Bioinformatics

CS300

Crash course:

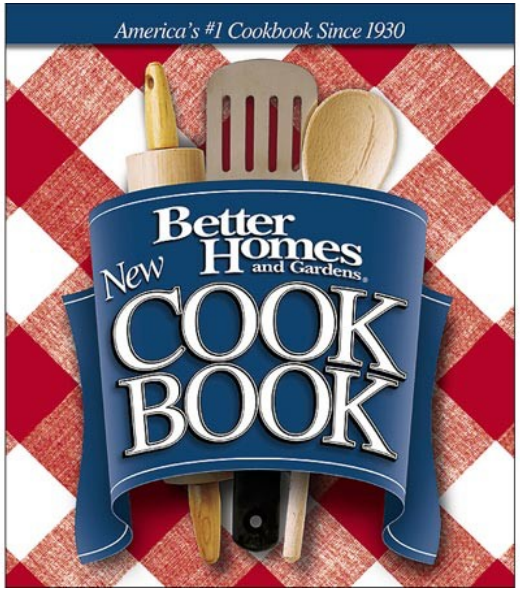
Transcription and Translation

Running Python in Docker or Online

Fall 2019

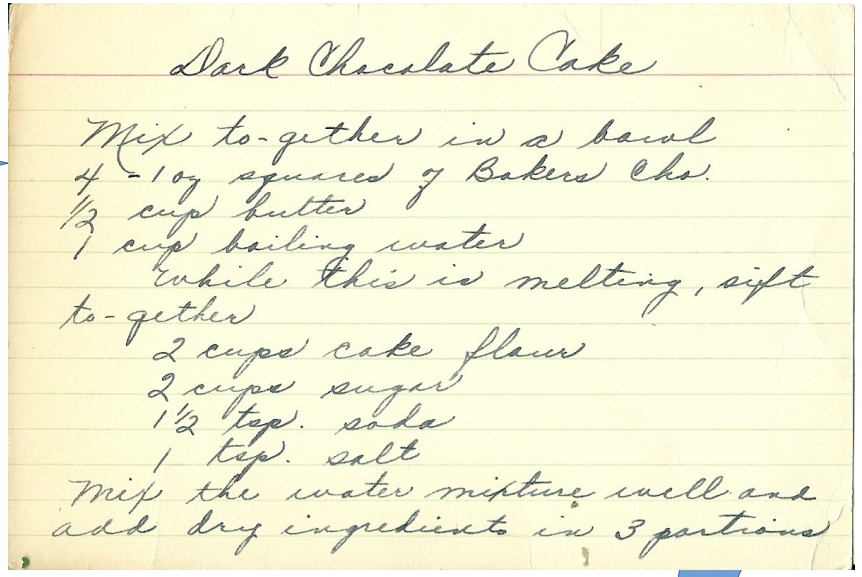
Oliver BONHAM-CARTER

Gene Expression
Transcription and Translation



Transcription

copy a set of ingredients/instructions from a cookbook to create a recipe



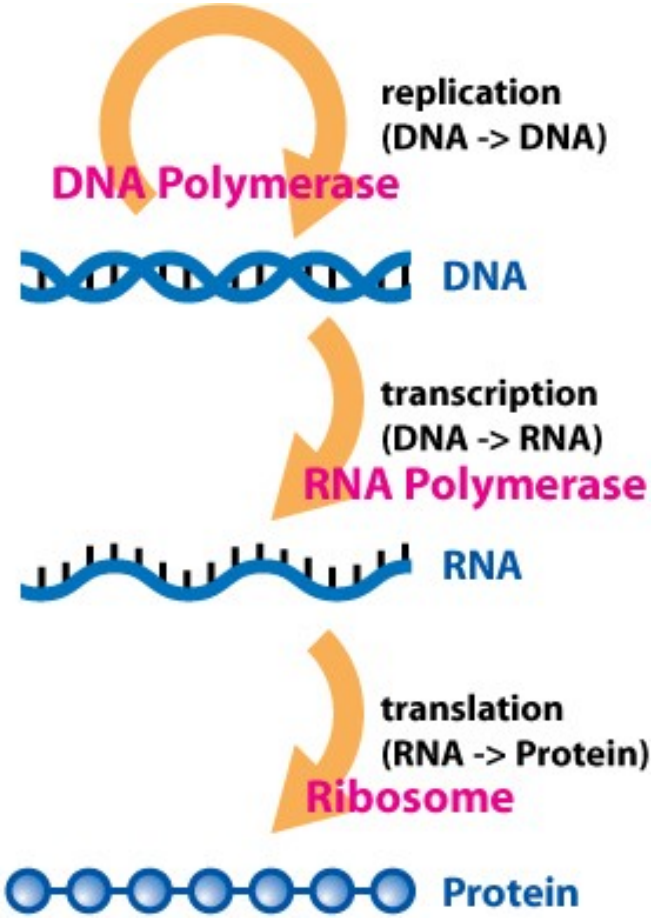
Dark Chocolate Cake

Mix to-gether in a bowl
4 - 1 oz squares of Bakers Choc.
½ cup butter
1 cup boiling water
while this is melting, sift to-gether
2 cups cake flour
2 cups sugar
1½ tsp. soda
1 tsp. salt
Mix the water mixture well and add dry ingredients in 3 portions

