

CSE 427 Computational Biology

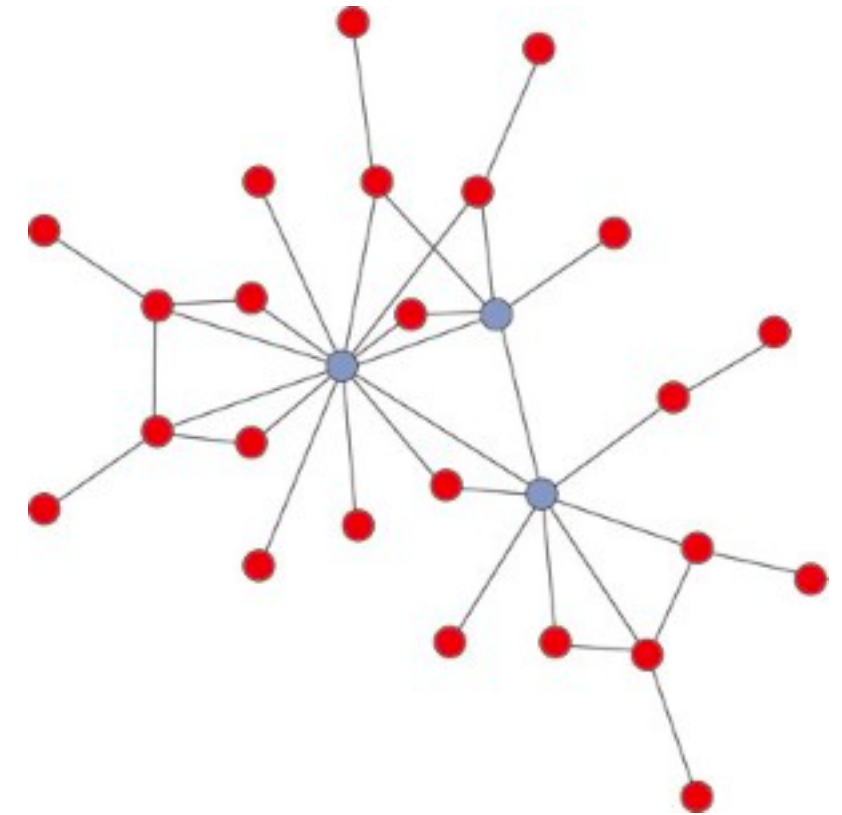
Lecture 1: Introduction

Goal for CSE427

- Learn how to collaborate with biologists and doctors to solve a biomedical problem using computational approaches
- We don't need to define the problem or propose an important problem
 - Our collaborators (biologists/doctors) will do it.
- Computational approaches
 - Algorithm: dynamic programming, graph shortest distance
 - Machine learning: LSTM, GPT, Graph neural network
- Learn how to communicate
 - How to understand and formulate a biomedical problem
 - How to explain and present our computational solution/results to others

A concrete example

- Biologists: I have lots of protein-protein interaction data. I would like to find which protein is the most important one.
- Translate to computer science language
 - Protein-protein interaction: a network of protein nodes
 - Which protein is important: find a machine learning method that can identify important nodes in a network
- Our goal: understand biomedical problem and find the appropriate computational solution



Goal for CSE427

- Understand the biomedical problem
 - A data structure perspective: understand the data first
- Find an off-the-shelf computational tool (comp bio course)
 - Comp bio research: propose/develop new computational solutions for existing biomedical problems that do not have an off-the-shelf computational solutions.
 - Advanced comp bio research: propose/identify new biomedical problems that can be addressed by emerging computational solutions (GPT can solve new biomedical problem)
- CSE427 offers a transition from comp bio course to comp bio research

Grading

- No exams, no quizzes
- Three homework assignments (60%)
 - HW1 20%, HW2 20%, HW 3 20%
 - Submit to **Gradescope**
 - Written assignments only, no programming.
- Discussion and attending five research showcase lectures (20%)
- Literature review (20%)
 - Optional: a research project

Research showcase

- lectures by Allen School PhD students working on comp bio projects at Allen School
- Learn more about research opportunities in the Allen School
- 45-minute presentation
- Discussion

Literature review

- Pick just one paper related to the topic presented at research showcase
- Submit a one-page review by the end of the quarter
 - How to understand a research paper
 - Significance: why is this problem important?
 - Novelty: what is the difference between this paper and others?
 - Approaches: Rigorous of the approach and technical contribution
 - Limitation: your thoughts on the paper

Course logistics

- Lecture time: Tuesday and Thursday 11:30-12:50pm
- Course mailing email: cse427a_au25@uw.edu

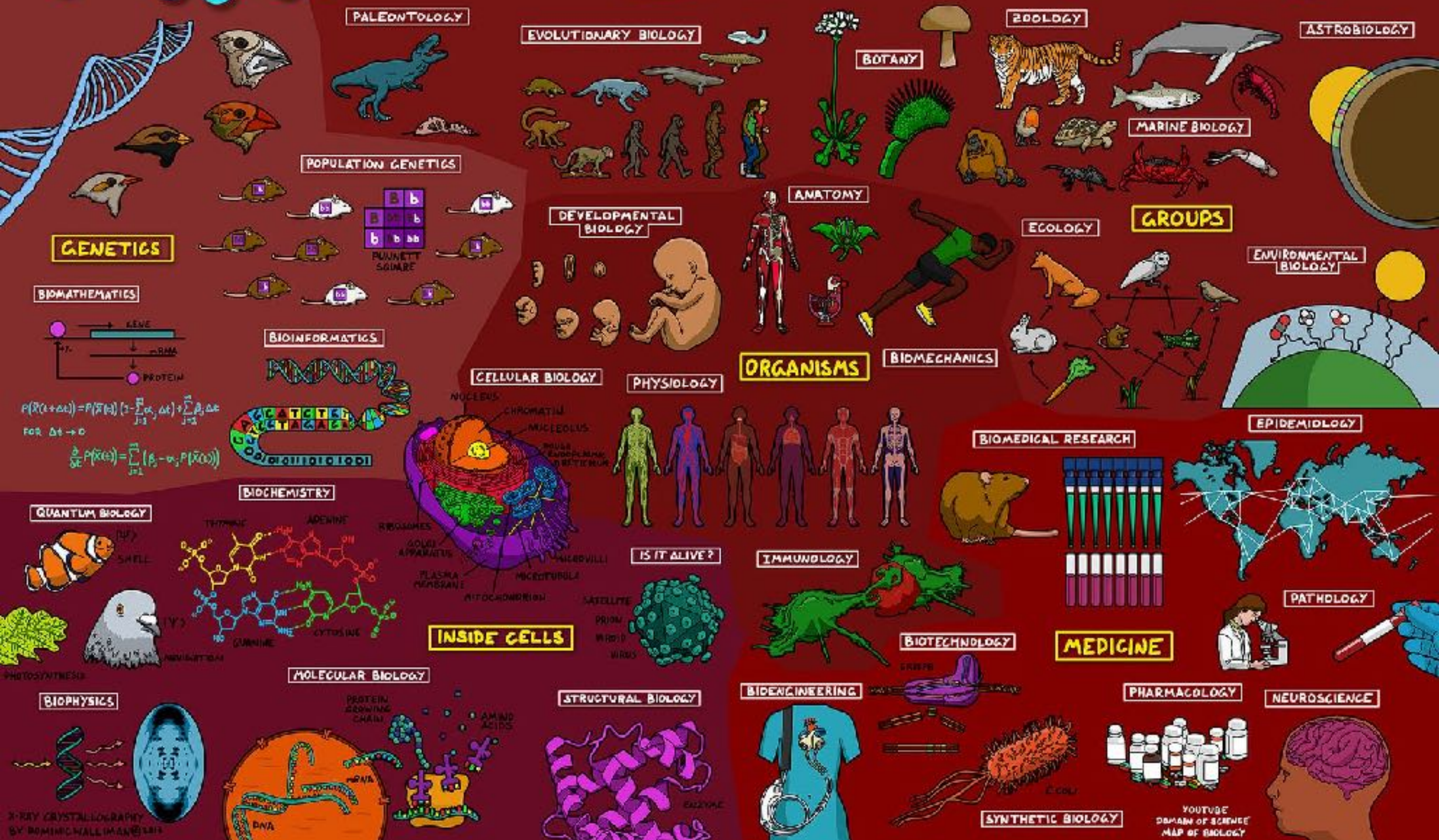
Instructor and TAs

- Instructor
 - Sheng Wang
 - <https://homes.cs.washington.edu/~swang/>
 - swang@cs.washington.edu
 - Office hour: Wed 12-1 pm (zoom)
- TA:
 - Hanwen Xu (zucksliu@cs.washington.edu)
 - Office hour: Thursday 10:00am - 11:00am
 - Yongkang Li(chentong@cs.washington.edu)
 - Office hour: Friday 10:00am - 11:00am
 - Zoom: <https://washington.zoom.us/j/93707841384>.

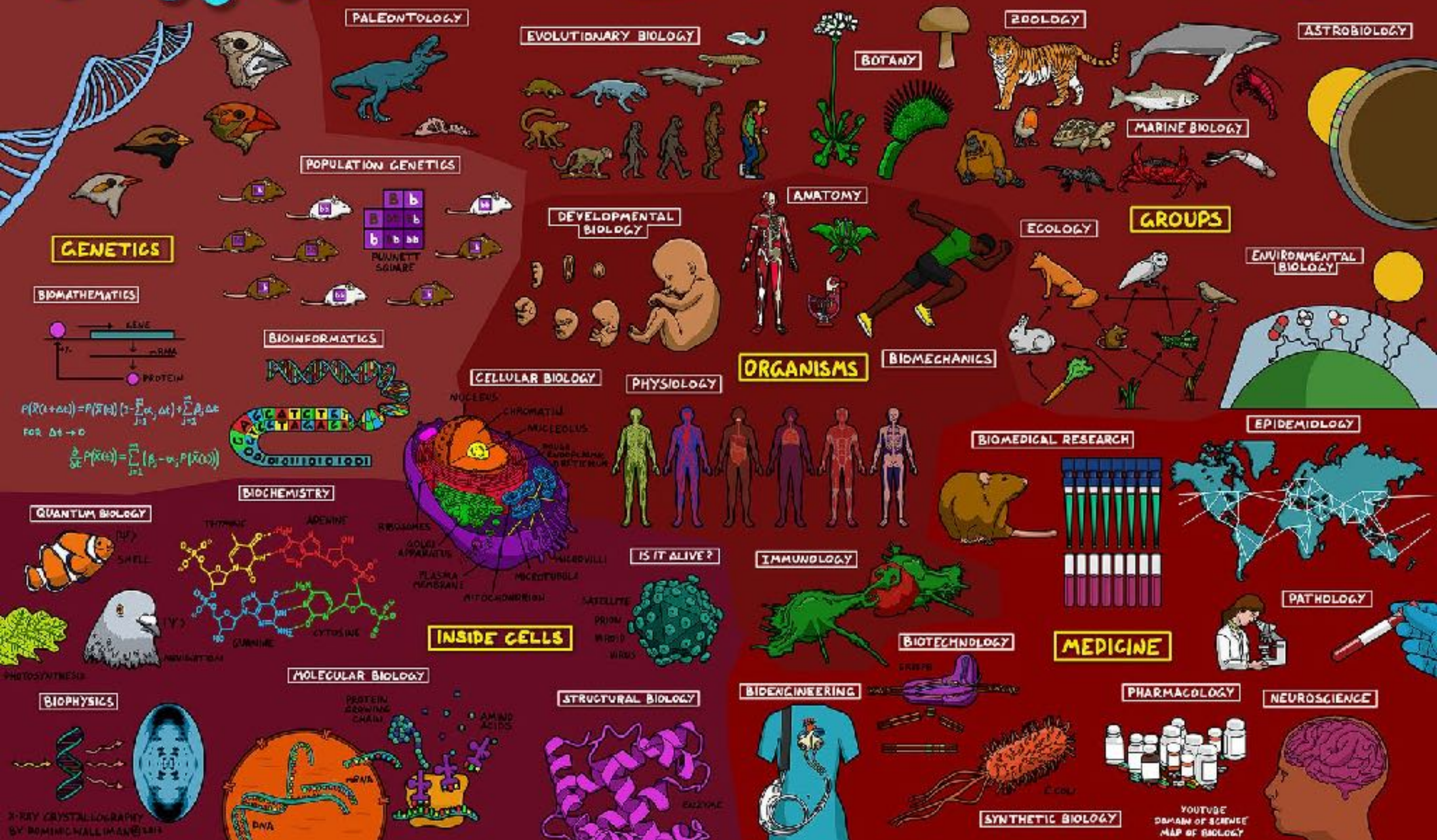
Date	Content
	Basics
9/25	Welcome/overview. Introduction to computational biology.
	Sequence
9/30	Sequence alignment and analysis (Part 1)
10/2	Sequence alignment and analysis (Part 2)
10/7	Protein function prediction (part 1)
10/9	Protein function prediction (part 2)
	Graph (systems biology)
10/14	Introduction to biomedical graph analysis (part 1)
10/16	Research showcase (Wei Qiu, generative AI for aging research)
10/21	Introduction to biomedical graph analysis (part 2)
10/23	Research showcase (Zixuan Liu, generative AI for retinal imaging analysis)
10/28	Biomedical graph diffusion (part 1)
10/30	Research showcase (Hanwen Xu, generative AI for cancer pathology)
11/4	Biomedical graph diffusion (part 2)
11/6	Research showcase (Yongkang Li, generative AI for label-free mouse brain imaging)
11/11	Research showcase (Shengqi Hang, generative AI for chromatin structure analysis)

	Genomics
11/13	Genomics for precision medicine (drug repurposing, part 1)
11/18	Genomics for precision medicine (drug combination, part 2)
11/20	Research showcase (Tangqi Fang, generative AI for genomic data analysis)
11/25	Research showcase (Chanwoo Kim, interpretable AI for pathology data analysis)
12/2	Research showcase (Patricia Xiao, generative AI for cardiovascular disease analysis)
12/4	Research showcase (Sheng Wang, generative AI for Medicine)

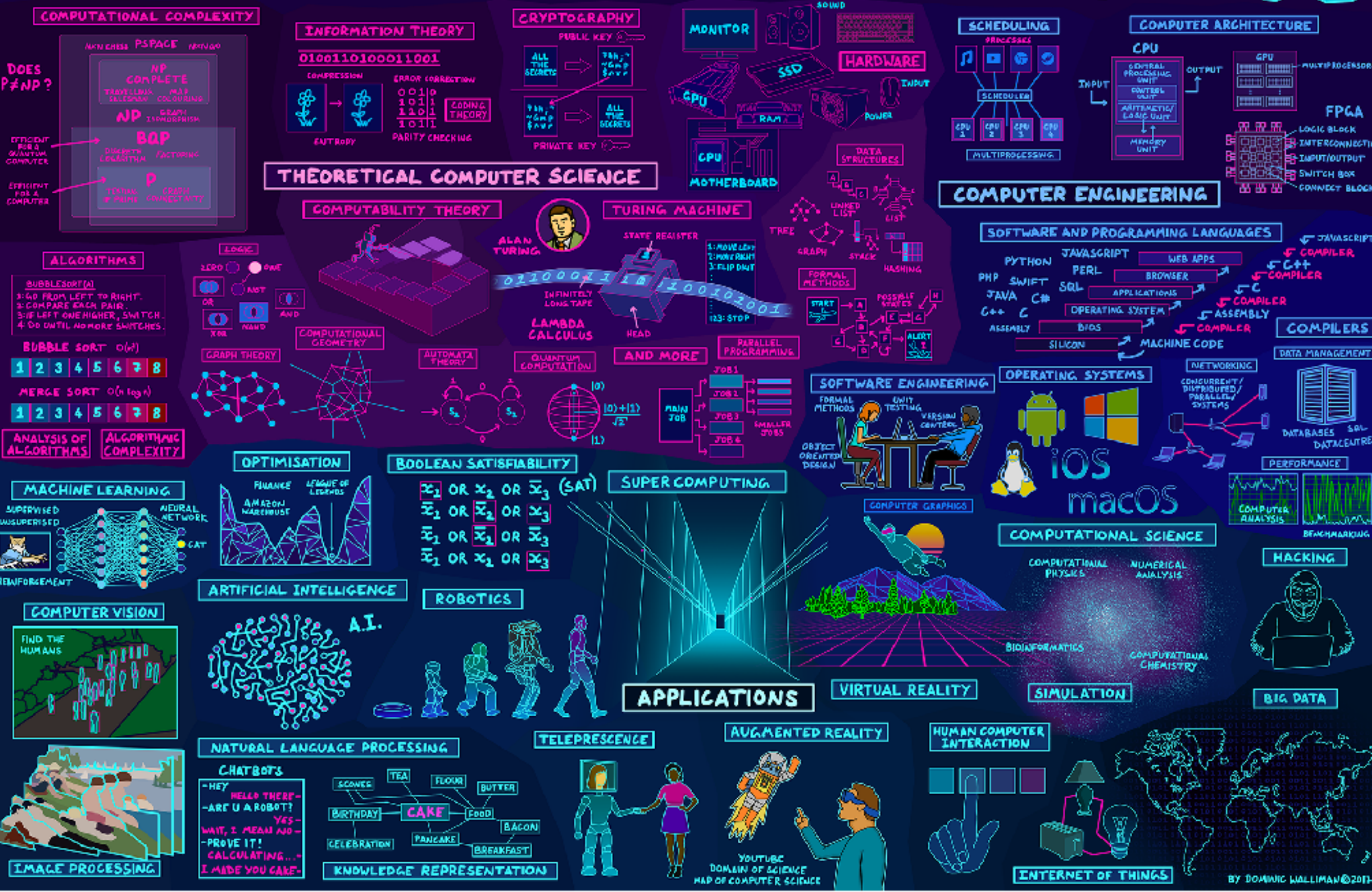
MAP OF BIOLOGY



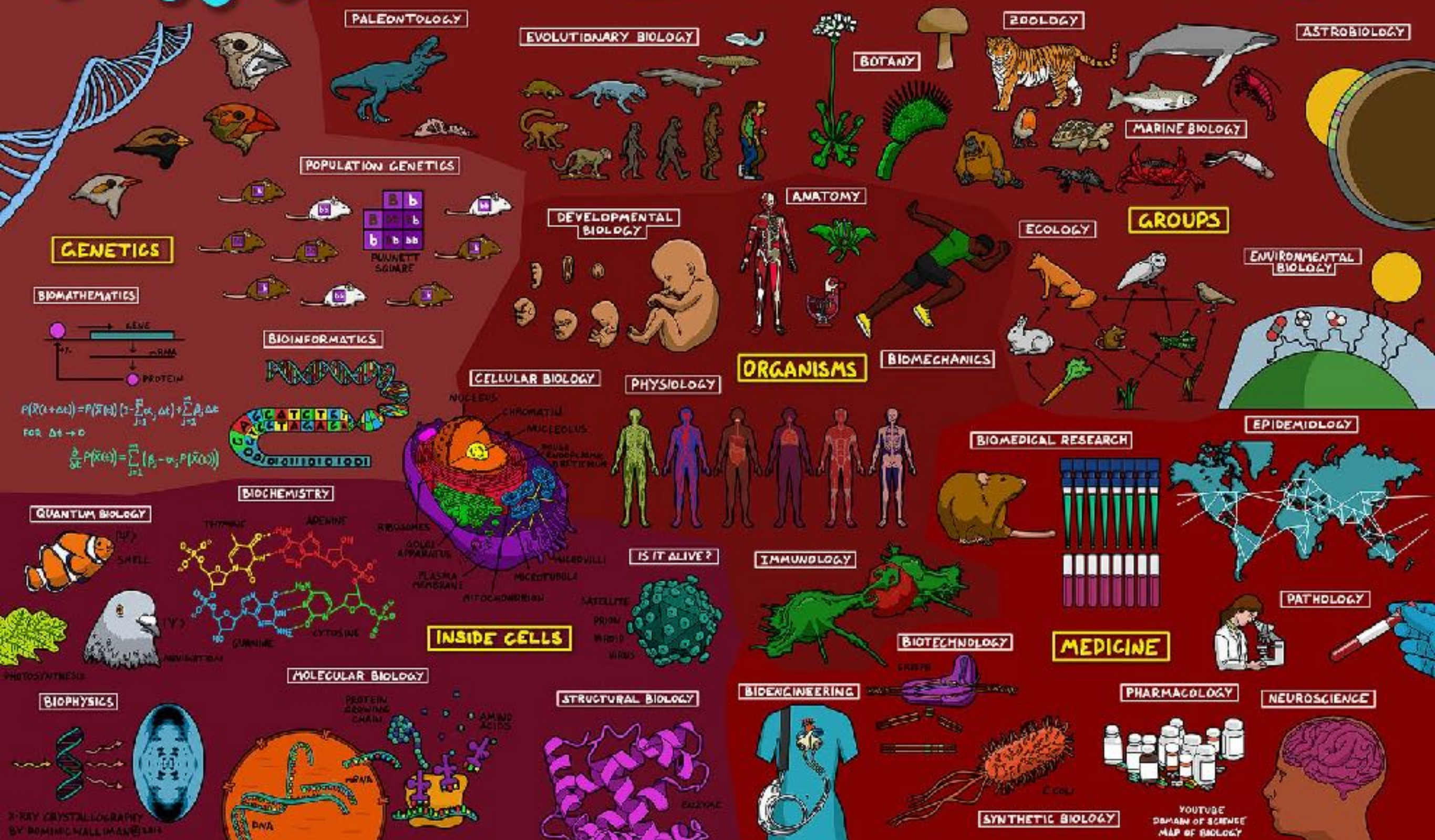
Question: How do we divide biology into subfield?



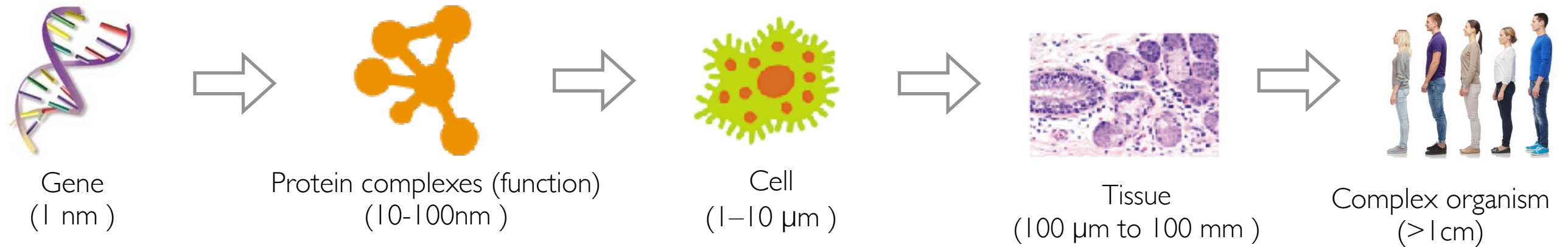
MAP OF COMPUTER SCIENCE



Subfield of biology are divided based on the **scale**



CSE427: Computational methods for biology at different scales

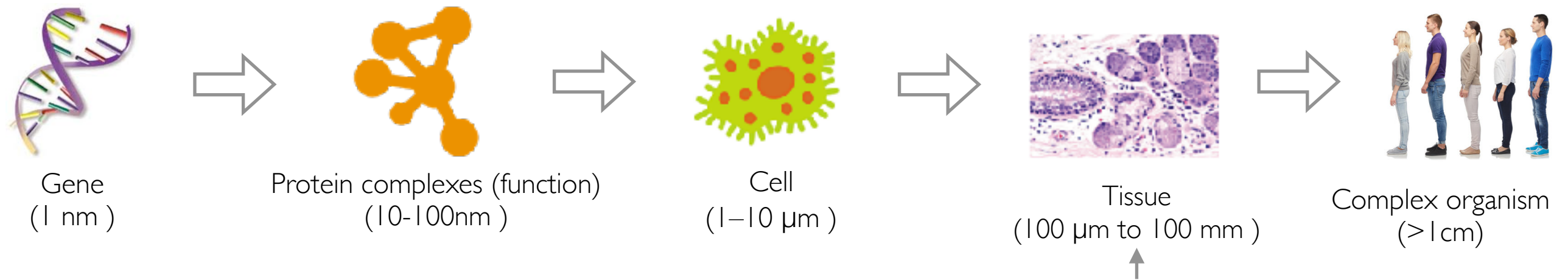


A rich hierarchy of biological subsystems at multiple scales: genotypic variations in nucleotides (1 nm scale) $\rightarrow$ proteins (1-10 nm) $\rightarrow$ protein complexes (10-100 nm), cellular processes (100 nm) $\rightarrow$ phenotypic behaviors of cells (1-10 μm), tissues (100 μm to 100 mm), $\rightarrow$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

How a computer scientist study comp bio?

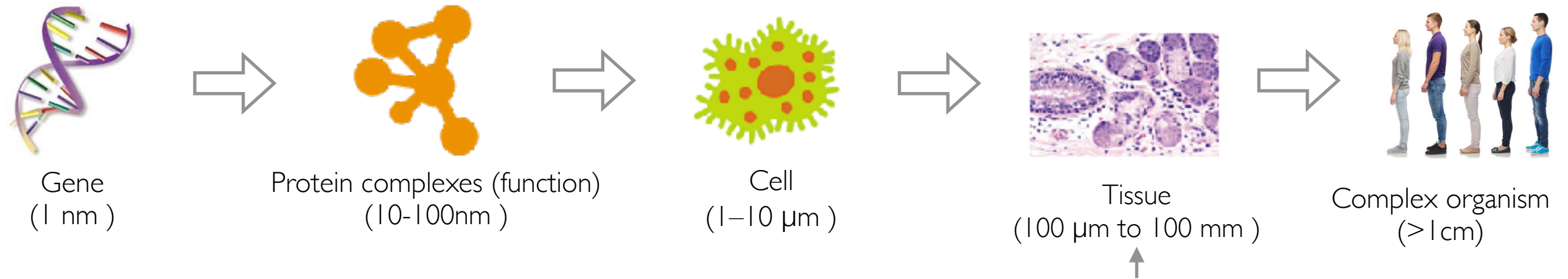
Understand the input and output first



Biologists: which input should I use for this problem? Gene expression? Tissue images?

Computer scientists: Given the input we have, which method should we use to solve this problem?

