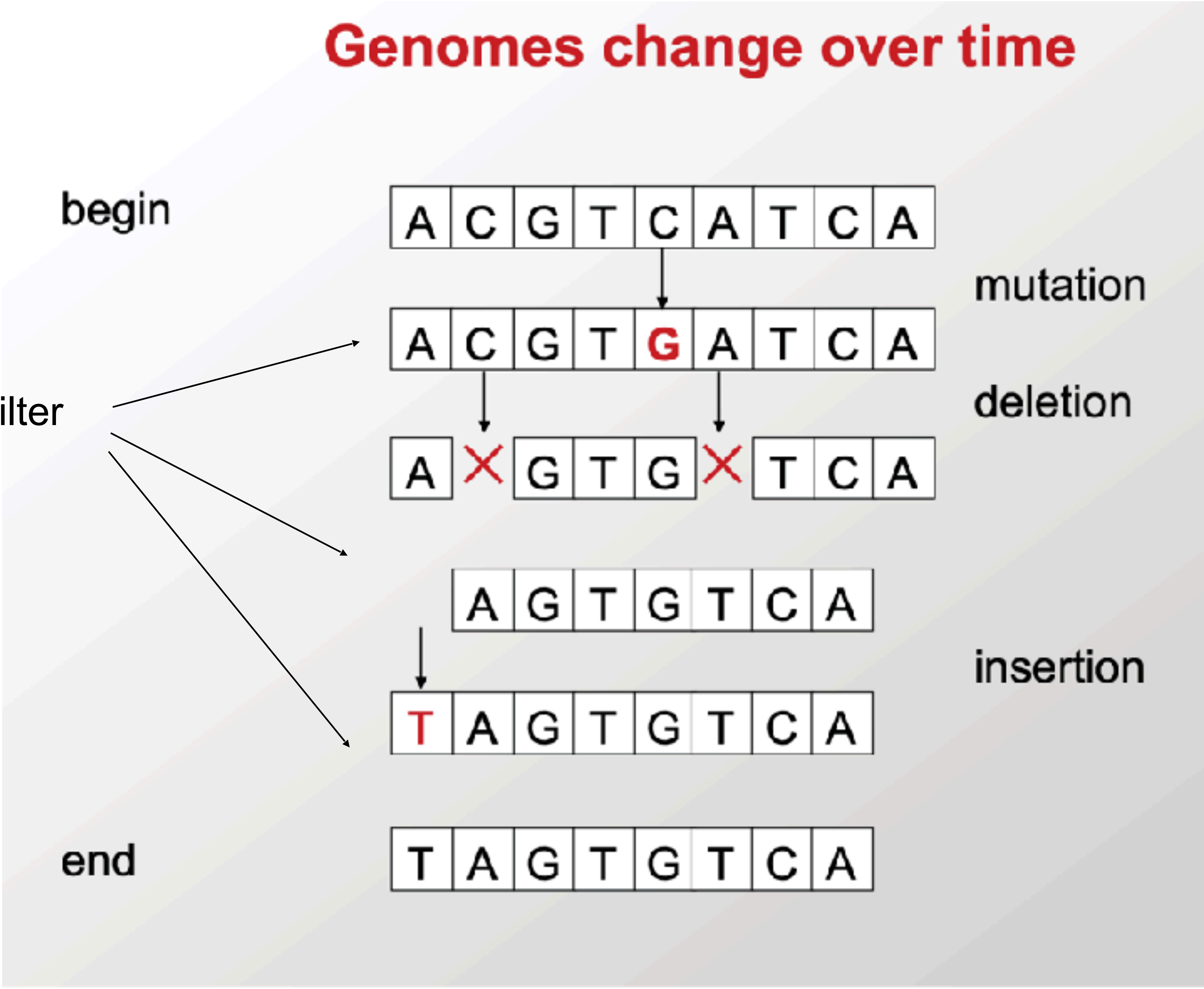
Computer vision-based solution



source:https://www.boredpanda.com/google-ai-amazing-image-enhancement/

Nothing in biology makes sense except in the light of evolution --

Theodosius Dobzhansky



Genomes change over time

A C G T C A T C A

A C G T G A T C A

A ✕ G T G ✕ T C A

A G T G T C A

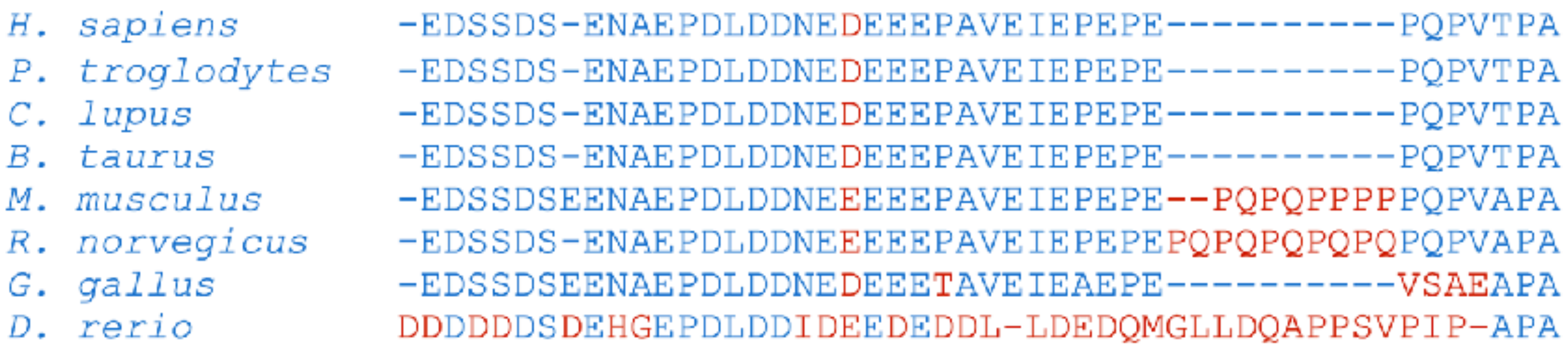
T A G T G T C A

T A G T G T C A

Evolution filter

That is why we want to compare sequences

Partial CTCF protein sequence in 8 organisms:



```
H. sapiens       -EDSSDS-ENAEPDLDDNEDEEEPAVEIEPEPE----------PQPVTPA
P. troglodytes   -EDSSDS-ENAEPDLDDNEDEEEPAVEIEPEPE----------PQPVTPA
C. lupus         -EDSSDS-ENAEPDLDDNEDEEEPAVEIEPEPE----------PQPVTPA
B. taurus        -EDSSDS-ENAEPDLDDNEDEEEPAVEIEPEPE----------PQPVTPA
M. musculus      -EDSSDSEENAEPDLDDNEEEEEPAVEIEPEPE--PQPQPPPPPQPVAPA
R. norvegicus    -EDSSDS-ENAEPDLDDNEEEEEPAVEIEPEPEPQPQPQPQPQPQPVAPA
G. gallus        -EDSSDSEENAEPDLDDNEDEEETAVEIEAEPE----------VSAEAPA
D. rerio         DDDDDDSDEHGEPDLDDIDEEDEDDL-LDEDQMGLLDQAPPSVPIP-APA
```

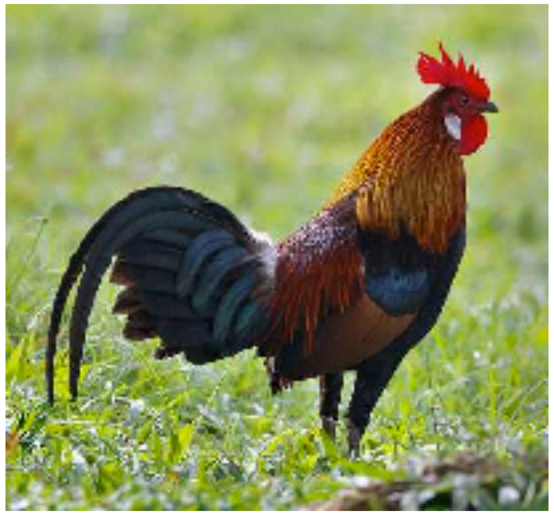
- Identify important sequences by finding conserved regions.
- Find genes similar to known genes.
- Understand evolutionary relationships and distances (D. rerio aka zebrafish is farther from humans than G. gallus aka chicken).
- Interface to databases of genetic sequences.
- As a step in genome assembly, and other sequence analysis tasks.
- Provide hints about protein structure and function

That is why we want to compare sequences

Partial CTCF protein sequence in 8 organisms:

```
H. sapiens       -EDSSDS-ENAEPDLDDNEDEEEPAVEIEPEPE----------PQPVTPA
P. troglodytes   -EDSSDS-ENAEPDLDDNEDEEEPAVEIEPEPE----------PQPVTPA
C. lupus         -EDSSDS-ENAEPDLDDNEDEEEPAVEIEPEPE----------PQPVTPA
B. taurus        -EDSSDS-ENAEPDLDDNEDEEEPAVEIEPEPE----------PQPVTPA
M. musculus      -EDSSDSEENAEPDLDDNEEEEEPAVEIEPEPE--PQPQPPPPPQPVAPA
R. norvegicus    -EDSSDS-ENAEPDLDDNEEEEEPAVEIEPEPEPQPQPQPQPQPQPVAPA
G. gallus        -EDSSDSEENAEPDLDDNEDEEETAVEIEAEPE----------VSAEAPA
D. rerio         DDDDDDSDEHGEPDLDDIDEEDEDDL-LDEDQMGLLDQAPPSVPIP-APA
```

D. rerio



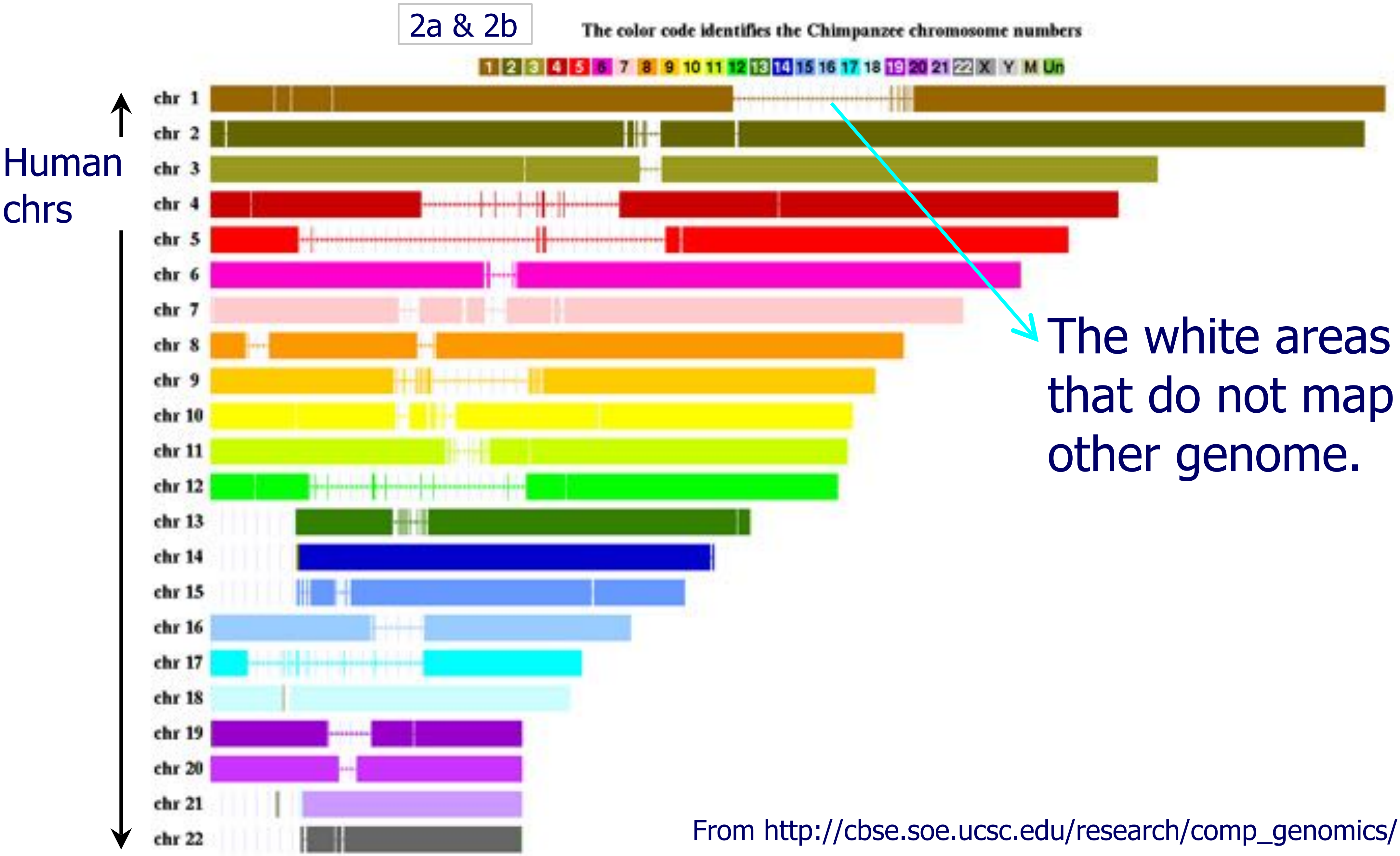
G. gallus

P. Troglodytes

C. lupus

Comparing Human, Chimp, and Mouse Genomes

- 95% of the chimp genome is mapped to identical sequence in the human genome.