Translation

use the recipe to create a dish

The Central Dogma of Molecular Biology



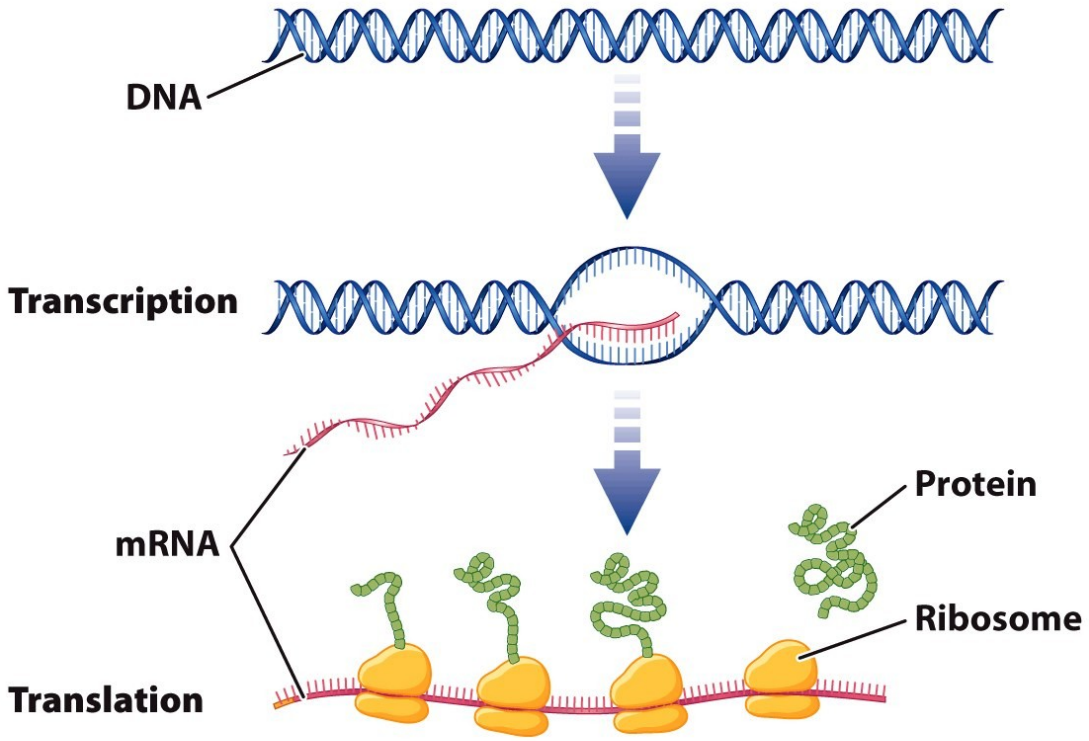
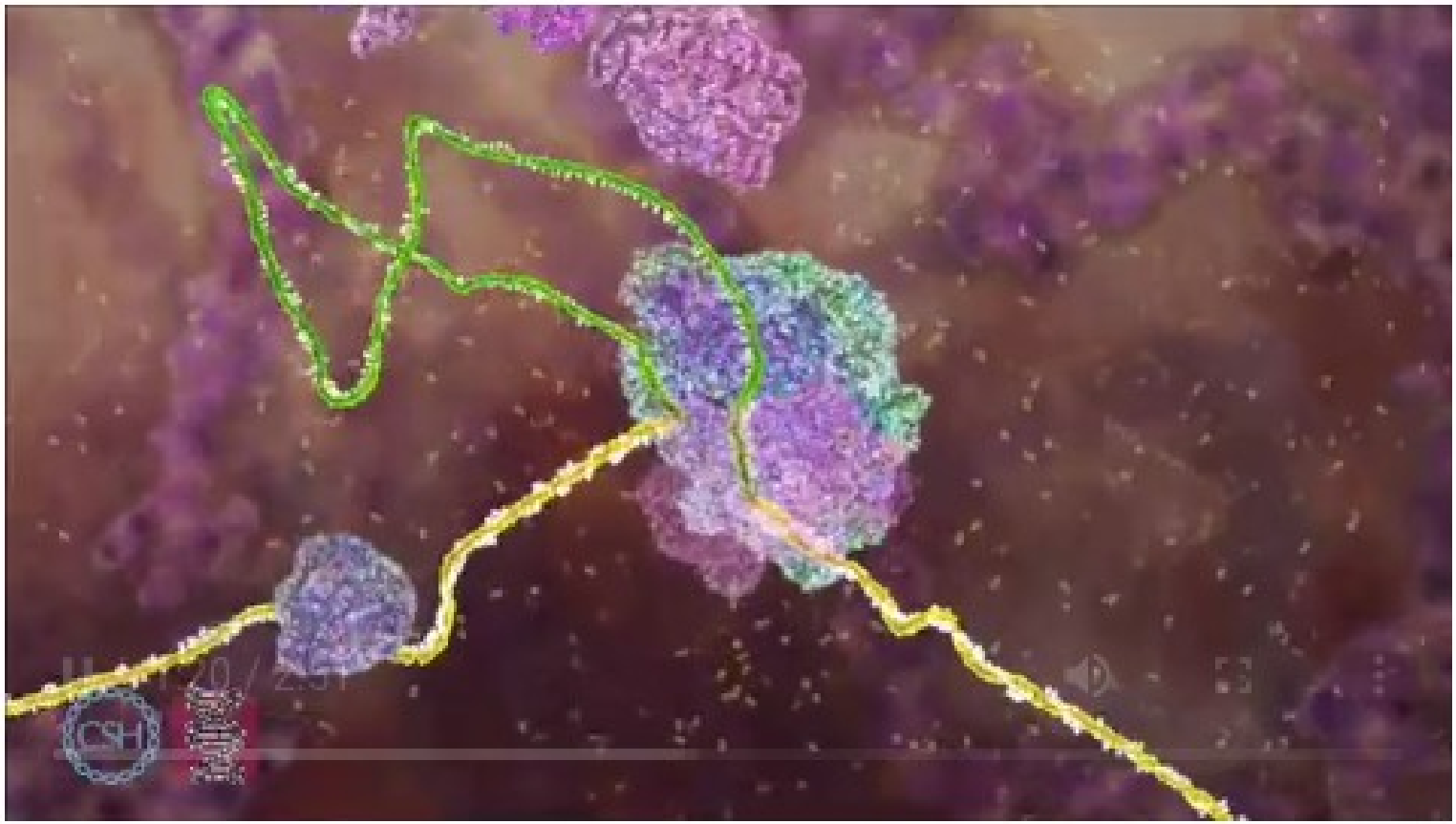