Disentangle the process of solving comp bio problems



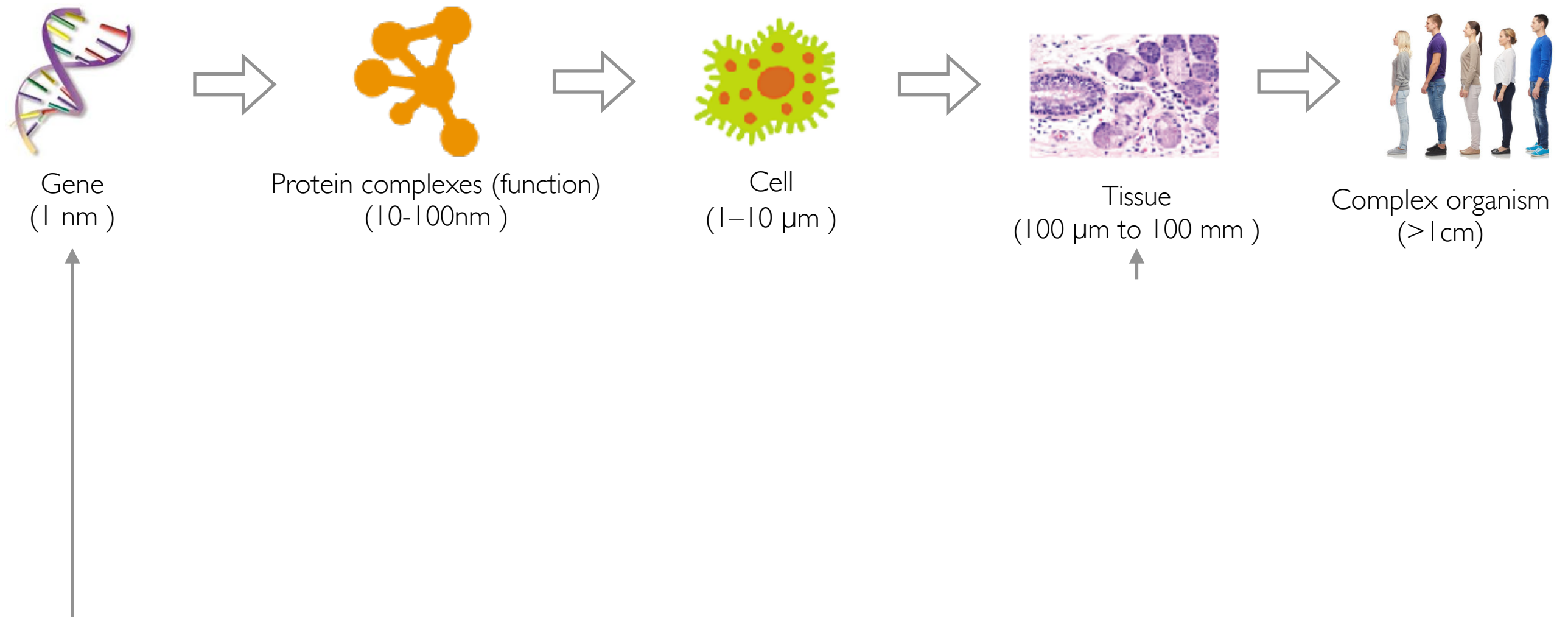
Step 1. Understand the data structure of input and output

Step 2. Develop methods based on the data structure

Step 3. Validate on existing data (cross-validation)

Step 4. Find new biology (literature evidence)

Data structure for each scale: protein

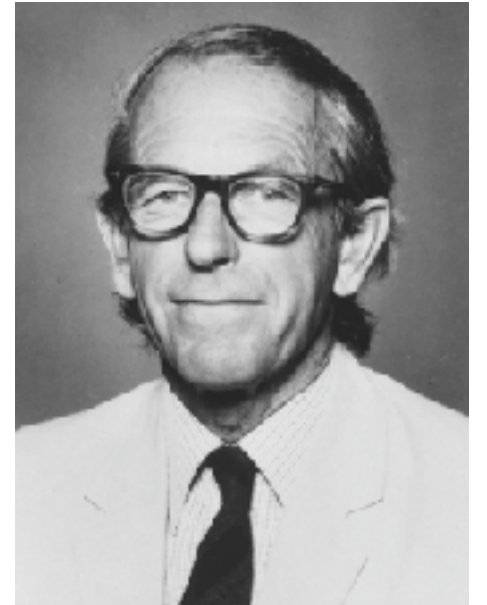


A sequence of amino acids/nucleic acids -> A sequence of word/character
NLP methods (edit distance, LSTM, BERT)

Computational challenge: modeling the order in the sequence

Next generation sequencing (NGS)

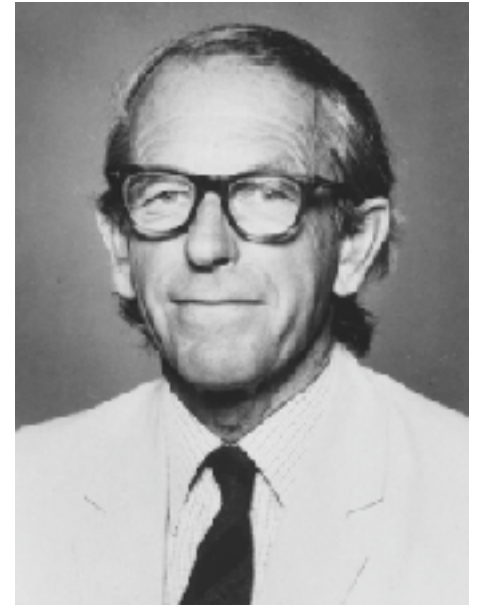
- What is NGS?
 - A fast and cheap experimental technology that can produce the entire DNA sequence of a person within a single day.



Dr. Frederick Sanger
Nobel prize in
Chemistry (1958, 1980)

Next generation sequencing (NGS)

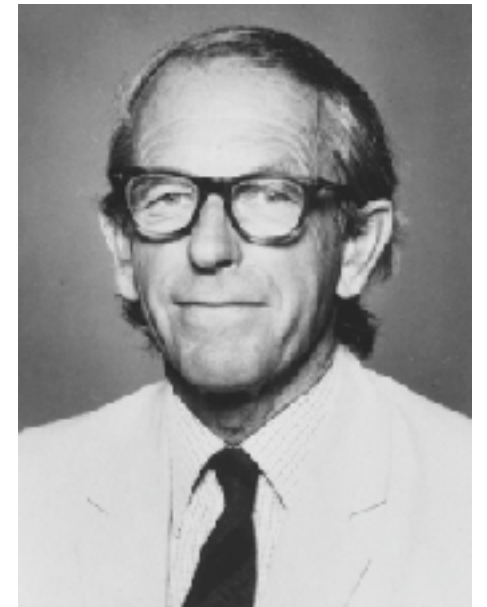
- What is NGS?
 - A fast and cheap experimental technology that can produce the entire DNA sequence of a person within a single day.
 - Good to know the technique details, but the algorithm are more important for CS people.
 - In human, DNA is a **3 billion-long string of As, Cs, Gs and Ts**



Dr. Frederick Sanger
Nobel prize in
Chemistry (1958, 1980)

Next generation sequencing (NGS)

- What is NGS?
 - A fast and cheap experimental technology that can produce the entire DNA sequence of a person within a single day.
 - Good to know the technique details, but the algorithm are more important for CS people.
 - In human, DNA is a **3 billion-long string of As, Cs, Gs and Ts**
- **Important Question:**
 - what algorithms should we develop for DNA sequence? (this technique emerged in 1994 and became commercially available since 2005)
 - Storage? Privacy? Compression?



Dr. Frederick Sanger
Nobel prize in
Chemistry (1958, 1980)

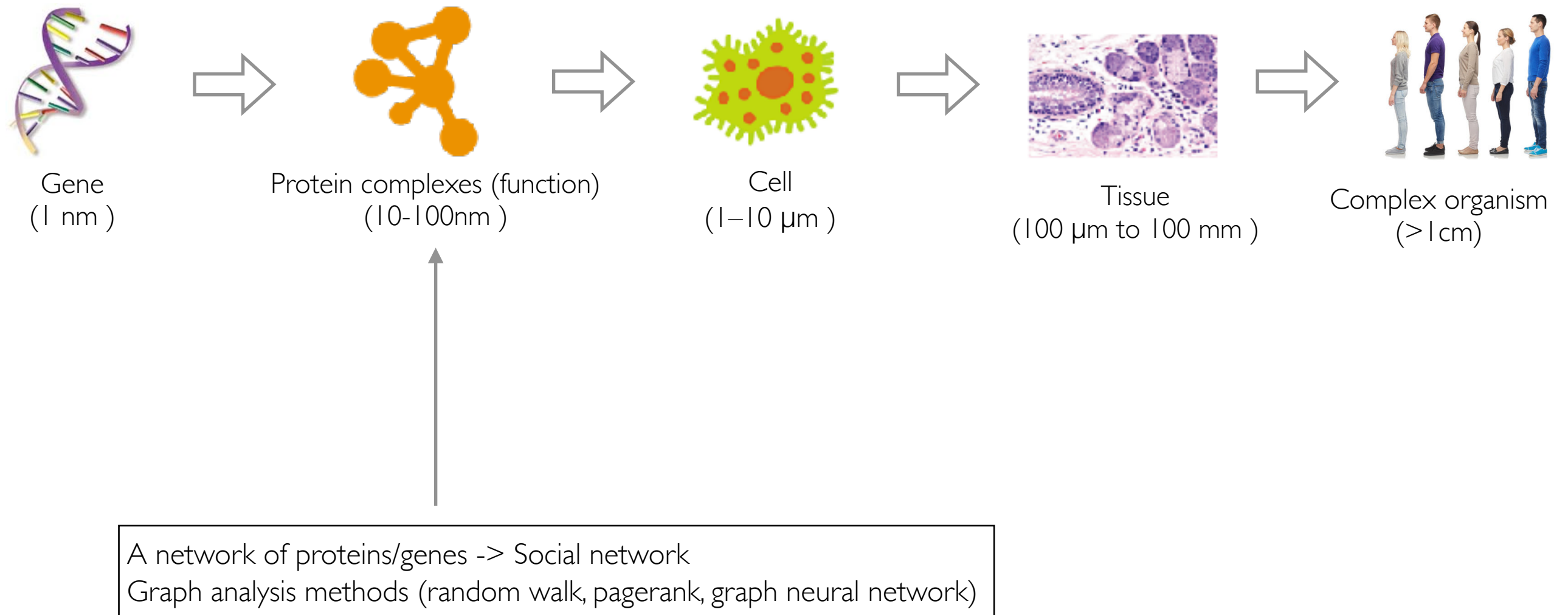
Common comp bio question: measure the similarity between two samples

- Measure the similarity between two DNA sequences (or two patients)
- Always think about it from two perspectives:
 - Algorithmic perspective: string match, Knuth-Morris-Pratt KMP String Matching Algorithm
 - Machine learning perspective: LSTM, RNN, CNN, Language model

Principle for computer scientists to work on a biomedicine problem

- Step 1. Understand the data structures of input and output
- Step 2. Find similar problem in algorithm and ML classes
 - Text string match -> DNA string match
- Step 3. Transfer that method to biology
- Step 4 (optional, PhD student research). Improve that method based on the unique property in bio data
 - Text strings are often short (a sentence only has ~20 words) and have clear structures (word, phrase, sentence, paragraph)
 - How to segment DNA sequence? DNA sequences are very long.
- Step 5 (optional. Suggest future research direction to biologists)
 - Ask the biologist. Can you segment the DNA sequence using some experimental techniques? If so, I have more powerful methods to analyze them.

Data structure for each scale: network



Computational challenge: interaction, synergistic effect

Yeast two-hybrid (Y2H)

- What is Y2H?
 - A molecular biology technique that can discovery protein-protein interactions (PPIs) and protein-DNA interactions.
- What is PPI?
 - A graph. Each node is a protein (about 20K nodes in human). Each edge is an interaction between two proteins.

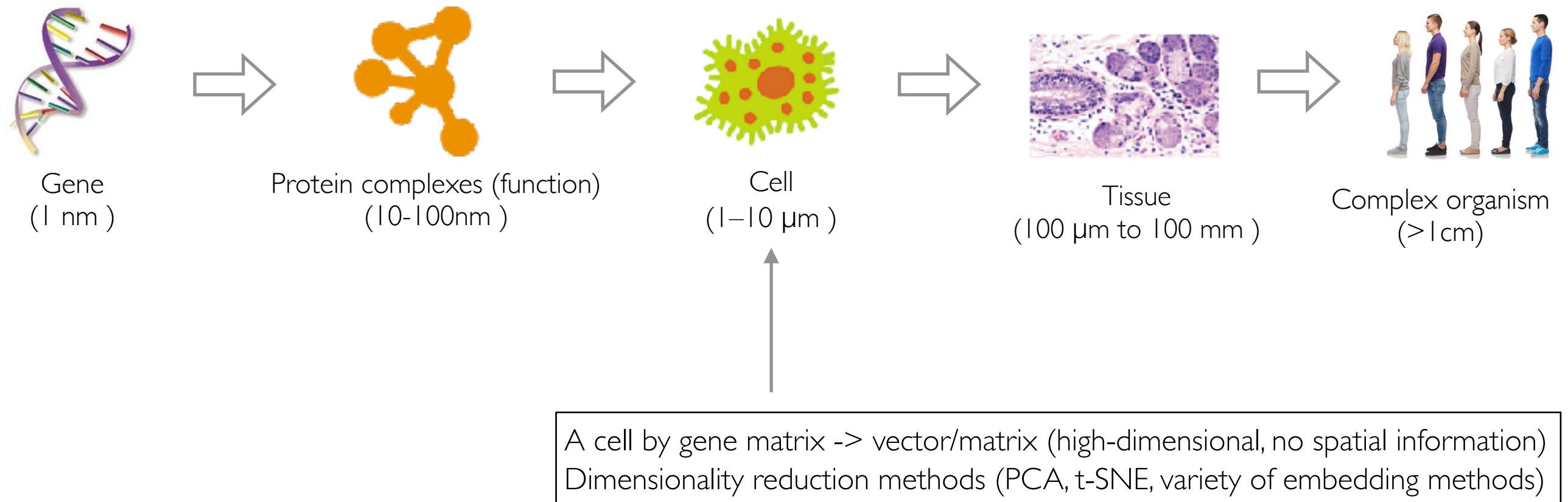
Yeast two-hybrid (Y2H)

- What is Y2H?
 - A molecular biology technique that can discovery protein-protein interactions (PPIs) and protein-DNA interactions.
- What is PPI?
 - A graph. Each node is a protein (about 20K nodes in human). Each edge is an interaction between two proteins.
- Analogy in other applications?
 - Facebook social network. Each user is a protein. User-user friendship relationship is an interaction between two proteins.
- **Important Question:**
 - what algorithms should we develop for Y2H and PPIs?
 - One interesting question in almost any bio subdomains.
 - How to measure the similarity?

What computational questions should we work on for Y2H and PPIs?

- Measure similarity between two proteins in the network
- Measure similarity between two users in the facebook
- Always think about it from two perspectives:
 - Algorithmic perspective: shortest distance (Dijkstra's algorithm)
 - Machine learning perspective: random walk, random walk with restart, graph neural network, graph embedding

Data structure for each scale: cell



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

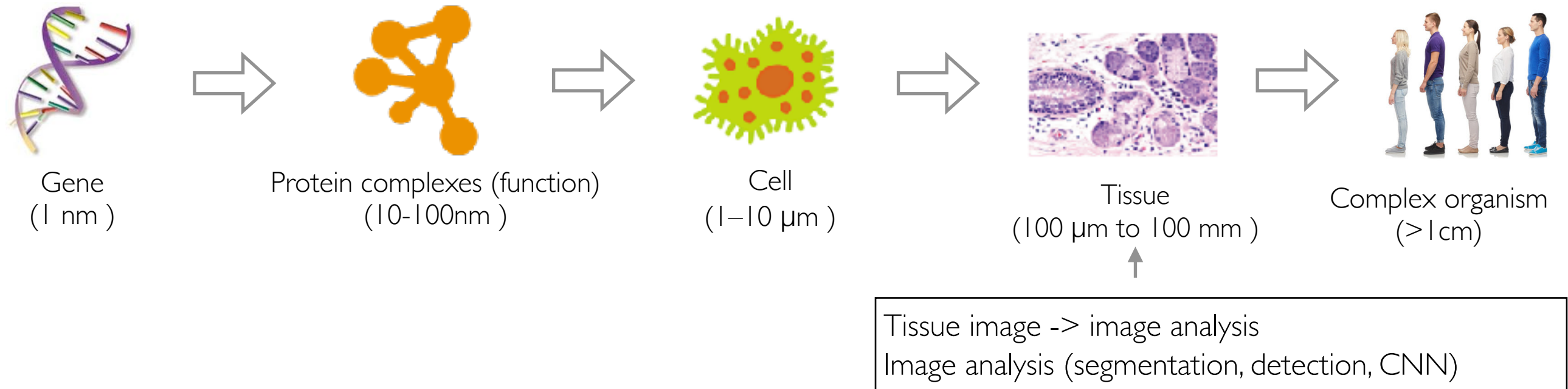
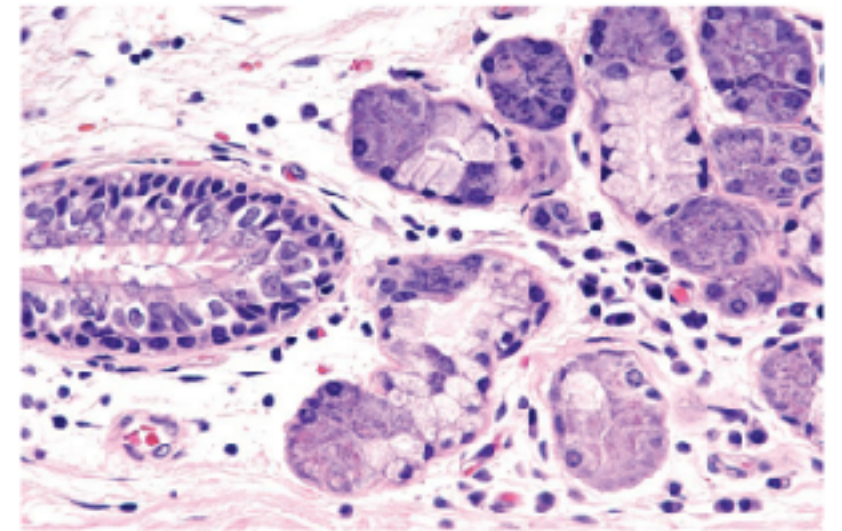


Image analysis, lack of high-quality annotations

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)

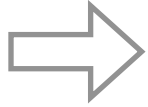


Tumor tissue image

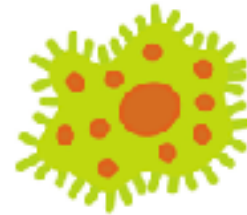
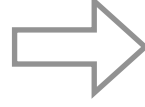
Data structure for each scale: organism



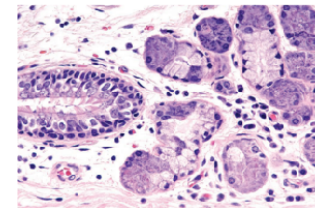
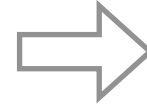
Gene
(1 nm)



Protein complexes (function)
(10-100nm)



Cell
(1-10 μ m)



Tissue
(100 μ m to 100 mm)



Complex organism
(> 1cm)



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

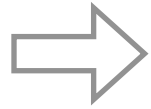
Multi-modality and heterogeneous

Computational methods for biology at different scales



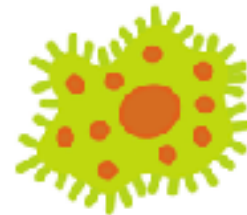
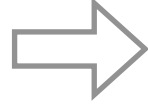
Gene
(1 nm)

Genetics



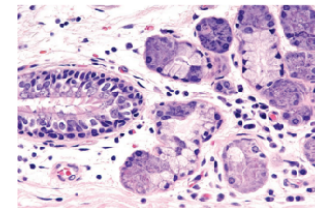
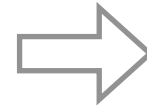
Protein complexes (function)
(10-100nm)

Systems biology



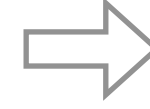
Cell
(1-10 μ m)

Cellular biology



Tissue
(100 μ m to 100 mm)

Medical imaging



Complex organism
(> 1cm)

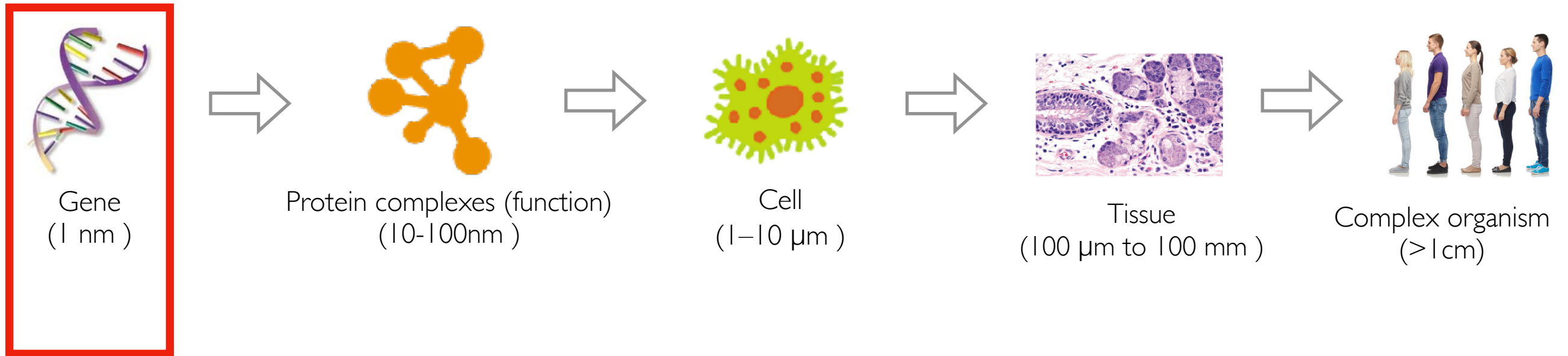
Computational medicine

Focus of CSE 427

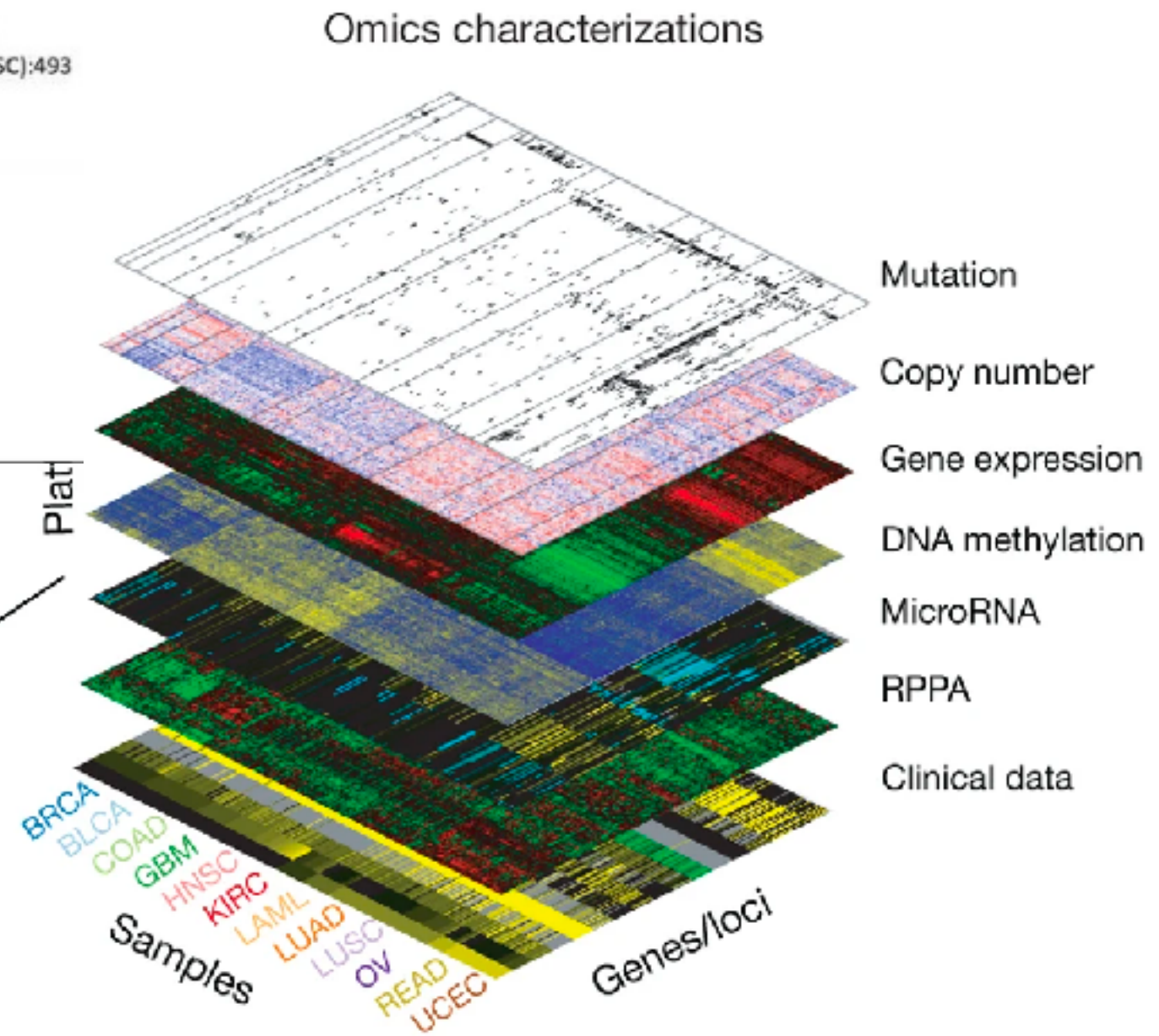
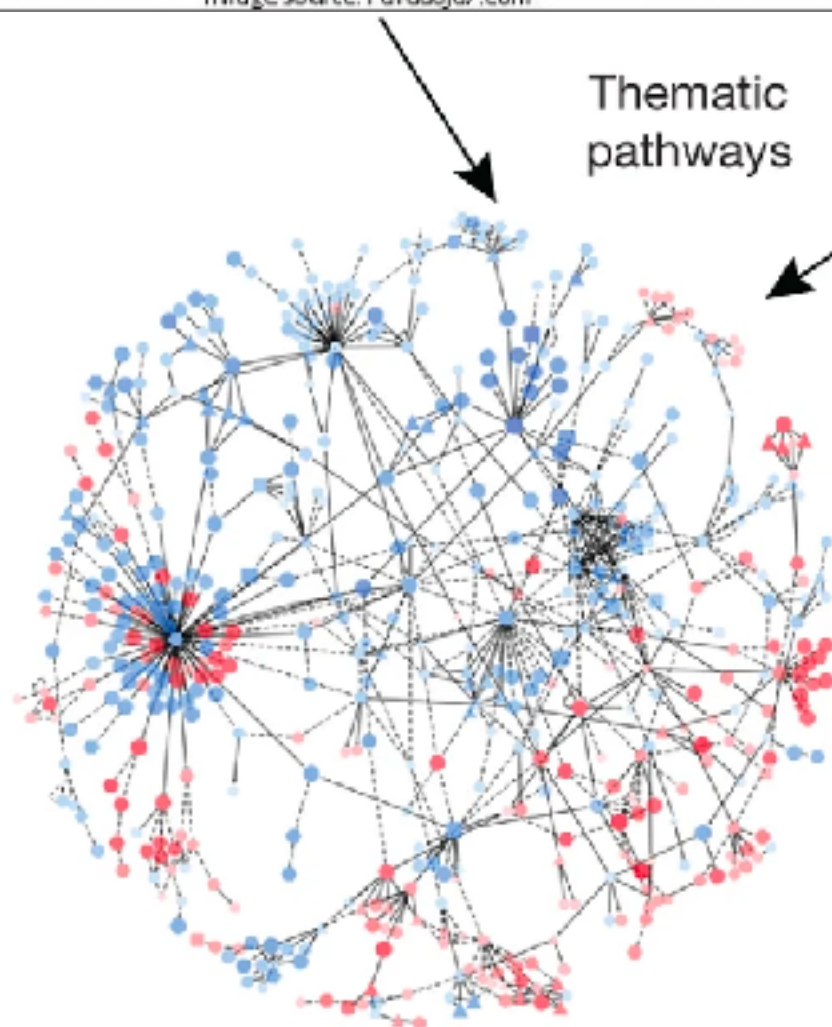
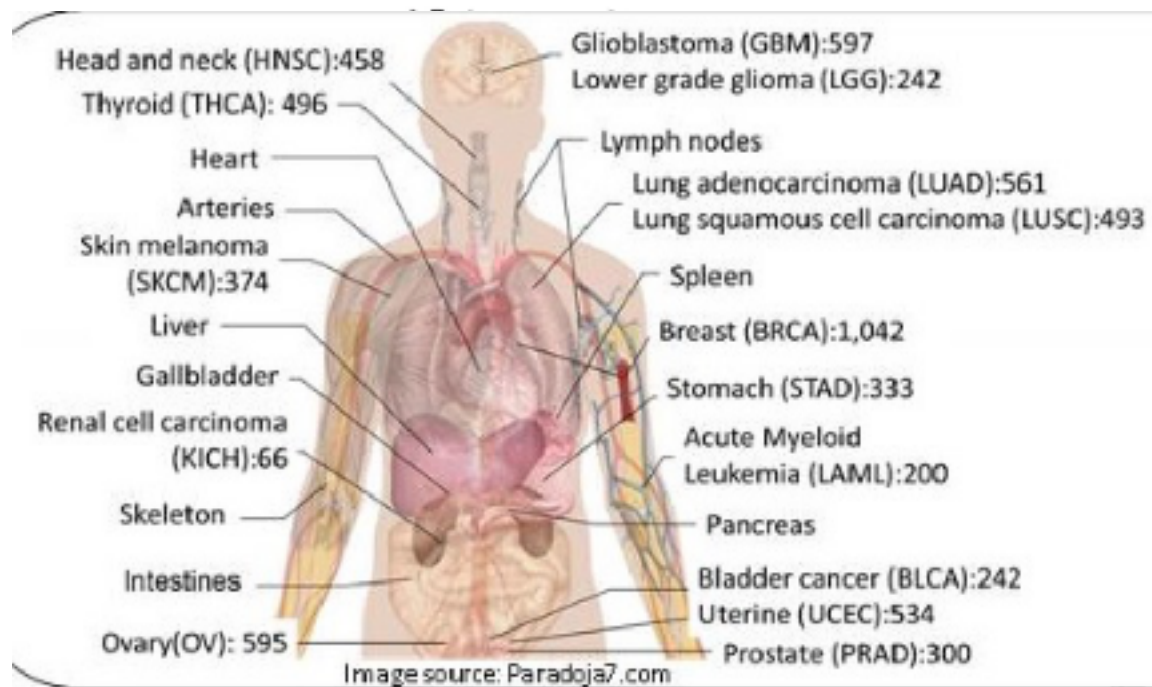
Real world research question: how to measure the similarity between two patients