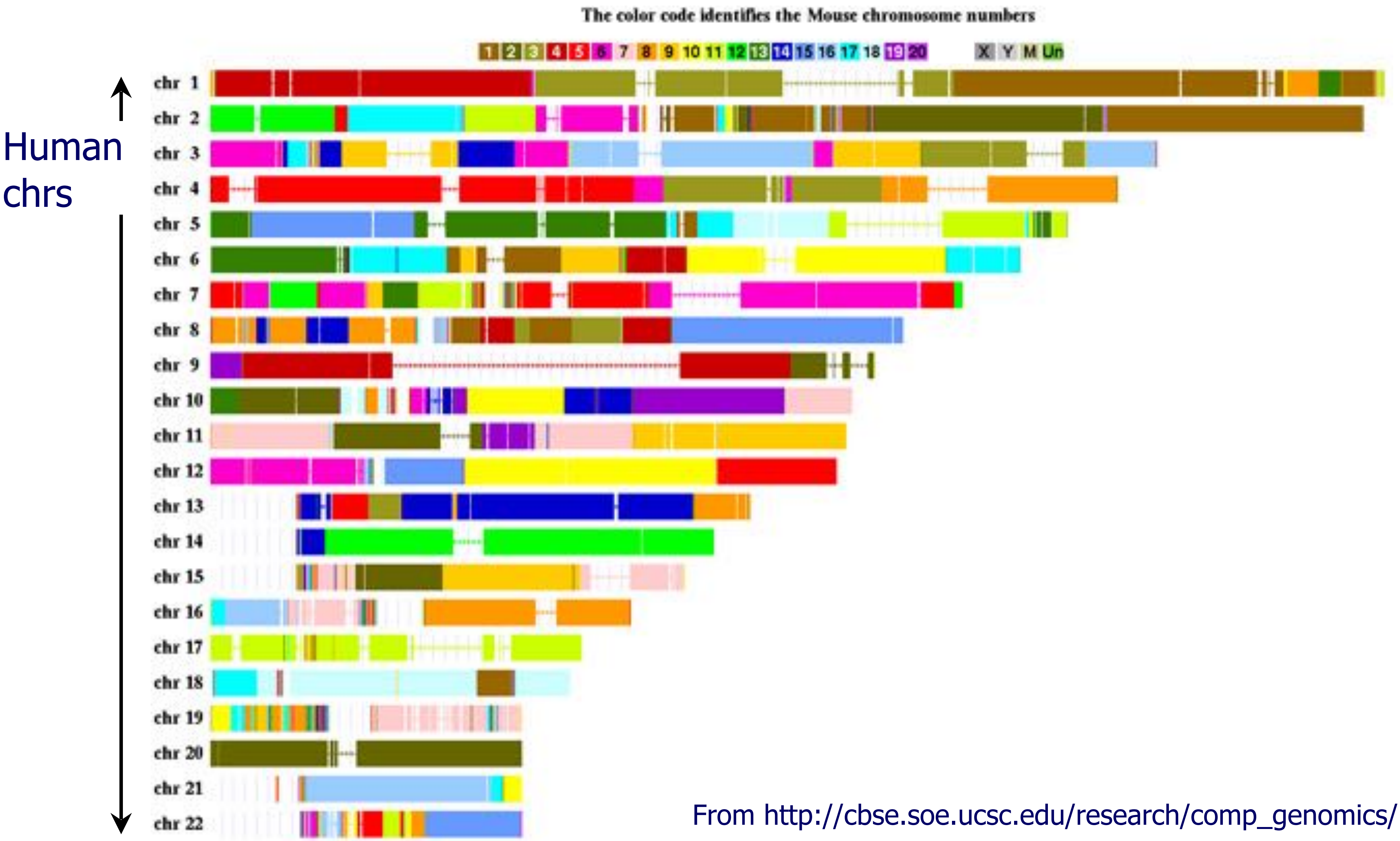


The white areas indicate areas that do not map well to the other genome.

From http://cbse.soe.ucsc.edu/research/comp_genomics/human_chimp_mouse

Comparing Human, Chimp, and Mouse Genomes

- 34% of the mouse genome is mapped to identical sequence in the human genome.



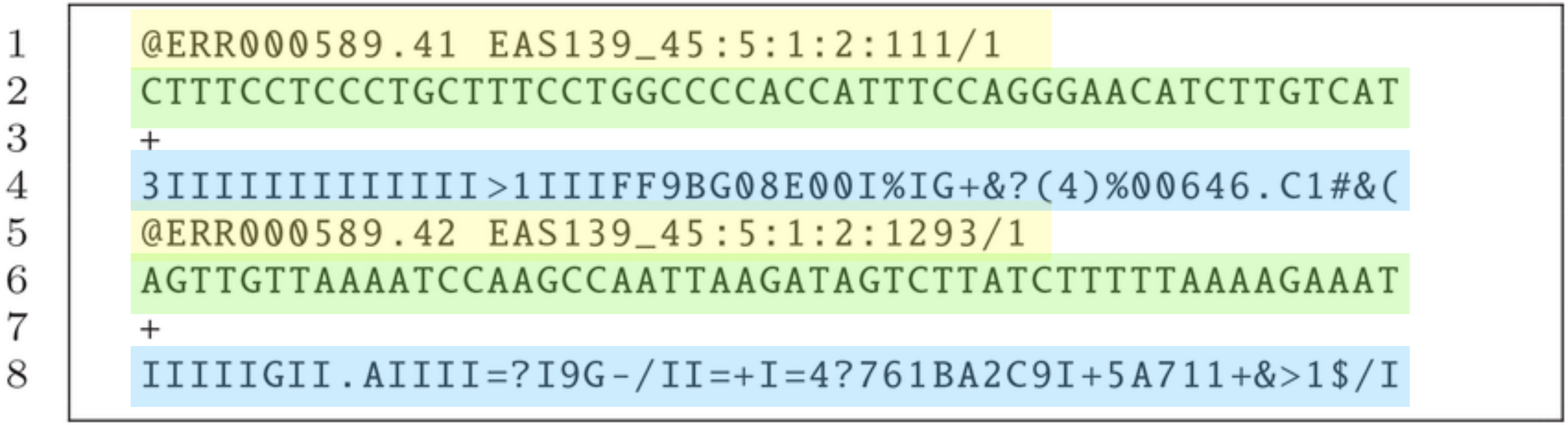
From http://cbse.soe.ucsc.edu/research/comp_genomics/human_chimp_mouse

What does a fastq file look like?

Quality

Sequence

Header



```
@ERR000589.41 EAS139_45:5:1:2:111/1
CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
+
3IIIIIIIIIIIII>1IIIFF9BG08E00I%IG+&?(4)%00646.C1#&(
@ERR000589.42 EAS139_45:5:1:2:1293/1
AGTTGTTAAAATCCAAGCCAATTAAGATAGTCTTATCTTTTTAAAAGAAAT
+
IIIIIGII.AIIII=?I9G-/II=+I=4?761BA2C9I+5A711+&>1$/I
```

Very large! ~300000000 lines
Quality: ASCII chars

What should we do? Map each short sequence (we call it read) to the entire human genome

What does a fastq file look like?

Reference genome: "average" human genome.
Most widely used human genome GRCh38: derived from 13 thirteen anonymous volunteers

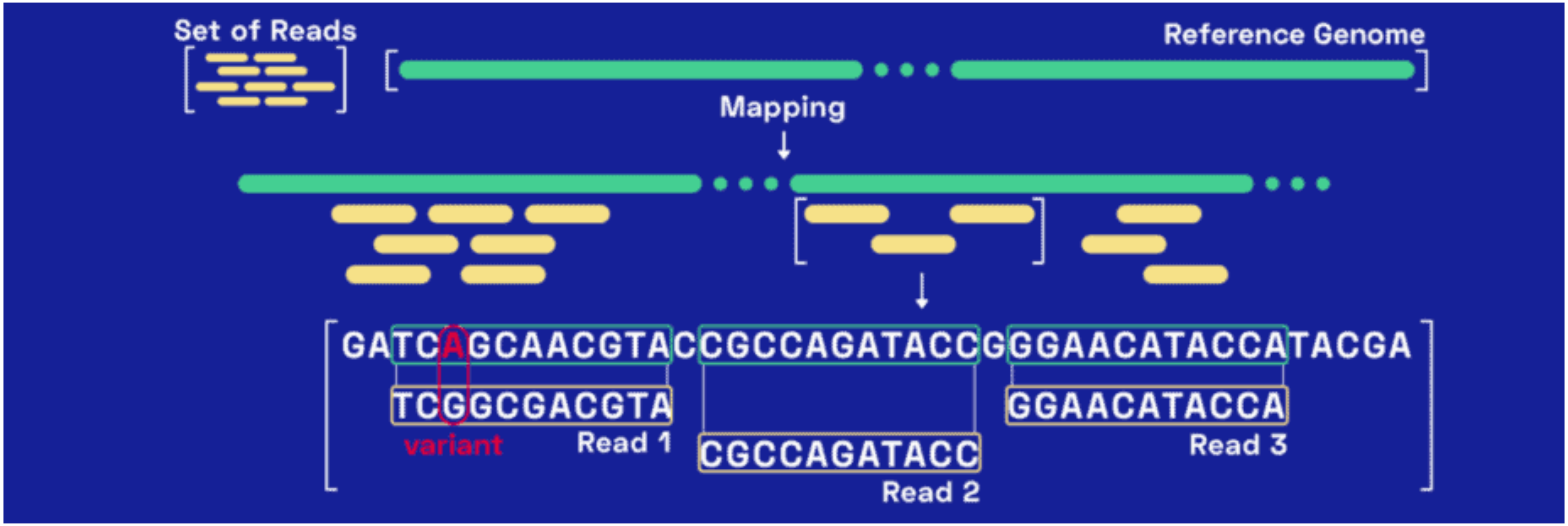


Image from Nebula Genomics

The Simplest String Comparison Problem

Given: Two strings

$$a = a_1a_2a_3a_4\ldots a_m$$
$$b = b_1b_2b_3b_4\ldots b_n$$

where a_i, b_i are letters from some alphabet like {A,C,G,T}.

Compute how similar the two strings are.

What do we mean by "similar"?

Edit distance between strings a and b = the smallest number of the following operations that are needed to transform a into b:

- mutate (replace) a character
- delete a character
- insert a character

$$\text{riddle} \xrightarrow{\text{delete}} \text{ridle} \xrightarrow{\text{mutate}} \text{riple} \xrightarrow{\text{insert}} \text{triple}$$

Dynamic Programming (DP)

- Dynamic programming is used to solve optimization problems, similar to greedy algorithms.

- DP problem can always be decomposed to a series of subproblems with the same structure.
  - Define proper subproblems.

  - Ensure the subproblem space is polynomial.

  - Define a table (matrix), called DP table, to store all the optimal score for each subproblem.

  - Need a traversal order. Subproblems must be ready (solved) when they are needed, so computation order matters.

  - Determine a recursive formula: A larger subproblem is typically solved as a function of its subparts.

  - Remember choices or the solution of each subproblem.

Dynamic Programming (DP)

- Once dynamic programming is setup, computation is typically straight-forward:
  - Systematically fill in the table of results (and usually traceback pointers) and find an optimal score.
  - Traceback from the optimal score through the pointers to determine an optimal solution.

- Example: Fibonacci Numbers
  - The Fibonacci sequence is recursively defined as $F(0) = F(1) = 1$,
    $F(n) = F(n-1) + F(n-2)$ for $n >= 2$.

Local and Global Alignment

- Sometimes we need to choose whether we want to align the entire sequence.

```
A   T   A   C   G   T   C   T
-   -   A   C   G   T   -   -
```

Local alignment: Smith-Waterman algorithm

```
A   T   A   C   G   T   C   T
A   -   -   C   G   -   -   T
```

Global alignment: Needleman-Wunsh algorithm

- They both contain four align positions and four gaps. Which one should we choose?

- Criteria
  - Do we want to check the whole sequence or a local region?
  - Is there a big length difference between two sequences?
  - Are the sequences distantly related during evolution?
  - Is your job about finding motifs, conserved domains?