Figure 3.3
How Life Works

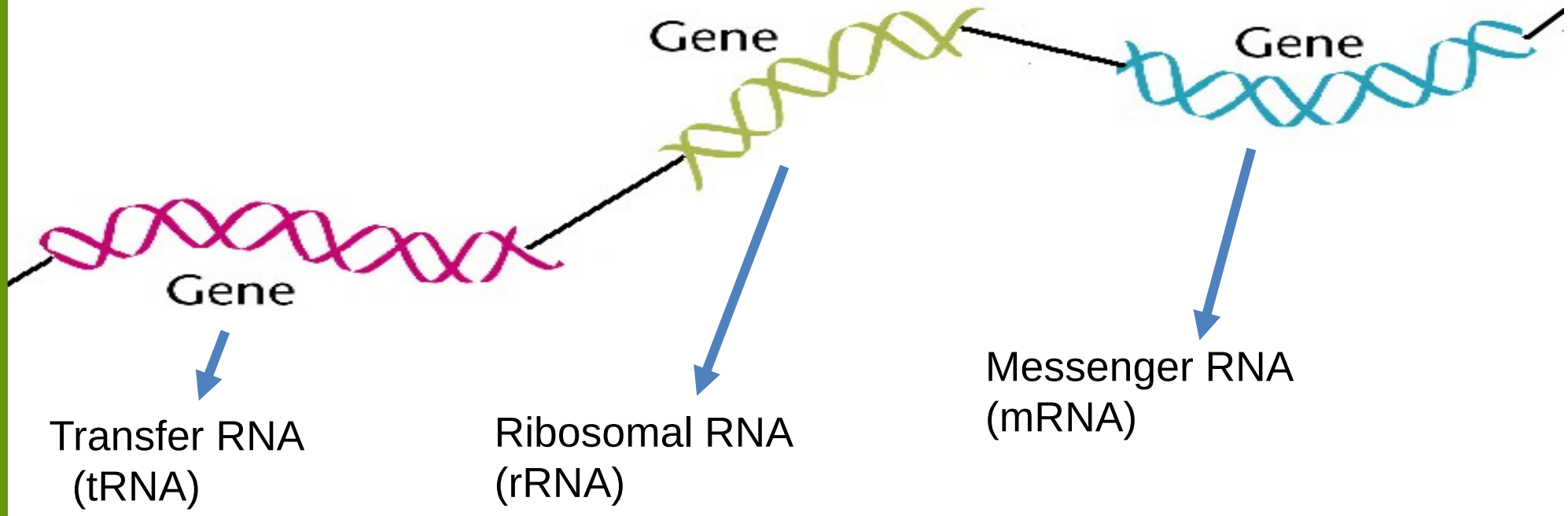
Animation:
Central Dogma of Biology



