CMSC 701: Computational Genomics

Spring 2024

UNIVERSITY OF MARYLAND

Course Info

Instructor: Rob Patro (rob@cs.umd.edu)

Office: 3220 IRB

Office Hours: By appointment (please include the string “[CMSC701_S24]” in your e-mail)

Website: https://umd-cmsc701.github.io/s2024

Piazza: https://piazza.com/umd/spring2024/cmsc701

Course Info

TA: Noor Pratap Singh

Office Hours: TBD

e-mail: npsingh@umd.edu

Syllabus stuff

Assumed prerequisites

The only restriction to enroll for this course is that you _"Must be a graduate student in CMSC or BISI program." If you do not meet these requirements, then please reach out to me and let me know your interest in enrolling in this course. If you have a sufficient background in the prerequisites, I will be happy to sign the form to help you enroll.

It is worth noting that this is a CMSC PhD qualifying course. Among other things, we will be covering advanced data structures, succinct data structures, and the algorithms used along with them. While I will attempt to cover each topic we discuss as independently as possible, I'll be assuming familiarity with:

- Basic data structures and algorithms (arrays, lists, trees, hashing, divide-and-conquer, dynamic programming, greedy algorithm design, etc.)
- Basic CS theory and asymptotic notation (the word-RAM model, familiarity wih Landau notation — primarily big O, little o, and Theta, and basic understanding of NP-completeness)
- Basic programming skills in a *non-managed* language (e.g. C, C++, Rust). This last requirement isn't a *hard* one, but in general, projects in the class should *at least* be done in a compiled language (e.g. Java or Go are OK, **Python is not**).

If there are aspects of these topics with which you are not comfortable, it is worthwhile to try and improve your background on these topics. Below are listed resources for the general CS background assumed that should cover these topics at a level sufficient for what will be assumed in this course.

Course Objectives

Course Objectives

The main objective of this course will be to provide an understanding of some of the algorithms, data structures, and methods that underlie *modern* computational genomics. This course is intended as a broad introduction to some problems in genomics, as well as a deep dive to the research edge of some (very narrow) aspects of the field. However, this is a huge field, so we will not cover everything, and what we do cover will not all be at the same depth. Our perspective will be a computational and algorithmic one, though we will take the time to understand the necessary biology and motivation for the problems we discuss. At the end of this course, you should have a good understanding of how new challenges in genomics drive algorithmic innovations and how algorithmic innovations enable new and improved biological analyses.

Coursework & Grading

Coursework and grading: The coursework will consist of 2-3 homework projects, a final project, and a final exam. Students will have an opportunity to select their final project in late Feb.; there will be a few projects to choose from, and students will also be allowed to propose their own projects. The projects are to be done, ideally, in teams of 3 (I will allow the project to be done solo with approval, an *may* approve a team of 4 if there is a compelling reason). Further, the grade for the final project will be broken down into components for the interim report, a final project presentation, and the final project delivery itself. For the final project, the final deliverables will consist of runnable code (including a link to a version-controlled repository containing the source), and a short (4-5 page) research-style paper describing the work you've done. The breakdown of weights for these different assignments will be as follows:

- Homeworks - 25%
- Final Project - 50%
  - Interim report 10%
  - Final presentation 10%
  - Final report 30%
- Final Exam — 25%

Late policy: Assignments that are turned in late will be docked 1% for each hour they are late up to the first 48 hours. After 48 hours, late assignments will not be accepted. Each student is allowed *one* free late assignment turnin (you can turn the assignment in up to 48 hours late with no penalty). However, you must let me know that you are using your free late assignment *when you turn in your assignment*, and the decision is non-revocable (if you decide to use the free late assignment for assignment 1, you can't than request to take the late penalty for 1 and use the late assignment for assignment 2).

Regrade policy: All requests to re-grade, re-check, or re-mark an assignment or exam question **must be made in writing**. When the assignment is re-graded, it will be re-checked in its entirety. This means that *it is possible to lose points on other problems if they were graded incorrectly or too leniently the first time*. Therefore, I urge you to thoroughly consider each regrade request you make.

Textbooks

Textbook(s): We will be making use of, in part, a new textbook by Carl Kingsford. A PDF of this text is provided behind authentication via ELMS. As this text is a (copyrighted) work in progress (near-final draft), and since it is kindly being provided to us free of charge for the purpose of instruction and feedback for this course, please do not distribute or share this text via any public medium. Other resources, where relevant, will be provided via links on the course website, accompanying the slides or lecture notes. However, this is a graduate-level course, and you should *absolutely* seek out other sources explaining these topics from different angles, using different notations and examples, etc. Of course, you should reach out to me if you are having trouble understanding a topic in the course and have been unable to become comfortable with it from the lectures and other sources. Here are some (non-required) textbooks that I personally recommend as references for different topics:

Textbooks (continued)

Basics of algorithms and data structures:

This course will assume familiarity with basic algorithms and data structures, though I will attempt to refresh everyone's memory on relevant concepts when we cover them. If you need a refresher on algorithmic basics, I recommend the following resources:

- Algorithms (Dasgupta, Papadimitriou, and Vazirani 2006)
- Algorithm Design (Kleinberg and Tardos 2006)
- Introduction to Algorithms, 3rd edition(Cormen, Leiserson, Rivest and Stein, 2009)

Genomics algorithms, data structures, and statisitcal models:

- Bioinformatics Algorithms: An Active Learning Approach
- Genome Scale Algorithm Design (Mäkinen, Belazzougui, Cunial, Tomescu 2015)
- Biological Sequence Analysis (Durbin, Eddy, Krogh, Mitchinson 1998)

Molecular biology:

We will cover the basic required molecular Biology in the course. However if you're not familiar with basic molecular Biology, there are some useful resources worth reading:

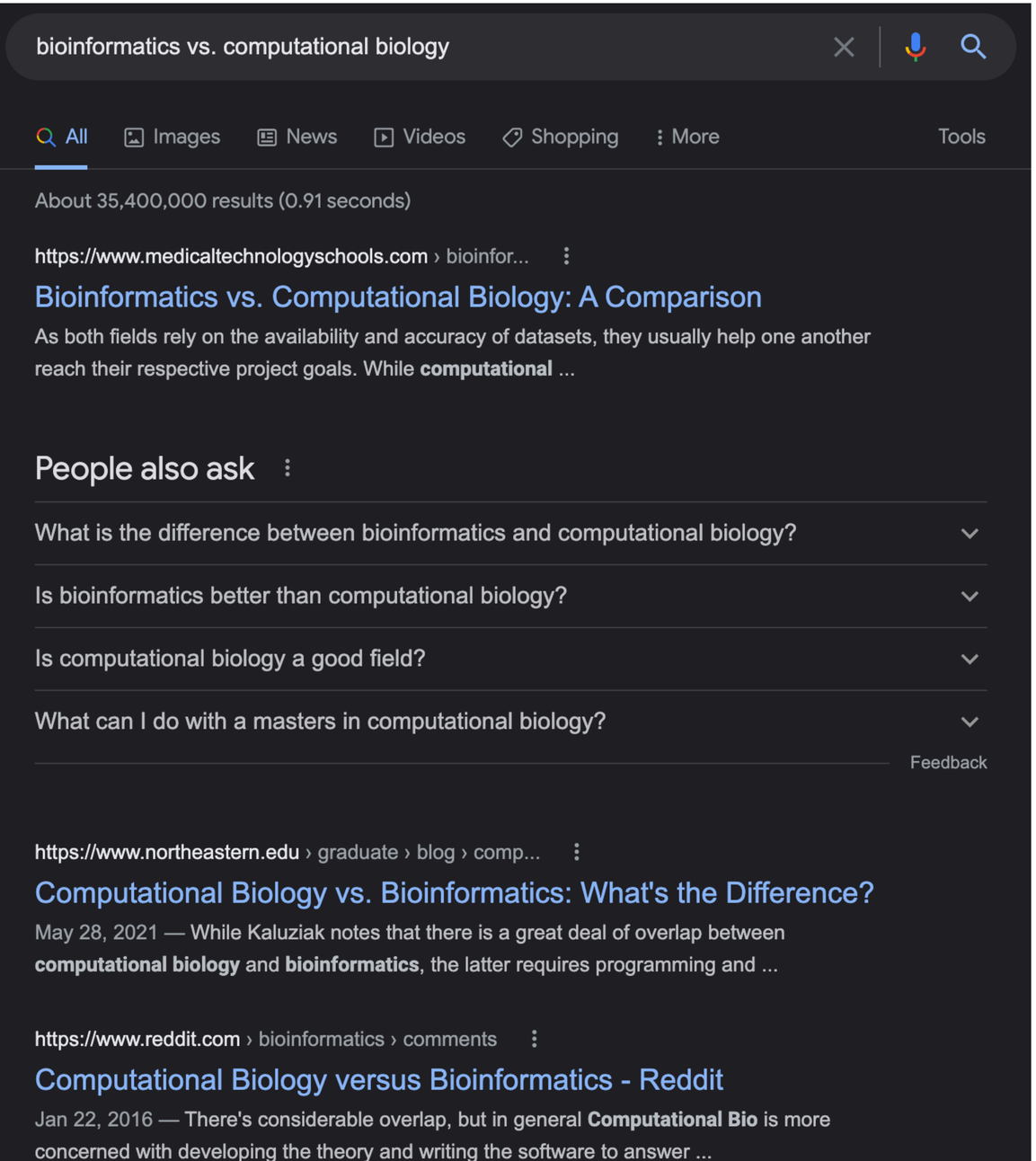
- Molecular Biology of the Cell (Alberts, Johnson, Lewis, Raff, Roberts and Walter, 2002)
- Molecular Biology: Principles of Genome Function 2nd Edition (Craig, Green, Greider, Storz, Wolberger, Cohen-Fix, 2014)
- Molecular Biology (Clark and Pazdernik 2012)

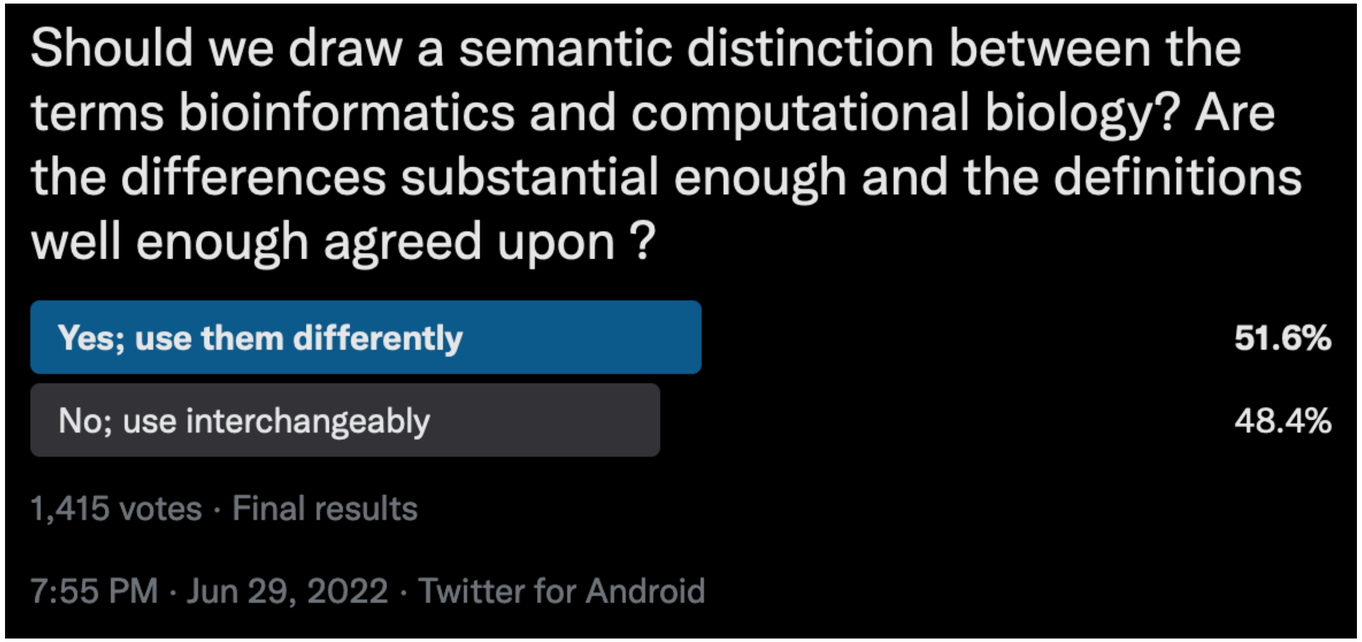
Academic Integrity maintain it!