Key difference

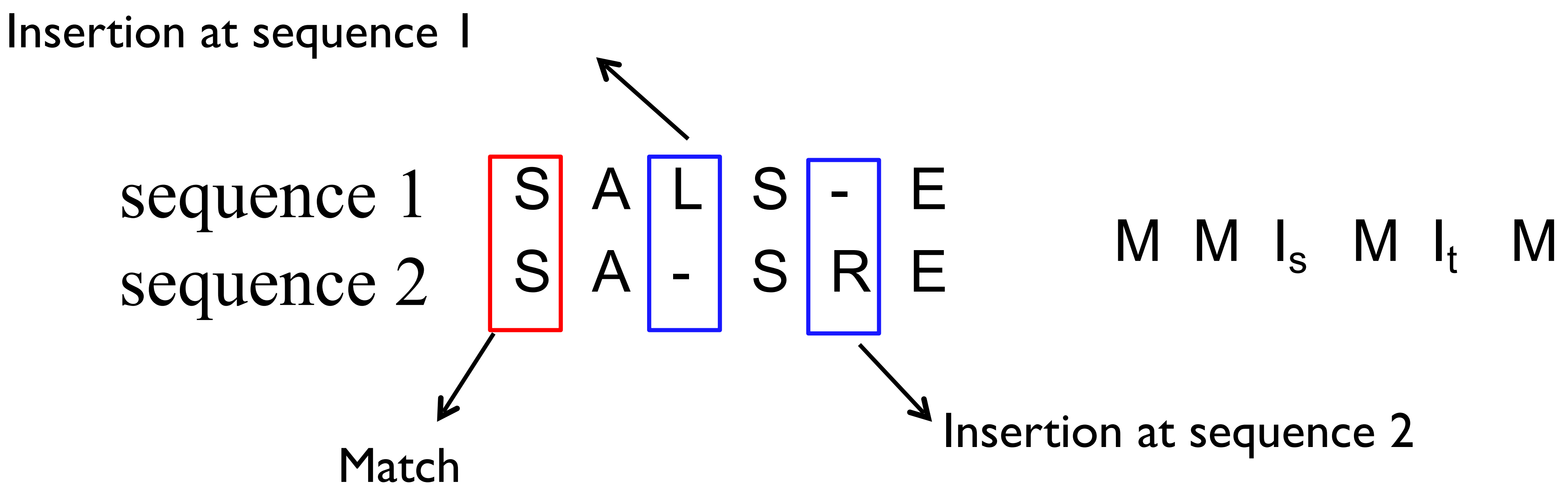
- Sometimes we need to choose whether we want to align the entire sequence.

```
A T A C G T C T
- - A C G T - -
```

```
A T A C G T C T
A - - C G - - T
```

We don’t want to punish the gap at the two ends!

We need to assign a score for each alignment



Insertion at sequence 1

sequence 1 S A L S - E

sequence 2 S A - S R E

M M I_s M I_t M

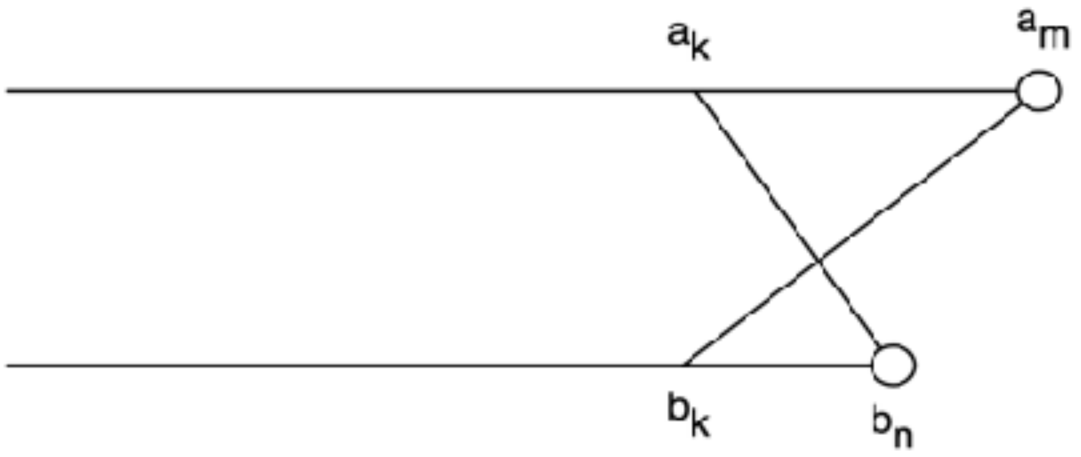
Match

Insertion at sequence 2

The score of an alignment is equal to the sum of the score contributed by each position.

Several rules must hold:

- Each position on sequence 1 can only be aligned to one position on sequence 2
- No crossing rule:



Sequence alignment

AGGCTATCACCTGACCTCCAGGCCGATGCCC

TAGCTATCACGACCGCGGTCGATTTGCCCGAC

-**AG**G**CTATCAC**CT**GACC**T**C**CA**GG**C**CGA**--**TGCCC**---

T**AG**-**CTATCAC**--**GACC**G**C**--**GG**T**CGA**TT**TGCCC**GAC

What is a good alignment?

```
AGGCTAGTT,
AGCGAAGTTT
```

```
AGGCTAGTT-
AGCGAAGTTT
```

6 matches, 3 mismatches, 1 gap

```
AGGCTA-GTT-
AG-CGAAGTTT
```

7 matches, 1 mismatch, 3 gaps

```
AGGC-TA-GTT-
AG-CG-AAGTTT
```

7 matches, 0 mismatches, 5 gaps

Scoring Function

Sequence edits:	AGGCCTC
Mutations	AGGACTC
Insertions	AGGGCCTC
Deletions	AGG . CTC

Alternative definition:

minimal edit distance

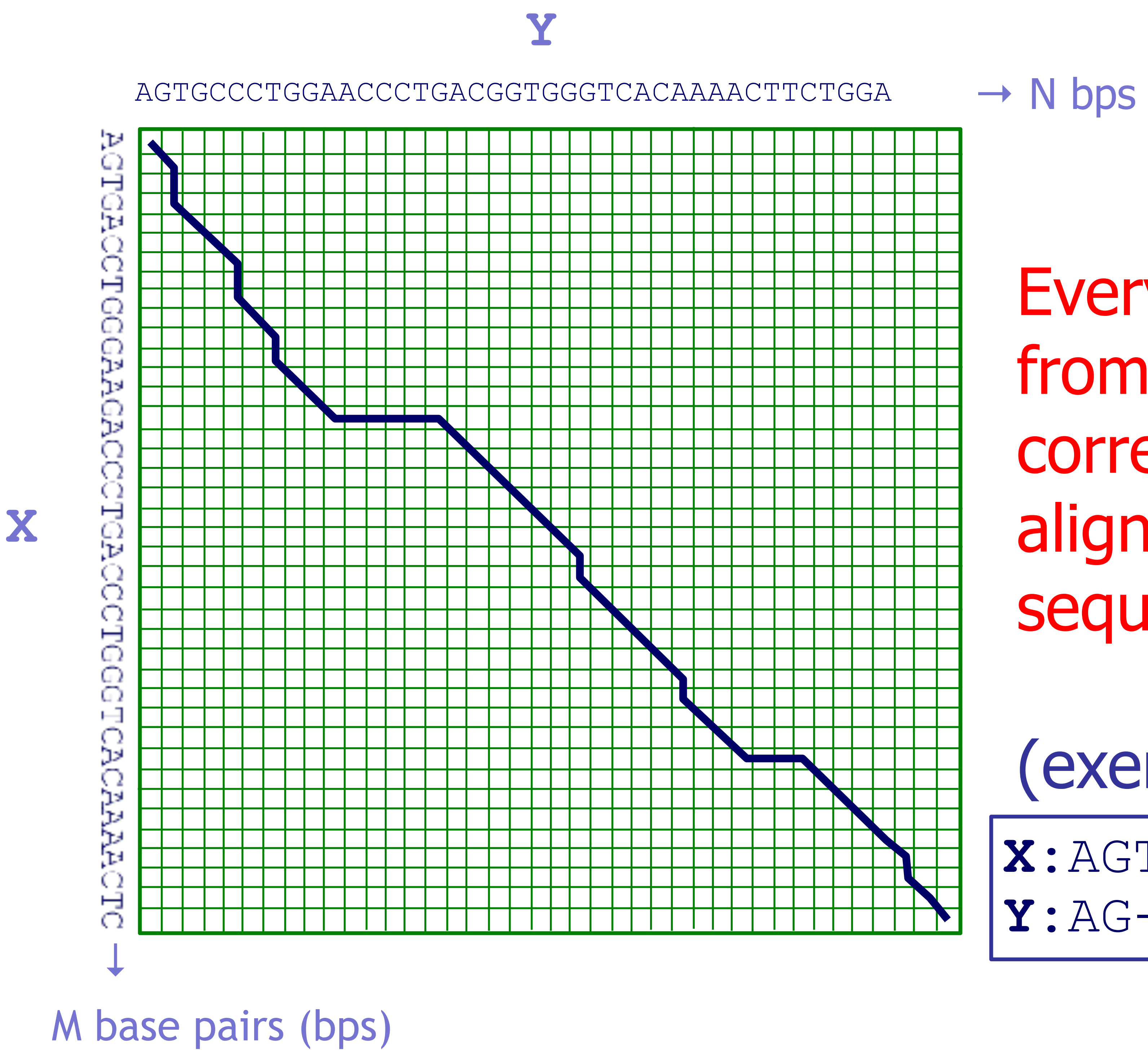
"Given two strings x, y, find minimum # of edits (insertions, deletions, mutations) to transform one string to the other"

Scoring Function:

Match:	+m
Mismatch:	-s
Gap:	-d

Score F = (# matches) × m - (# mismatches) × s – (#gaps) × d

How do we compute the best alignment?

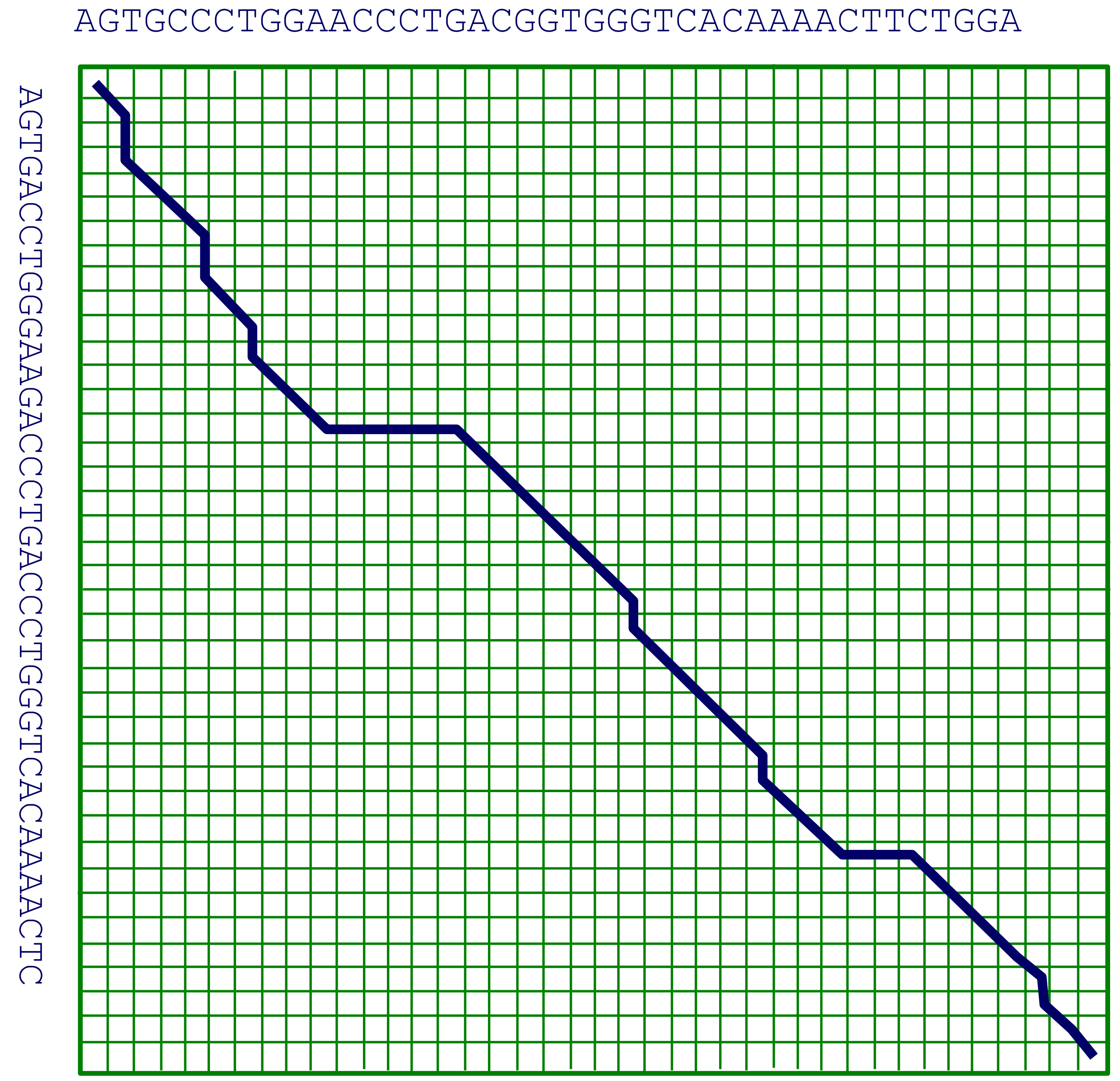

Y
AGTGCCCTGGAACCCTGACGGTGGGTCACAAAACTTCTGGA
→ N bps
X
AGTGACCTGGGAAGACCCTGACCCTGGGTCACAAAACTC
↓
M base pairs (bps)

Every non-decreasing path from (0,0) to (M, N) corresponds to an alignment of the two sequences, and vice versa.

(exercise)

```
X:AGTGACCTGGGAAGA-----C…
Y:AG--TGC--CC-TGGAACCCT…
```

How do we compute the best alignment?