- 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



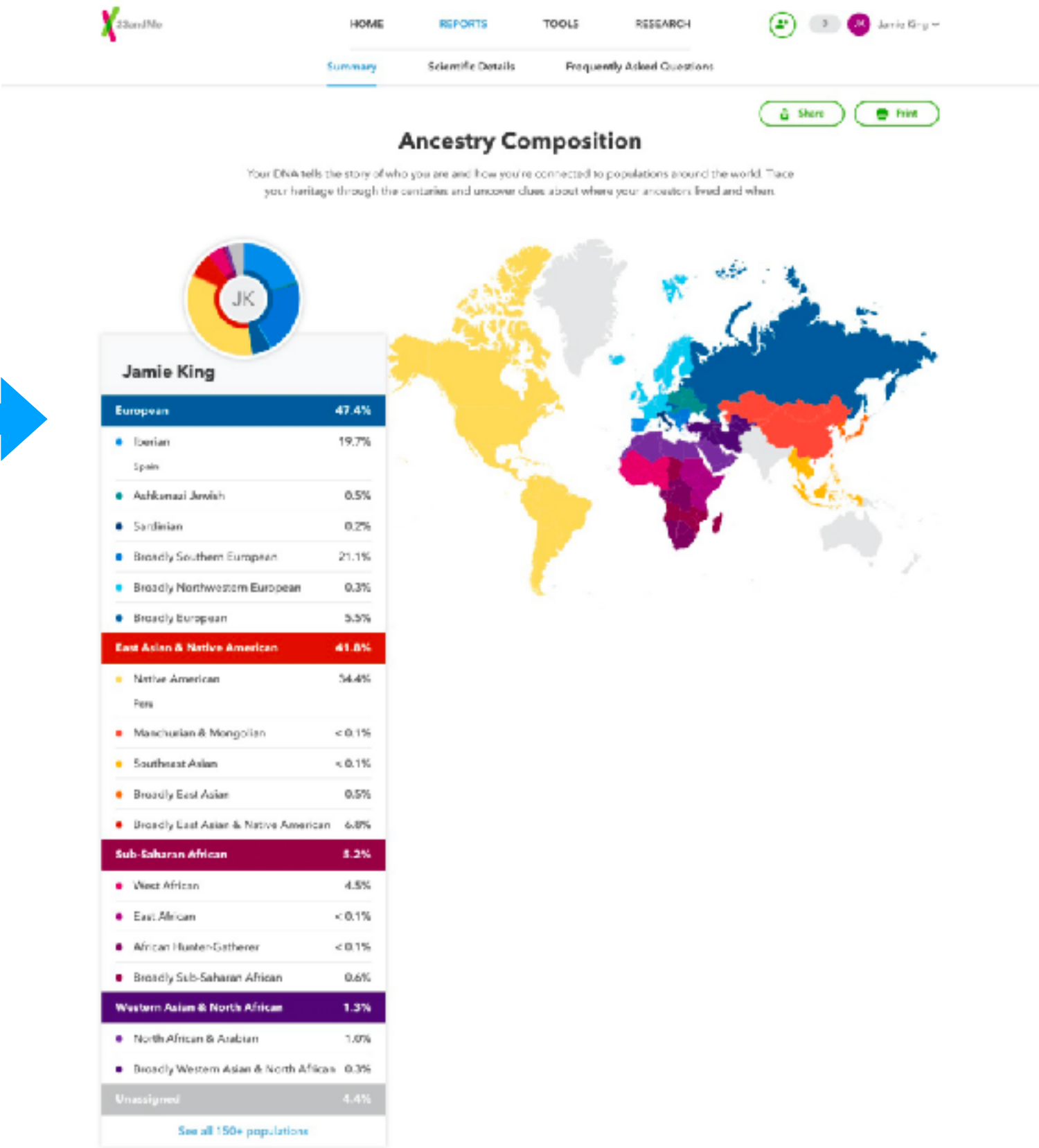
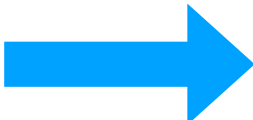
A concrete example: The Cancer Genome Atlas Program



DNA sample analysis by 23andMe



DNA sample



How did they do this?



DNA sample



Sequencing machine
~2000 dollars

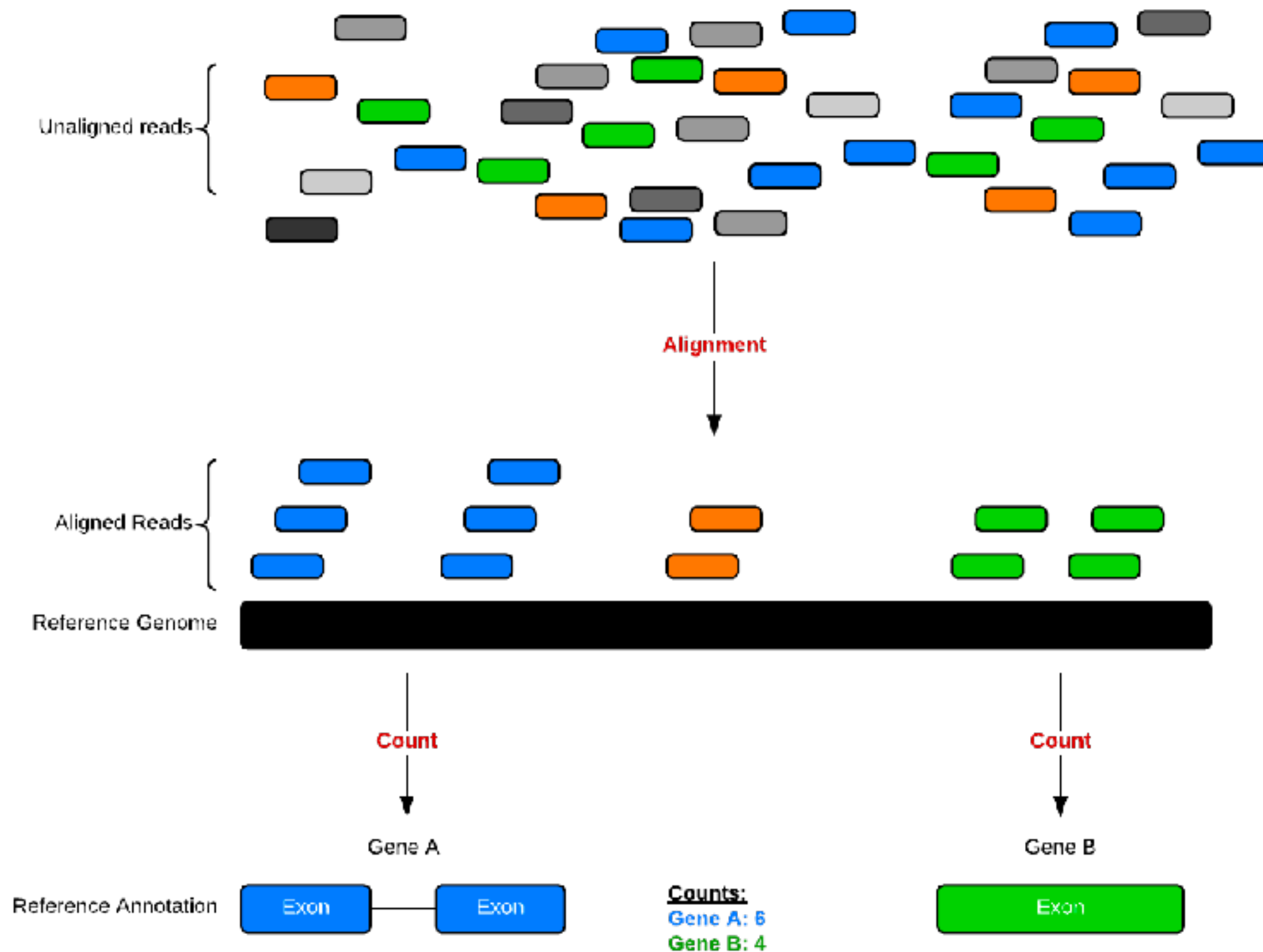


Name	Size
26455-P_2.fastq	25.84 GB

Your entire genome sequence
*.fastq file

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

Process raw data using sequence alignment (dynamic programming)



What does a fastq file look like?

```
1 @ERR000589.41 EAS139_45:5:1:2:111/1
2 CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
3 +
4 3IIIIIIIIIIII>1IIIF9BG08E00I%IG+&?(4)%00646.C1#&(
5 @ERR000589.42 EAS139_45:5:1:2:1293/1
6 AGTTGTTAAAATCCAAGCCAATTAAGATAGTCTTATCTTTTAAAAGAAAT
7 +
8 IIIIIGII.AIIII=?I9G-/II=+I=4?761BA2C9I+5A711+&>1$/I
```

Very large! ~3000000000 lines

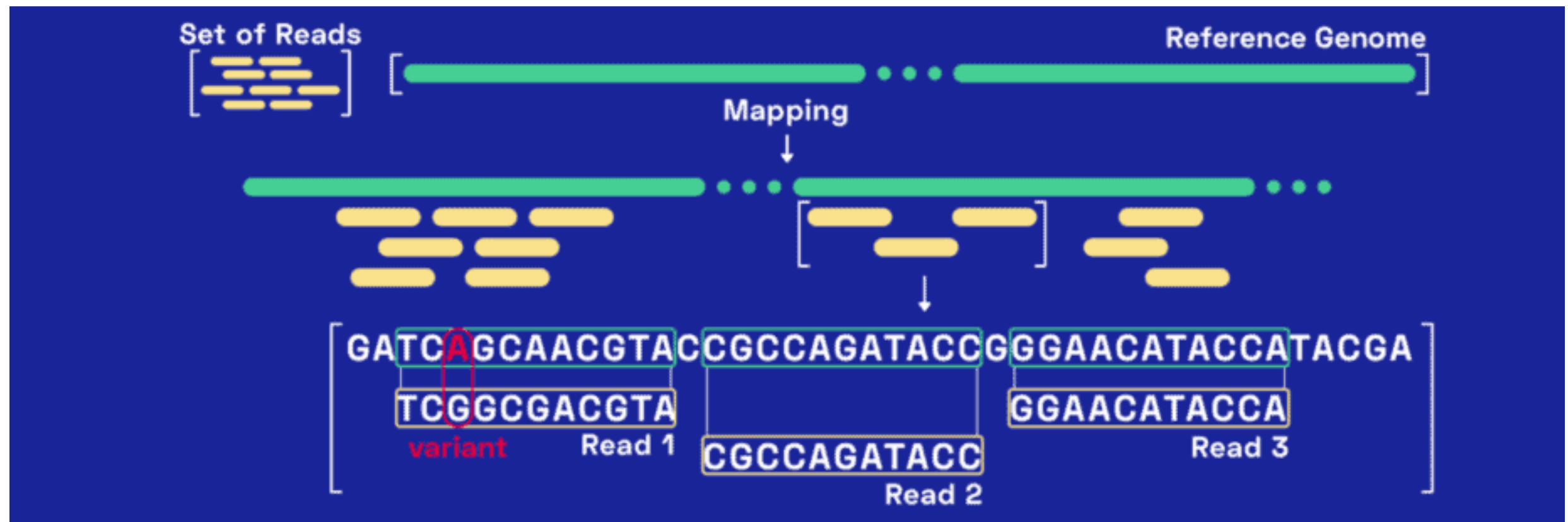
Quality: ASCII chars

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



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

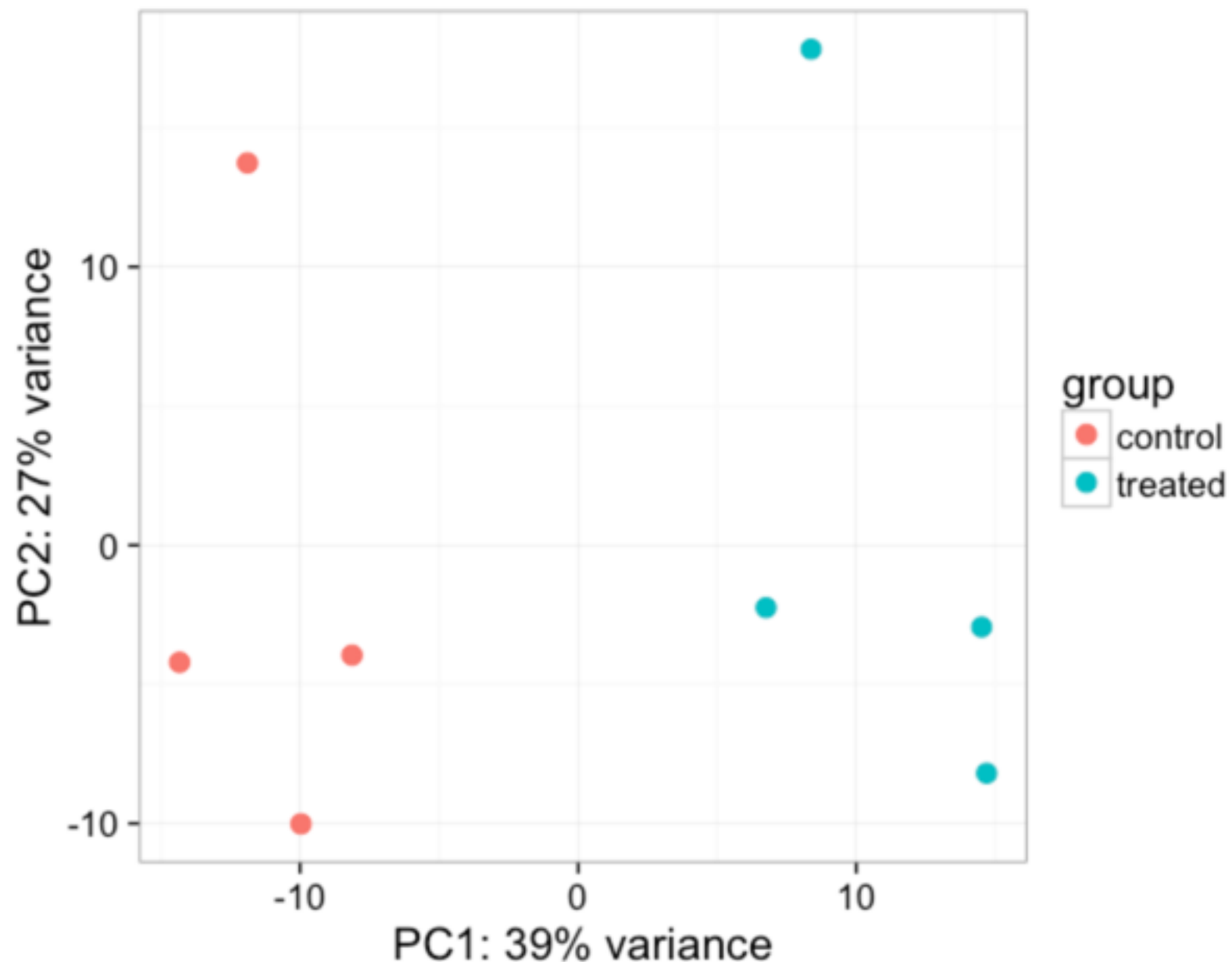
colData

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

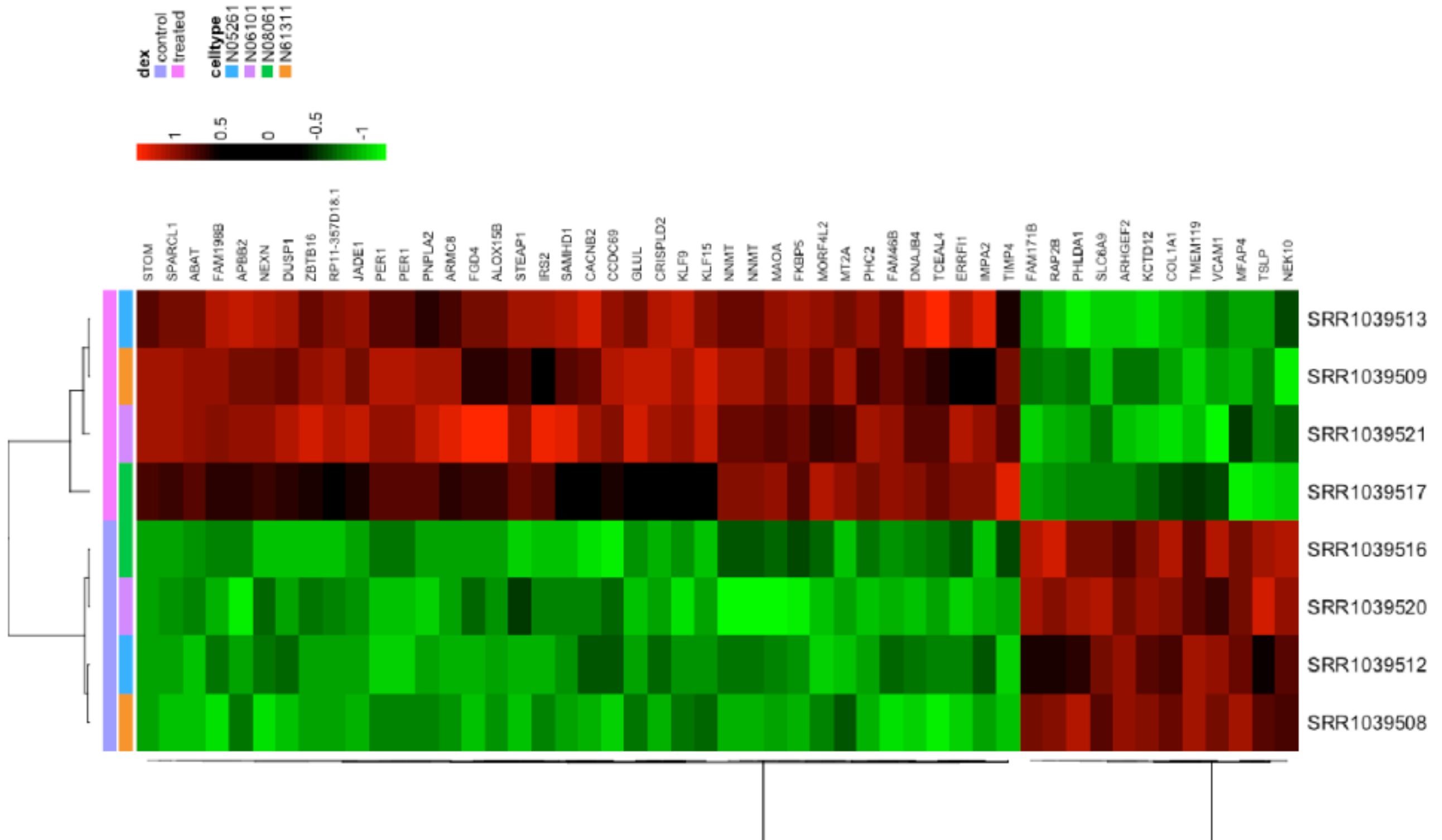
Sample names:

ctrl_1, ctrl_2, exp_1, exp_2

Clustering analysis using dimensionality reduction



Heatmap for visualization

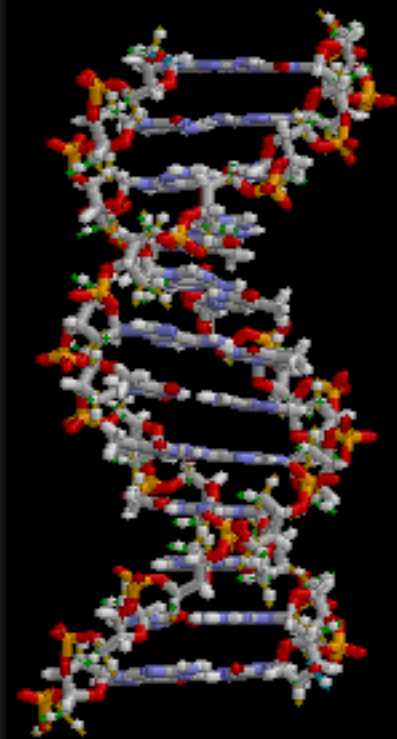


Each individual has a slightly different version of the DNA sequence



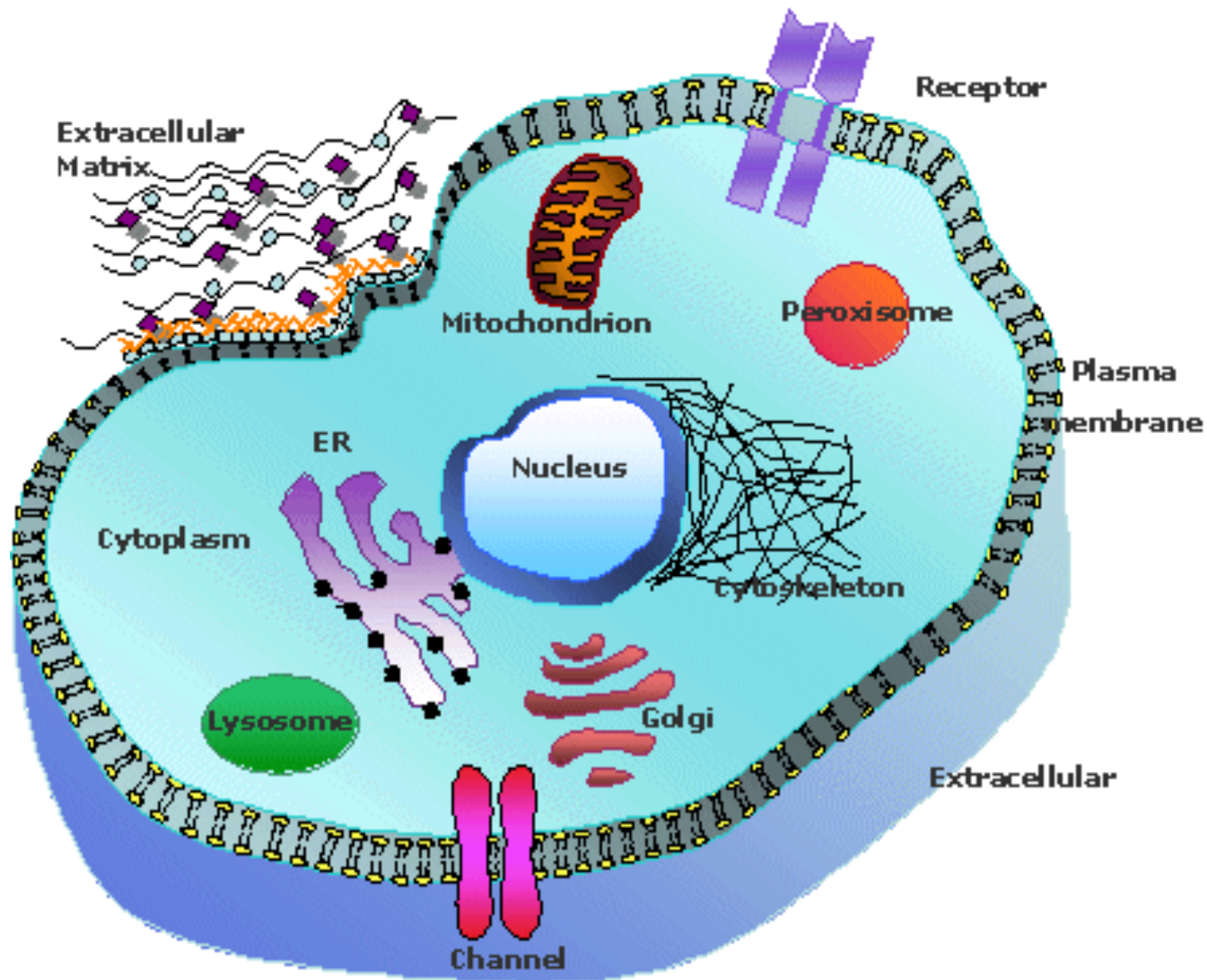
DNA: “Blueprints” for a cell

- Genetic information encoded in long strings of double-stranded DNA (Deoxyribo Nucleic Acid)
- DNA comes in only four flavors: **A**denine, **C**ytosine, **G**uanine, **T**hymine
 - 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