TLDR : Don't cheat. Don't copy code from friends, classmates, or the internet for the short programming assignments or the projects. Don't provide code to classmates for any of the assignments or projects. Don't cheat on the exams. Be cool, and everything will be cool.

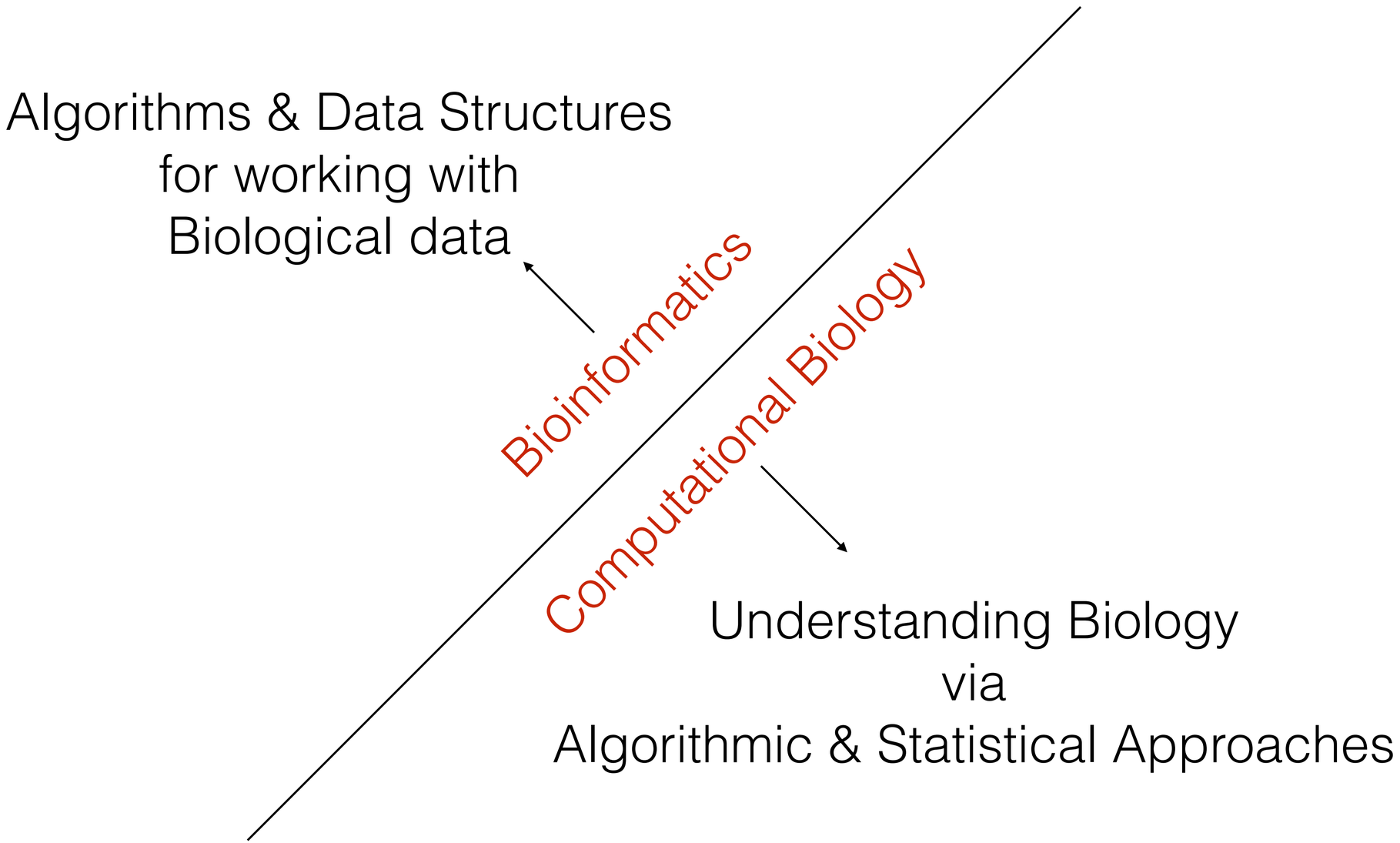
Academic integrity is a very serious issue. Any assignment, project or exam you complete in this course is expected to be your own work. If you are allowed to discuss the details of or work together on an assignment, this will be made explicit. Otherwise, you are expected to complete the work yourself. *Plagarism is not just the outright copying of content*. If you paraphrase someone else's thoughts, words, or ideas and you don't cite your source, this constitues plagraism. It is always much better to turn in an incorrect or incomplete assignment representing your own efforts than to attempt to pass off the work of another as your own. **If you are academically dishonest in this course, you will recieve a grade of XF, and you will be reported to the university's Office of Student Conduct**.

A rose by any other name





Bioinformatics & Computational Biology



Why Genomics?

Our capabilities for *high-throughput* measurement of Biological data has been transformative

1990 - 2000

Sequencing the first human genome took ~10 years and cost ~$2.7 **billion**

Today

Sequencing a genome costs ~$100 - 1,000✢ (depending on how you count)

~18 Tb per “run” at maximum capacity

Progression of sequencing capacity

$3,000,000,000	**2003** Human Genome Project	
$20,000,000	**2006** 1st individual genome	
$2,000,000	**2007** 1st NGS Genome	
$200,000	**2008** 1st 30x genome	
$10,000	**2010** 1st sub-10K genome	
$1,000	**2014** 1st $1,000 genome	
$100	**2017** 1st $100 genome	

https://blog.dnanexus.com/2017-04-24-dnanexus-celebrates-national-dna-day/

Tons of Data, but we need Knowledge

We'll discuss a bit about how sequencing works soon. But the hallmark *limitations* are:

- Short "reads" (75 — 250) characters when the texts we're interested in are 1,000s to 1,000,000,000s of characters long.

- Imperfect "reads" — results in infrequent but considerable "errors"; modifying, inserting or deleting one or more characters in the "read"

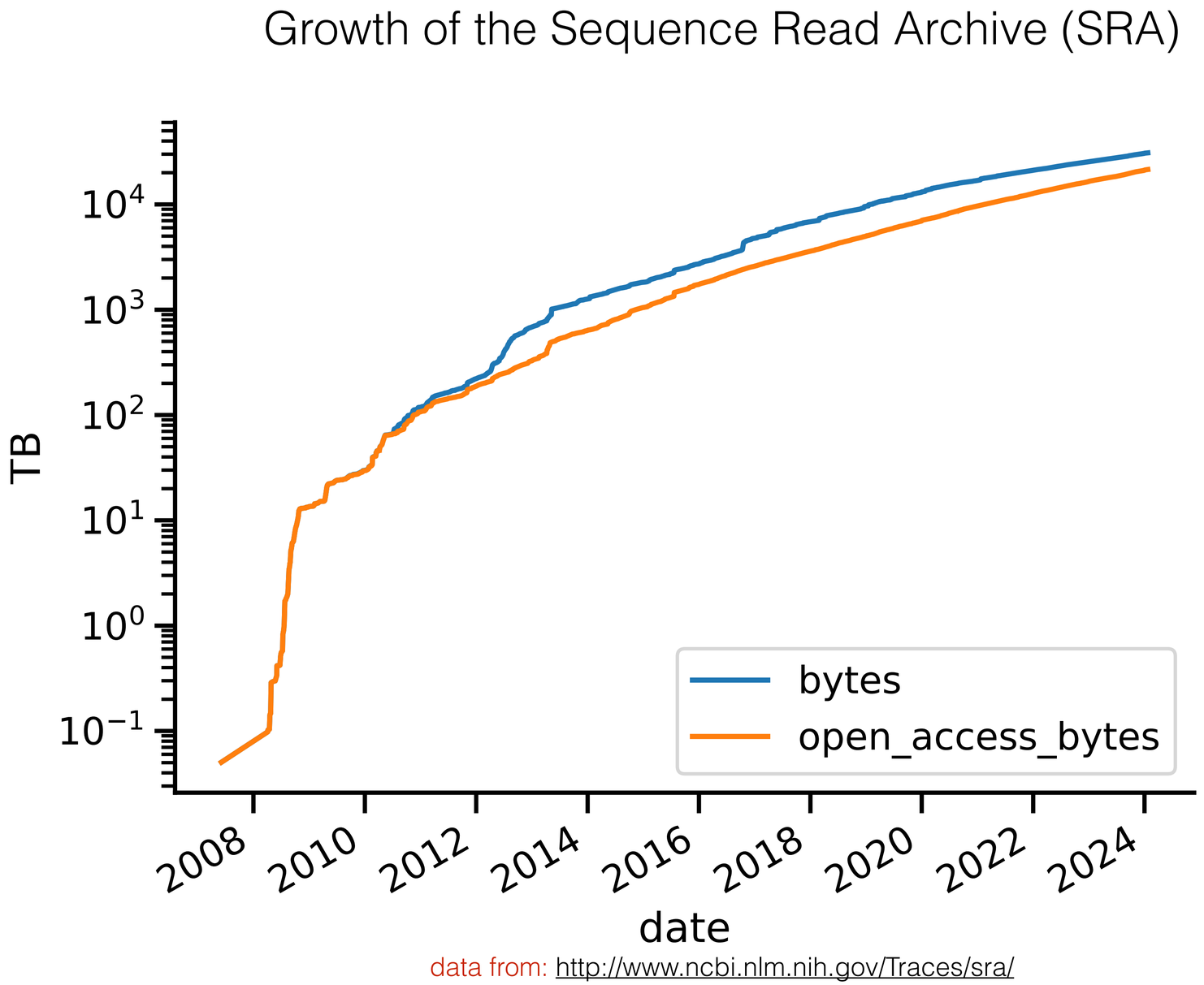
- Biased "reads" — as a result of the underlying chemistry & physics, sampling is not perfectly uniform and random. Biases are not always known.

- Emerging "long read" technologies exist, but have their own set of limitations.

How we get our data (FASTA & FASTQ formats)

How we get our data (FASTA & FASTQ formats)