AGTGCCCTGGAACCCTGACGGTGGGTCACAAAACTTCTGGA
AGTGACCTGGGAAGACCCTGACCCTGGGTCACAAAACTC

Too many possible alignments:

$>> 3^{\min(M,N)}$

Alignment is additive

Observation:

The score of aligning $x_1 \ldots\ldots x_M$

$y_1 \ldots\ldots y_N$

is additive

Say that $x_1 \ldots x_i$ $x_{i+1} \ldots x_M$

aligns to $y_1 \ldots y_j$ $y_{j+1} \ldots y_N$

The two scores add up:

$$F(x[1:M], y[1:N]) = F(x[1:i], y[1:j]) + F(x[i+1:M], y[j+1:N])$$

Dynamic Programming

- Consider subproblems for $i \leq M$ and $j \leq N$
  - Align $x_1 \ldots x_i$ to $y_1 \ldots y_j$

- Original problem is one of the subproblems
  - Align $x_1 \ldots x_M$ to $y_1 \ldots y_N$

- Each subproblem is easily solved from smaller subproblems
  - We will show next

- Then, we can apply Dynamic Programming!!!

Let $F(i, j)$ = optimal score of aligning

$x_1 \ldots\ldots x_i$

$y_1 \ldots\ldots y_j$

F is the DP "Matrix" or "Table"

"Memorization"

Scoring Function

- Sequence edits: AGGCCTC
  - Mutations AGGACTC
  - Insertions AGGGCCTC
  - Deletions AGG . CTC

Alternative definition:

minimal edit distance

"Given two strings x, y, find minimum # of edits (insertions, deletions, mutations) to transform one string to the other"

Scoring Function:

Match:	+m
Mismatch:	-s
Gap:	-d

Score F = (# matches) × m - (# mismatches) × s – (#gaps) × d

Dynamic Programming (cont'd)

Notice three possible cases:

1. x_i aligns to y_j

   $x_1 \ldots\ldots x_{i-1} \quad x_i$

   $y_1 \ldots\ldots y_{j-1} \quad y_j$

   $$F(i, j) = F(i-1, j-1) + \begin{cases} m, \text{ if } x_i = y_j \\ -s, \text{ if not} \end{cases}$$

2. x_i aligns to a gap

   $x_1 \ldots\ldots x_{i-1} \quad x_i$

   $y_1 \ldots\ldots y_j \quad -$

   $$F(i, j) = F(i-1, j) - d$$

3. y_j aligns to a gap

   $x_1 \ldots\ldots x_i \quad -$

   $y_1 \ldots\ldots y_{j-1} \quad y_j$

   $$F(i, j) = F(i, j-1) - d$$

Dynamic Programming (cont'd)

How do we know which case is correct?

Inductive assumption:

F(i, j – 1), F(i – 1, j), F(i – 1, j – 1) are optimal

Then,

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) \\ F(i-1, j) - d \\ F(i, j-1) - d \end{cases}$$

where

$$s(x_i, y_j) = \begin{cases} m, \text{ if } x_i = y_j \\ -s, \text{ if not} \end{cases}$$

Example

x = ACGCTG match: +2
y = CATGT mismatch, gap: -1

- $F(i,j)$ = optimal score of aligning $x_1, \ldots, x_i$ to $y_1, \ldots, y_j$

F =

	j	0	1	2	3	4	5	
i			C	A	T	G	T	←Y
0		0						
1	A							
2	C							
3	G							
4	C							
5	T							
6	G							

↑
X

x = ACGCTG match: +2

y = CATGT mismatch, gap: -1

	j	0	1	2	3	4	5
i			**C**	**A**	**T**	**G**	**T**
0		0					
1	**A**						
2	**C**						
3	**G**						
4	**C**						
5	**T**						
6	**G**						

←**Y**

↑

X

x = ACGCTG match: +2
y = CATGT mismatch, gap: -1

	j	0	1	2	3	4	5	
i			C	A	T	G	T	←Y
0		0	-1					
1	A							
2	C							
3	G							
4	C							
5	T							
6	G							
	↑ X							

$\boxed{\begin{matrix}-\\ \mathrm{C}\end{matrix}}$ s(-,C) = -1

x = ACGCTG match: +2

y = CATGT mismatch, gap: -1

	j	0	1	2	3	4	5	
i			C	A	T	G	T	←Y
0		0	-1	-2	-3	-4	-5	
1	A	-1						
2	C	-2						
3	G	-3						
4	C	-4						
5	T	-5						
6	G	-6						

↑
X

$$\boxed{\begin{array}{c} - \\ C \end{array}} \quad s(-,C) = -1$$

x = ACGCTG match: +2
y = CATGT mismatch, gap: -1

	j	0	1	2	3	4	5
i			C	A	T	G	T
0		0	-1	-2	-3	-4	-5
1	A	-1					
2	C	-2					
3	G	-3					
4	C	-4					
5	T	-5					
6	G	-6					

←Y

↑
X

A
-

s(A,-) = -1

x = ACGCTG match: +2

y = CATGT mismatch, gap: -1

	j	0	1	2	3	4	5
i			C	A	T	G	T
0		0	-1	-2	-3	-4	-5
1	A	-1					
2	C	-2					
3	G	-3					
4	C	-4					
5	T	-5					
6	G	-6					

←Y

↑
X

A C
- -
-1

s(C,-) = -1

x = ACGCTG   match: +2

y = CATGT   mismatch, gap: -1

	j	0	1	2	3	4	5
i			C	A	T	G	T
0		0	-1	-2	-3	-4	-5
1	A	-1					
2	C	-2					
3	G	-3					
4	C	-4					
5	T	-5					
6	G	-6					

←Y

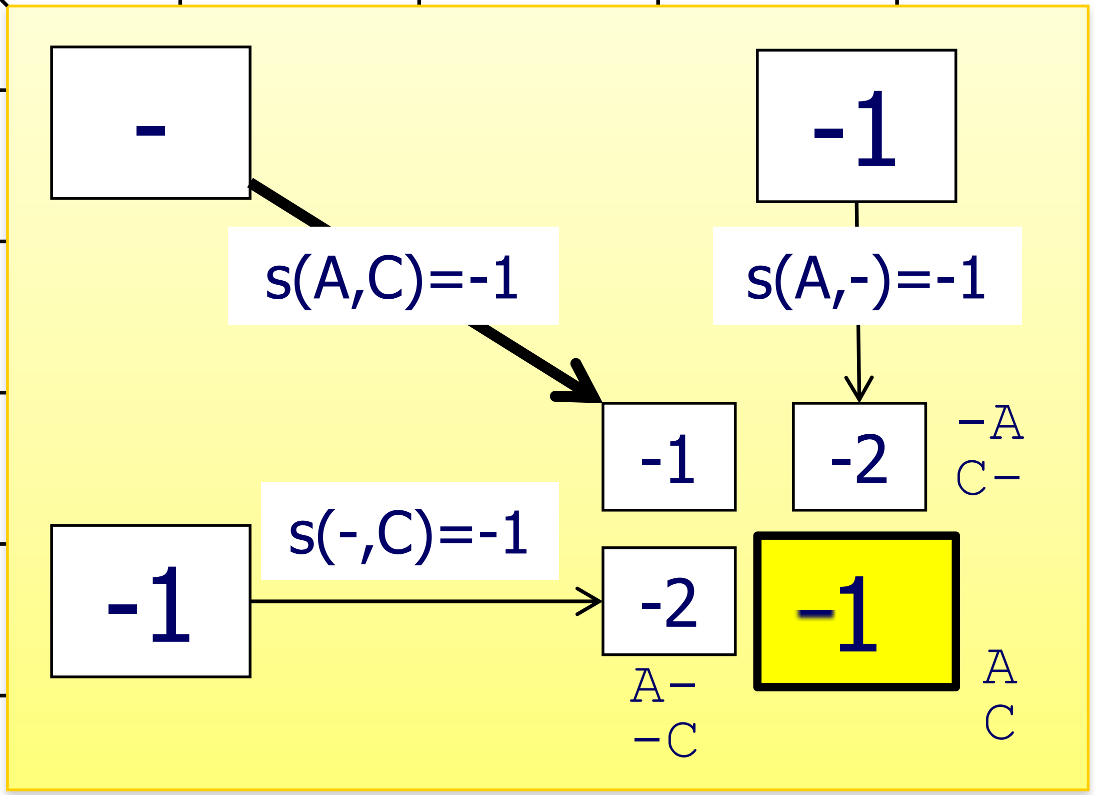
↑
X

x = ACGCTG match: +2

y = CATGT mismatch, gap: -1

	j	0	1	2	3	4	5	
i			C	A	T	G	T	←Y
0		0	-1	-2	-3	-4	-5	
1	A	-1	-1					
2	C	-2						
3	G	-3						
4	C	-4						
5	T	-5						
6	G	-6						

↑
X



x = ACGCTG match: +2

y = CATGT mismatch, gap: -1

	j	0	1	2	3	4	5
i			C	A	T	G	T
0		0	-1	-2	-3	-4	-5
1	A	-1	-1				
2	C	-2					
3	G	-3					
4	C	-4					
5	T	-5					
6	G	-6					

←Y

↑

X

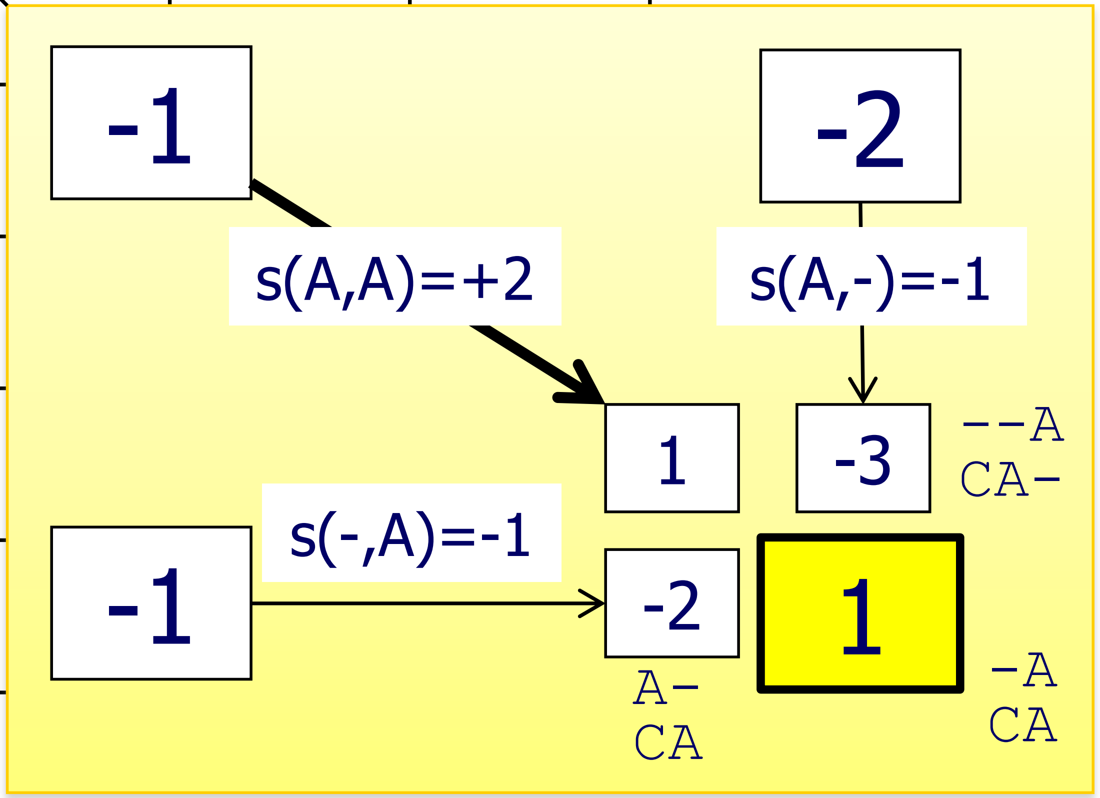
x = ACGCTG match: +2

y = CATGT mismatch, gap: -1

	j	0	1	2	3	4	5
i			C	A	T	G	T
0		0	-1	-2	-3	-4	-5
1	A	-1	-1	1			
2	C	-2					
3	G	-3					
4	C	-4					
5	T	-5					
6	G	-6					

←Y

↑
X



x = ACGCTG match: +2

y = CATGT mismatch, gap: -1

	j	0	1	2	3	4	5
i			C	A	T	G	T
0		0	-1	-2	-3	-4	-5
1	A	-1	-1	1	0	-1	-2
2	C	-2	1	0	0	-1	-2
3	G	-3	0	0	-1	2	1
4	C	-4	-1	-1	-1	1	1
5	T	-5	-2	-2	1	0	3
6	G	-6	-3	-3	0	3	2