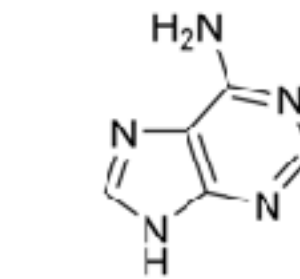
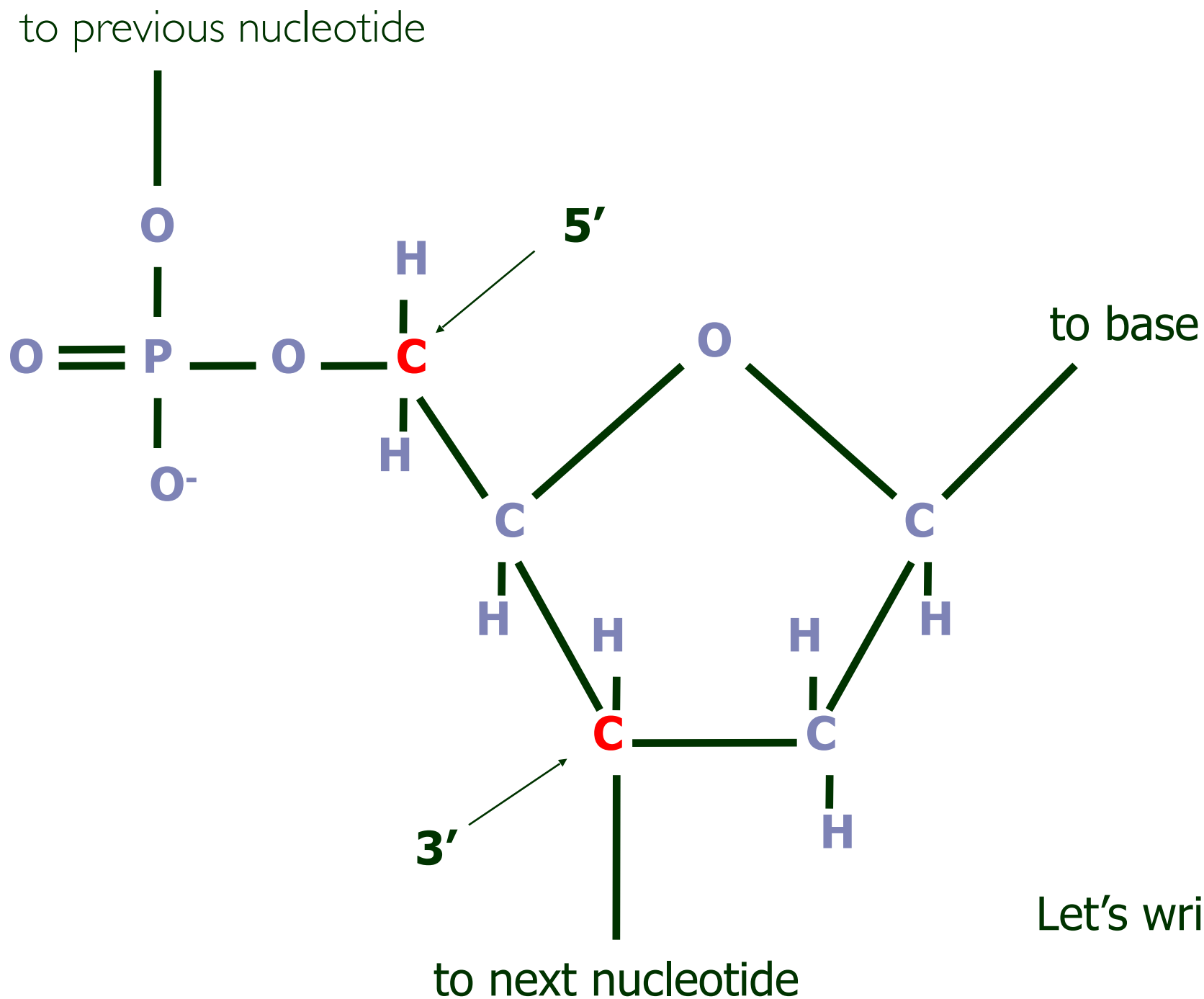
Cell

Cell, nucleus, cytoplasm, mitochondrion

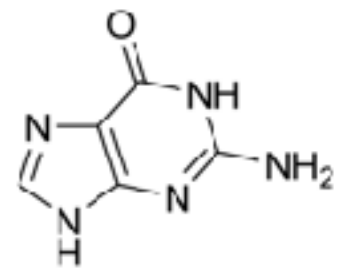


Nucleotide

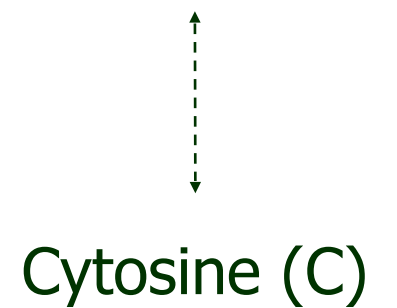
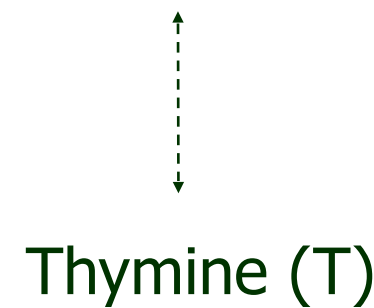
Nucleotide, base, A, C, G, T, 3', 5'



Adenine (A)

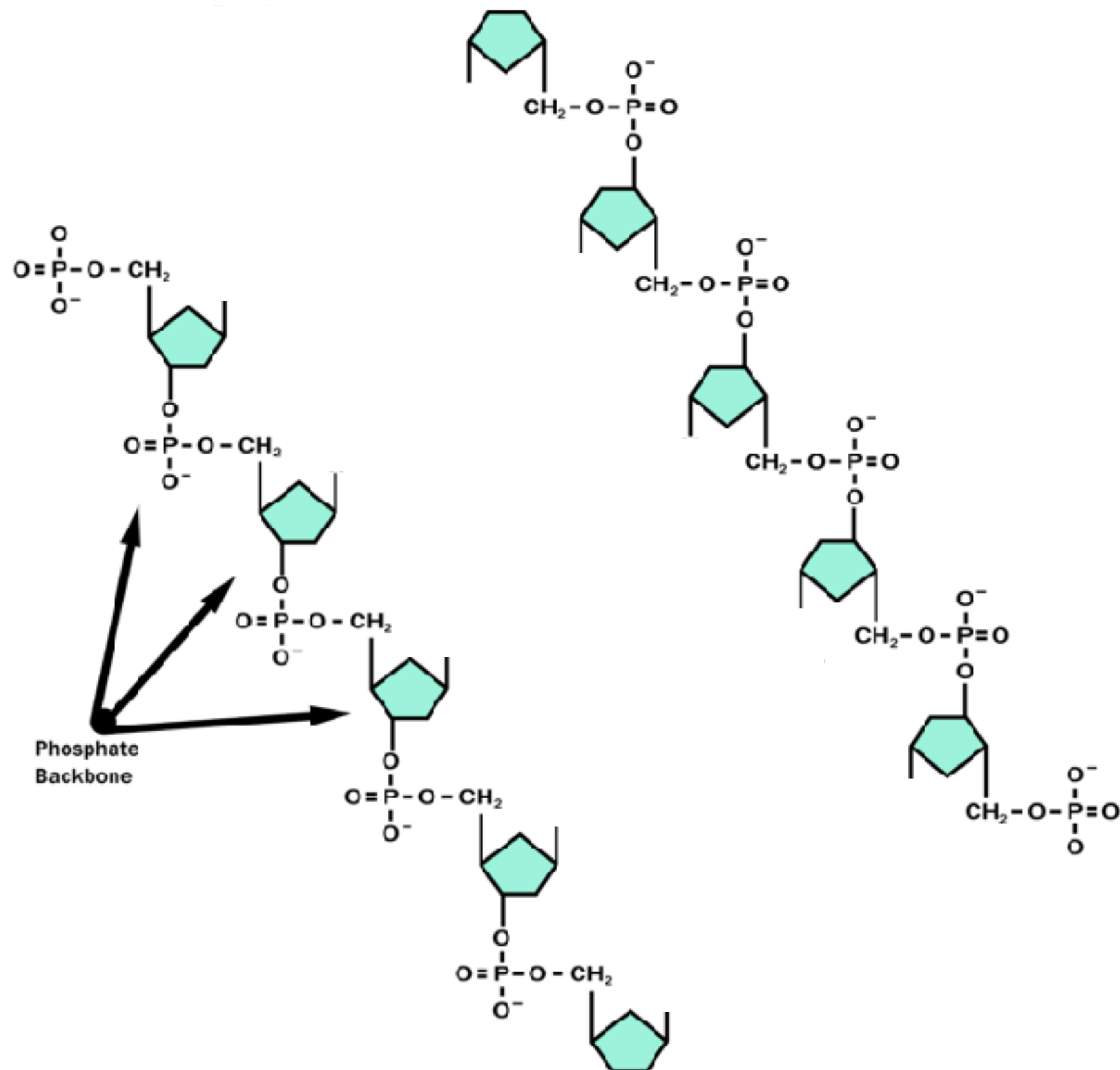


Guanine (G)

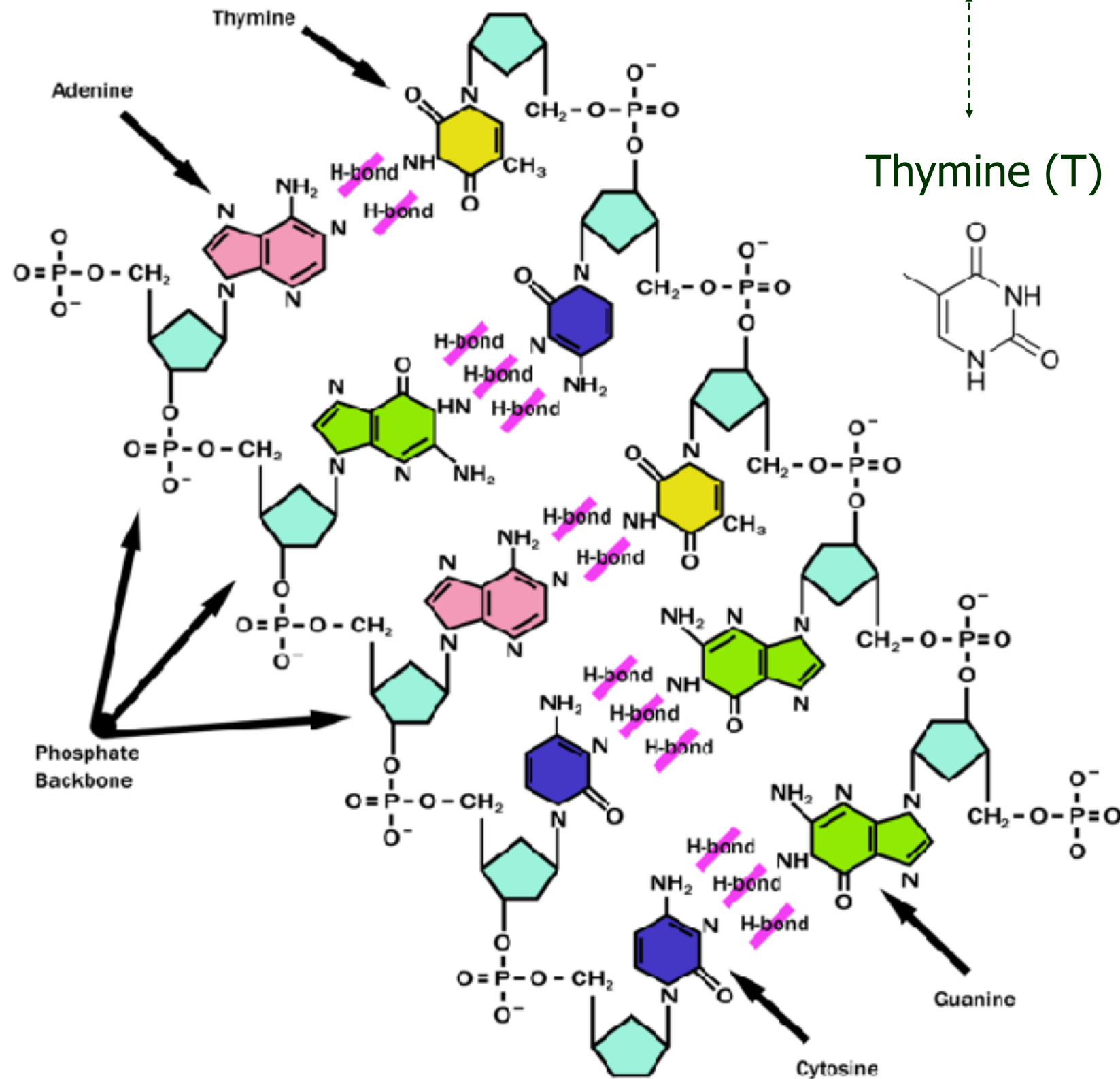


Let's write "AGACC"!

“AGACC” (backbone)



“AGACC” (DNA)

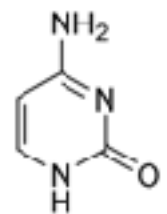
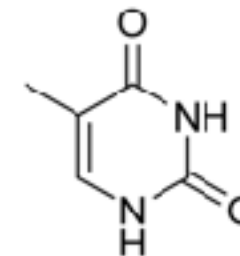
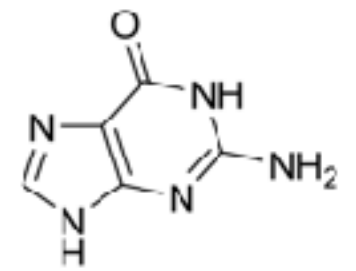
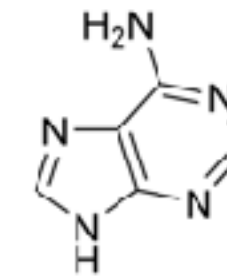


Adenine (A)

Guanine (G)

Thymine (T)

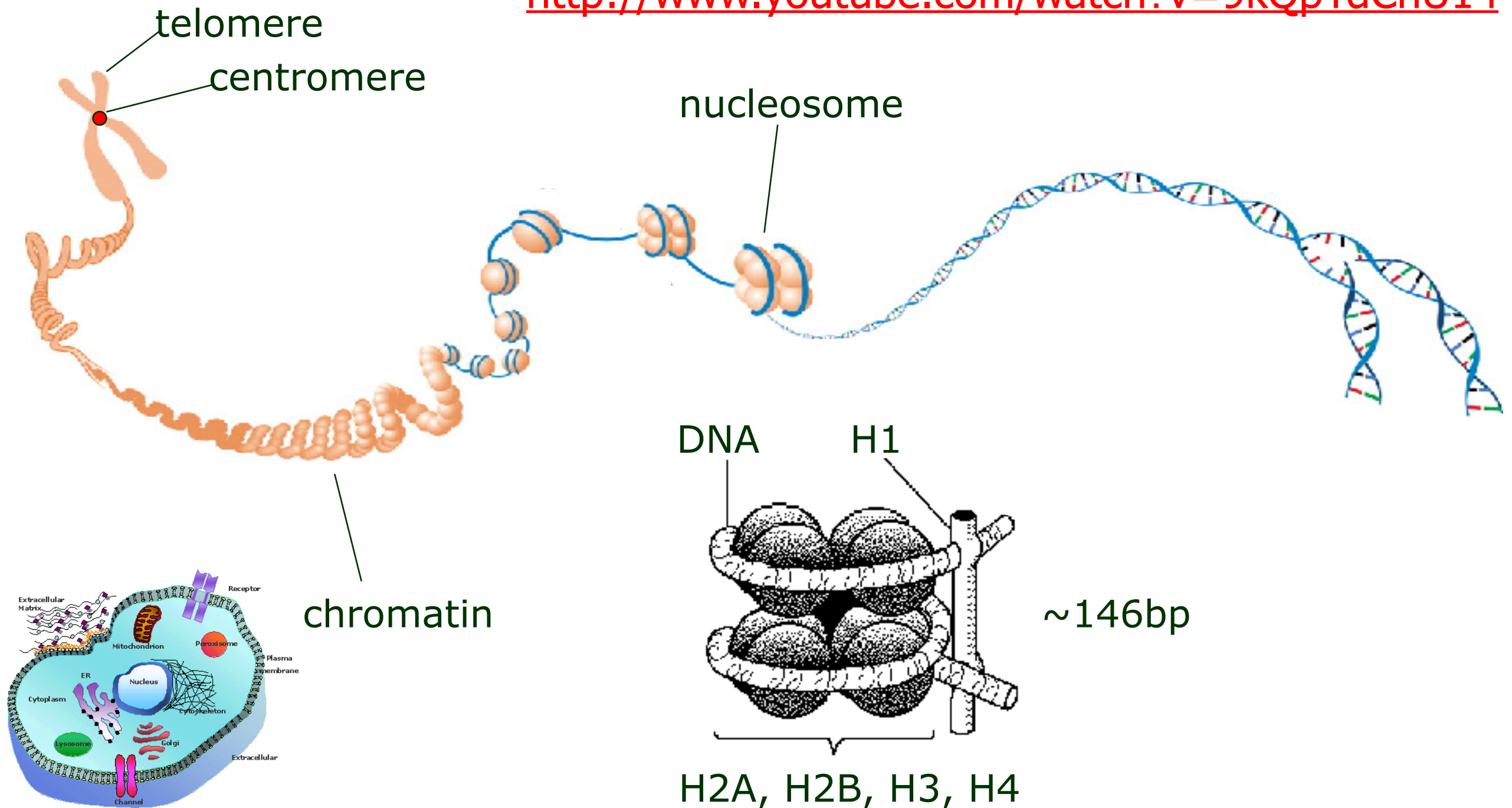
Cytosine (C)



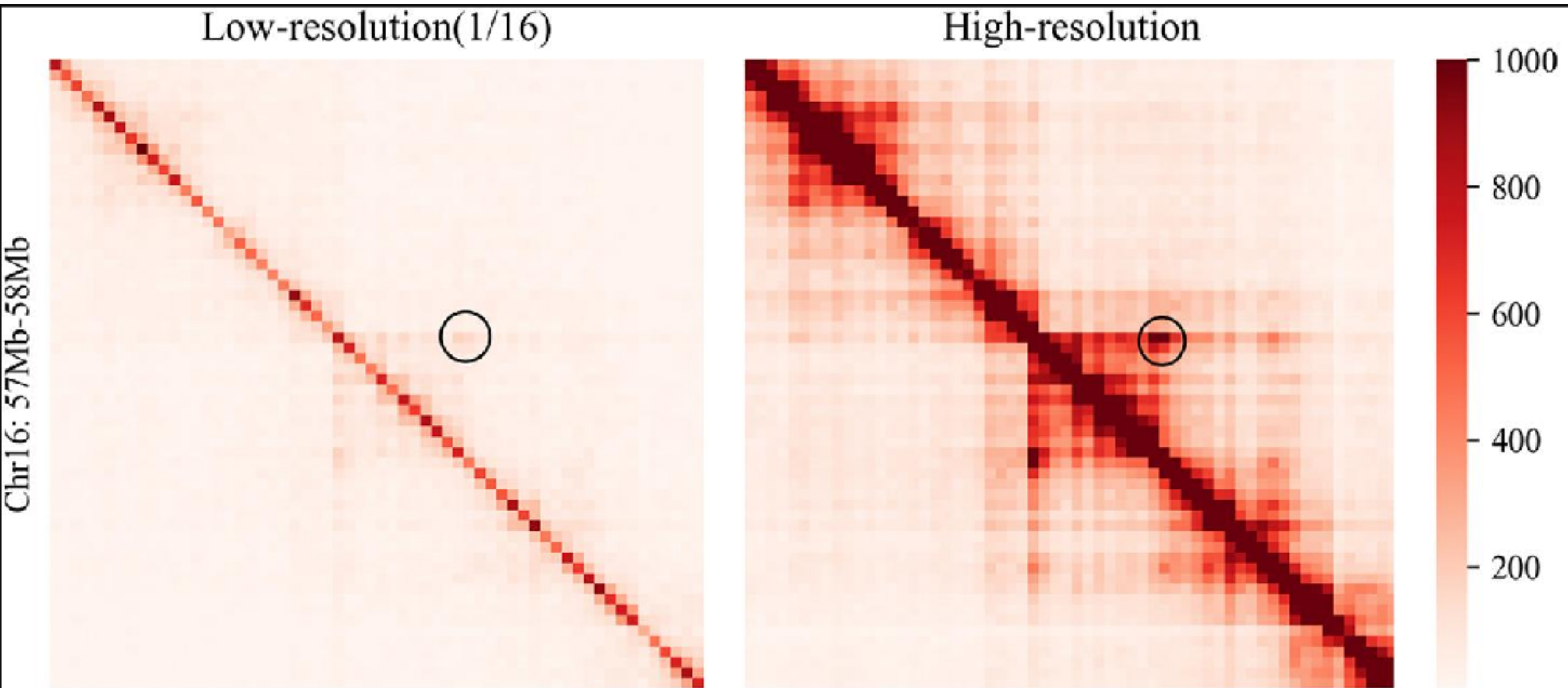
DNA packaging (DNA is 6 feet long!)

Histone, nucleosome, chromatin, chromosome, centromere, telomere

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



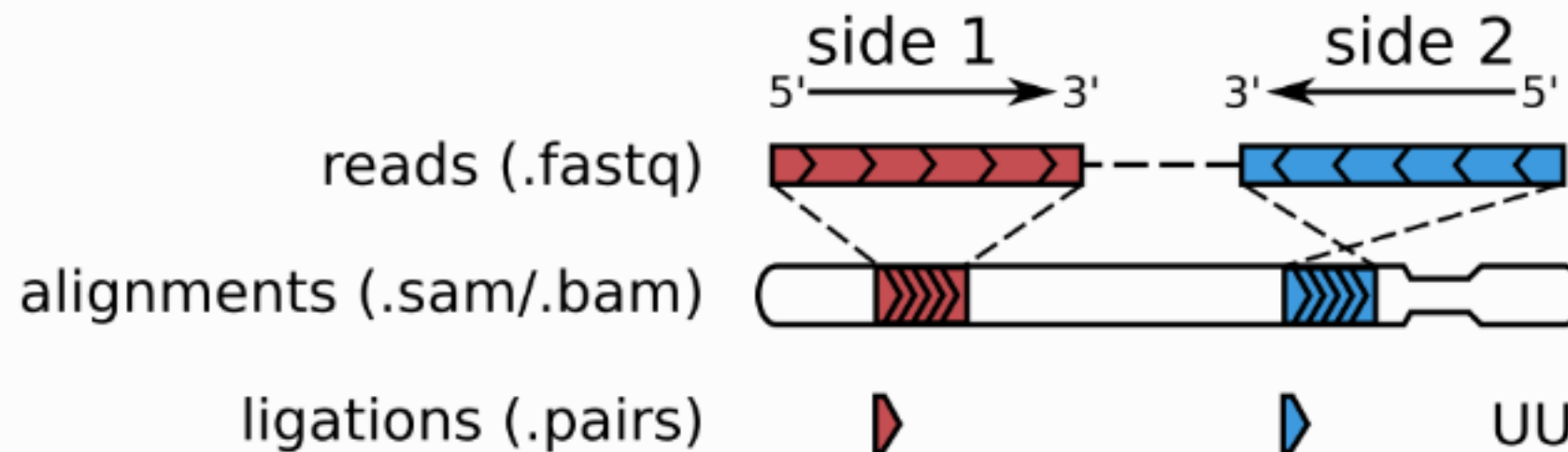
Data structure and computational problem



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

What will the data look like?