despite these limitations, scientists have used sequencing at a breakneck pace



data from: http://www.ncbi.nlm.nih.gov/Traces/sra/

Finally…

SPECIAL SECTION COMPLETING THE HUMAN GENOME

RESEARCH ARTICLE

HUMAN GENOMICS

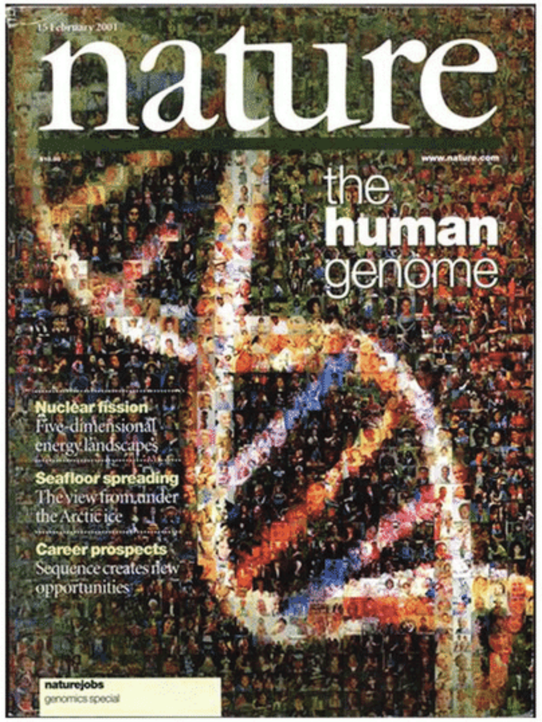
The complete sequence of a human genome

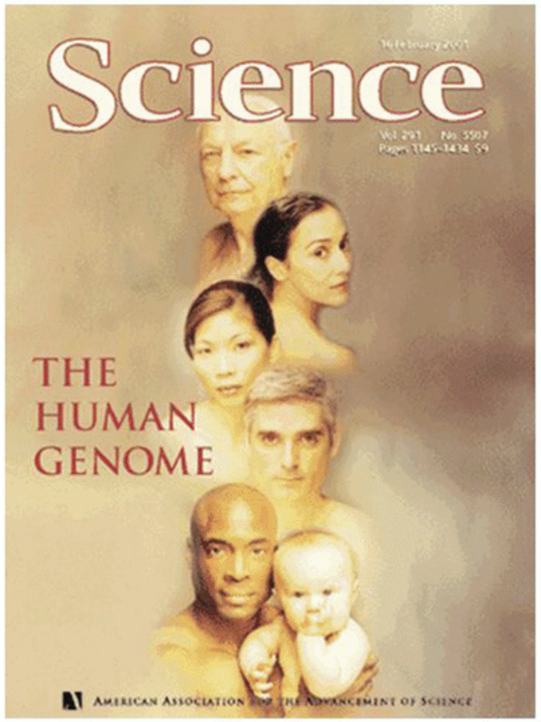
Sergey Nurk[1]†, Sergey Koren[1]†, Arang Rhie[1]†, Mikko Rautiainen[1]†, Andrey V. Bzikadze[2], Alla Mikheenko[3], Mitchell R. Vollger[4], Nicolas Altemose[5], Lev Uralsky[6,7], Ariel Gershman[8], Sergey Aganezov[9]‡, Savannah J. Hoyt[10], Mark Diekhans[11], Glennis A. Logsdon[4], Michael Alonge[9], Stylianos E. Antonarakis[12], Matthew Borchers[13], Gerard G. Bouffard[14], Shelise Y. Brooks[14], Gina V. Caldas[15], Nae-Chyun Chen[9], Haoyu Cheng[16,17], Chen-Shan Chin[18], William Chow[19], Leonardo G. de Lima[13], Philip C. Dishuck[4], Richard Durbin[19,20], Tatiana Dvorkina[3], Ian T. Fiddes[21], Giulio Formenti[22,23], Robert S. Fulton[24], Arkarachai Fungtammasan[18], Erik Garrison[11,25], Patrick G. S. Grady[10], Tina A. Graves-Lindsay[26], Ira M. Hall[27], Nancy F. Hansen[28], Gabrielle A. Hartley[10], Marina Haukness[11], Kerstin Howe[19], Michael W. Hunkapiller[29], Chirag Jain[1,30], Miten Jain[11], Erich D. Jarvis[22,23], Peter Kerpedjiev[31], Melanie Kirsche[9], Mikhail Kolmogorov[32], Jonas Korlach[29], Milinn Kremitzki[26], Heng Li[16,17], Valerie V. Maduro[33], Tobias Marschall[34], Ann M. McCartney[1], Jennifer McDaniel[35], Danny E. Miller[4,36], James C. Mullikin[14,28], Eugene W. Myers[37], Nathan D. Olson[35], Benedict Paten[11], Paul Peluso[29], Pavel A. Pevzner[32], David Porubsky[4], Tamara Potapova[13], Evgeny I. Rogaev[6,7,38,39], Jeffrey A. Rosenfeld[40], Steven L. Salzberg[9,41], Valerie A. Schneider[42], Fritz J. Sedlazeck[43], Kishwar Shafin[11], Colin J. Shew[44], Alaina Shumate[41], Ying Sims[19], Arian F. A. Smit[45], Daniela C. Soto[44], Ivan Sović[29,46], Jessica M. Storer[45], Aaron Streets[5,47], Beth A. Sullivan[48], Françoise Thibaud-Nissen[42], James Torrance[19], Justin Wagner[35], Brian P. Walenz[1], Aaron Wenger[29], Jonathan M. D. Wood[19], Chunlin Xiao[42], Stephanie M. Yan[49], Alice C. Young[14], Samantha Zarate[9], Urvashi Surti[50], Rajiv C. McCoy[49], Megan Y. Dennis[44], Ivan A. Alexandrov[3,7,51], Jennifer L. Gerton[13,52], Rachel J. O'Neill[10], Winston Timp[8,41], Justin M. Zook[35], Michael C. Schatz[9,49], Evan E. Eichler[4,53]*, Karen H. Miga[11,54]*, Adam M. Phillippy[1]*

Since its initial release in 2000, the human reference genome has covered only the euchromatic fraction of the genome, leaving important heterochromatic regions unfinished. Addressing the remaining 8% of the genome, the Telomere-to-Telomere (T2T) Consortium presents a complete 3.055 billion–base pair sequence of a human genome, T2T-CHM13, that includes gapless assemblies for all chromosomes except Y, corrects errors in the prior references, and introduces nearly 200 million base pairs of sequence containing 1956 gene predictions, 99 of which are predicted to be protein coding. The completed regions include all centromeric satellite arrays, recent segmental duplications, and the short arms of all five acrocentric chromosomes, unlocking these complex regions of the genome to variational and functional studies.

Actually Completing the Human Genome

2001





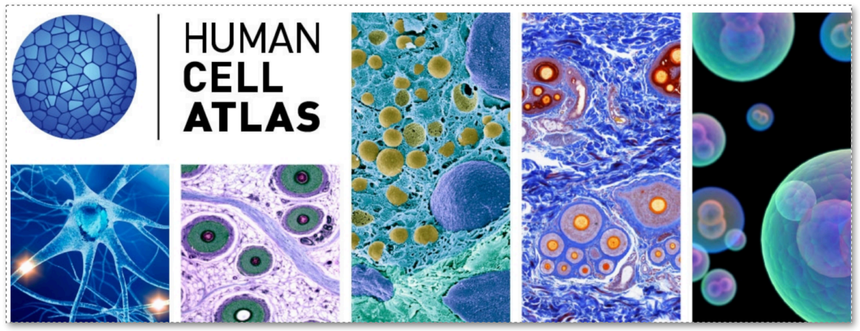
algorithmic
&
biotech
progress

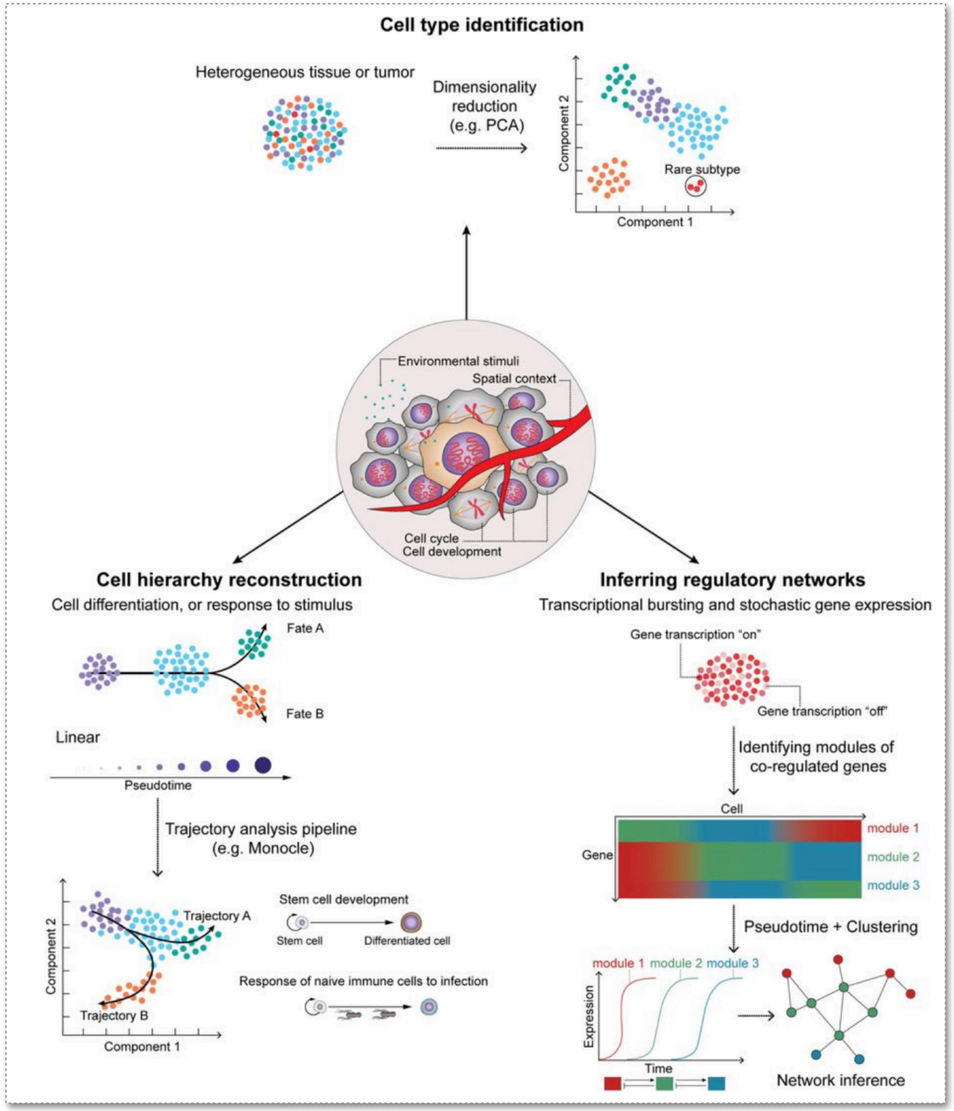
2022

Single-cell sequencing – cell-level profiling

Explore biology at *unprecedented resolution*.
Some *transformative* applications:

- Study *treatment-resistant* cells in disease (cancer)
- Understand tissue / organism *development* (cell fate)
- Understand immune response at the cellular level
- Understand dynamic cellular processes
- Learn how expression is regulated (regulatory net.)
- Characterize new cell types & cell states (HCA)





Hwang, B., Lee, J. H., & Bang, D. (2018). Single-cell RNA sequencing technologies and bioinformatics pipelines. *Experimental & molecular medicine*, *50*(8), 96.
https://www.broadinstitute.org/research-highlights-human-cell-atlas

Answer questions “in the large”

What is the genome of the terrapin? (genomics)

Which genes are expressed in healthy vs. diseased tissue? (transcriptomics)

How do environment changes affect the microbial ecosystem of the Chesapeake bay? (metagenomics)

How do genome changes lead to changes & diversity in a population? (population genetics/genomics)

How related are two species if we look at their whole genomes? (phylogenetics / phylogenomics)

Some Computational Challenges

Answering questions on such a scale becomes a *fundamentally* computational endeavor:

Assembly — Find a likely "super string" that parsimoniously explains 200M short sub-strings (string processing, graph theory)

Alignment — Find an *approximate* match for 50M short string in a 5GB corpus of text (string processing, data structure & algorithm design)

Expression / Abundance Estimation — Find the most probable mixture of genes / microbes that explain the results of a sequencing experiment (statistics & ML)

Phylogenomics — Given a set of related gene sequences, and an assumed model of sequence evolution, determine how these sequences are related to each other (statistics & ML)

CS & Biology: “Scientific” differences

Biology deals with *very* complex natural systems that arise through evolution

Biological systems can be indirect, redundant and counterintuitive

Nothing is “always” true/false — Biological laws are not like Physical or Mathematical laws; more stochastic truths or rules of thumb.

Biological laws *are* a result of Physical laws, but treating them that way is computationally infeasible

Try to understand mechanisms by probing and measuring complex systems and obtaining (often noisy) measurements

Experiments often *very* expensive

CS & Biology: “Scientific” differences

Computer Science deals with *less* complex (won’t say simple) systems that arise through design

CS is more about invention than discovery (philosophy aside)

Things are always formally true or false in CS & detailed theoretical analysis allows precise description

Computational outcomes *are* a result of mathematical laws & effective algorithms often have an intuitive explanation

Some subfields of CS (e.g. network measurement) do bear a resemblance to the natural sciences — many are much closer to math.

Experiments often dirt cheap and easy to re-run

“Cultural” differences

Biology	CS
Only journals matter	Selective conferences often preferred to journals
Larger labs: PI → postdoc→grad students	Smaller labs: PI → a few grad students
Student may study a specific gene for their entire PhD	Students typically work on a wide variety of projects in PhD
Focus on being “right” and discovering something interesting about the natural world. (focus on knowledge)	More weight given to being “different”. Need not be 1st often just be “best”, fastest or simplest. (focus on methods)

Many of these differences start to vanish at the interface of data-intensive Biology, where computational savvy is a necessity.

Immense Spatial & Time Scales

The scale, in both space and time, of the Biological systems we're interested in studying are **truly expansive**.

Time:

Protein folding can happen on the order of microseconds

Evolution works over the span of hundreds, thousands and tens of thousands of years

Space:

A cell nucleus is measured in micrometers

Population migrations happen over tens of thousands of miles

Computational Biology encompasses the study of all of these problems.

Establishing a basic lexicon

This course will focus on algorithms and data structures (with some probability and statistics), but the ***problems*** we will explore derive directly from biological questions.

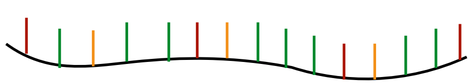
In order to motivate our Computer Science, we will need a basic understanding of the Molecular Biology in which the problems are phrased.