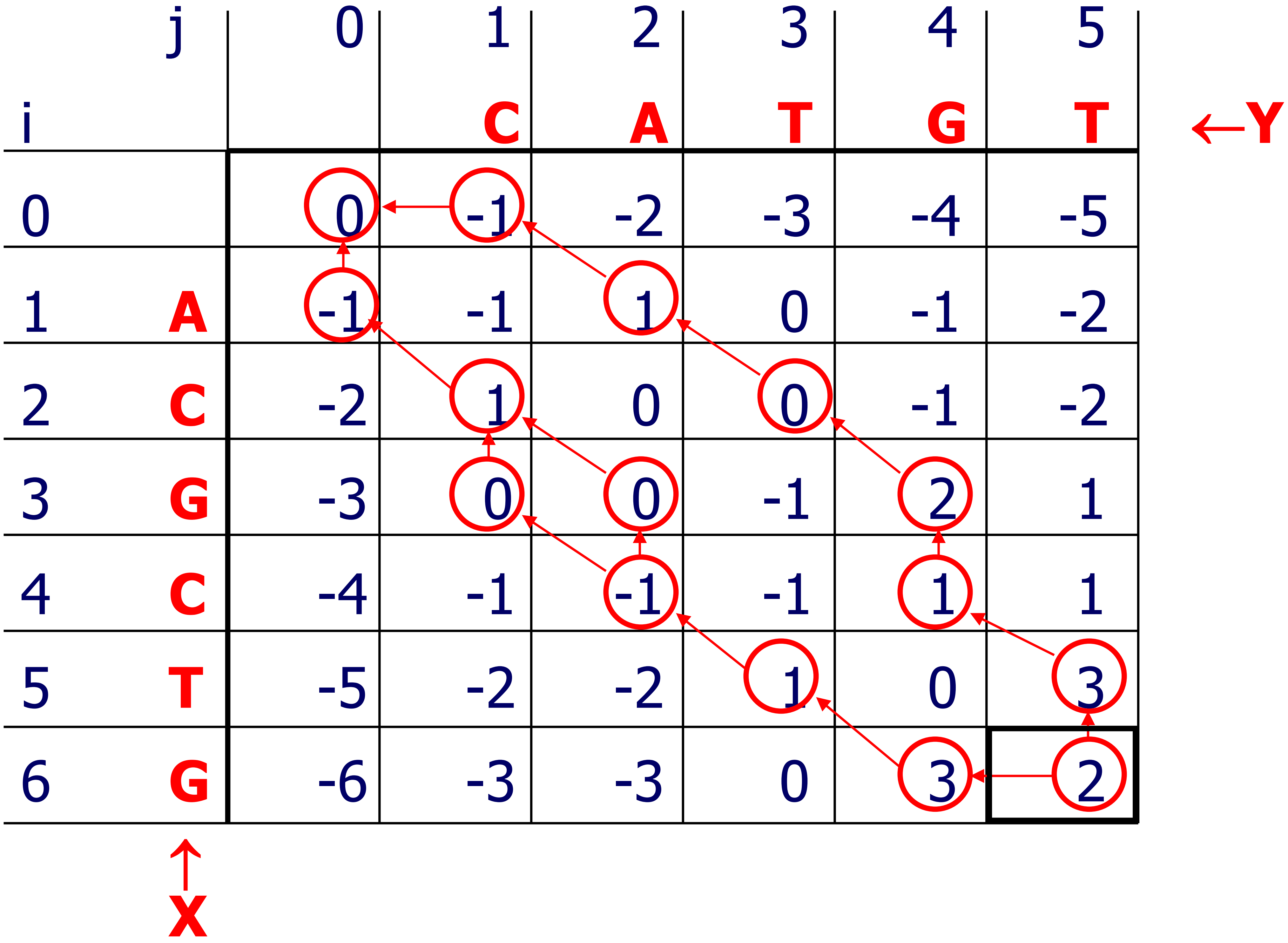
←Y

↑
X

Time
= O(MN)

Finding alignments: trace back

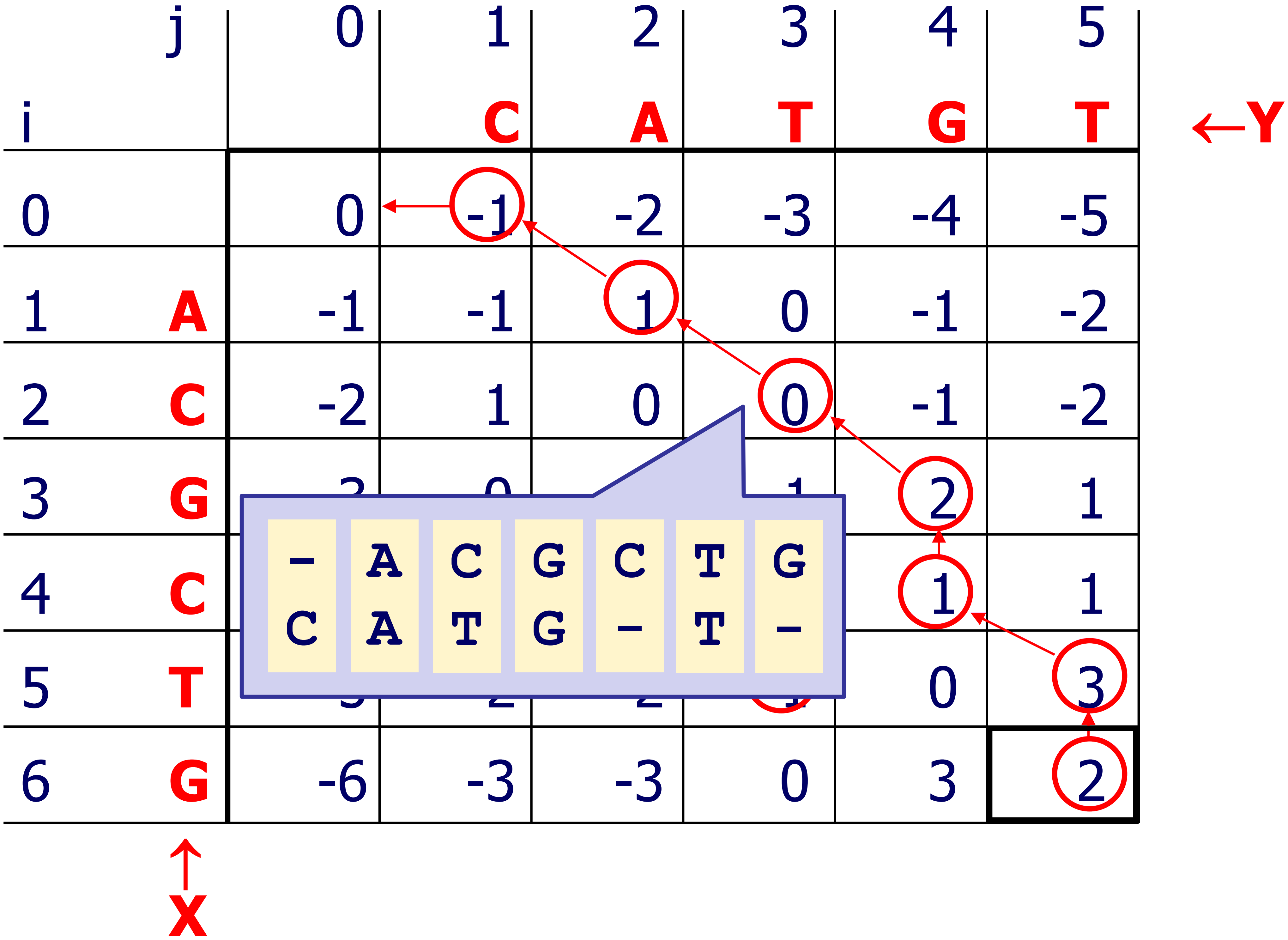
Arrows = (ties for) max in F(i,j); 3 LR-to-UL paths = 3 optimal alignments



	j	0	1	2	3	4	5	
i			C	A	T	G	T	←Y
0		0	-1	-2	-3	-4	-5	
1	A	-1	-1	1	0	-1	-2	
2	C	-2	1	0	0	-1	-2	
3	G	-3	0	0	-1	2	1	
4	C	-4	-1	-1	-1	1	1	
5	T	-5	-2	-2	1	0	3	
6	G	-6	-3	-3	0	3	2	

↑
X

Finding alignments: trace back



i \ j		0	1	2	3	4	5	
			C	A	T	G	T	←Y
0		0	-1	-2	-3	-4	-5	
1	A	-1	-1	1	0	-1	-2	
2	C	-2	1	0	0	-1	-2	
3	G	[illegible]	0	[illegible]	[illegible]	2	1	
4	C	[illegible]	[illegible]	[illegible]	[illegible]	1	1	
5	T	[illegible]	[illegible]	[illegible]	[illegible]	0	3	
6	G	-6	-3	-3	0	3	2	
	↑ X							

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

The Needleman-Wunsch Algorithm

1. Initialization.
   a. $F(0, 0) = 0$
   b. $F(0, j) = - j \times d$
   c. $F(i, 0) = - i \times d$

2. Main Iteration. Filling-in partial alignments

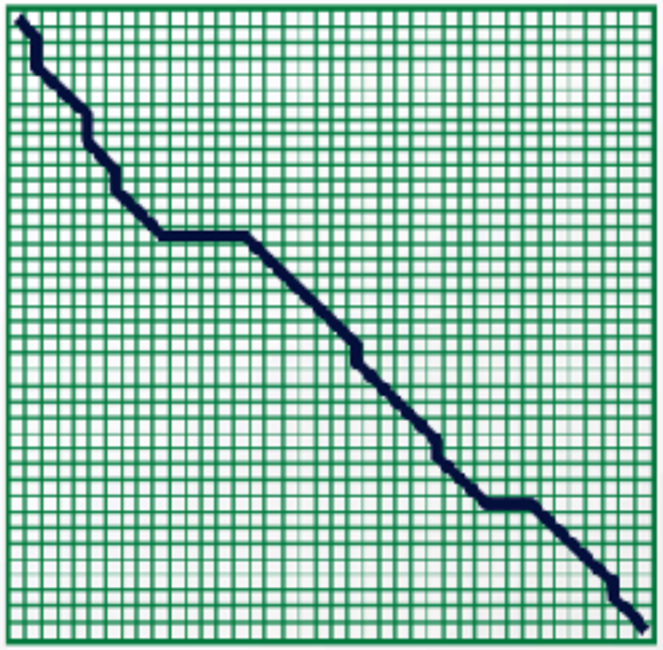
```
For each   i = 1……M
  For each   j = 1……N
```

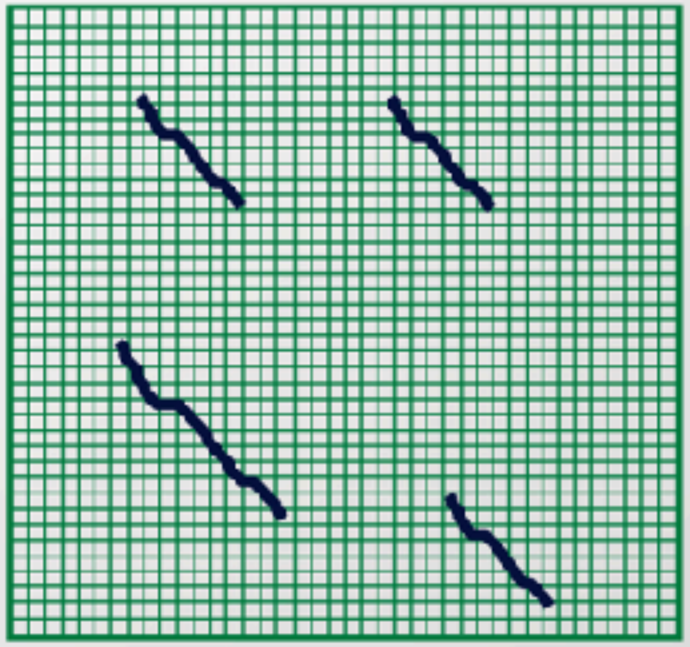
$$F(i, j) = \max \begin{cases} F(i-1, j-1) + s(x_i, y_j) & \text{[case 1]} \\ F(i-1, j) - d & \text{[case 2]} \\ F(i, j-1) - d & \text{[case 3]} \end{cases}$$

$$Ptr(i, j) = \begin{cases} \text{DIAG,} & \text{if [case 1]} \\ \text{UP,} & \text{if [case 2]} \\ \text{LEFT,} & \text{if [case 3]} \end{cases}$$

3. Termination. F(M, N) is the optimal score, and from Ptr(M, N) can trace back optimal alignment

Global Alignment vs. Local alignment





Needleman-Wunsch algorithm

Initialization: $F(0, 0) = 0$

Iteration:

$$F(i, j) = \max \begin{cases} F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: Bottom right

Smith-Waterman algorithm

Initialization: $F(0, j) = F(i, 0) = 0$

Iteration:

$$F(i, j) = \max \begin{cases} 0 \\ F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: **Anywhere**

Performance

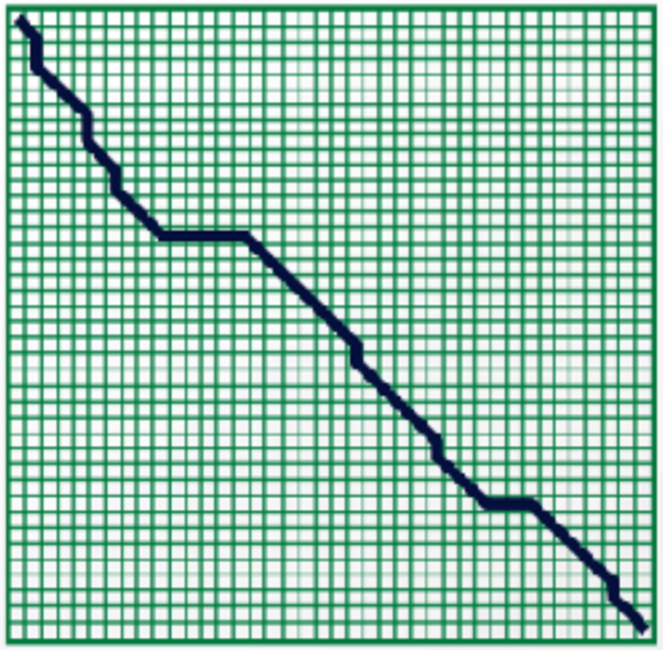
- Time:

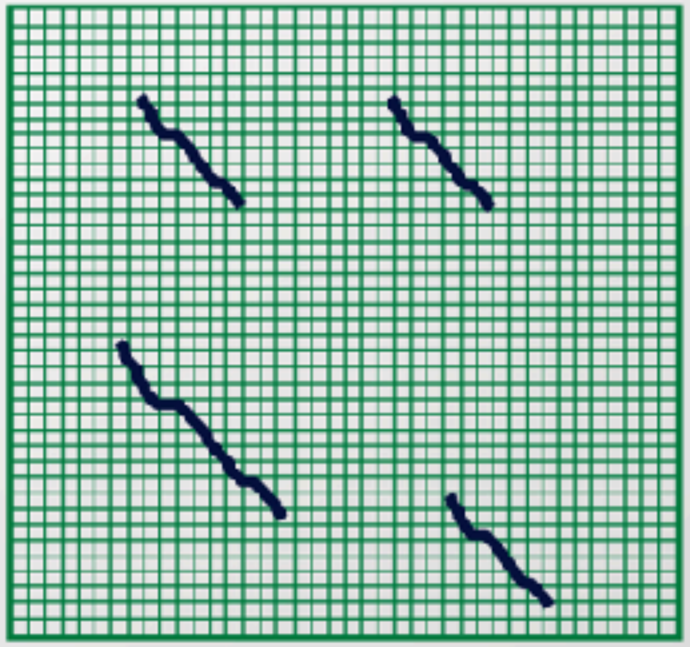
  O(NM)

- Space:

  O(NM)

Global Alignment vs. Local alignment





Needleman-Wunsch algorithm

Initialization: $F(0, 0) = 0$

Iteration:

$$F(i, j) = \max \begin{cases} F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: Bottom right

Smith-Waterman algorithm

Initialization: $F(0, j) = F(i, 0) = 0$

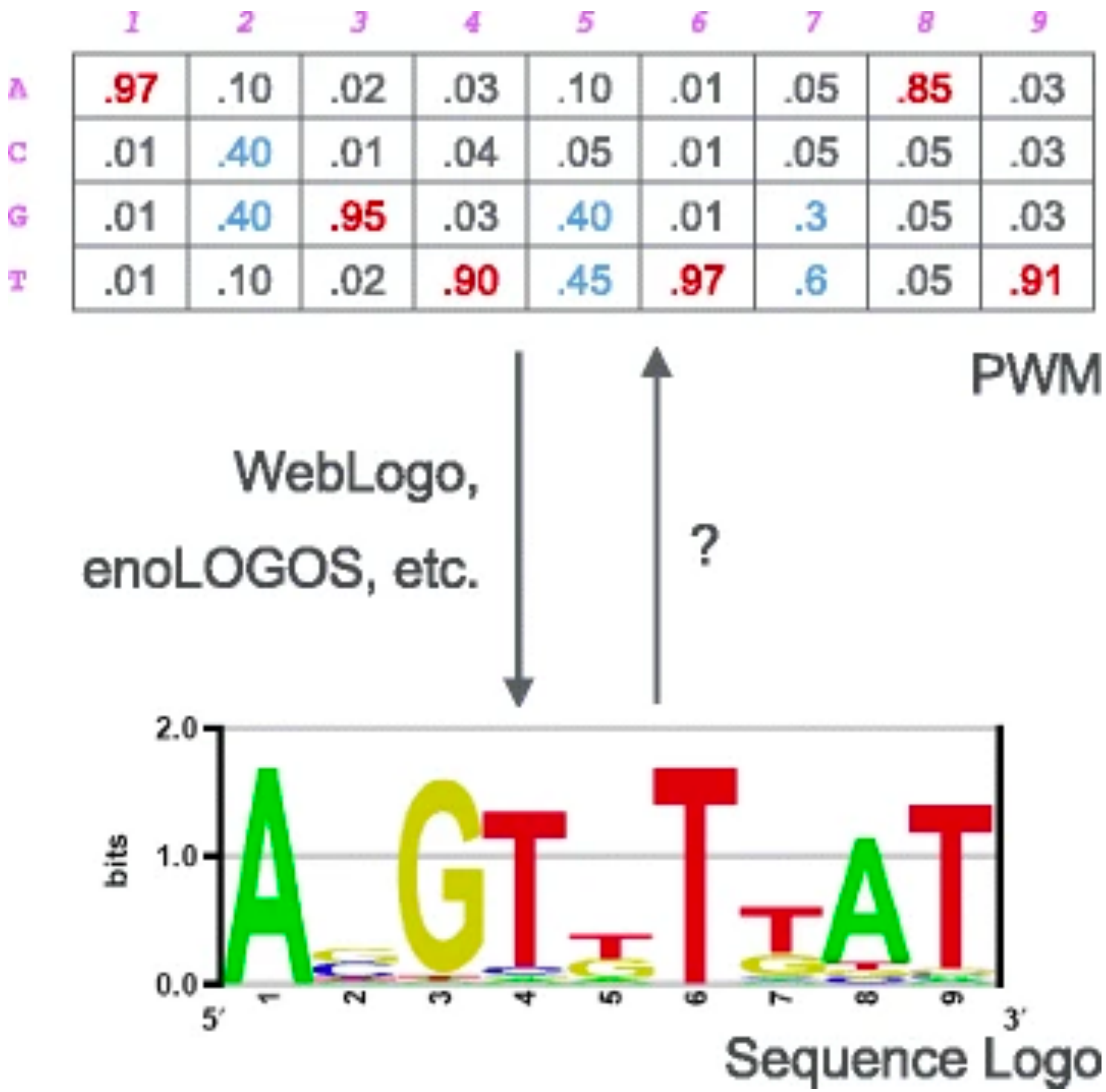
Iteration:

$$F(i, j) = \max \begin{cases} 0 \\ F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(x_i, y_j) \end{cases}$$

Termination: **Anywhere**

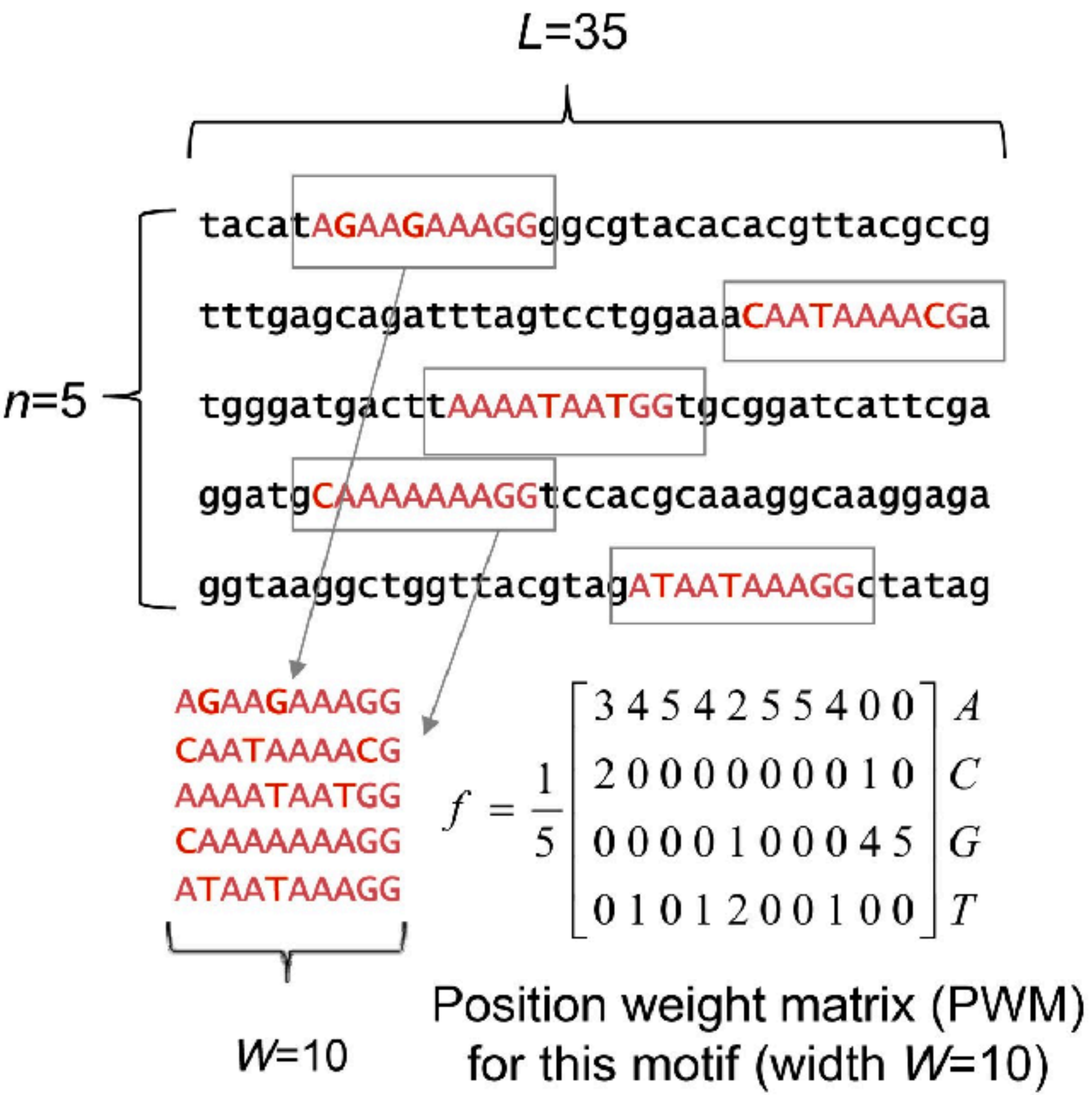
- What if we only penalize the gap at the beginning
- What if we only penalize the gap at the end

Motif: probabilistic representation of a sequence



	1	2	3	4	5	6	7	8	9
A	.97	.10	.02	.03	.10	.01	.05	.85	.03
C	.01	.40	.01	.04	.05	.01	.05	.05	.03
G	.01	.40	.95	.03	.40	.01	.3	.05	.03
T	.01	.10	.02	.90	.45	.97	.6	.05	.91

https://bmcgenomics.biomedcentral.com/articles/10.1186/s12864-017-4023-9/figures/1



L=35

n=5

```
tacatAGAAGAAAGGggcgtacacacgttacgccg
tttgagcagatttagtcctggaaaCAATAAAACGa
tgggatgacttAAAATAATGGtgcggatcattcga
ggatgCAAAAAAAAGGtccacgcaaaggcaagaga
ggtaaggctggttacgtagATAATAAAGGctatag
```

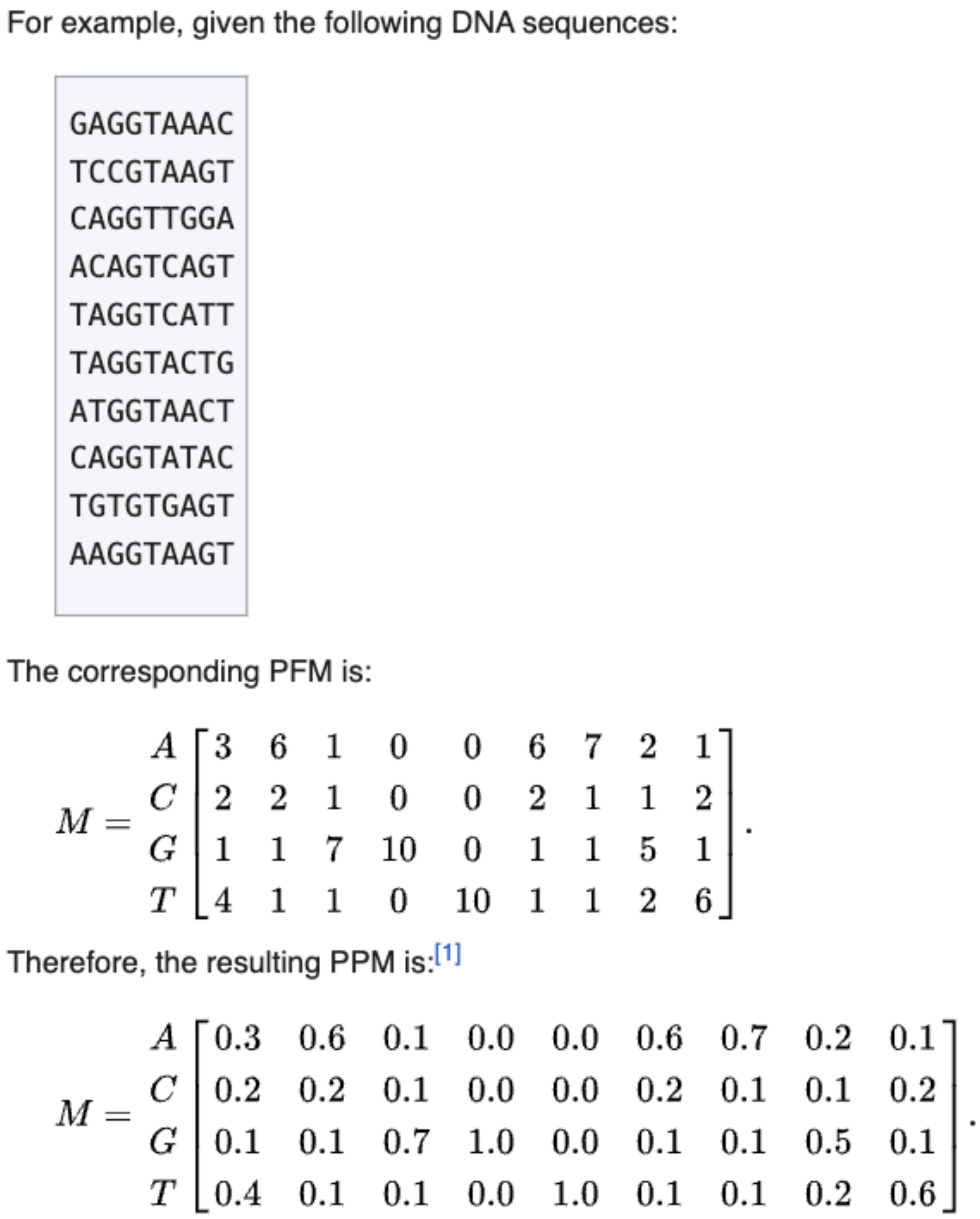
```
AGAAGAAAGG
CAATAAAACG
AAAATAATGG
CAAAAAAAGG
ATAATAAAGG
```

W=10

$$f = \frac{1}{5}\begin{array}{c}\left[\begin{array}{cccccccccc} 3&4&5&4&2&5&5&4&0&0 \\ 2&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&1&0&0&0&4&5 \\ 0&1&0&1&2&0&0&1&0&0 \end{array}\right]\end{array}\begin{array}{c} A \\ C \\ G \\ T \end{array}$$