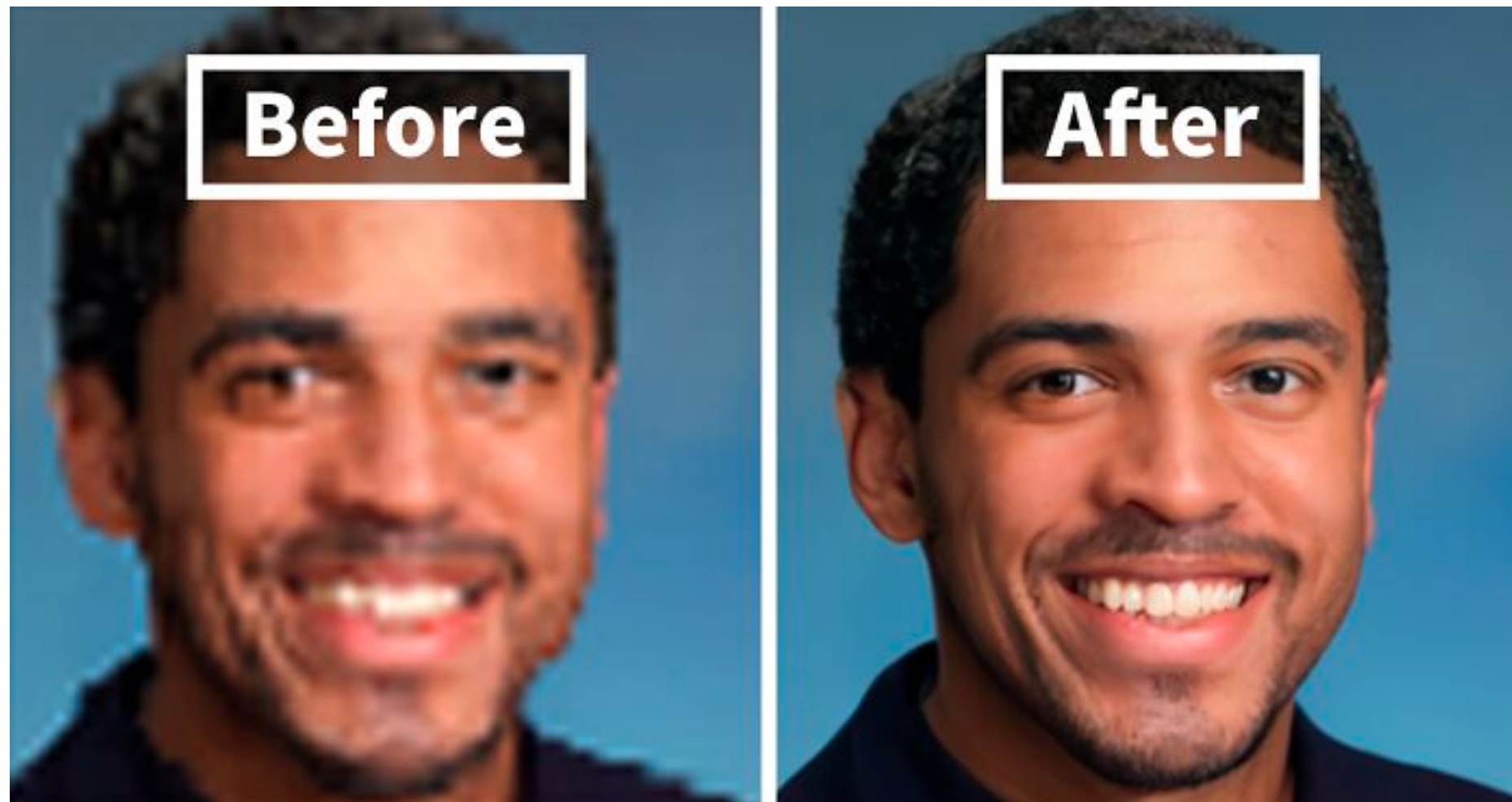
Two .fastq files. Lines correspond to each other



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
```

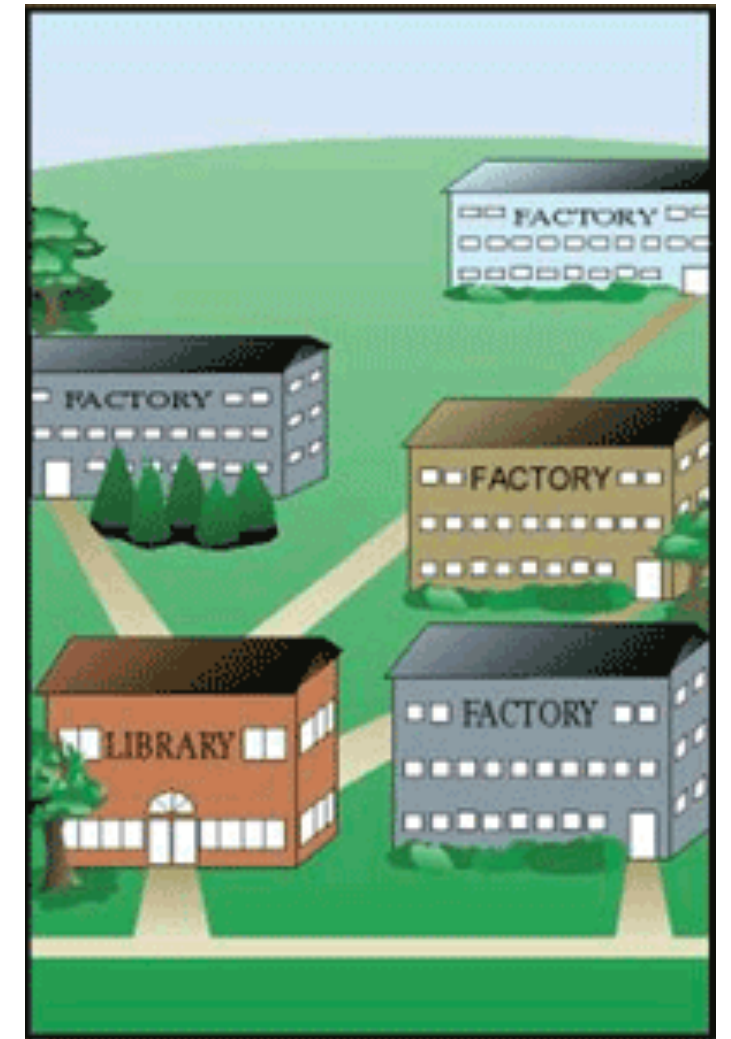
Computer vision-based solution



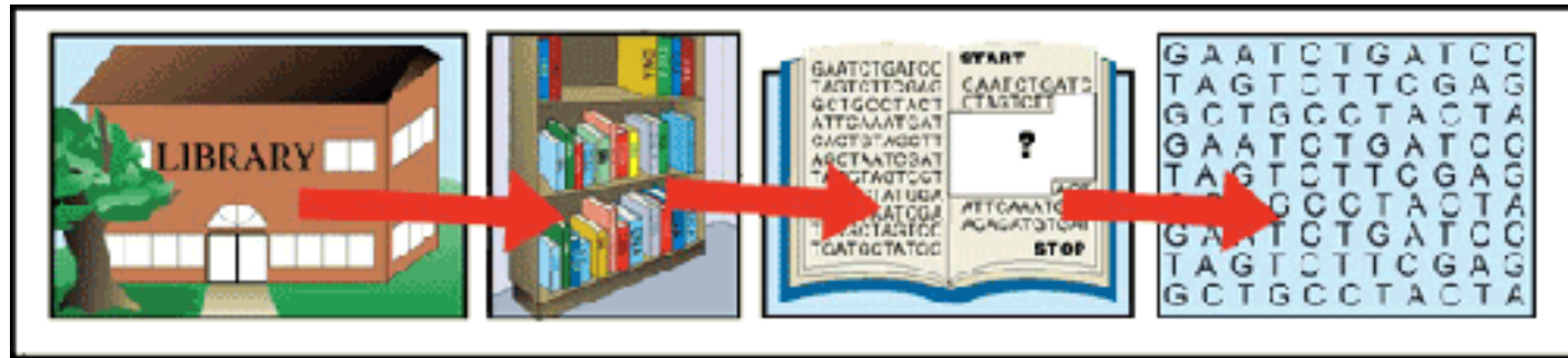
source:<https://www.boredpanda.com/google-ai-amazing-image-enhancement/>

Cell: a town

- Cell is a town. It has many factories and one library.
- Library (**nucleus**)
 - The most important part of this town
 - Contains genetic information in the cell
- Many factories
 - Retrieve receipts from library and then produce different kinds of goods
 - Goods are **proteins**

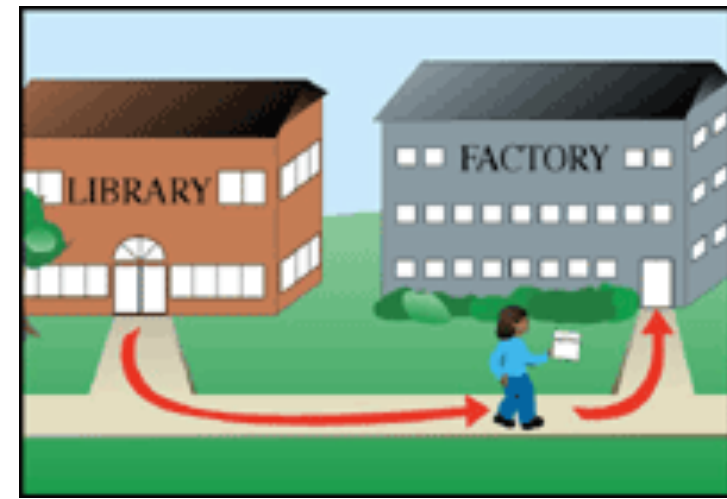


Nucleus: a library



- Books in the library (**DNA**)
 - Genetical material that determines physical characteristics of the cell and ultimately the organism
 - Books are always in the library. We can only make copy of it and send the copy to factories.
- Copy from books (**messenger RNA (mRNA)**)
 - Retrieve instructions from nucleus (only copy, not remove)
 - A “copy” of the information contained in the sequences of DNA
 - This copy is transported to a separate region of the cell (e.g., factory) where proteins are made
- Copy machine (**transcription**)
 - mRNA takes the instructions within the nucleus and bring it to the factory
- Turning the instructions into a product in the factory (**translation, cytoplasm**)

Translation: a factory



- Factory gets information from library
 - Nucleic acids
- Factory generates goods based on the information
 - Goods are proteins (amino acid language)
 - Essential for the cell and our human body.

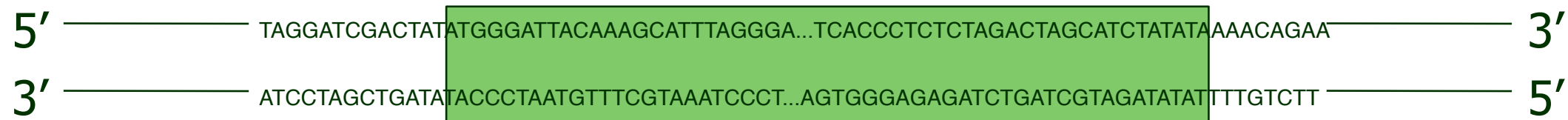
Summarization

- A town has one library and many factories. Factory gets instructions from library and use introduction to produce goods.
- A cell has one nucleic and many other components. mRNA sends information from nucleic to each component. Each component uses it to produce proteins.

Genes & proteins

gene, transcription, translation, protein

Double-stranded DNA



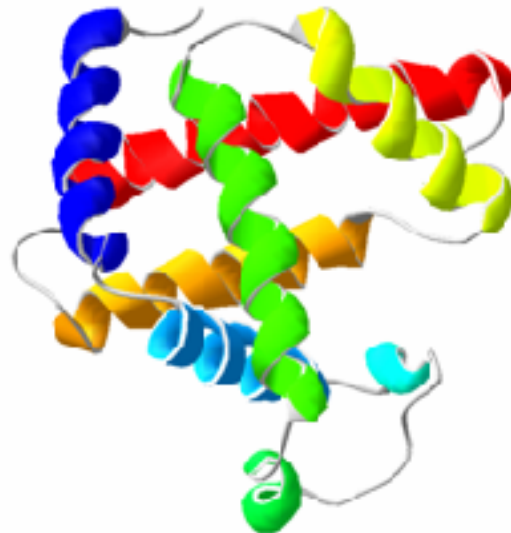
transcription

Single-stranded RNA

AUGGGAUUACAAAGCAUUUAGGGA...UCACCCUCUCUAGACUAGCAUCUAUAUAA

translation

protein



The genetic code

- Mapping from a codon to an amino acid

The Genetic Code

	U	C	A	G	
U	UUU Phenylalanine UUC alanine UUG Leucine UUA Leucine	UCU Serine UCC Serine UCA Serine UCG Serine	UAU Tyrosine UAC Tyrosine UAA Stop UAG Stop	UGU Cysteine UGC Cysteine UGA Stop UGG Tryptophan	U C A G
C	CUU Leucine CUC Leucine CUA Leucine CUG Leucine	CCU Proline CCC Proline CCA Proline CCG Proline	CAU Histidine CAC Histidine CAA Glutamine CAG Glutamine	CGU Arginine CGC Arginine CGA Arginine CGG Arginine	U C A G
A	AUU Isoleucine AUC Isoleucine AUA Isoleucine AUG Methionine	ACU Threonine ACC Threonine ACA Threonine ACG Threonine	AAU Asparagine AAC Asparagine AAA Lysine AAG Lysine	AGU Serine AGC Serine AGA Arginine AGG Arginine	U C A G
G	GUU Valine GUC Valine GUA Valine GUG Valine	GCU Alanine GCC Alanine GCA Alanine GCG Alanine	GAU Aspartic acid GAC Aspartic acid GAA Glutamic acid GAG Glutamic acid	GGU Glycine GGC Glycine GGA Glycine GGG Glycine	U C A G

Translation

- Always start from Met

5' . . . A U U A U G G C C U G G A C U U G A . . . 3'

UTR

Met

Ala

Trp

Thr

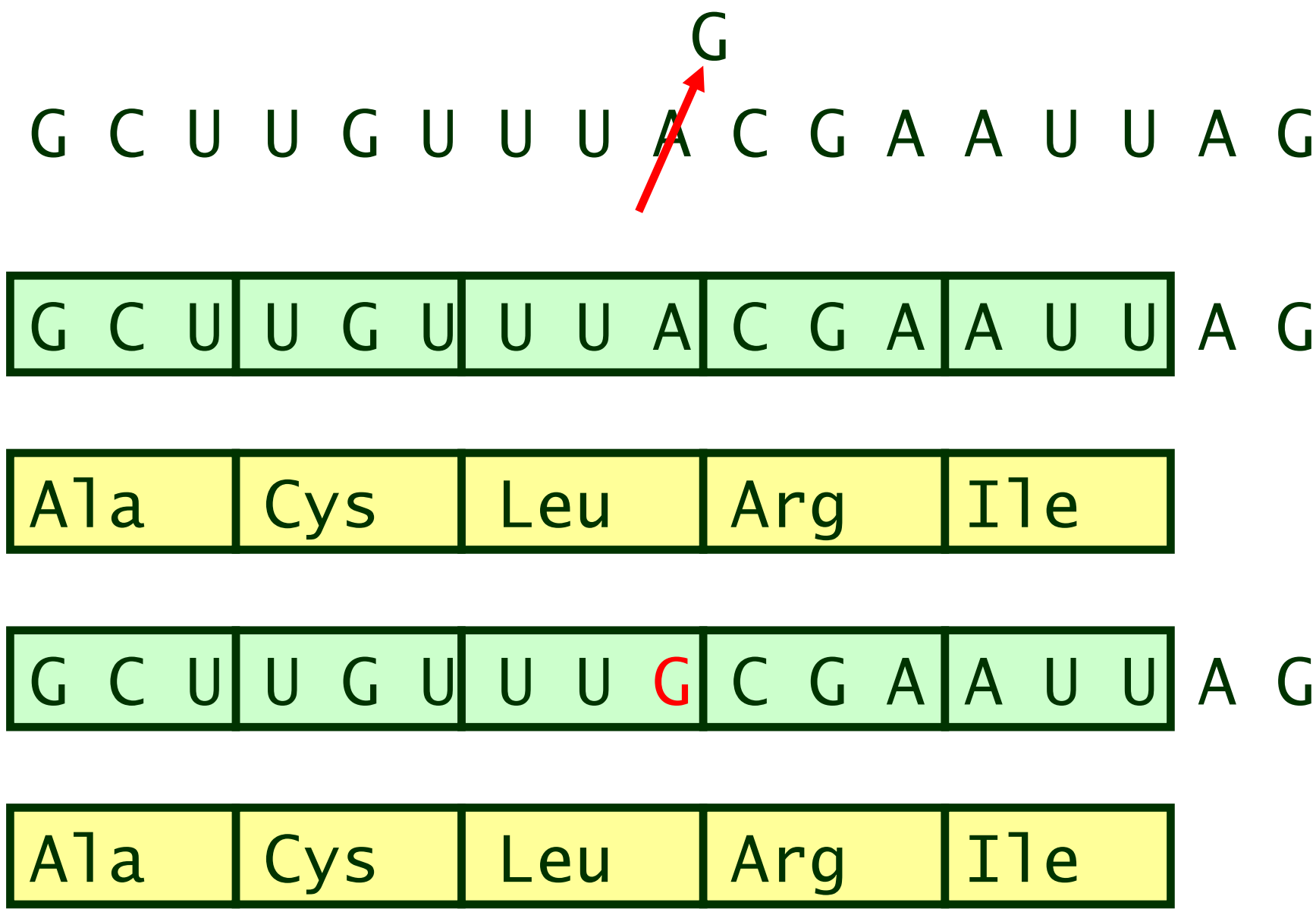
Start
Codon

Stop
Codon

Errors?

- What if the transcription / translation machinery makes mistakes?
- What is the effect of **mutations** in coding regions?

Synonymous mutation



Missense mutation

G C U U G **G** U U A C G A A U U A G



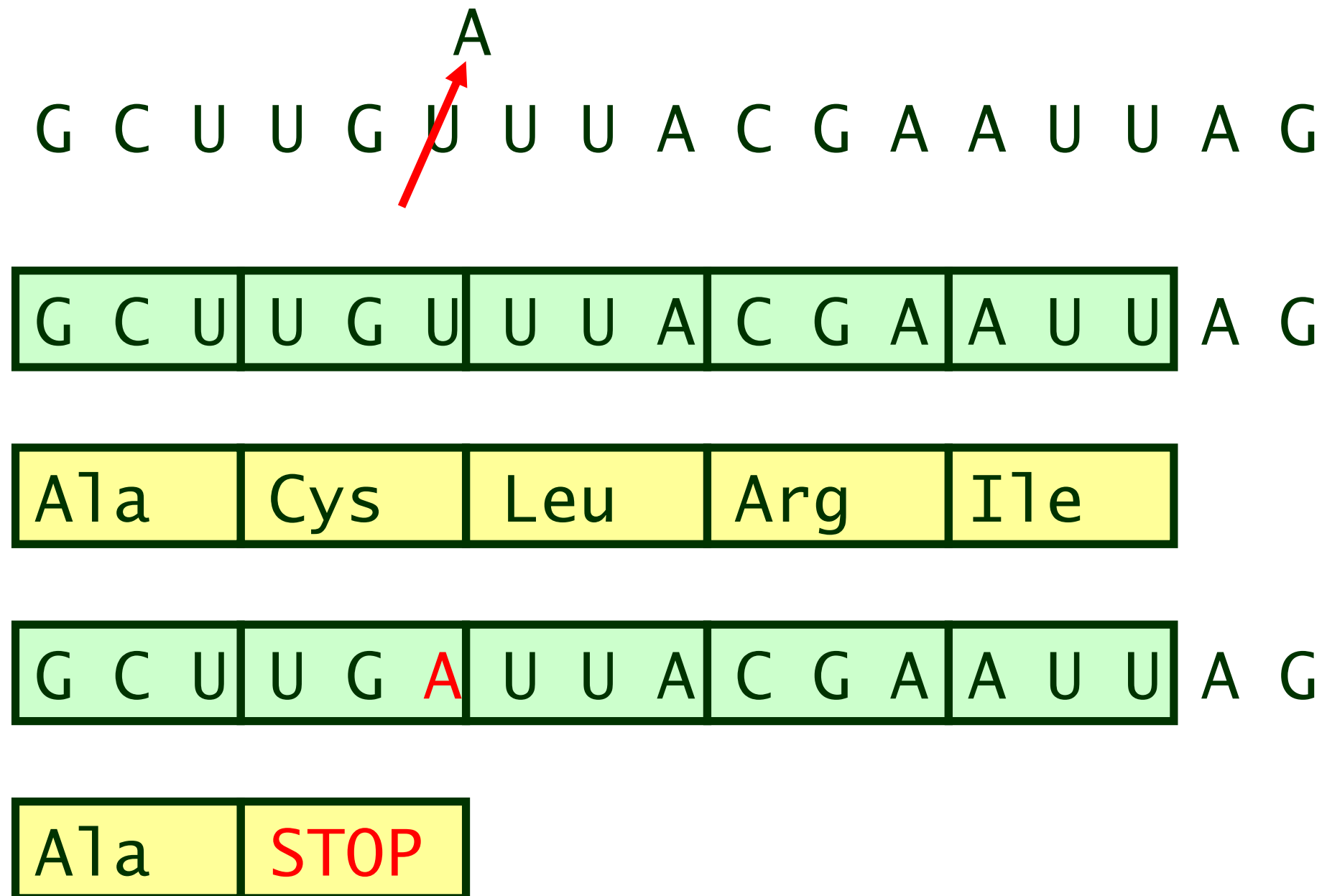
G C U	U G U	U U A	C G A	A U U	A G
-------	-------	-------	-------	-------	-----

Ala	Cys	Leu	Arg	Ile
-----	-----	-----	-----	-----

G C U	U G G	U U A	C G A	A U U	A G
-------	--------------	-------	-------	-------	-----

Ala	Trp	Leu	Arg	Ile
-----	------------	-----	-----	-----

Nonsense mutation



Frameshift

G C U U G U ~~U~~ U A C G A A U U A G

G C U	U G U	U U A	C G A	A U U	A G
-------	-------	-------	-------	-------	-----

Ala	Cys	Leu	Arg	Ile
-----	-----	-----	-----	-----

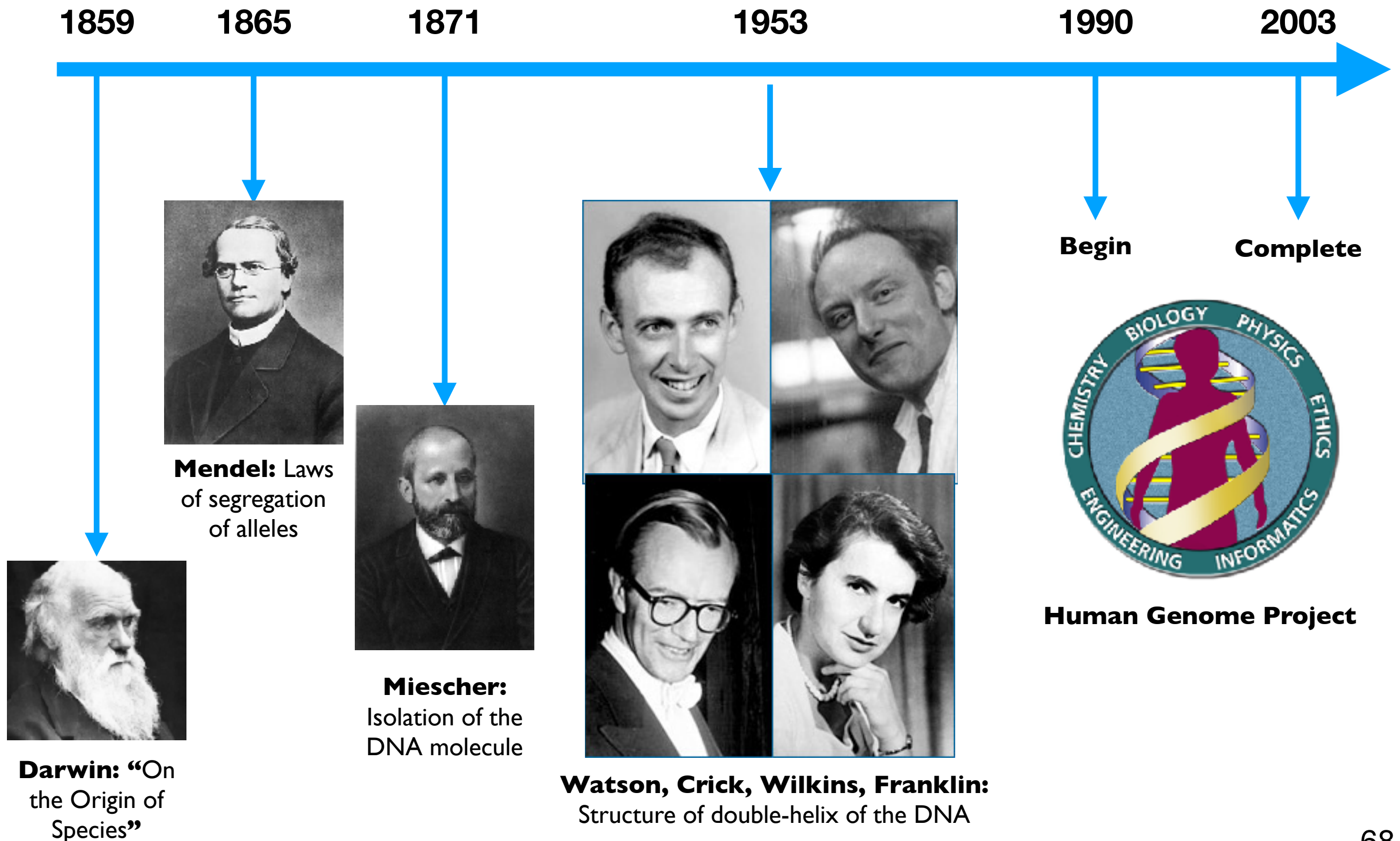
G C U	U G U	U A C	G A A	U U A	G
-------	-------	-------	-------	-------	---

Ala	Cys	Tyr	Glu	Leu
-----	-----	-----	-----	-----

Goal for today

- Human genome project
- Dynamic programming
- Needleman-Wunsch Algorithm

History of Molecular Biology



Human Genome Project



The February 2001 cover of Nature



Science

3 billion basepairs

\$3 billion

1990: Start

Most important scientific discovery in the 20th century.

2000: Bill Clinton:

2001: Draft

2003: Finished

2021: now what?

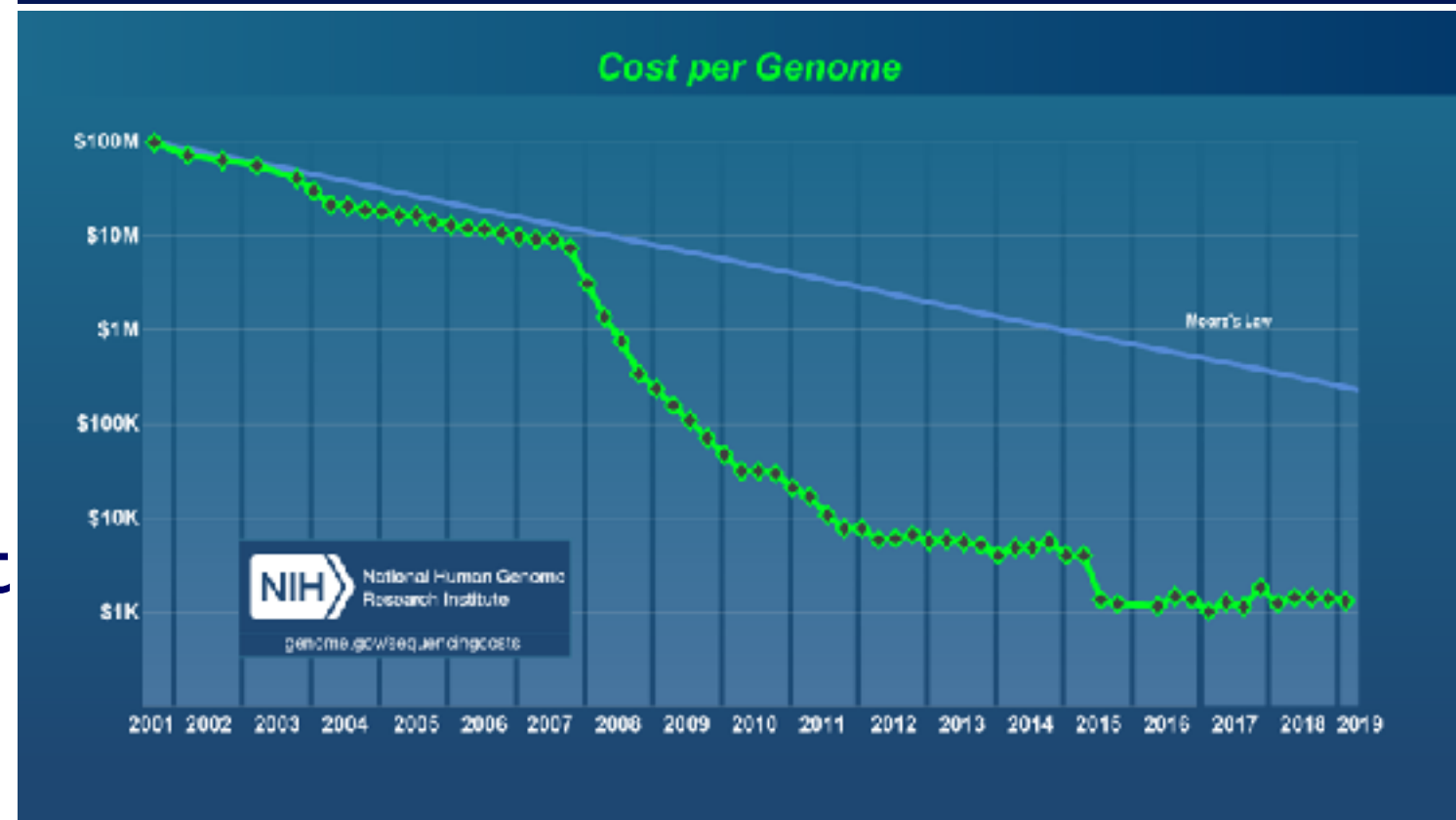
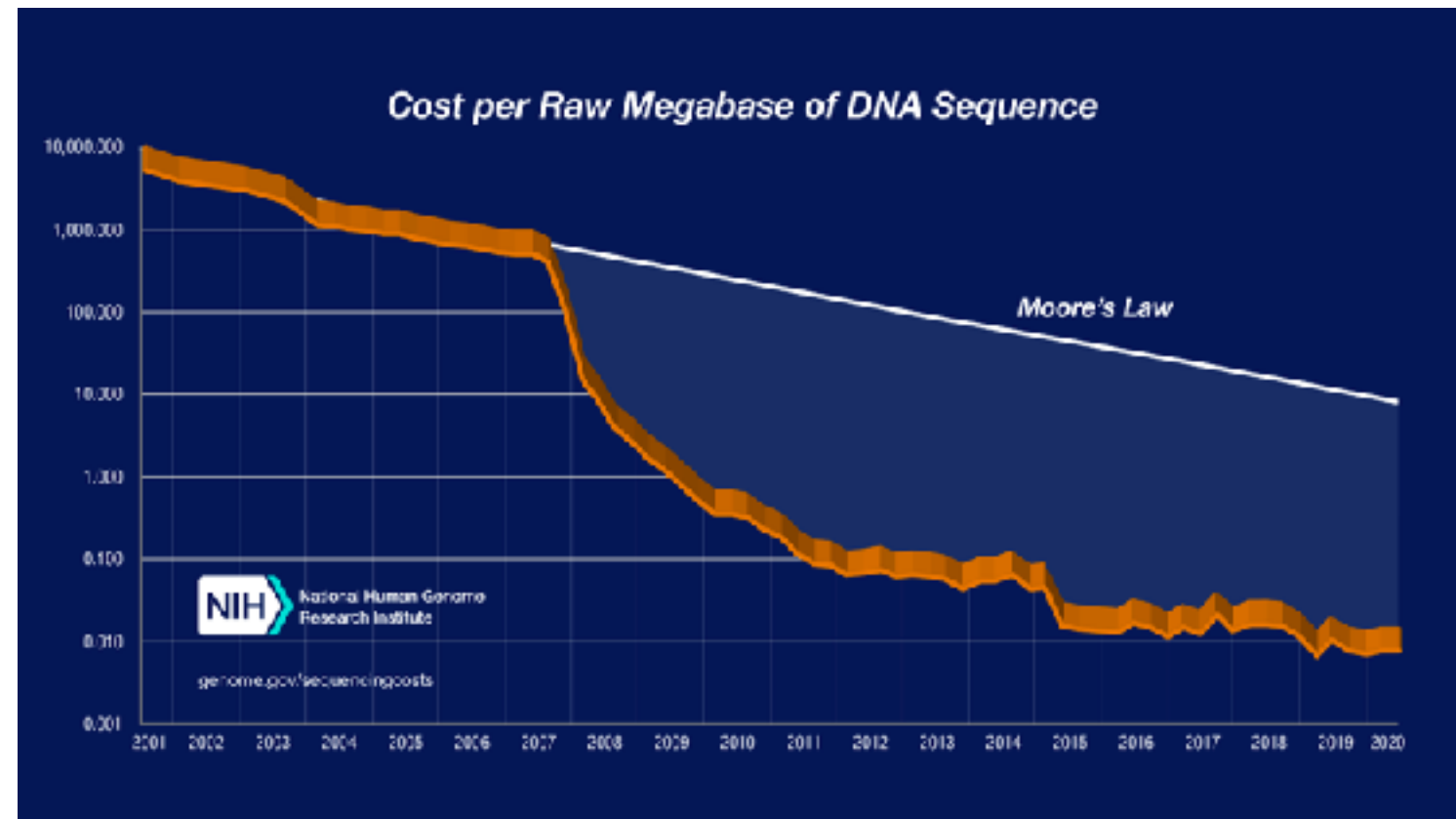
Sequencing Growth

Cost of one human genome

- 2004: \$30,000,000
- 2008: \$100,000
- 2010: \$10,000