“Flow” of information in the cell

DNA

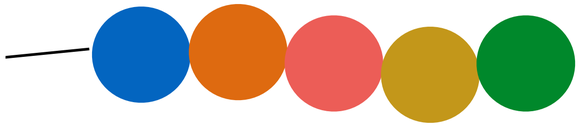
RNA Polymerase
(transcription)

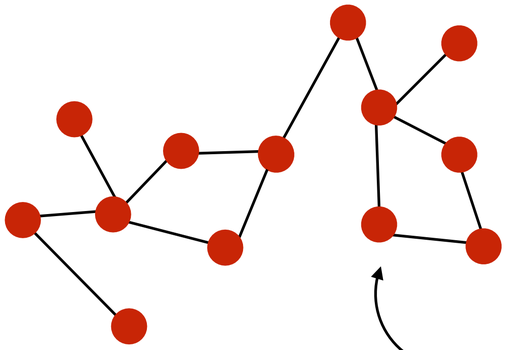
RNA



Ribosomes
(translation)

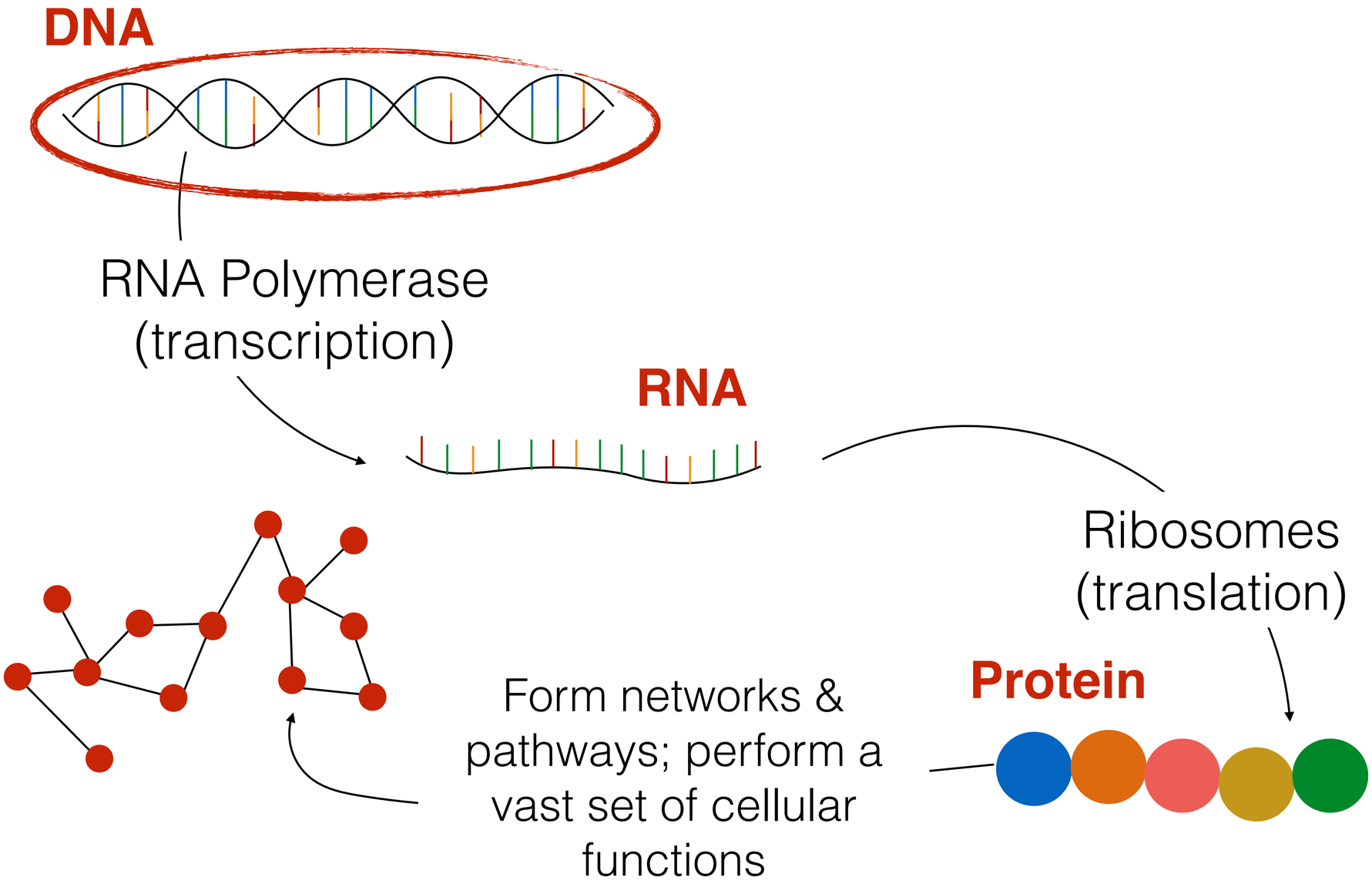
Protein



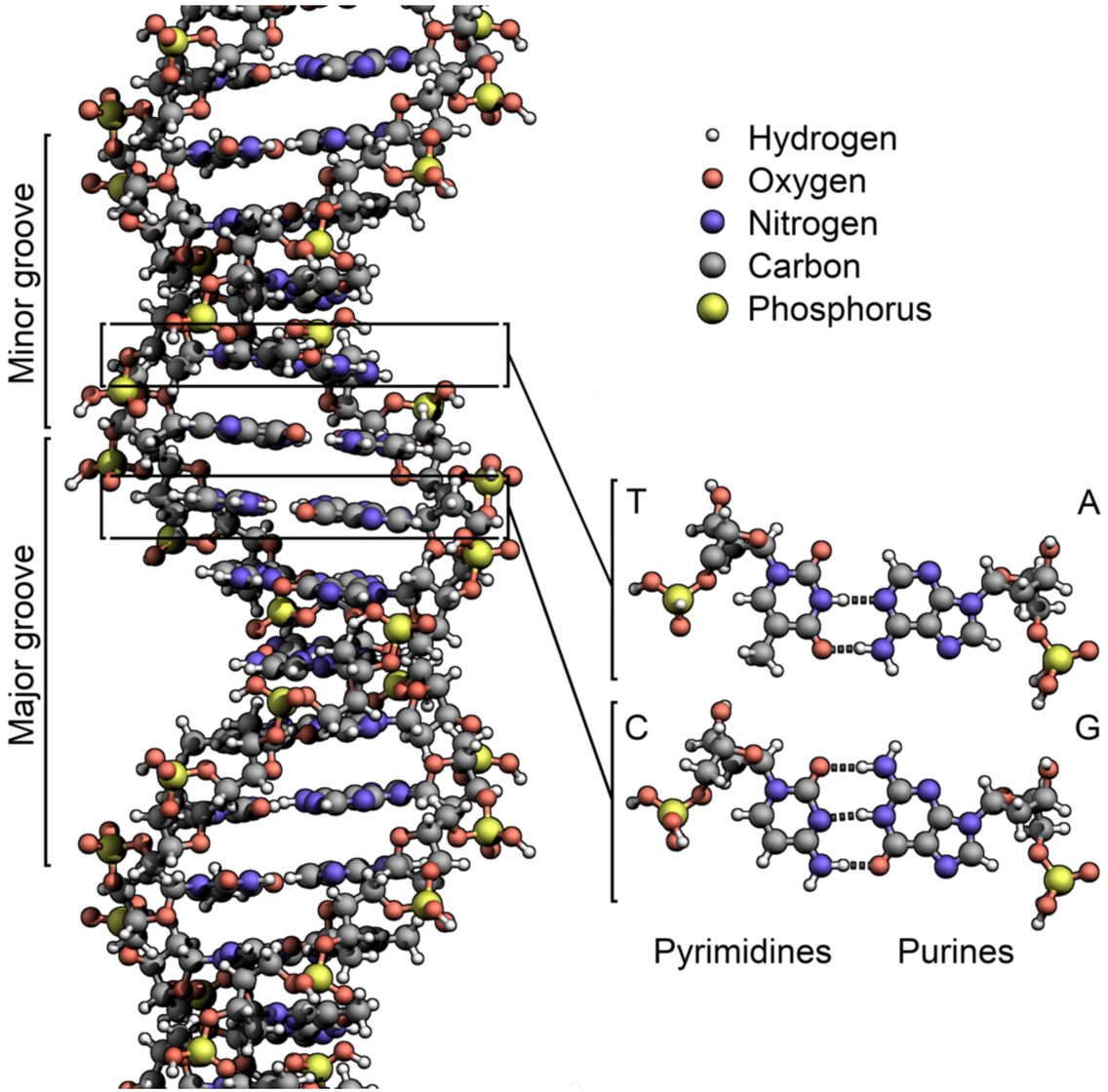


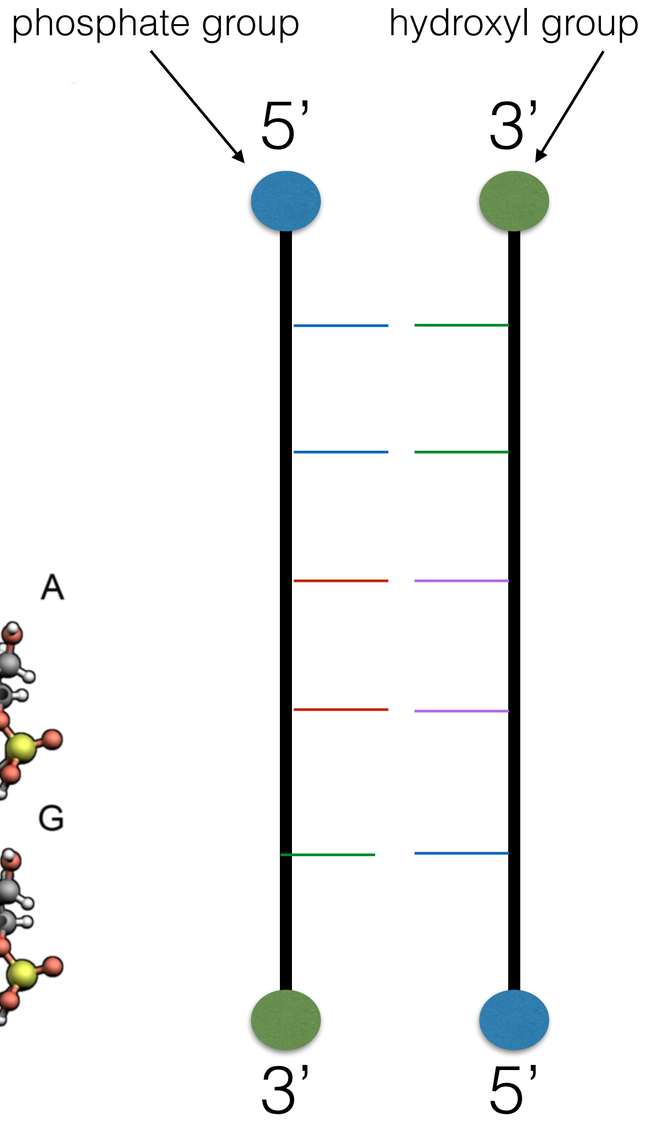
Form networks &
pathways; perform a
vast set of cellular
functions