Position weight matrix (PWM) for this motif (width W=10)

https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-13-151/figures/1



For example, given the following DNA sequences:

```
GAGGTAAAC
TCCGTAAGT
CAGGTTGGA
ACAGTCAGT
TAGGTCATT
TAGGTACTG
ATGGTAACT
CAGGTATAC
TGTGTGAGT
AAGGTAAGT
```

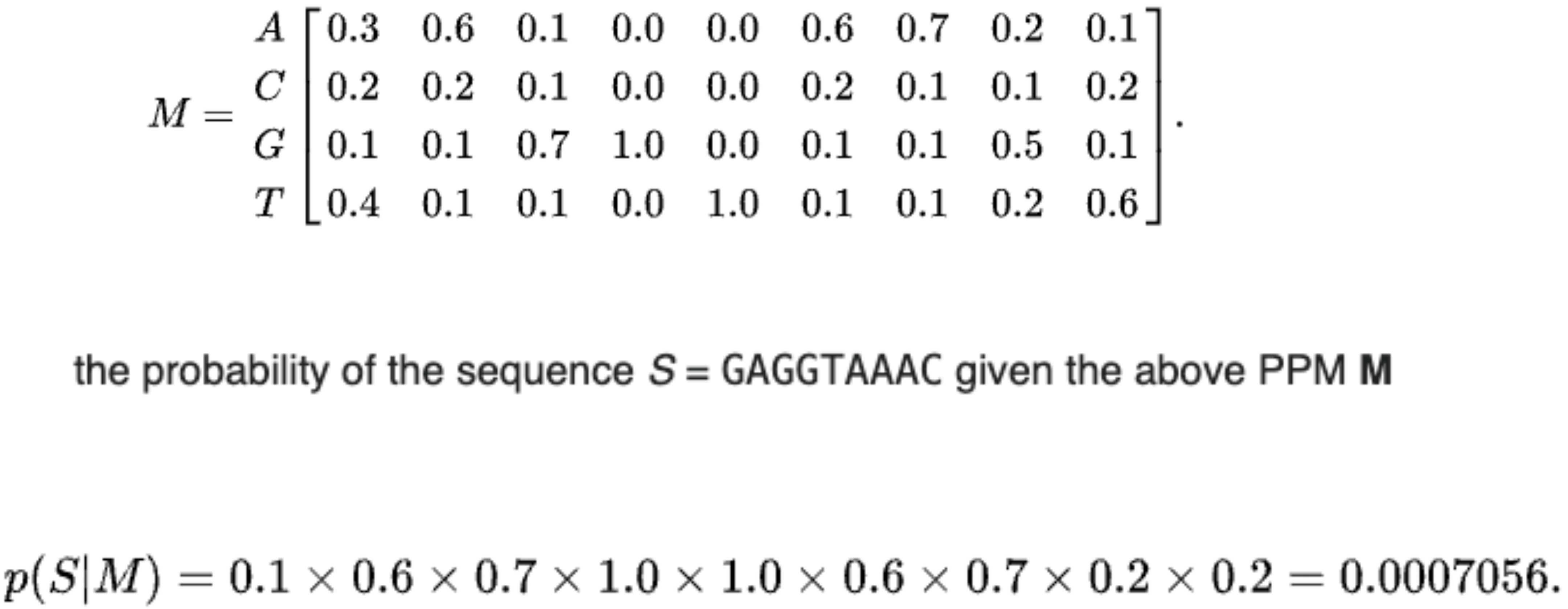
The corresponding PFM is:

$$M = \begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{bmatrix} 3 & 6 & 1 & 0 & 0 & 6 & 7 & 2 & 1 \\ 2 & 2 & 1 & 0 & 0 & 2 & 1 & 1 & 2 \\ 1 & 1 & 7 & 10 & 0 & 1 & 1 & 5 & 1 \\ 4 & 1 & 1 & 0 & 10 & 1 & 1 & 2 & 6 \end{bmatrix}.$$

Therefore, the resulting PPM is:[1]

$$M = \begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{bmatrix} 0.3 & 0.6 & 0.1 & 0.0 & 0.0 & 0.6 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.1 & 0.0 & 0.0 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.7 & 1.0 & 0.0 & 0.1 & 0.1 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.1 & 0.0 & 1.0 & 0.1 & 0.1 & 0.2 & 0.6 \end{bmatrix}.$$

https://en.wikipedia.org/wiki/Position_weight_matrix#:~:text=A%20position%20weight%20matrix%20(PWM,represented%20graphically%20as%20sequence%20logos.



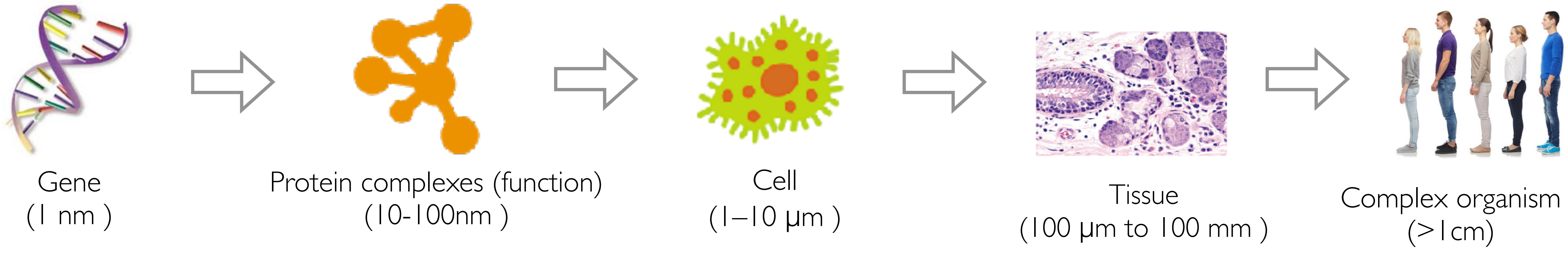
$$M = \begin{array}{c} A \\ C \\ G \\ T \end{array} \begin{bmatrix} 0.3 & 0.6 & 0.1 & 0.0 & 0.0 & 0.6 & 0.7 & 0.2 & 0.1 \\ 0.2 & 0.2 & 0.1 & 0.0 & 0.0 & 0.2 & 0.1 & 0.1 & 0.2 \\ 0.1 & 0.1 & 0.7 & 1.0 & 0.0 & 0.1 & 0.1 & 0.5 & 0.1 \\ 0.4 & 0.1 & 0.1 & 0.0 & 1.0 & 0.1 & 0.1 & 0.2 & 0.6 \end{bmatrix}.$$

the probability of the sequence S = GAGGTAAAC given the above PPM **M**

$$p(S|M) = 0.1 \times 0.6 \times 0.7 \times 1.0 \times 1.0 \times 0.6 \times 0.7 \times 0.2 \times 0.2 = 0.0007056.$$

https://en.wikipedia.org/wiki/Position_weight_matrix#:~:text=A%20position%20weight%20matrix%20(PWM,represented%20graphically%20as%20sequence%20logos.

Computational methods for biology at different scales



Gene
(1 nm)

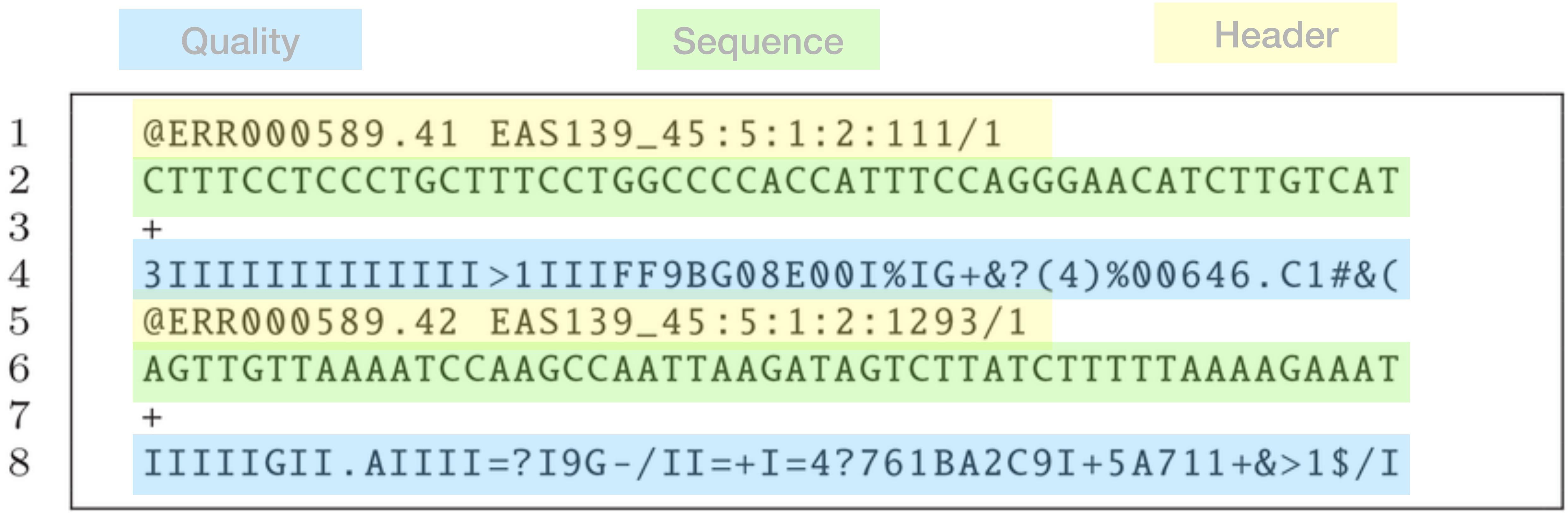
Protein complexes (function)
(10-100nm)

Cell
(1–10 µm)

Tissue
(100 µm to 100 mm)

Complex organism
(>1cm)

What does a fastq file look like?


Quality
Sequence
Header

```
@ERR000589.41 EAS139_45:5:1:2:111/1
CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
+
3IIIIIIIIIIIII>1IIIFF9BG08E00I%IG+&?(4)%00646.C1#&(
@ERR000589.42 EAS139_45:5:1:2:1293/1
AGTTGTTAAAATCCAAGCCAATTAAGATAGTCTTATCTTTTTAAAAGAAAT
+
IIIIIGII.AIIII=?I9G-/II=+I=4?761BA2C9I+5A711+&>1$/I
```

Very large! ~300000000 lines
Quality: ASCII chars

What should we do? Map each short sequence (we call it read) to the entire human genome

What does a fastq file look like?