- 2011: \$4,000
- 2015: \$1,000
- 2020: \$1,000



How much would
you pay for a smart
phone?



Uses of Genomes

- Medicine
 - Mendelian diseases
 - Cancer
 - Drug dosage (eg. Warfarin)
 - Disease risk
 - Diagnosis of infections
 - ...
- Ancestry
- Genealogy
- Nutrition
-



Sampling of traits reported in 23andme



- Ability to match musical pitch
- Asparagus odor detection
- Back hair (men only)
- Bald spot (men only)
- Bunions
- Cilantro Taste Aversion
- Early Hair Loss (men only)
- Fear of Heights
- Fear of Public Speaking
- Ice Cream Flavor Preference
- Misophonia
- Mosquito Bite Frequency
- Photic Sneeze Reflex
- Sweet vs. Salty
- Toe Length Ratio
- Unibrow
- Wake-Up Time
- Widow's Peak

Sampling of traits reported in 23andme



- Ability to match musical pitch
- Asparagus odor detection
- Back hair (men only)
- Bald spot (men only)
- Bunions
- Cilantro Taste Aversion
- Early Hair Loss (men only)
- Fear of Heights
- Fear of Public Speaking
- Ice Cream Flavor Preference
- **Misophonia**
- Mosquito Bite Frequency
- Photic Sneeze Reflex
- Sweet vs. Salty
- Toe Length Ratio
- Unibrow
- Wake-Up Time
- Widow's Peak

Sampling of traits reported in 23andme



- Ability to match musical pitch
- Asparagus odor detection
- Back hair (men)
- Bald spot (men)
- Bunions
- Cilantro Taste Aversion
- Early Hair Loss
- Fear of Heights
- Fear of Public Speaking
- **Ice Cream Flavor Preference**
- Misophonia

23andMe's New Trait Report Puts a Cherry on Top of Your Ice Cream Preference

June 28, 2019 By 23andMe under Health and Traits

- Unibrow
- Wake-Up Time
- Widow's Peak

Biological discovery: data-driven + literature evidence

You Scream, I Scream, We all Scream for Ice Cream

By using a statistical model and data from more than 980,000 23andMe research participants, our scientists were able to identify 739 genetic markers associated with preferring vanilla ice cream to chocolate. Pulling those genetic markers together with non-genetic factors – such as age and sex – we developed a model to estimate the likelihood of preferring vanilla ice cream to chocolate.

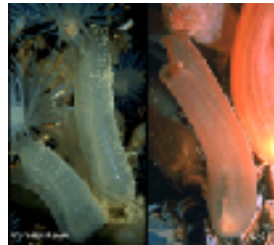
Obviously your ice cream flavor preference is influenced by far more than genetics – culture and environment for instance – but as with other types of food preferences, your genetics is the cherry on top. A person's preference may be related to their sense of smell. Indeed many of the genetic variants we found associated with ice cream preference are in or near olfactory receptor genes, like OR10A6 and OR5M8. Those genes contain instructions for proteins that help detect odors. While you're eating, your brain combines information from odors and your taste buds to perceive flavor.

Sampling of diseases reported in 23andme

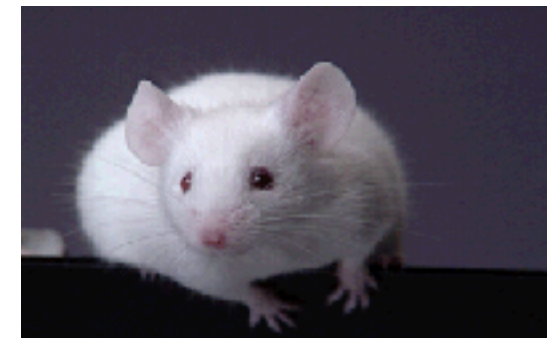
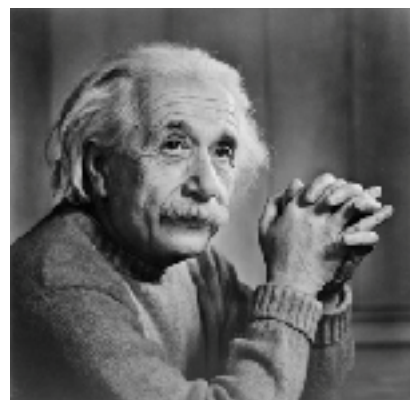
- Type 2 Diabetes
- Age-related macular degeneration
- Celiac Disease
- Late-Onset Alzheimer's Disease
- Parkinson's Disease



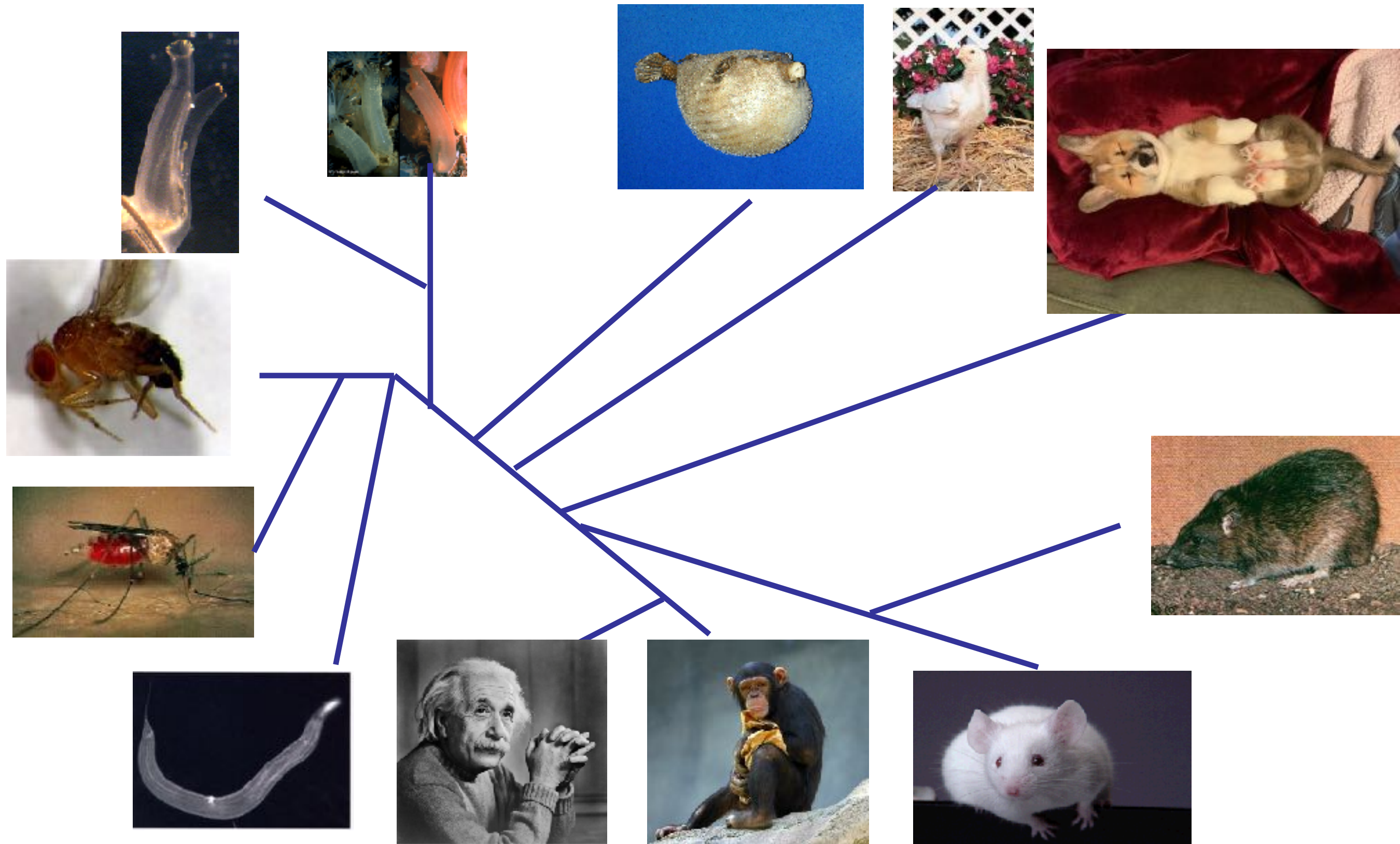
Complete DNA Sequences



More than 1000 complete
genomes have been
sequenced



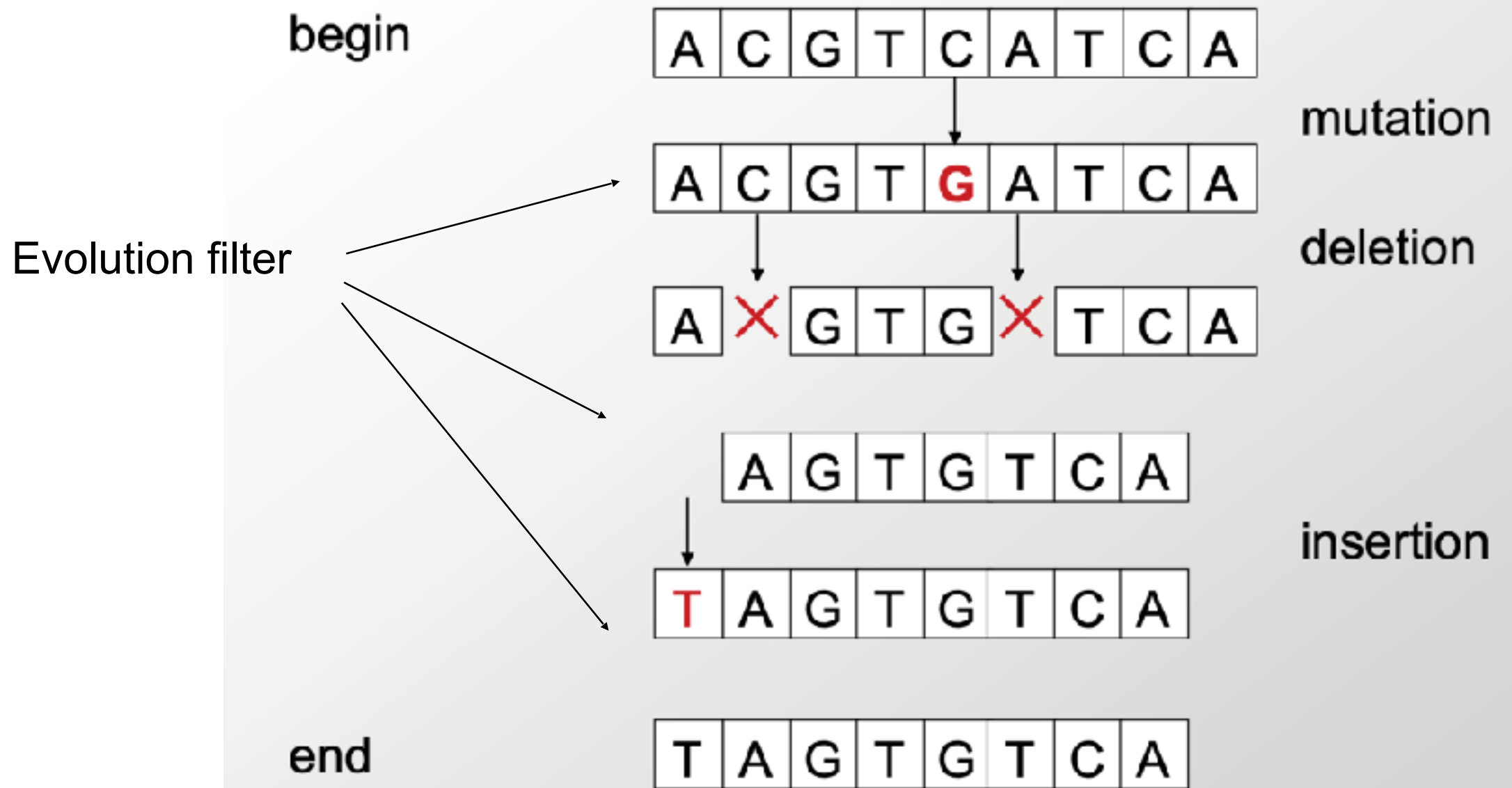
Evolution



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

Theodosius Dobzhansky

Genomes change over time



That is why we want to compare sequences

Partial CTCF protein sequence in 8 organisms:

<i>H. sapiens</i>	-EDSSDS-ENAEPLDLDNEDEEEPAVEIEPEPE-----PQPVTPA
<i>P. troglodytes</i>	-EDSSDS-ENAEPLDLDNEDEEEPAVEIEPEPE-----PQPVTPA
<i>C. lupus</i>	-EDSSDS-ENAEPLDLDNEDEEEPAVEIEPEPE-----PQPVTPA
<i>B. taurus</i>	-EDSSDS-ENAEPLDLDNEDEEEPAVEIEPEPE-----PQPVTPA
<i>M. musculus</i>	-EDSSDSEENAEPLDLDNEEEEPAVEIEPEPE--PQPQPQPQPQPQPVAPA
<i>R. norvegicus</i>	-EDSSDS-ENAEPLDLDNEEEEPAVEIEPEPEPQPQPQPQPQPQPVAPA
<i>G. gallus</i>	-EDSSDSEENAEPLDLDNEDEEETAVEIEAEPE-----VSAEAPA
<i>D. rerio</i>	DDDDDDSDDEHGEPDLDDIDEEDEDDL-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:

<i>H. sapiens</i>	-EDSSDS-ENAEPLDLDNEDEEPAVEIEPEPE-----PQPVTTPA
<i>P. troglodytes</i>	-EDSSDS-ENAEPLDLDNEDEEPAVEIEPEPE-----PQPVTTPA
<i>C. lupus</i>	-EDSSDS-ENAEPLDLDNEDEEPAVEIEPEPE-----PQPVTTPA
<i>B. taurus</i>	-EDSSDS-ENAEPLDLDNEDEEPAVEIEPEPE-----PQPVTTPA
<i>M. musculus</i>	-EDSSDSEENAEPLDLDNEEEPAVEIEPEPE--PQPQPPPPQPQVAPA
<i>R. norvegicus</i>	-EDSSDS-ENAEPLDLDNEEEPAVEIEPEPEPQPQPPQPQPQVAPA
<i>G. gallus</i>	-EDSSDSEENAEPLDLDNEDEEETAVEIEAEPE-----VSAEAPA
<i>D. rerio</i>	DDDDDDSDDEHGEPDLDDIDEEDEDDL-LDEDQMGLLDQAPPSVPIP-APA



D. rerio



G. gallus



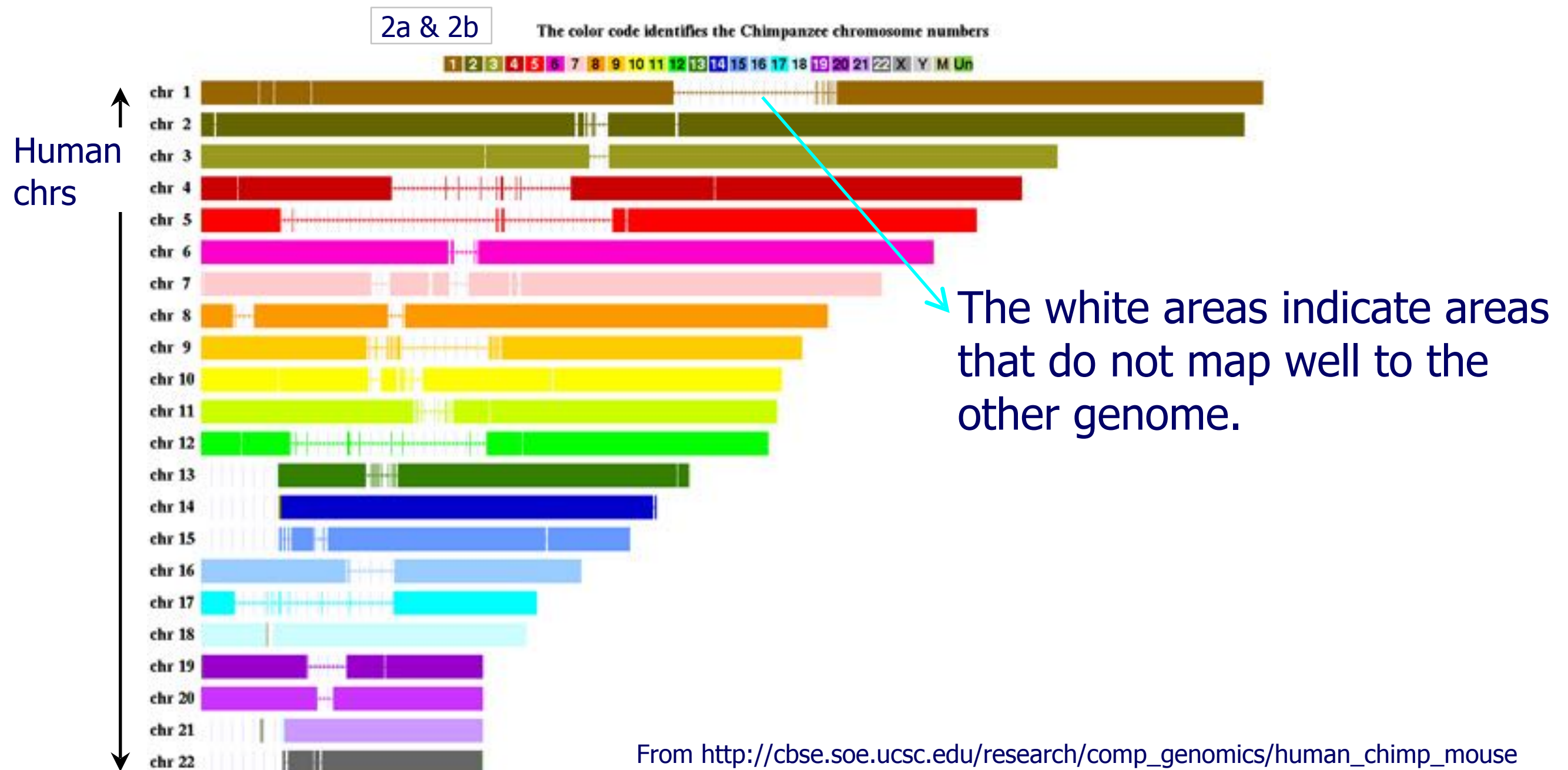
P. Troglodytes



C. lupus

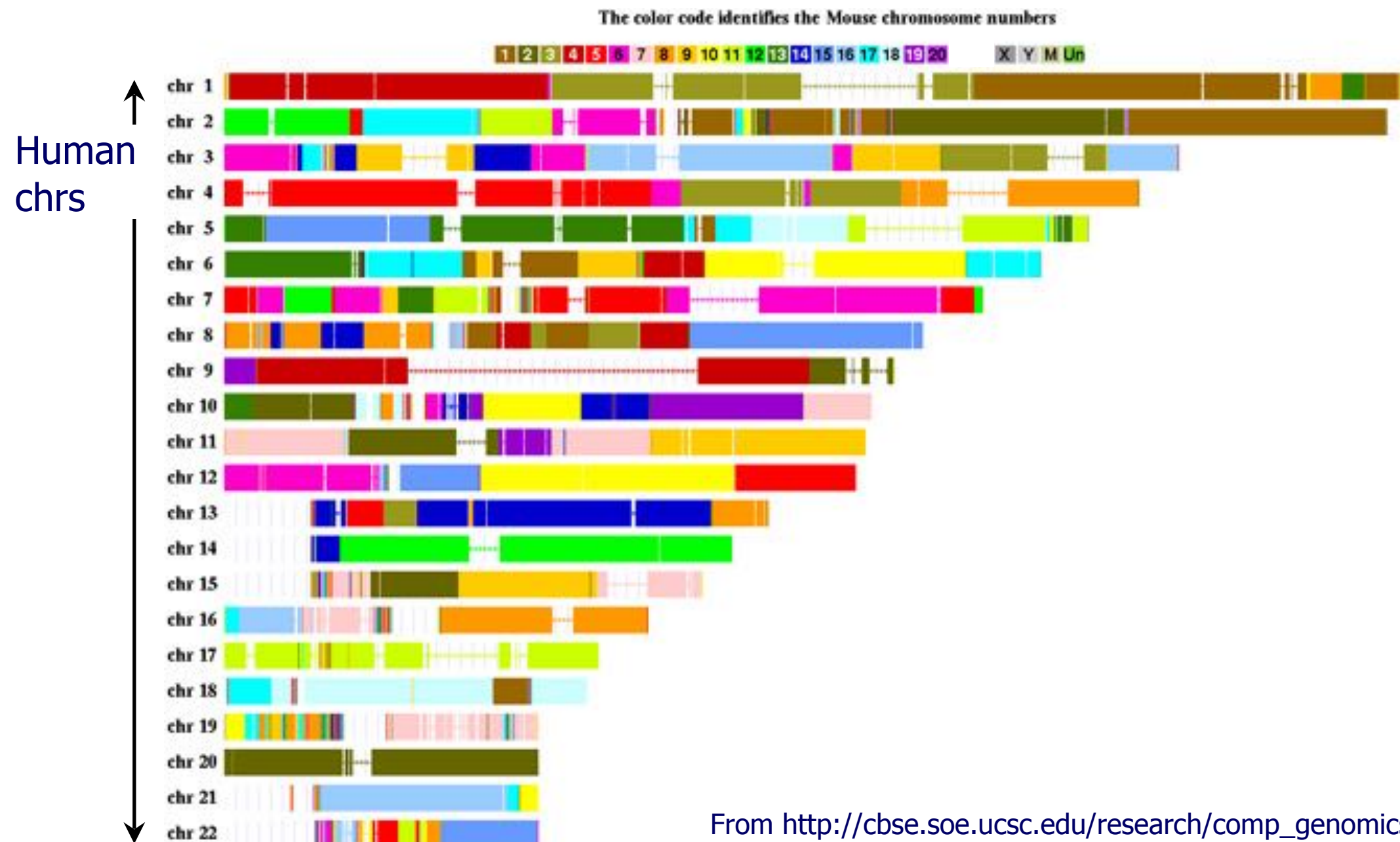
Comparing Human, Chimp, and Mouse Genomes

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



Comparing Human, Chimp, and Mouse Genomes

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



What does a fastq file look like?