“Flow” of information in the cell

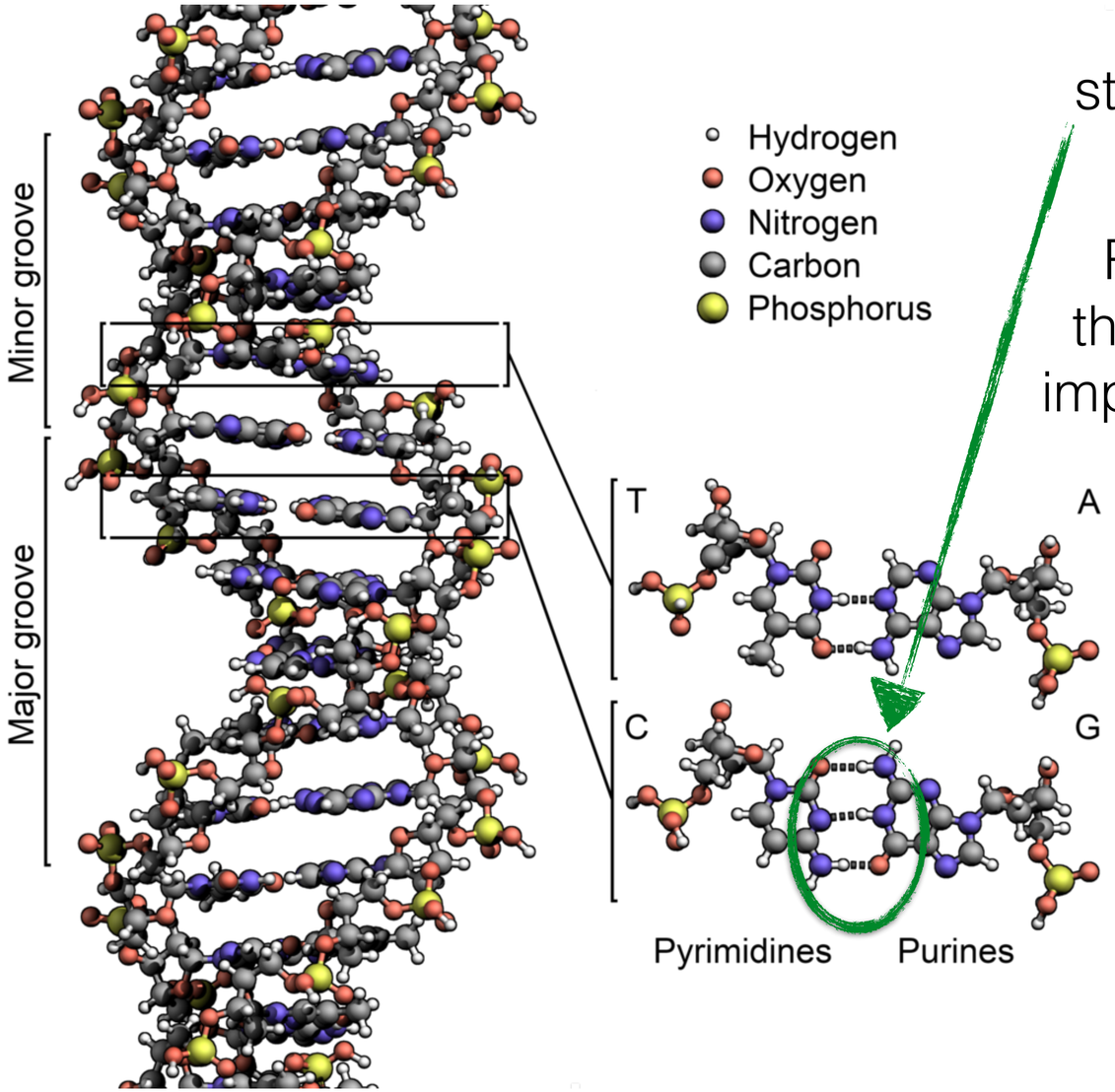


DNA (the genome)





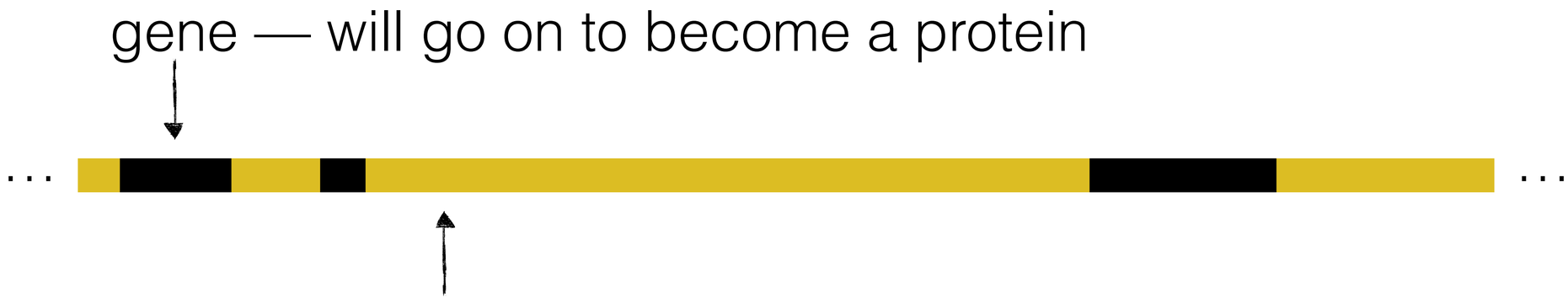
DNA (the genome)



G-C pairing generally stronger than A-T pairing

Ratio of G+C bases — the “GC content” — is an important sequence feature

DNA (the genome)



In humans, most DNA is “non-coding” ~98%

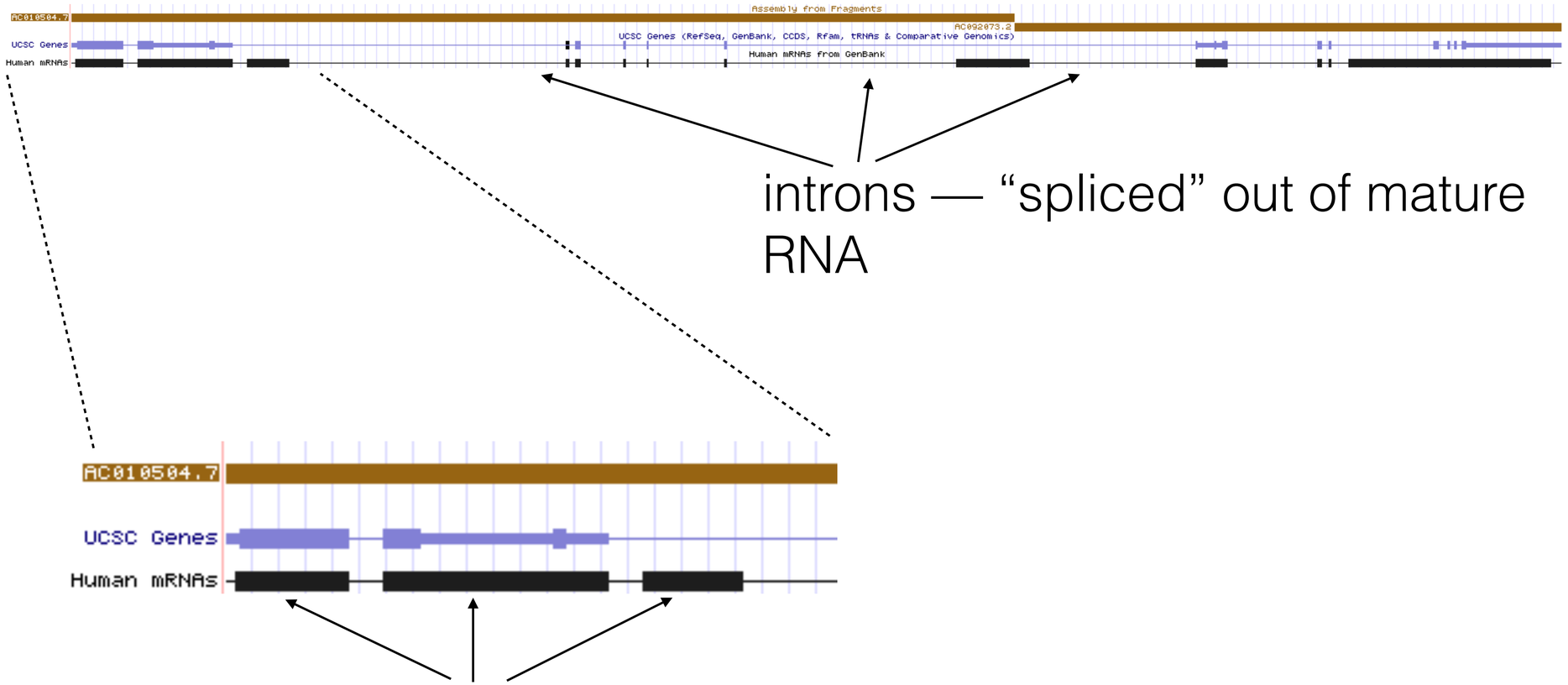
In typical bacterial genome, only small fraction — ~2% — of DNA is “non-coding”

Sometimes referred to as “junk” DNA — much is not, in any way, “junk”

DNA (the genome)

In **prokaryotes**, genes are typically contiguous DNA segment

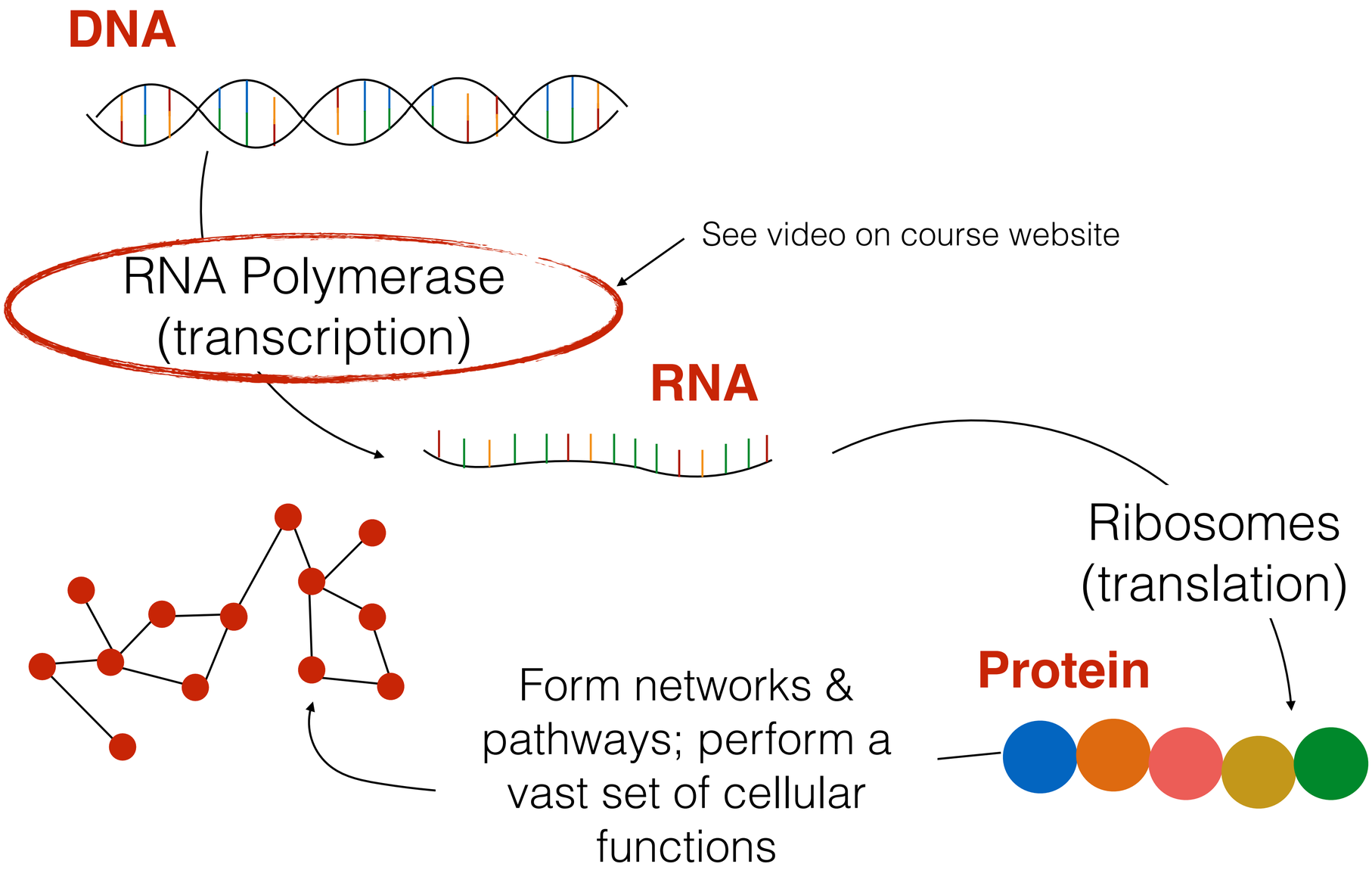
In **eukaryotes**, genes can have complex structure