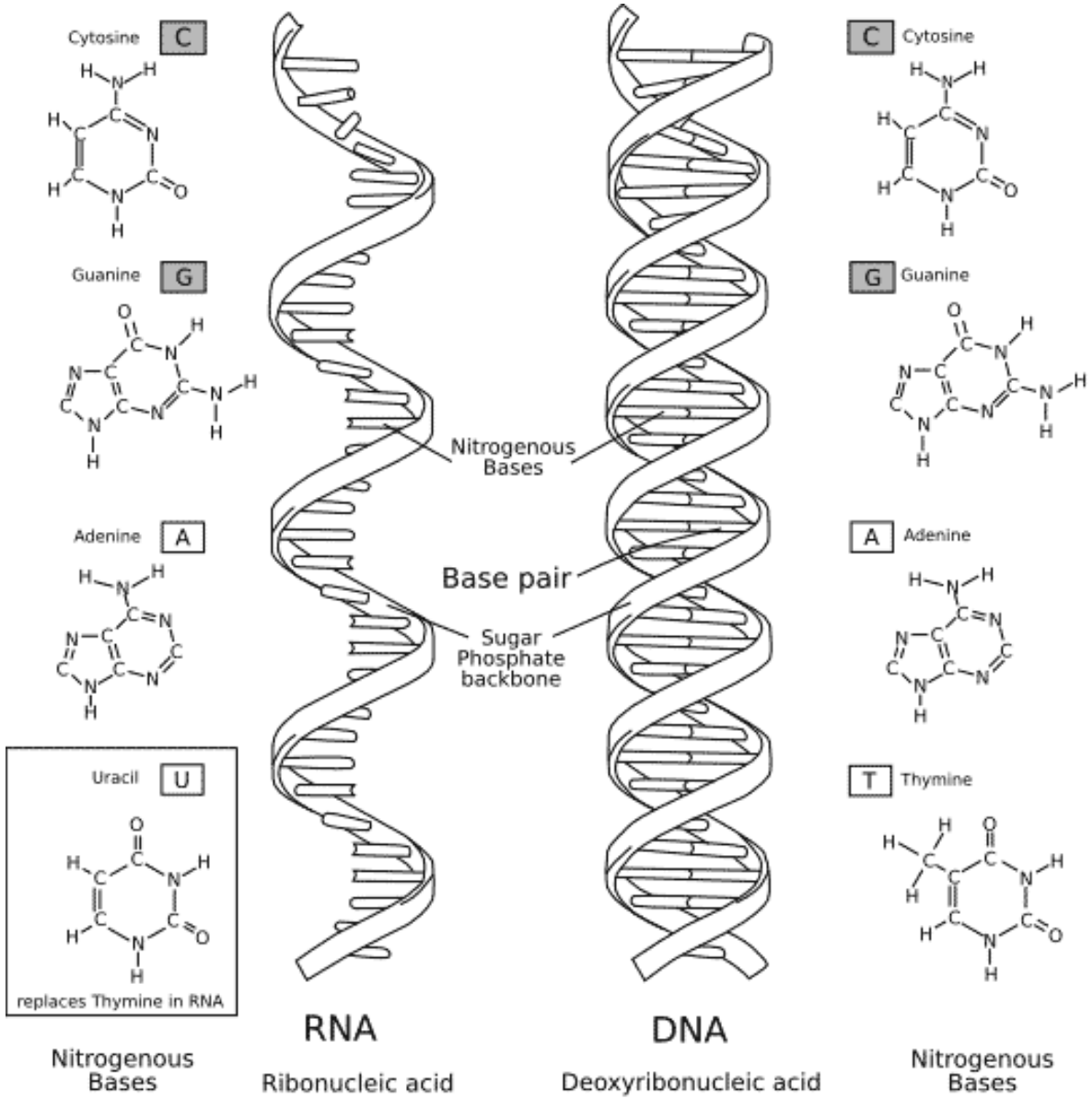
https://dnalc.cshl.edu/view/16933-3D-Animation-of-DNA-to-RNA-to-Protein.html