```
1 @ERR000589.41 EAS139_45:5:1:2:111/1
2 CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
3 +
4 3IIIIIIIIIIII>1IIIF9BG08E00I%IG+&?(4)%00646.C1#&(
5 @ERR000589.42 EAS139_45:5:1:2:1293/1
6 AGTTGTTAAAATCCAAGCCAATTAAGATAGTCTTATCTTTTAAAAGAAAT
7 +
8 IIIIIGII.AIIII=?I9G-/II=+I=4?761BA2C9I+5A711+&>1$/I
```

Very large! ~3000000000 lines

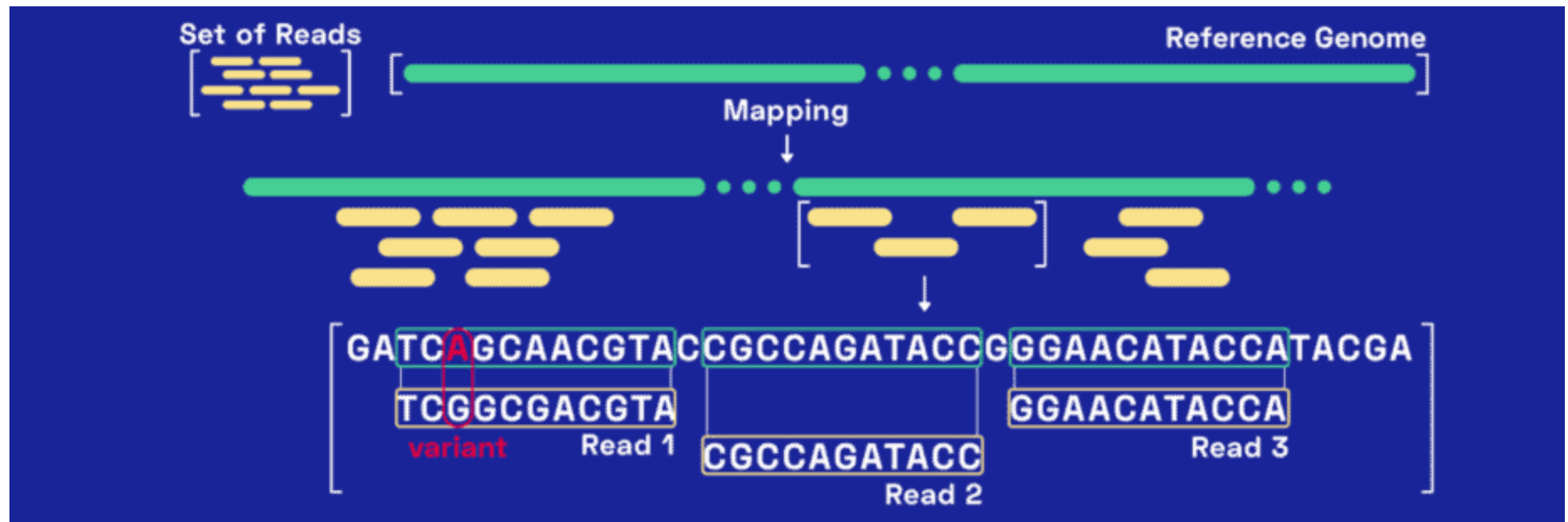
Quality: ASCII chars

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



The Simplest String Comparison Problem

Given: Two strings

$$a = a_1a_2a_3a_4\dots a_m$$

$$b = b_1b_2b_3b_4\dots b_n$$

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

riddle $\xrightarrow{\text{delete}}$ ridle $\xrightarrow{\text{mutate}}$ riple $\xrightarrow{\text{insert}}$ 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 $n \geq 2$.

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

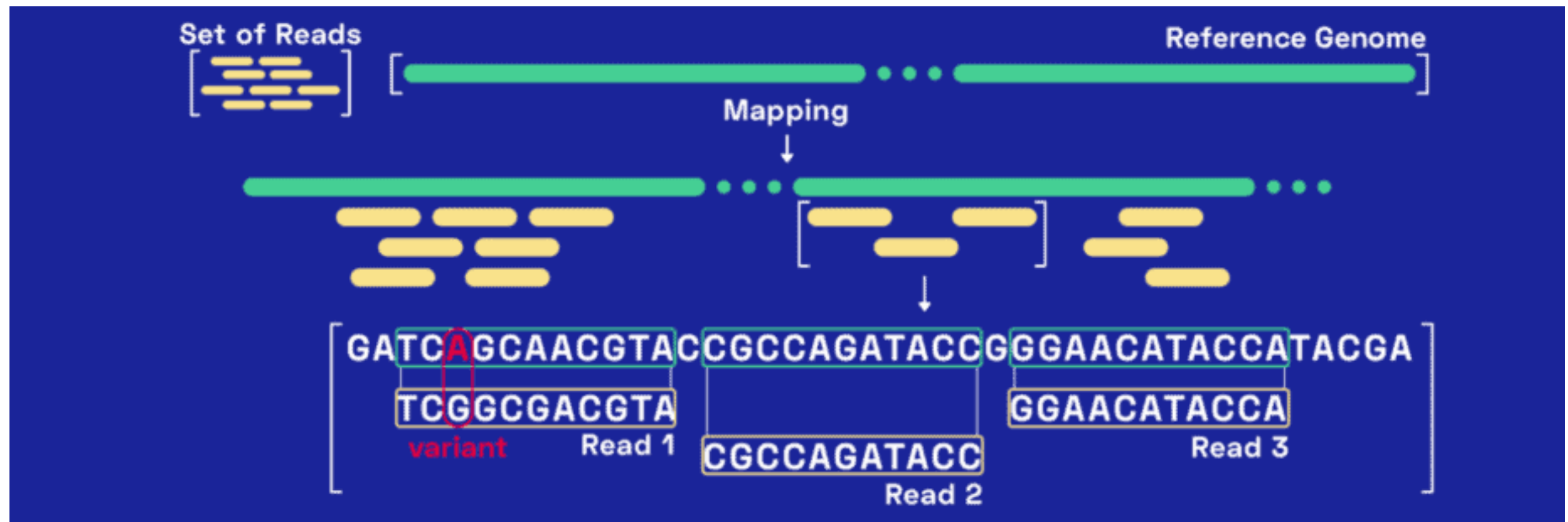
Global alignment: Needleman-Wunsh algorithm

- 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?

What does a fastq file look like?

Reference genome: "average" human genome.

Most widely used human genome GRCh38: derived from 13 thirteen anonymous volunteers



Key difference

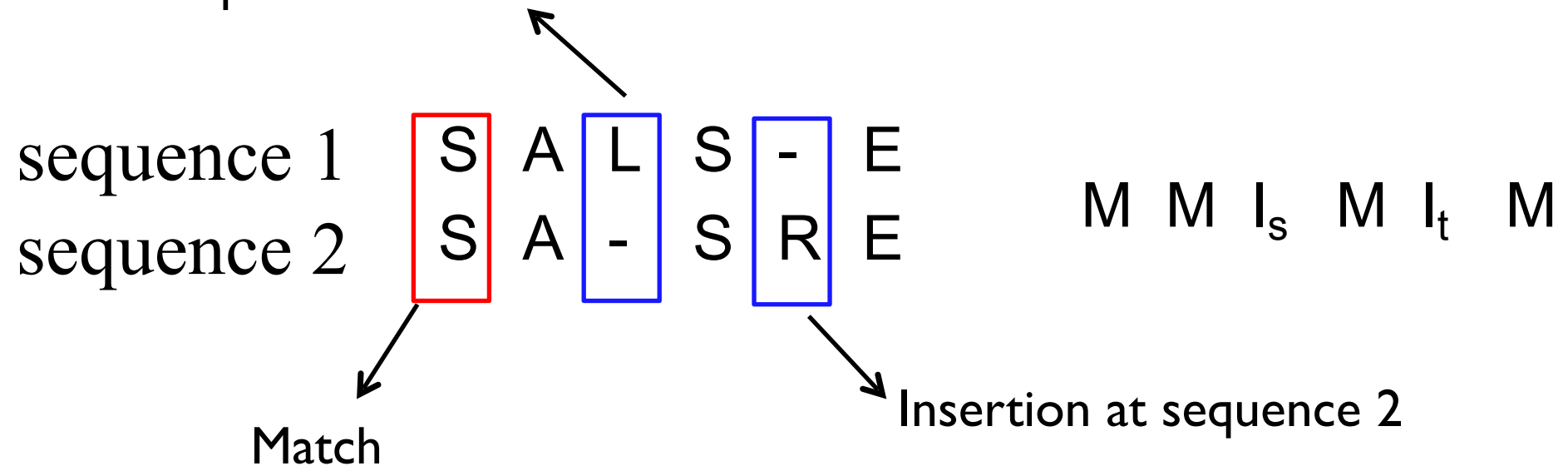
- Sometimes we need to choose whether we want to align the entire sequence.

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

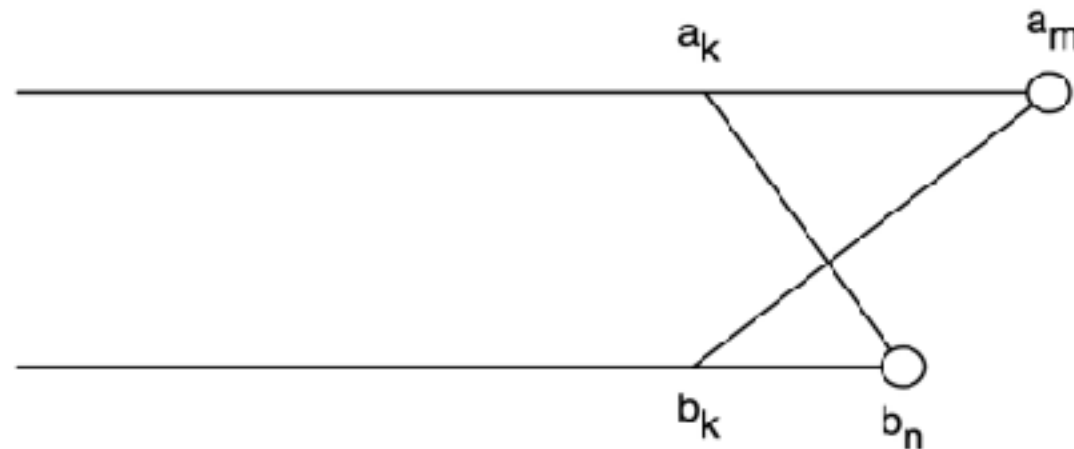
Insertion at sequence 1



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
TAGCTATCACGACCGCGGGTCGATTTGCCCGAC

–AGGCTATCACCTGACCTCCAGGCCGA––TGCCC–––
TAG–CTATCAC––GACCGC––GGTCGATTTGCCCGAC

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:

- Mutations
- Insertions
- Deletions

AGGCCTC

AGG**A**CTC

AGG**G**CCTC

AGG . CTC

Scoring Function:

Match: +m

Mismatch: -s

Gap: -d

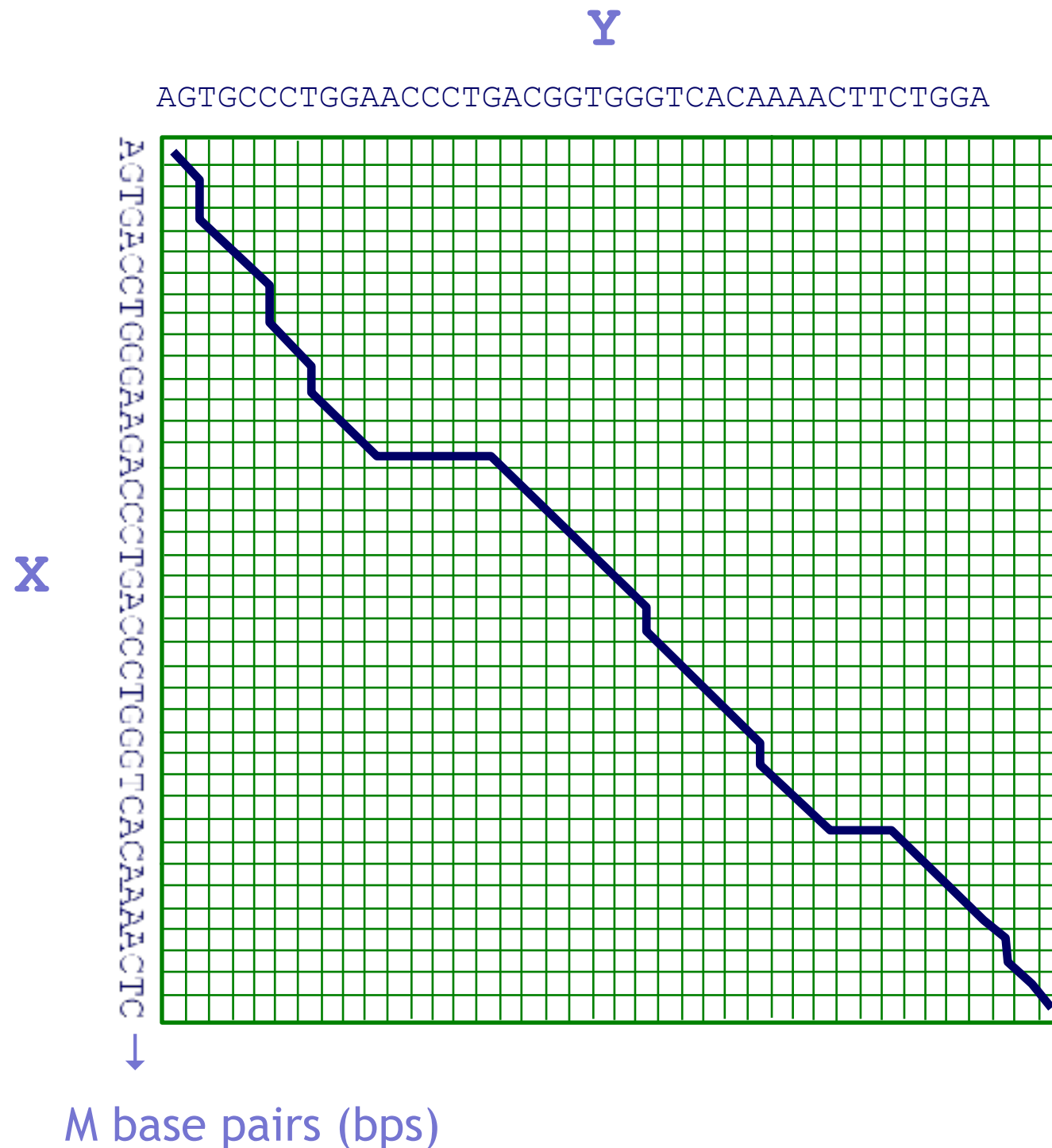
$$\text{Score } F = (\# \text{ matches}) \times m - (\# \text{ mismatches}) \times s - (\# \text{ gaps}) \times d$$

Alternative definition:

minimal edit distance

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

How do we compute the best alignment?

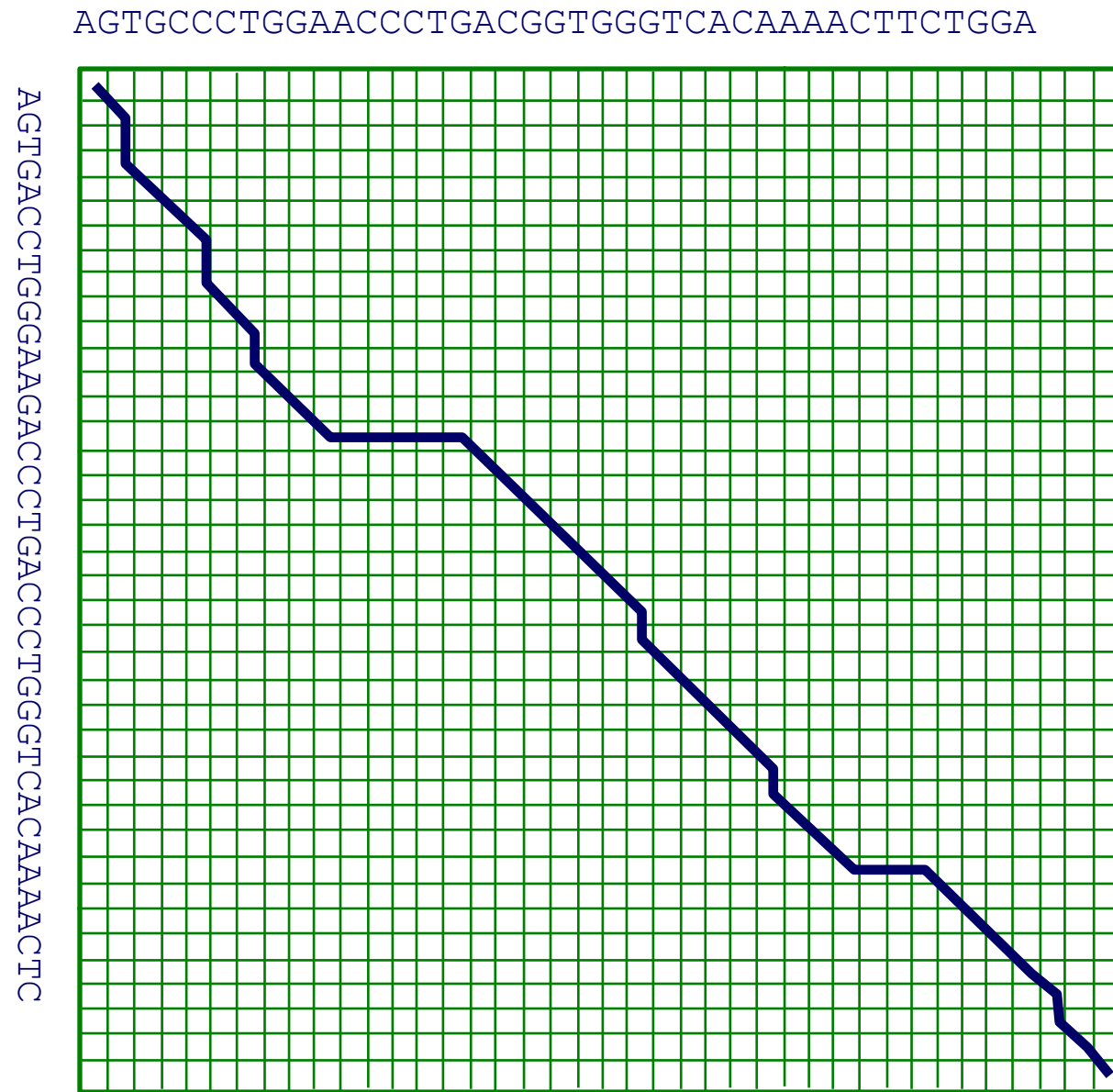


Every non-decreasing path from (0,0) to (M, N) corresponds to an alignment of the two sequences, and vice versa.

(exercise)

X: AGTGACCTGGGAAGA-----C...
Y: AG--TGC--CC-TGGAACCCT...

How do we compute the best alignment?



Too many possible alignments:

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

Alignment is additive

Observation:

The score of aligning

$x_1 \dots x_M$

$y_1 \dots y_N$

is additive

Say that

$x_1 \dots x_i$

$x_{i+1} \dots x_M$

aligns to

$y_1 \dots y_j$

$y_{j+1} \dots y_N$

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

$x_1 \dots x_i$

$y_1 \dots y_j$

F is the DP “Matrix” or “Table”

“Memorization”

Scoring Function

- Sequence edits:

- Mutations
- Insertions
- Deletions

AGGCCTC

AGG**A**CTC

AGG**G**CCTC

AGG . CTC

Scoring Function:

Match: +m

Mismatch: -s

Gap: -d

$$\text{Score } F = (\# \text{ matches}) \times m - (\# \text{ mismatches}) \times s - (\# \text{ gaps}) \times d$$

Alternative definition:

minimal edit distance

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

Dynamic Programming (cont'd)

Notice three possible cases:

1. x_i aligns to y_j

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

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

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

2. x_i aligns to a gap

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

$y_1 \dots y_j \quad -$

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

3. y_j aligns to a gap

$x_1 \dots x_i \quad -$

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

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

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

$y = \text{CATGT}$

match: +2

mismatch, gap: -1

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

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

↑
X

match: +2

mismatch, gap: -1

← Y

↑
X

x = ACGCTG

match: +2

y = CATGT

mismatch, gap: -1

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

-
C

$s(-,C) = -1$

↑
X

x = ACGCTG

match: +2

y = CATGT

mismatch, gap: -1

		j	0	1	2	3	4	5	←Y
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							

-
C

$s(-,C) = -1$

↑
X

x = ACGCTG

match: +2

y = CATGT

mismatch, gap: -1

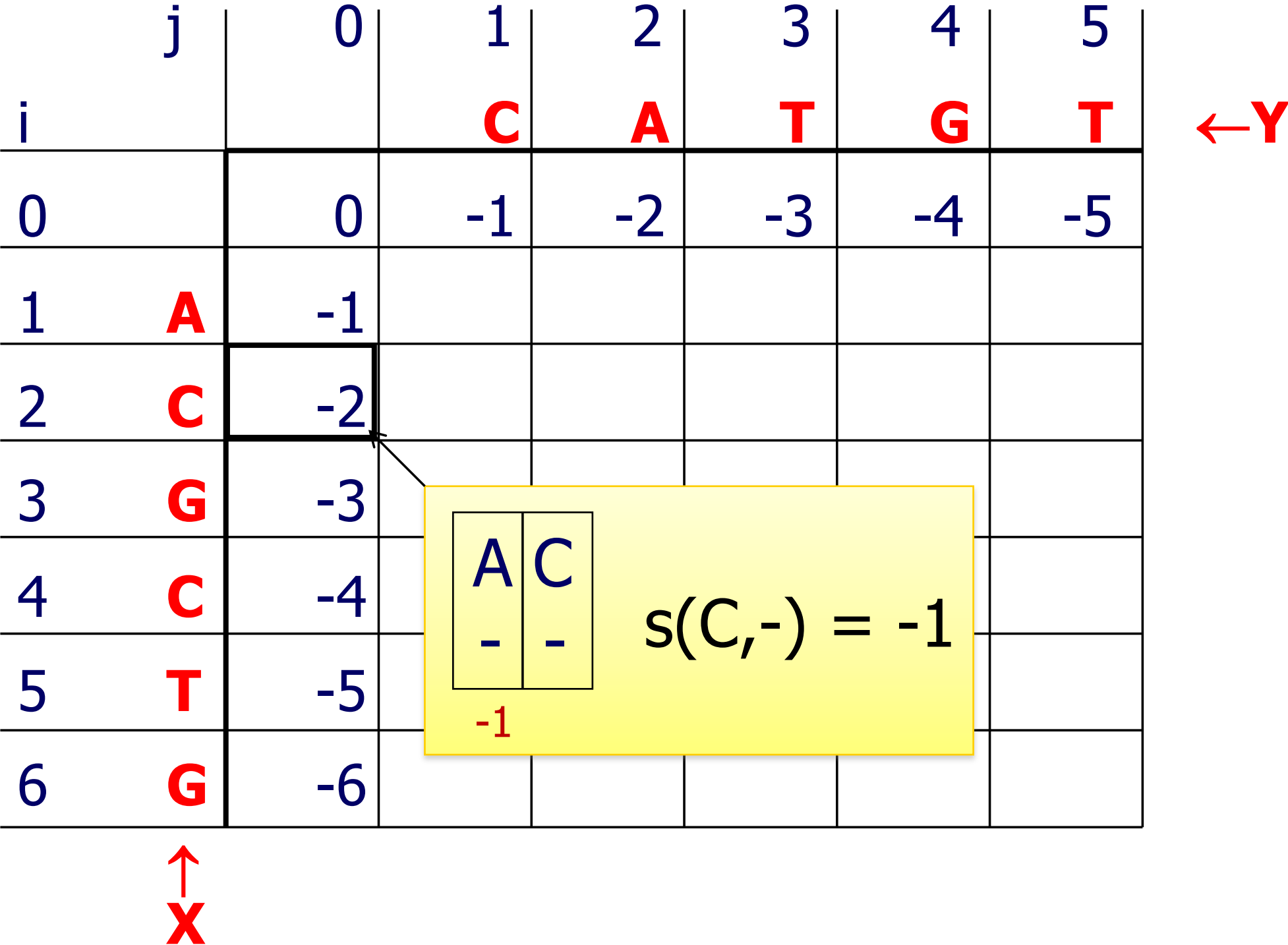
		j	0	1	2	3	4	5	←Y
i				C	A	T	G	T	
0			0	-1	-2	-3	-4	-5	
1	A	-1							
2	C	-2							
3	G	-3							
4	C	-4							
5	T	-5							
6	G	-6							

A
-

$s(A, -) = -1$

↑
X

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



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

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

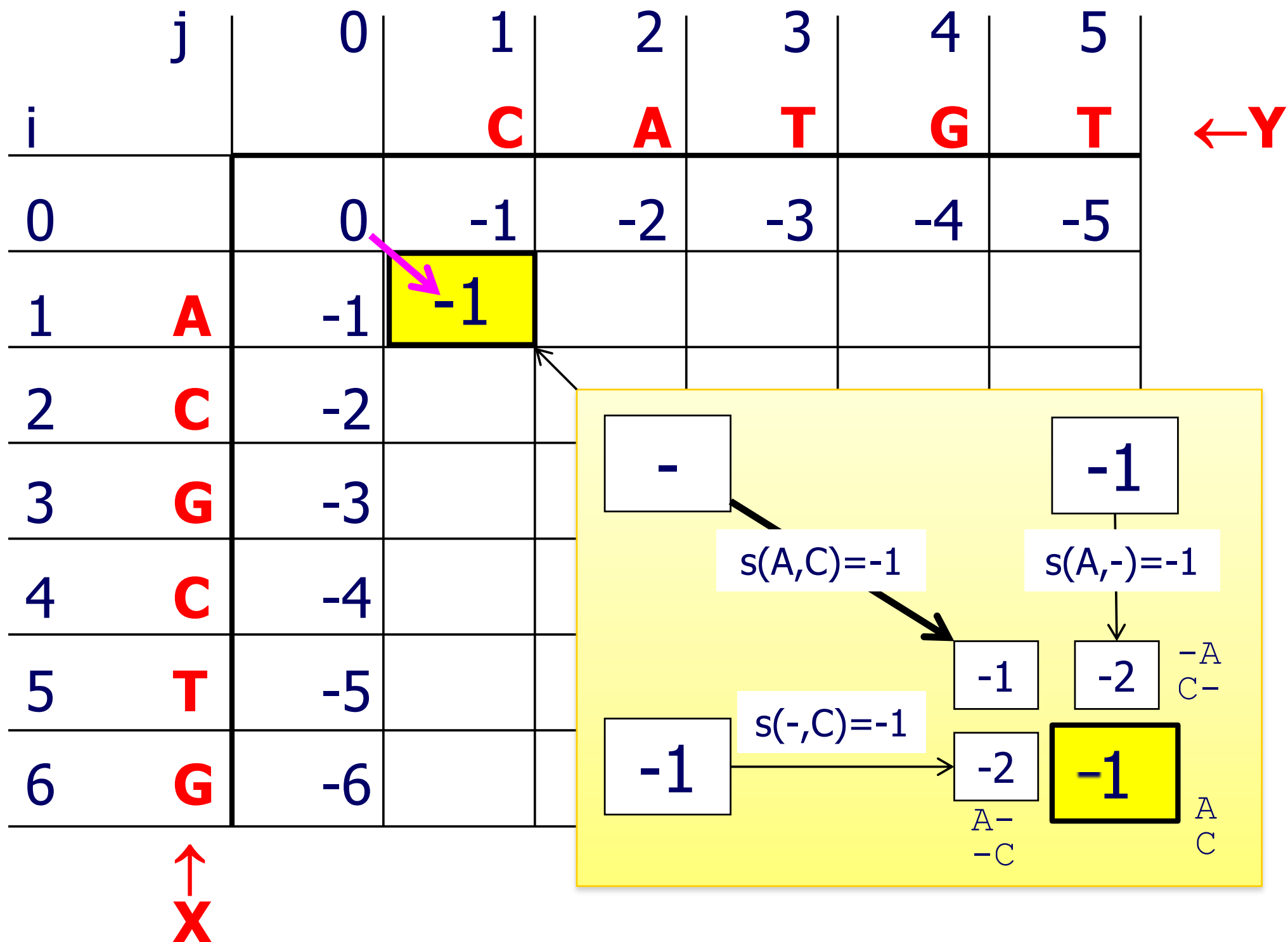
↑
X

x = ACGCTG

match: +2

y = CATGT

mismatch, gap: -1



x = ACGCTG

match: +2

y = CATGT

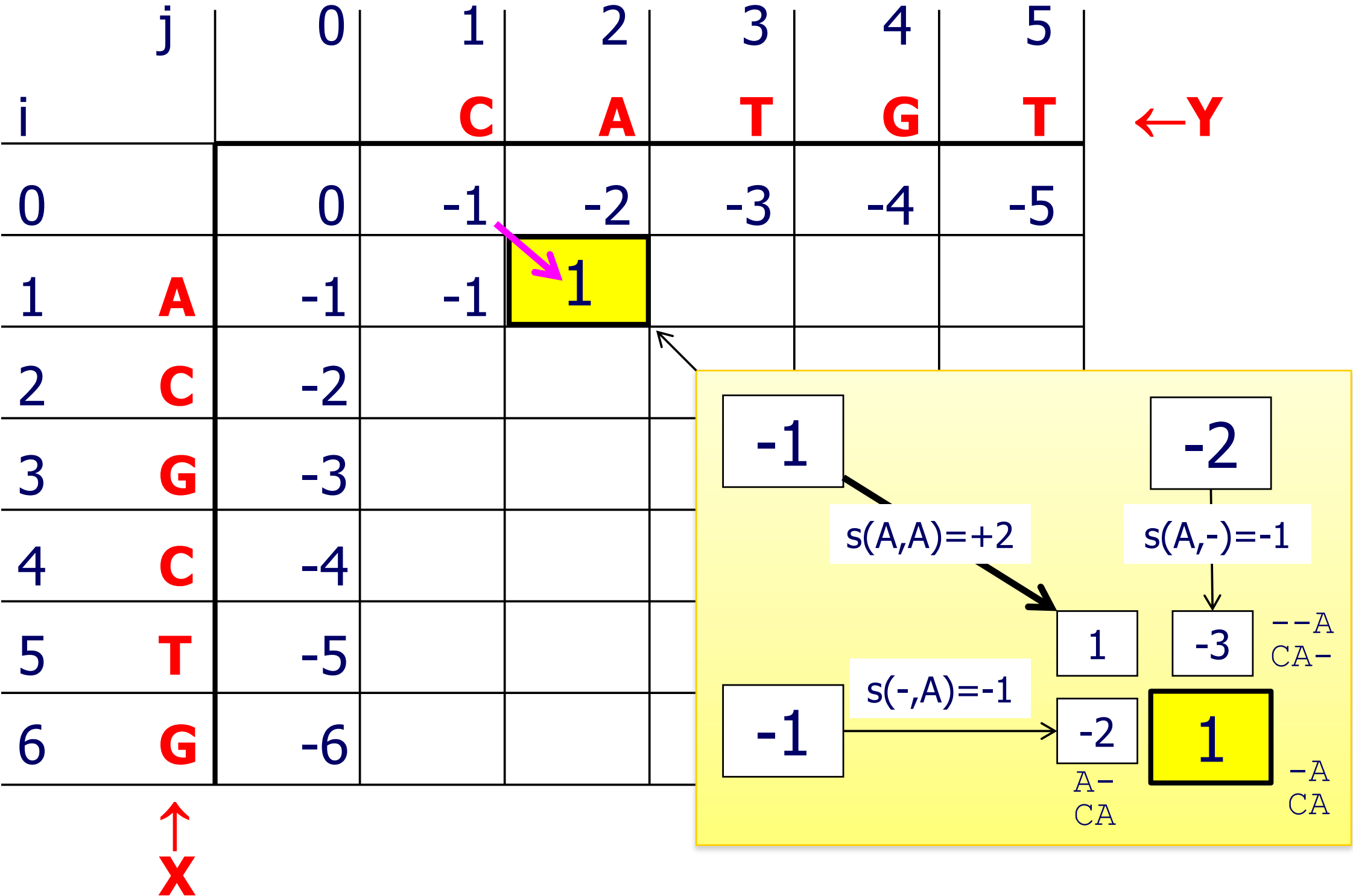
mismatch, gap: -1

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



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

Time
= O(MN)

↑
X

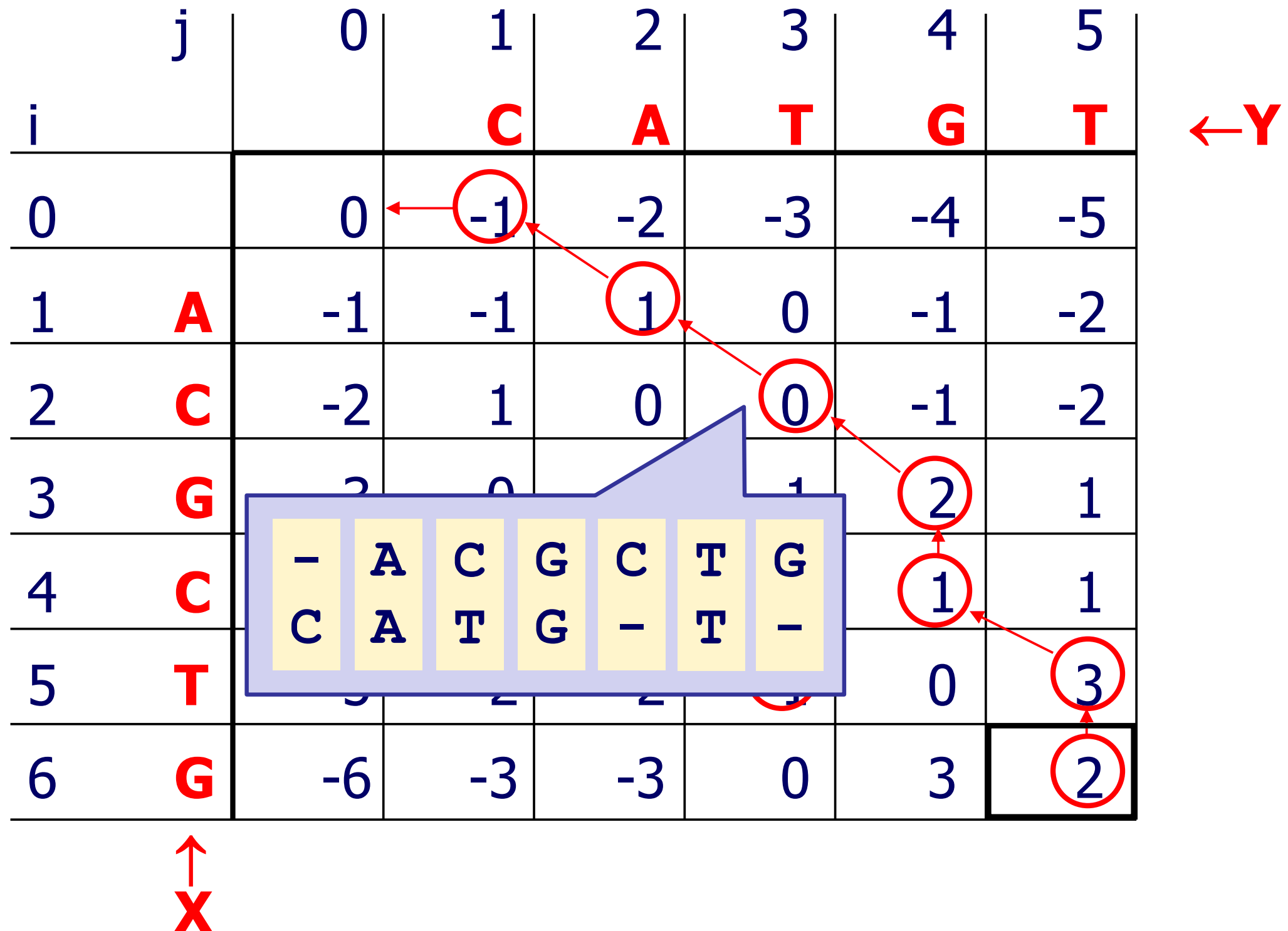
Finding alignments: trace back

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

		j	0	1	2	3	4	5	←Y
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	

↑X

Finding alignments: trace back



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

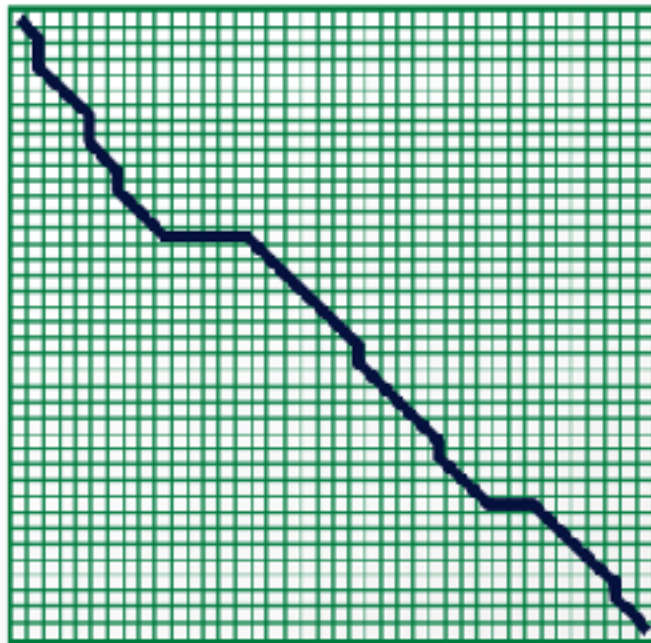
For each $i = 1 \dots M$
For each $j = 1 \dots N$

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

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

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

Global Alignment



Needleman-Wunsch algorithm

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

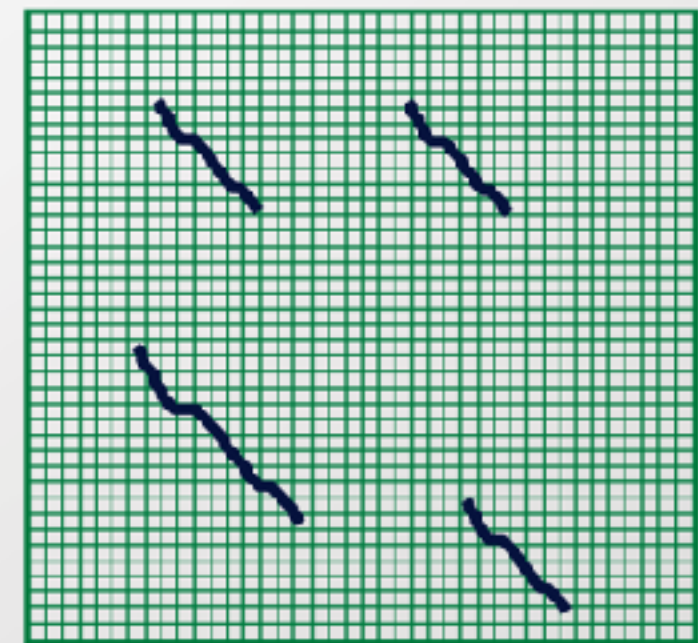
Iteration:

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

Termination: Bottom right

vs.

Local alignment



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

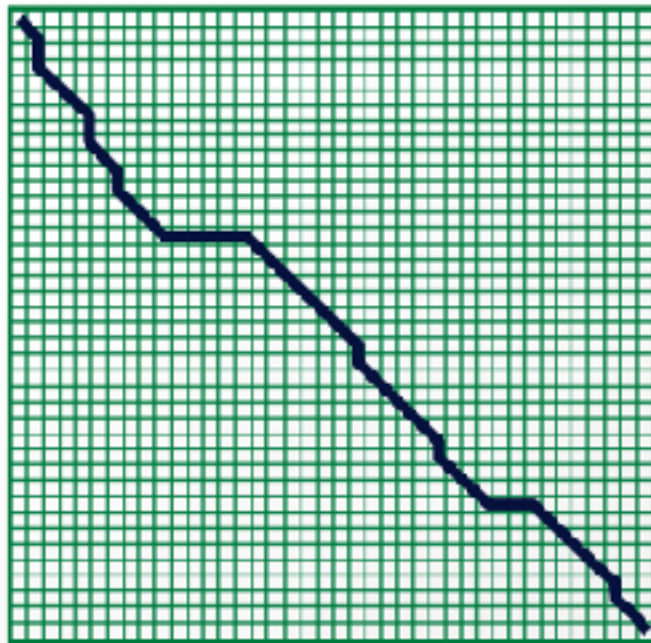
- Time:

$O(NM)$

- Space:

$O(NM)$

Global Alignment



Needleman-Wunsch algorithm

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

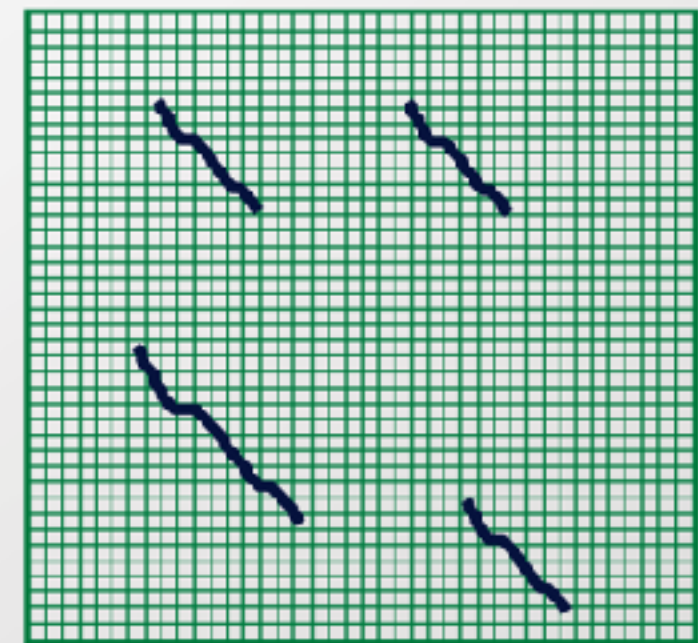
Iteration:

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

Termination: Bottom right

vs.

Local alignment



Smith-Waterman algorithm

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

Iteration:

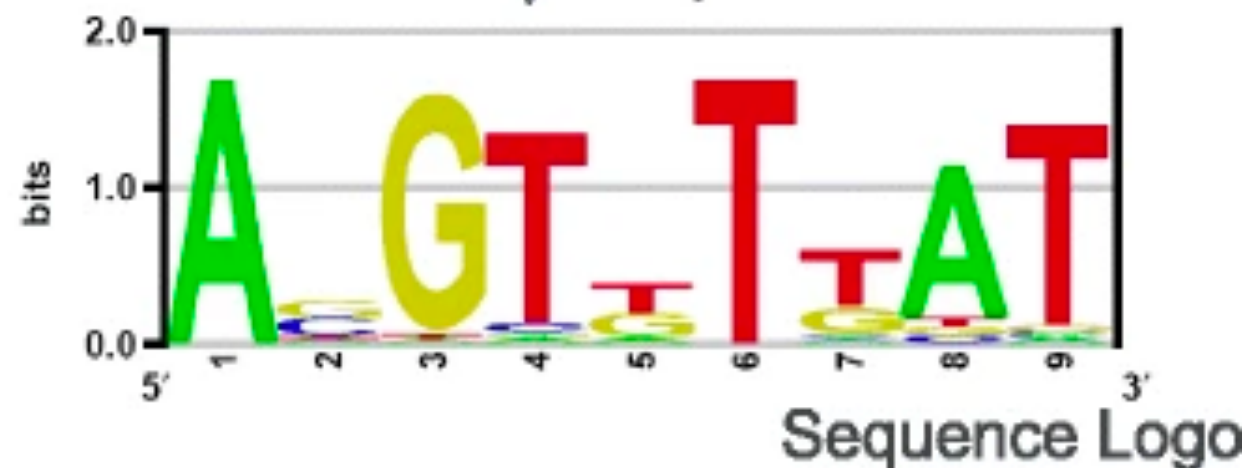
$$F(i, j) = \max \begin{cases} 0 \\ F(i-1, j) - d \\ F(i, j-1) - d \\ F(i-1, j-1) + s(x_i, y_j) \end{cases}$$

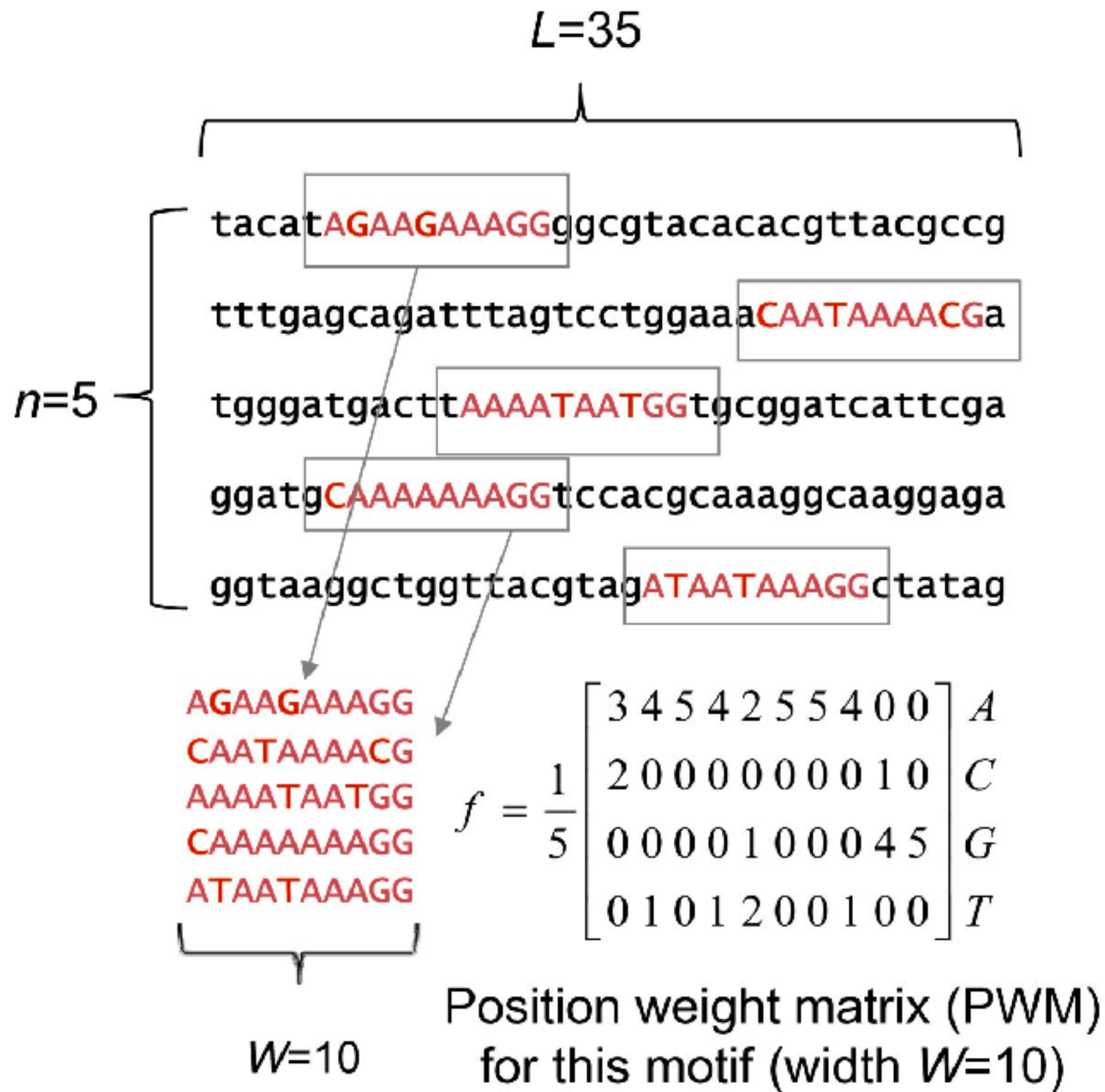
Termination: Anywhere

- 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





For example, given the following DNA sequences:

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

The corresponding PFM is:

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

Therefore, the resulting PPM is:^[1]

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

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

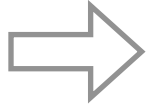
the probability of the sequence $S = \text{GAGGTAAAC}$ given the above PPM M

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

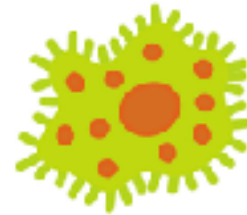
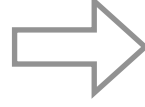
Computational methods for biology at different scales



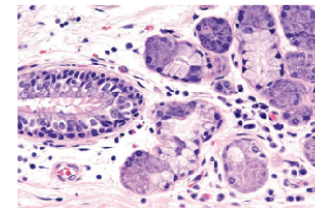
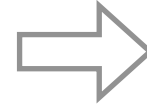
Gene
(1 nm)



Protein complexes (function)
(10-100nm)



Cell
(1-10 μ m)



Tissue
(100 μ m to 100 mm)



Complex organism
(> 1cm)

What does a fastq file look like?