Reference genome: "average" human genome.
Most widely used human genome GRCh38: derived from 13 thirteen anonymous volunteers

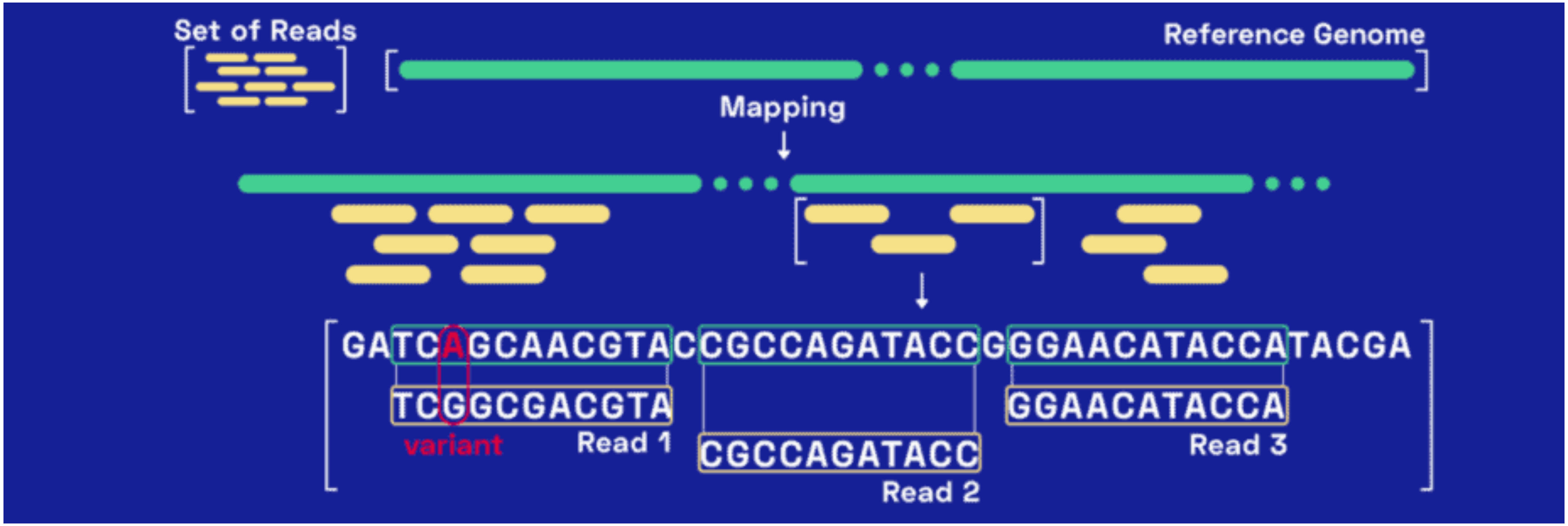
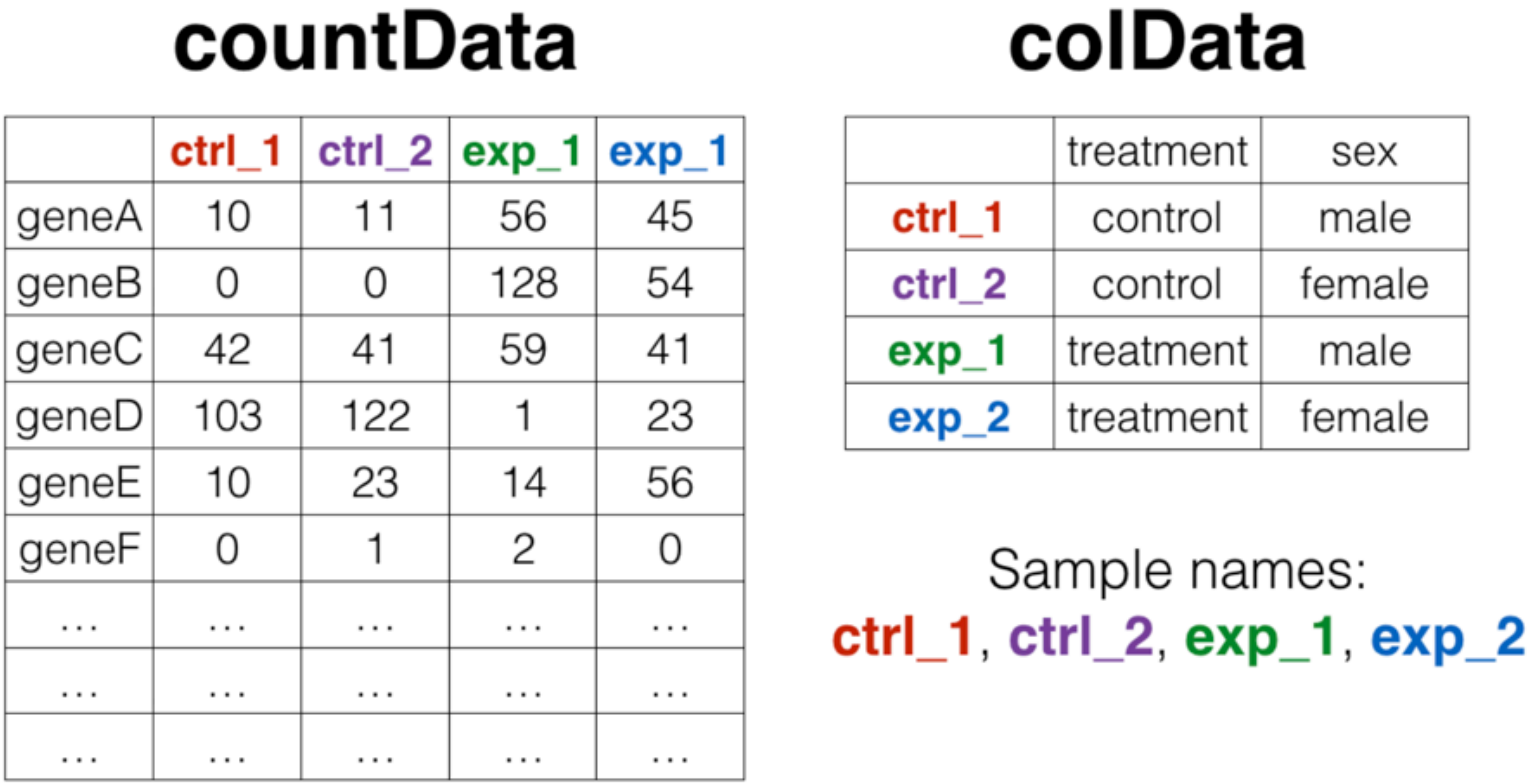


Image from Nebula Genomics

Processed data



countData

	ctrl_1	ctrl_2	exp_1	exp_1
geneA	10	11	56	45
geneB	0	0	128	54
geneC	42	41	59	41
geneD	103	122	1	23
geneE	10	23	14	56
geneF	0	1	2	0
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...

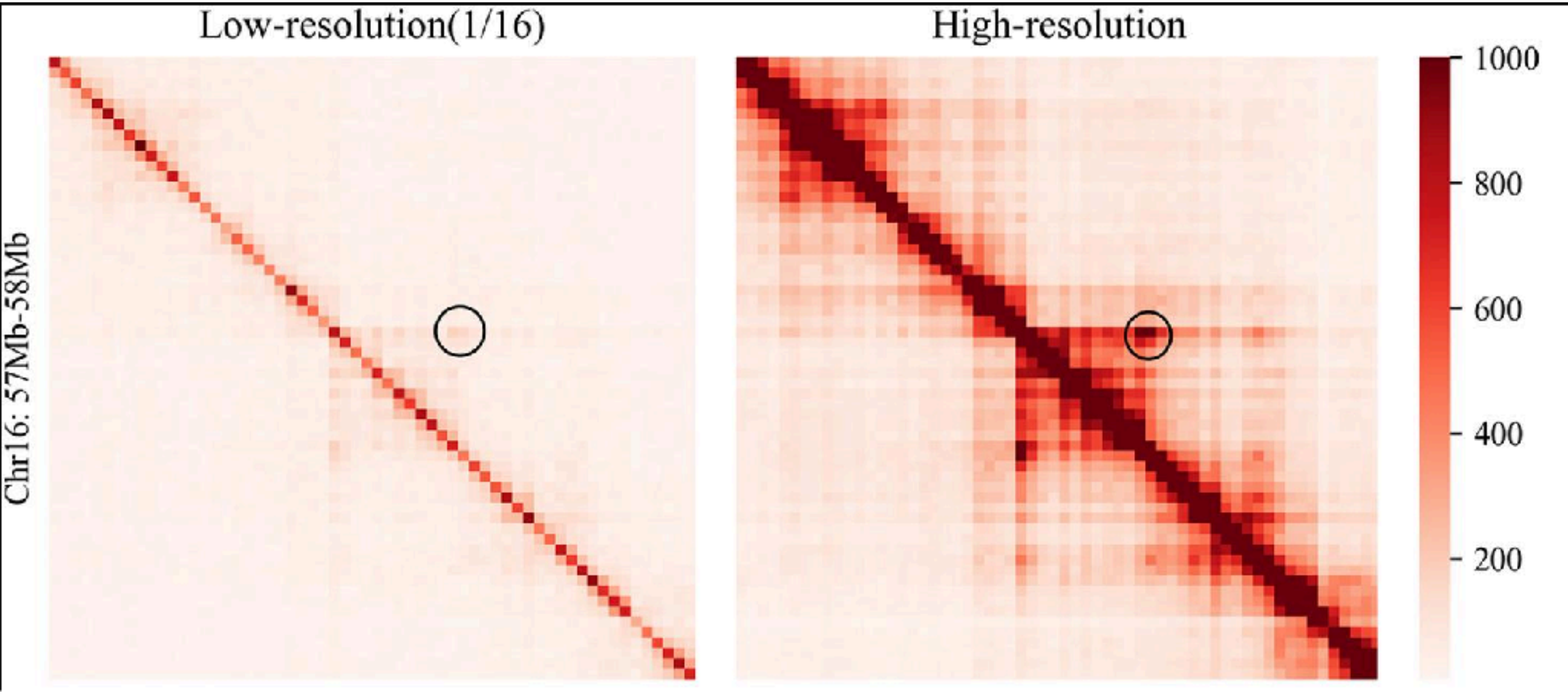
colData

	treatment	sex
ctrl_1	control	male
ctrl_2	control	female
exp_1	treatment	male
exp_2	treatment	female

Sample names:
ctrl_1, **ctrl_2**, **exp_1**, **exp_2**

source: https://bioconnector.github.io/bims8382/r-rnaseq-airway.html

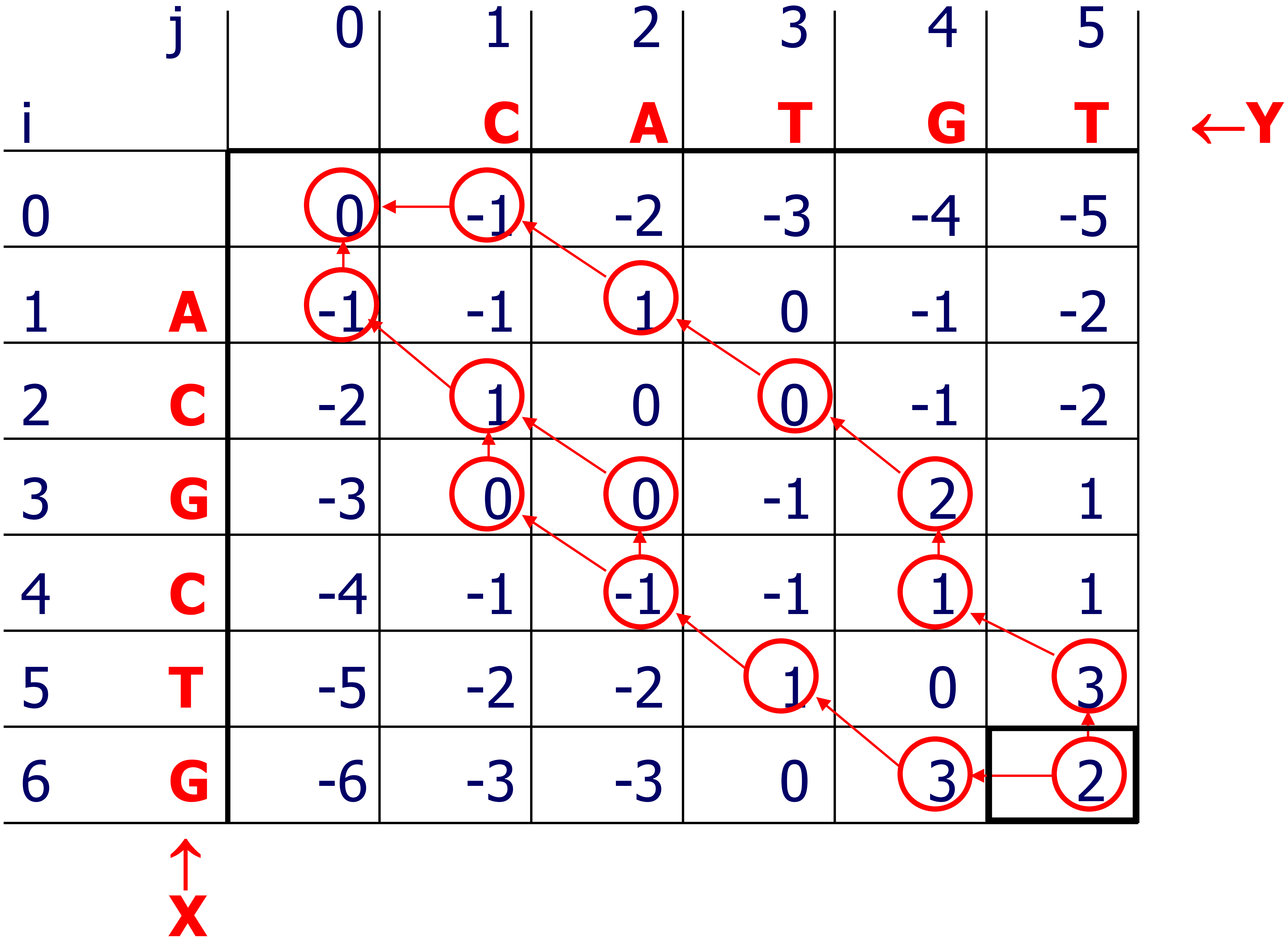
Data structure and computational problem



source: SRHiC: A Deep Learning Model to Enhance the Resolution of Hi-C Data

Finding alignments: trace back

Arrows = (ties for) max in F(i,j); 3 LR-to-UL paths = 3 optimal alignments



	j	0	1	2	3	4	5	
i			C	A	T	G	T	←Y
0		0	-1	-2	-3	-4	-5	
1	A	-1	-1	1	0	-1	-2	
2	C	-2	1	0	0	-1	-2	
3	G	-3	0	0	-1	2	1	
4	C	-4	-1	-1	-1	1	1	
5	T	-5	-2	-2	1	0	3	
6	G	-6	-3	-3	0	3	2	
	↑ X							