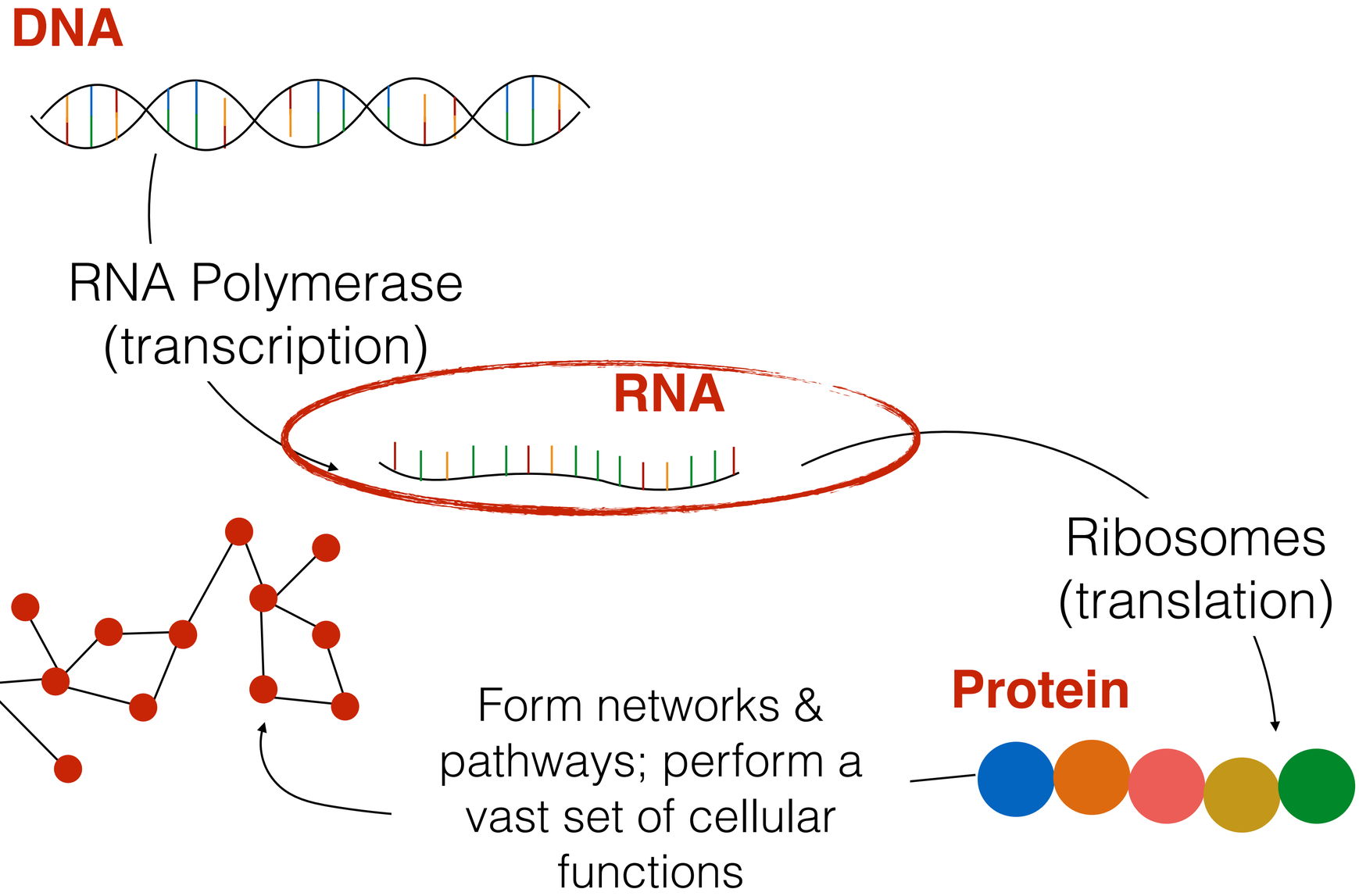
introns — “spliced” out of mature RNA

exons — appear in the *mature* RNA transcript

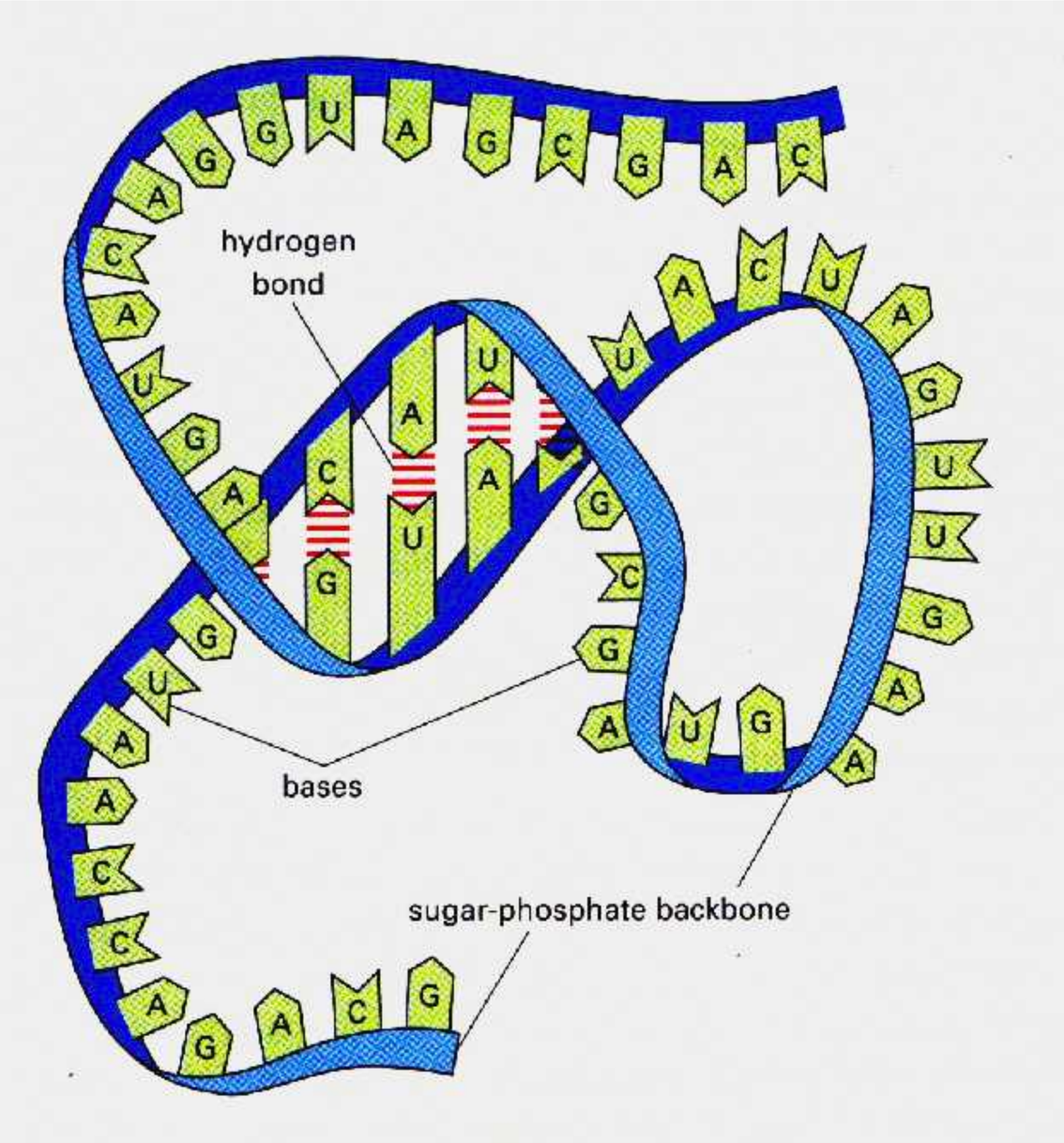
“Flow” of information in the cell



“Flow” of information in the cell



RNA



Less regular structure than DNA

Generally a single-stranded molecule

Secondary & tertiary structure can affect function

Act as transcripts for protein, but <u>also perform important functions themselves</u>

Same “alphabet” as DNA, except thymine replaced by uracil

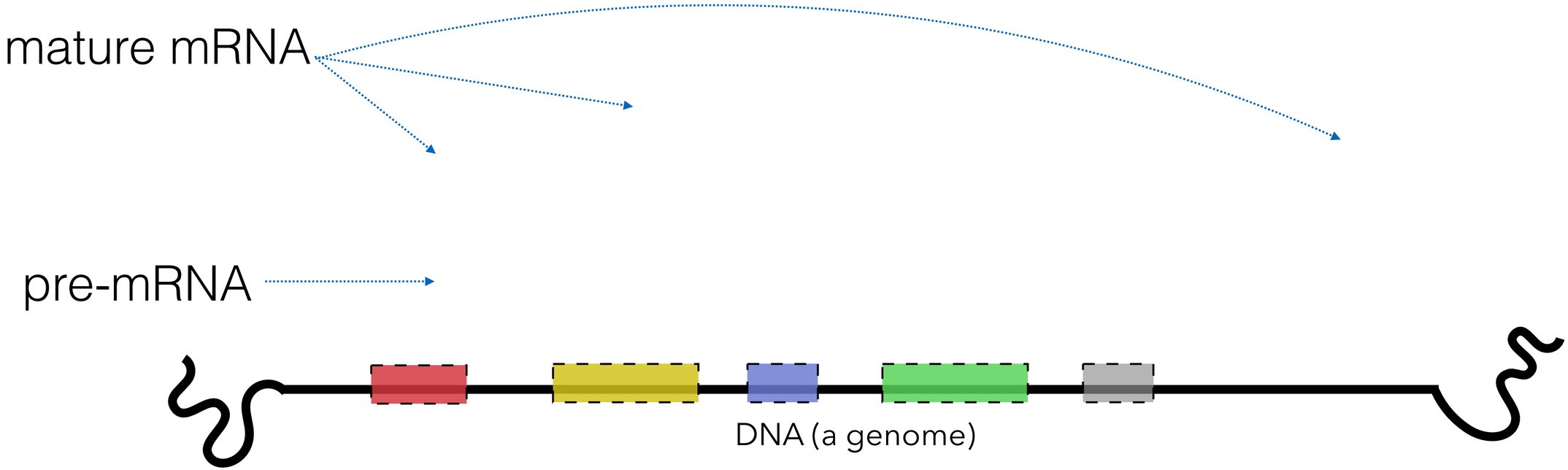
http://tigger.uic.edu/classes/phys/phys461/phys450/ANJUM04/RNA_sstrand.jpg