```
1 @ERR000589.41 EAS139_45:5:1:2:111/1
2 CTTTCCTCCCTGCTTTCCTGGCCCCACCATTTCCAGGGAACATCTTGTCAT
3 +
4 3IIIIIIIIIIII>1IIIF9BG08E00I%IG+&?(4)%00646.C1#&(
5 @ERR000589.42 EAS139_45:5:1:2:1293/1
6 AGTTGTTAAAATCCAAGCCAATTAAGATAGTCTTATCTTTTAAAAGAAAT
7 +
8 IIIIIGII.AIIII=?I9G-/II=+I=4?761BA2C9I+5A711+&>1$/I
```

Very large! ~3000000000 lines

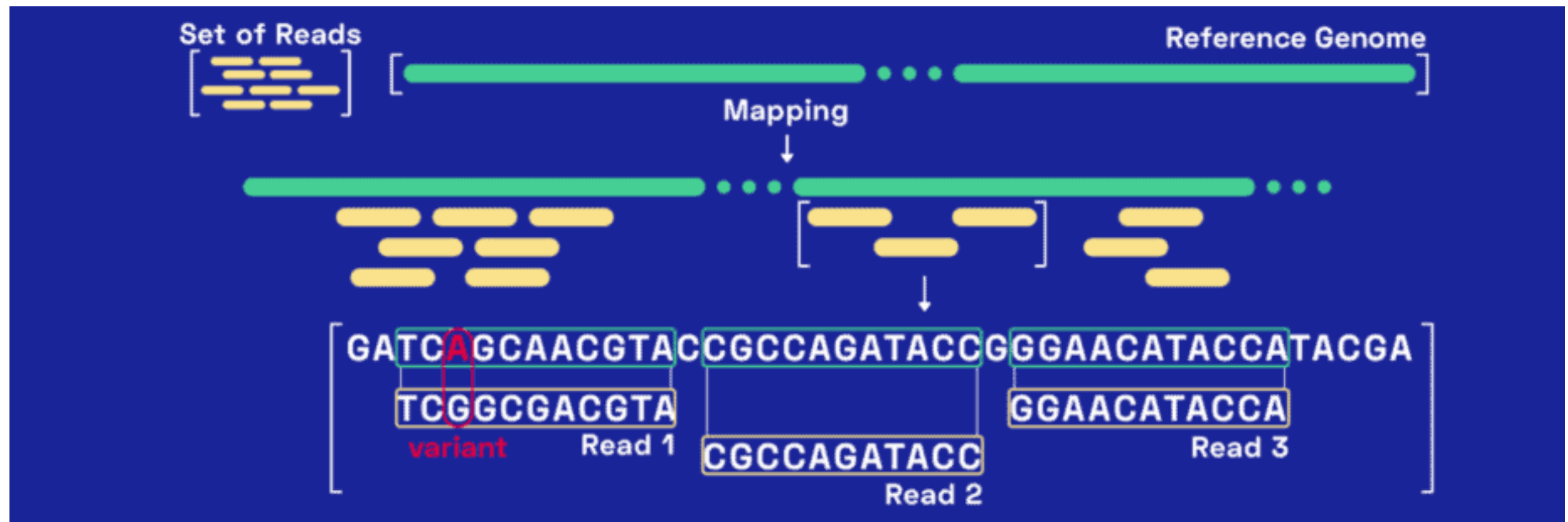
Quality: ASCII chars

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



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

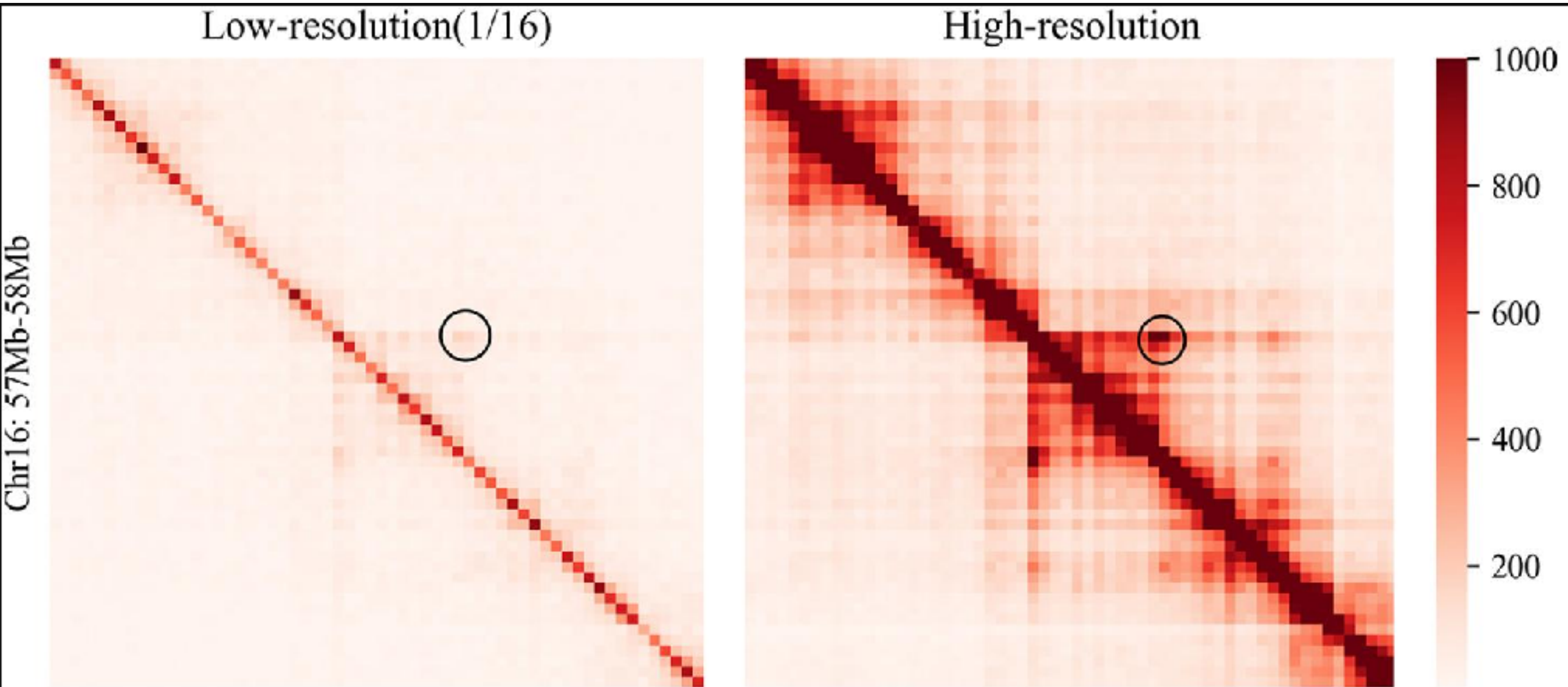
colData

	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

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	←Y
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	

↑X