RNA Splicing

DNA transcribed into pre-mRNA

Some “processing” occurs **capping** & **polyadenylation**

Introns removed from pre-mRNA resulting in mature mRNA

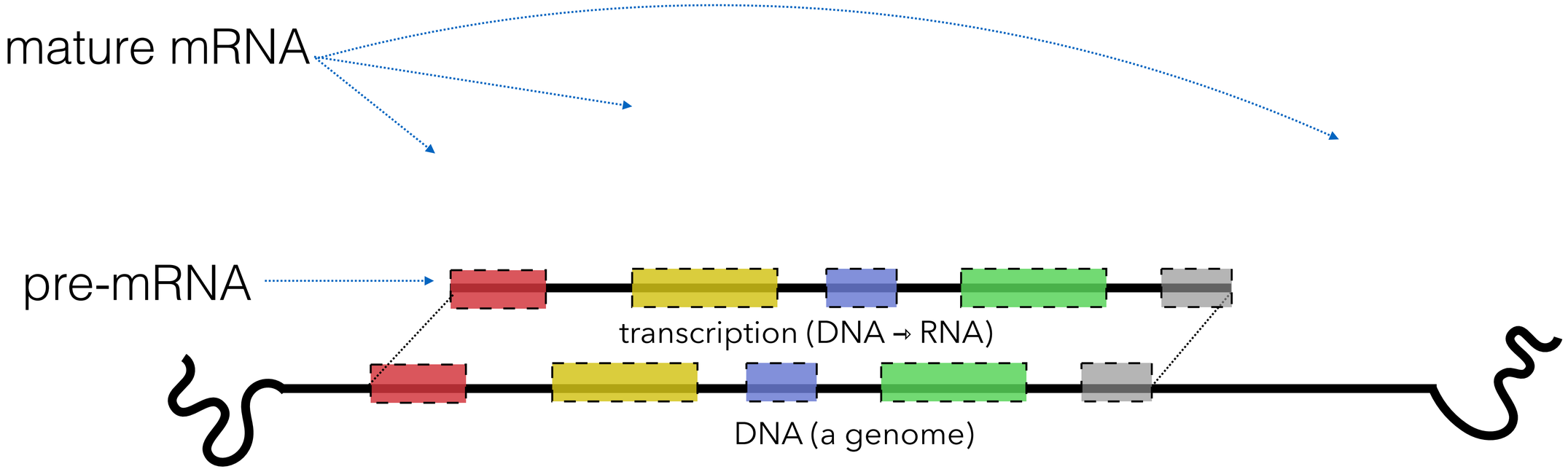


RNA Splicing

DNA transcribed into pre-mRNA

Some “processing” occurs **capping** & **polyadenylation**

Introns removed from pre-mRNA resulting in mature mRNA

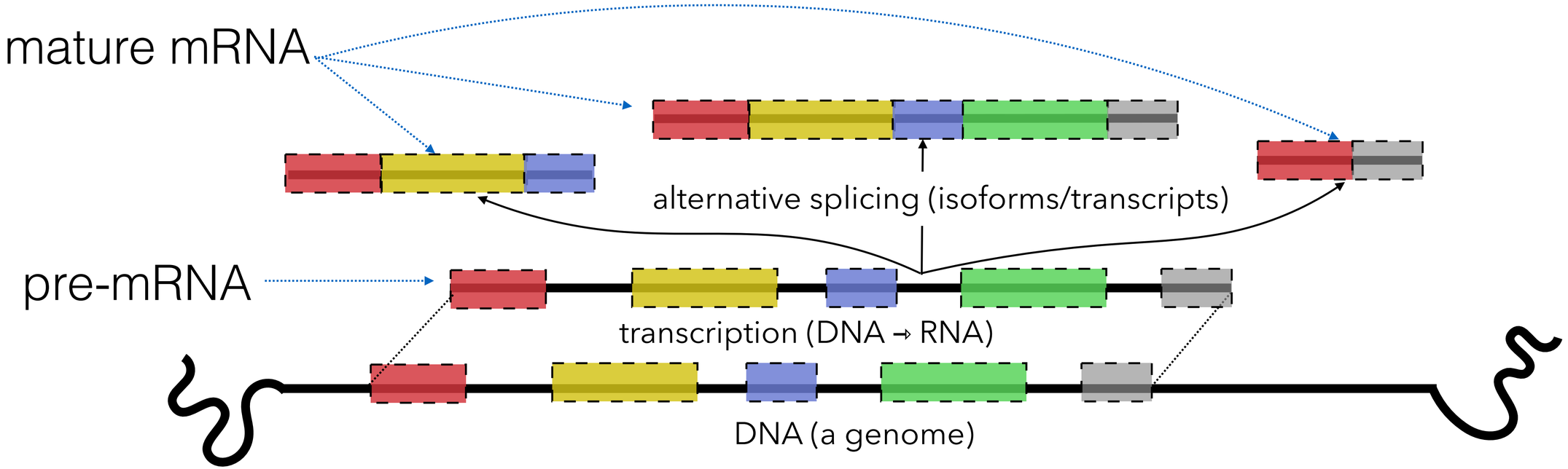


RNA Splicing

DNA transcribed into pre-mRNA

Some “processing” occurs **capping** & **polyadenylation**

Introns removed from pre-mRNA resulting in mature mRNA



“Flow” of information in the cell

DNA

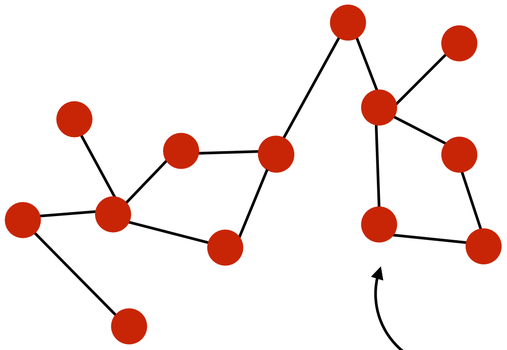
RNA Polymerase
(transcription)

RNA

See video on course website

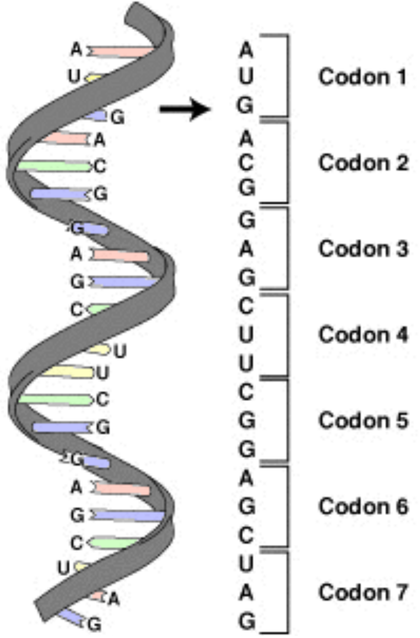
Ribosomes
(translation)

Protein



Form networks &
pathways; perform a
vast set of cellular
functions

Protein



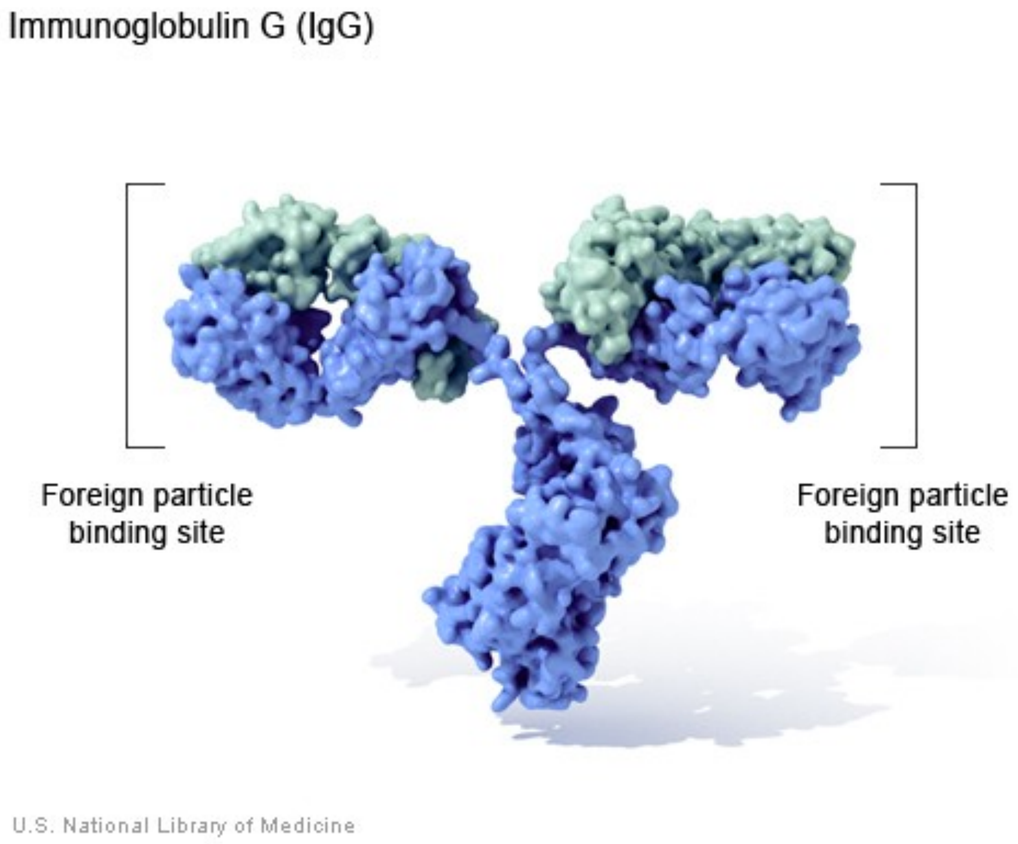
Triplets of mRNA bases (codons) correspond to specific amino acids

This mapping is known as the “genetic code” — an *almost* law of molecular Biology

Inverse table (compressed using IUPAC notation)

Amino acid	Codons	Compressed	Amino acid	Codons	Compressed
Ala/A	GCU, GCC, GCA, GCG	GCN	**Leu/L**	UUA, UUG, CUU, CUC, CUA, CUG	YUR, CUN
Arg/R	CGU, CGC, CGA, CGG, AGA, AGG	CGN, MGR	**Lys/K**	AAA, AAG	AAR
Asn/N	AAU, AAC	AAY	**Met/M**	AUG	
Asp/D	GAU, GAC	GAY	**Phe/F**	UUU, UUC	UUY
Cys/C	UGU, UGC	UGY	**Pro/P**	CCU, CCC, CCA, CCG	CCN
Gln/Q	CAA, CAG	CAR	**Ser/S**	UCU, UCC, UCA, UCG, AGU, AGC	UCN, AGY
Glu/E	GAA, GAG	GAR	**Thr/T**	ACU, ACC, ACA, ACG	ACN
Gly/G	GGU, GGC, GGA, GGG	GGN	**Trp/W**	UGG	
His/H	CAU, CAC	CAY	**Tyr/Y**	UAU, UAC	UAY
Ile/I	AUU, AUC, AUA	AUH	**Val/V**	GUU, GUC, GUA, GUG	GUN
START	AUG		**STOP**	UAA, UGA, UAG	UAR, URA

Protein



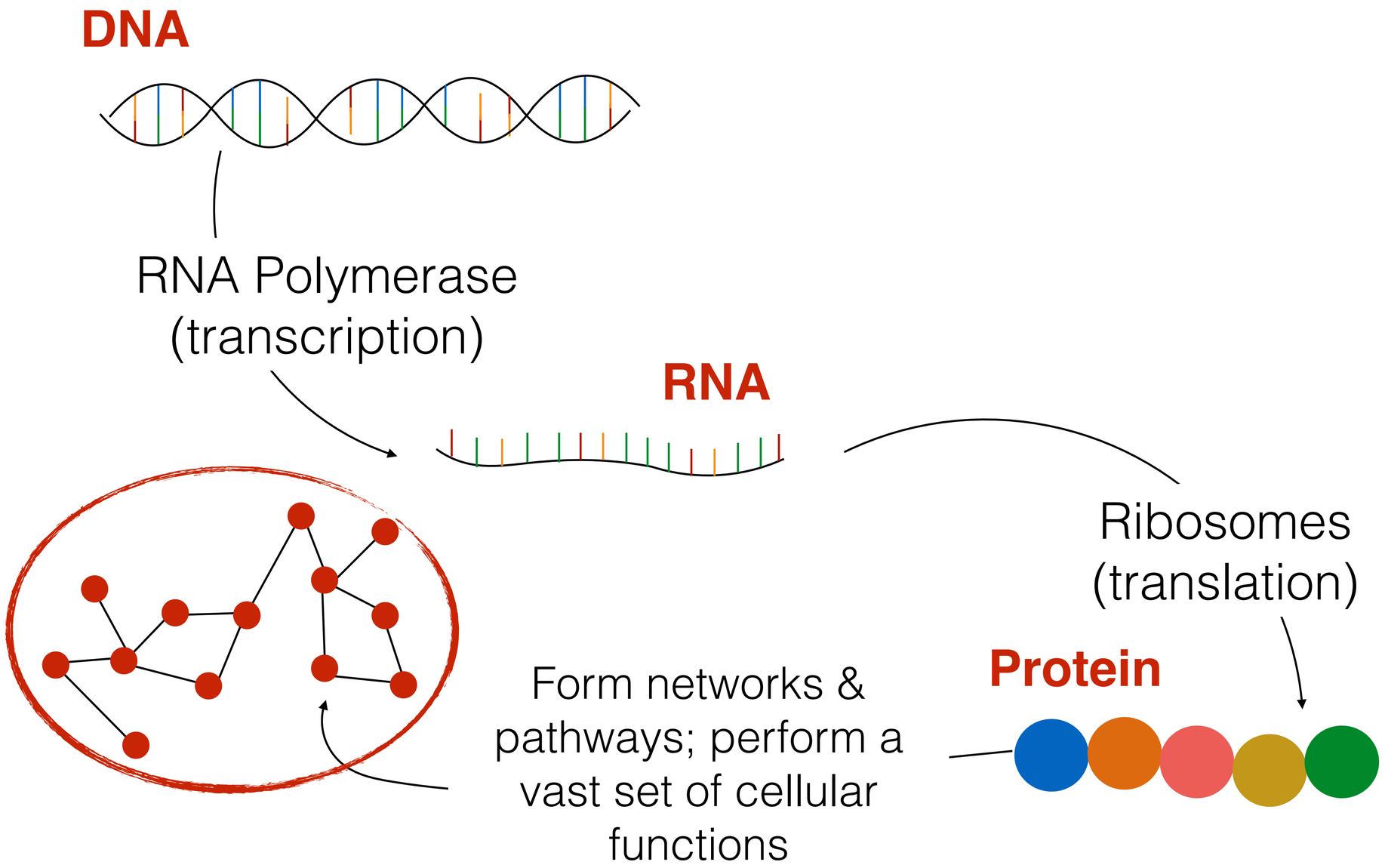
Perform vast majority of intra & extra cellular functions

Can range from a few amino acids to *very* large and complex molecules

Can bind with other proteins to form protein complexes

The shape or *conformation* of a protein is intimately tied to its function. Protein shape, therefore, is strongly conserved through evolution — even more so than sequence. A protein can undergo sequence mutations, but fold into the same or a similar shape and still perform the same function.

“Flow” of information in the cell



One way in which this “central dogma” is violated … retroviruses

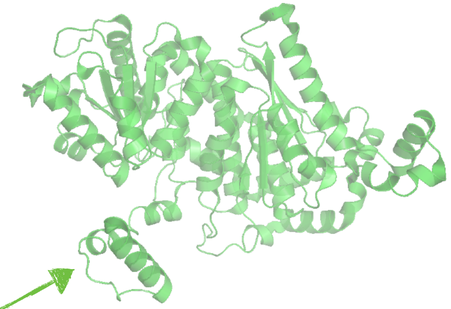
Glycolysis Pathway

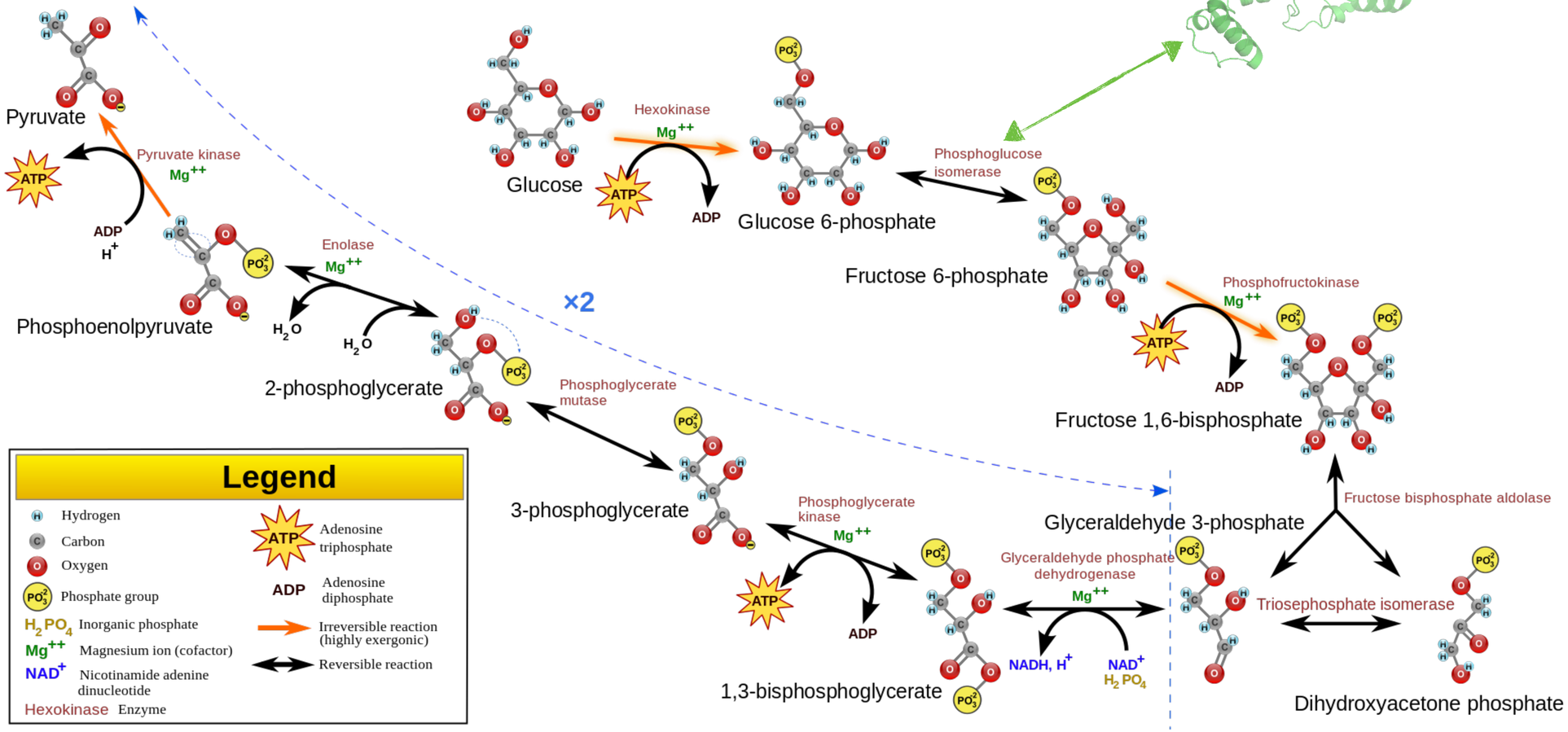
Converts glucose → pyruvate

Generates ATP (“energy currency” of the cell)

this is an **example**, no need to memorize this Bio.

phosphoglucose isomerase





Some Interesting Facts

Organism	Genome size	# of genes
ϕX174 (*E. coli* virus)	~5kb	11
E. coli K-12	~4.6Mb	~4,300
Fruit Fly	~122Mb	~17,000
Human	~3.3Gb	~21,000
Mouse	~2.8Gb	~23,000
P. abies (a spruce tree)	~19.6Gb	~28,000

No strong link between genome size & phenotypic complexity

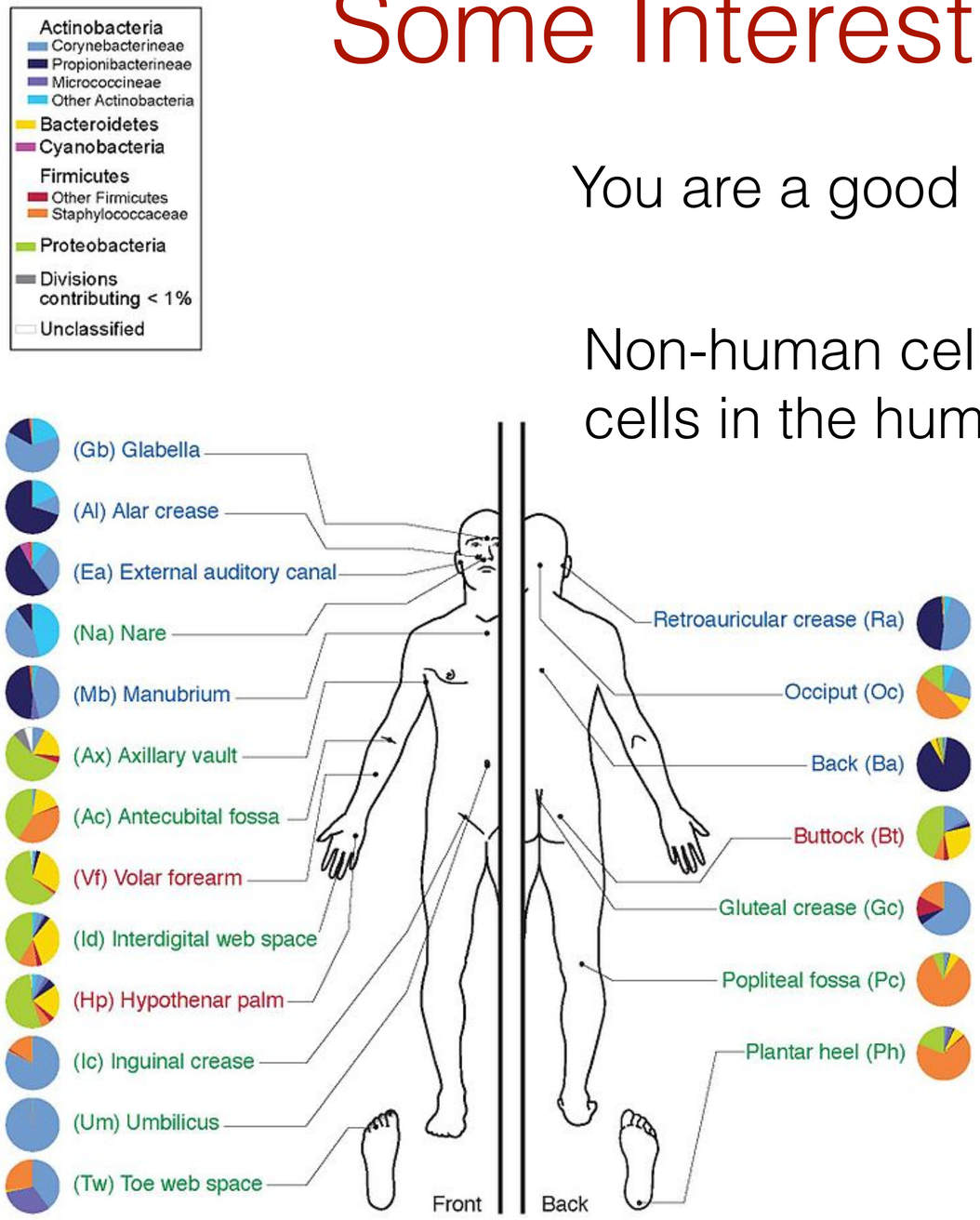
Plants can have **huge** genomes (adapt to environment while stationary!)

http://users.rcn.com/jkimball.ma.ultranet/BiologyPages/G/GenomeSizes.html

Some Interesting Facts

You are a good part non-human cells (e.g. bacteria)

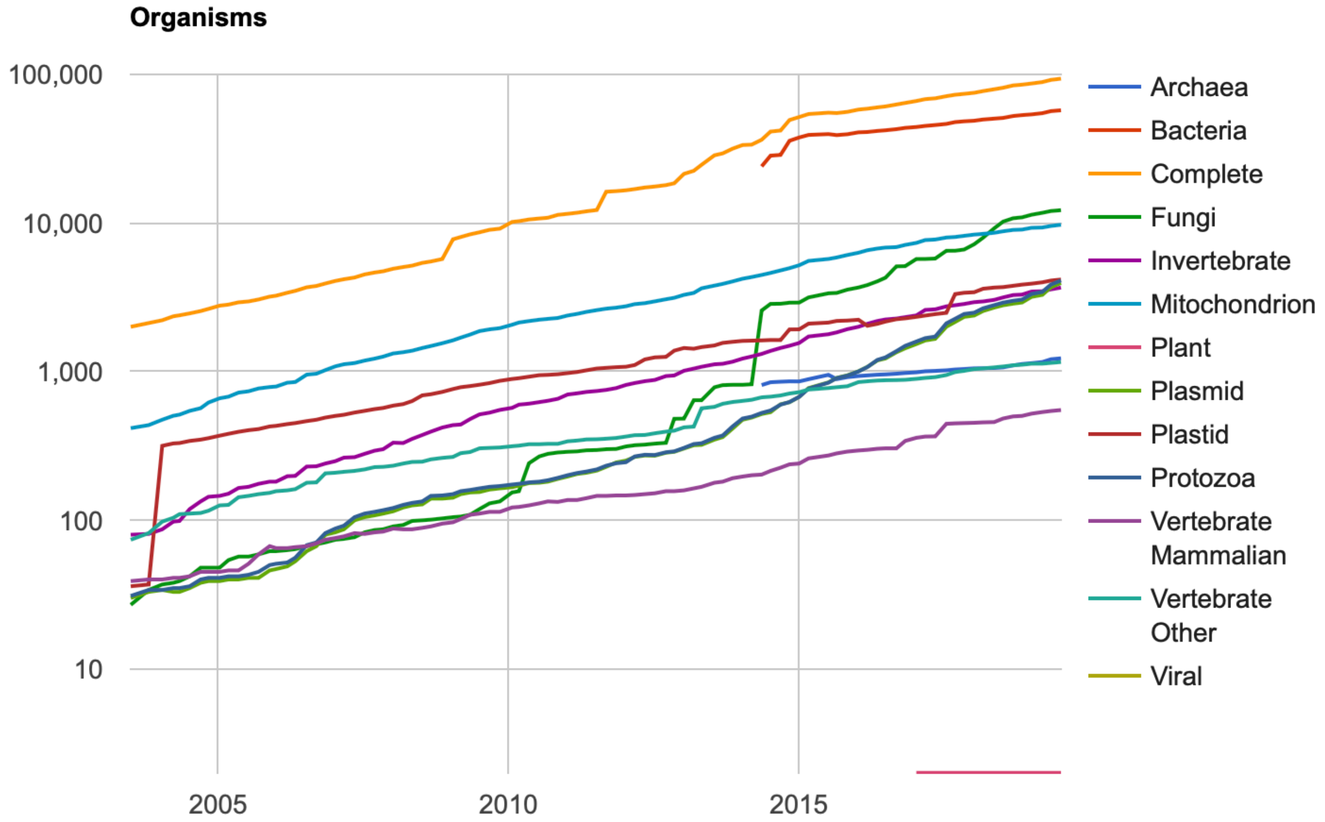
Non-human cells equal or outnumber human cells in the human body



This population of organisms is called the microbiome

en.wikipedia.org : public domain

Some Interesting Facts



https://www.ncbi.nlm.nih.gov/refseq/statistics/

. . . Out of 8.7 ± 1.3 Mil*

Vast majority of species unsequenced & *can not be cultivated in a lab* (one of the many motivations for metagenomics)

*Mora, Camilo, et al. "How many species are there on Earth and in the ocean?." PLoS biology 9.8 (2011): e1001127.