Transcription

- Transcribe specific regions of DNA – genes
  - Human genome ~25,000 genes (just 1.5% of genome)
- RNA is the direct product of transcribing a gene (DNA)
  - DNA -> RNA
  - same language (nucleotides)



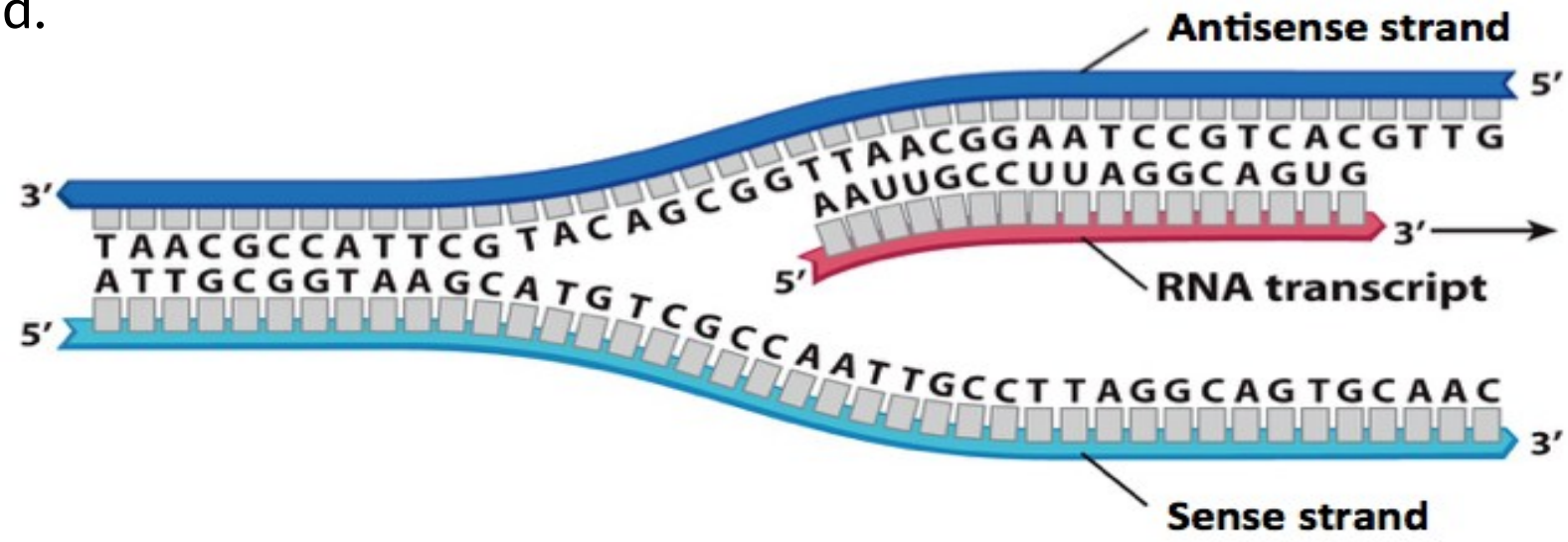
RNA vs DNA



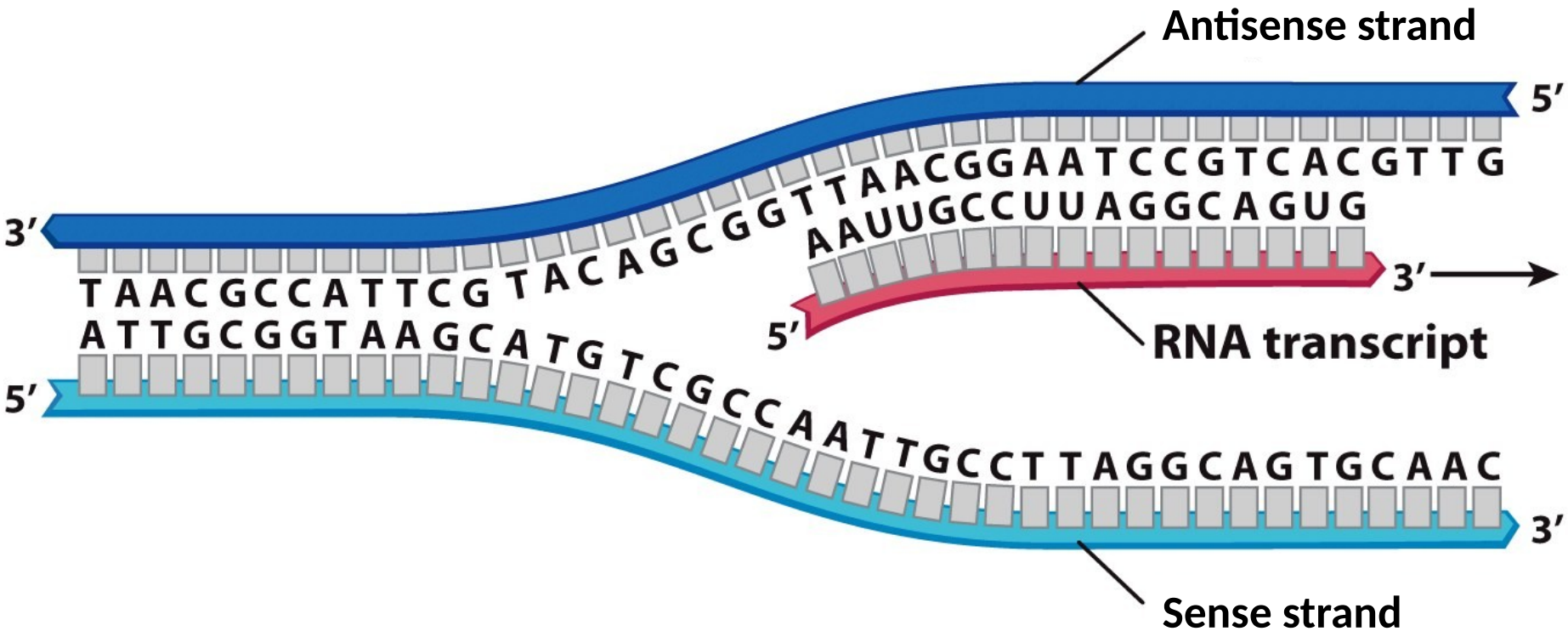
- RNA – uracil replaces thymine (no Ts in RNA)
- RNA – single stranded (one backbone, no basepairs)
- (RNA – slightly different sugar)

Genes exists on both strands of DNA…

- Transcription occurs on the strand containing the gene whose product is needed.
- The strand containing the gene is the antisense strand.
- The RNA transcript is the complement of the antisense strand.



Antisense and Sense Strands of DNA –
relative to the gene being transcribed



Genes have beginnings and ends - promoters and terminators

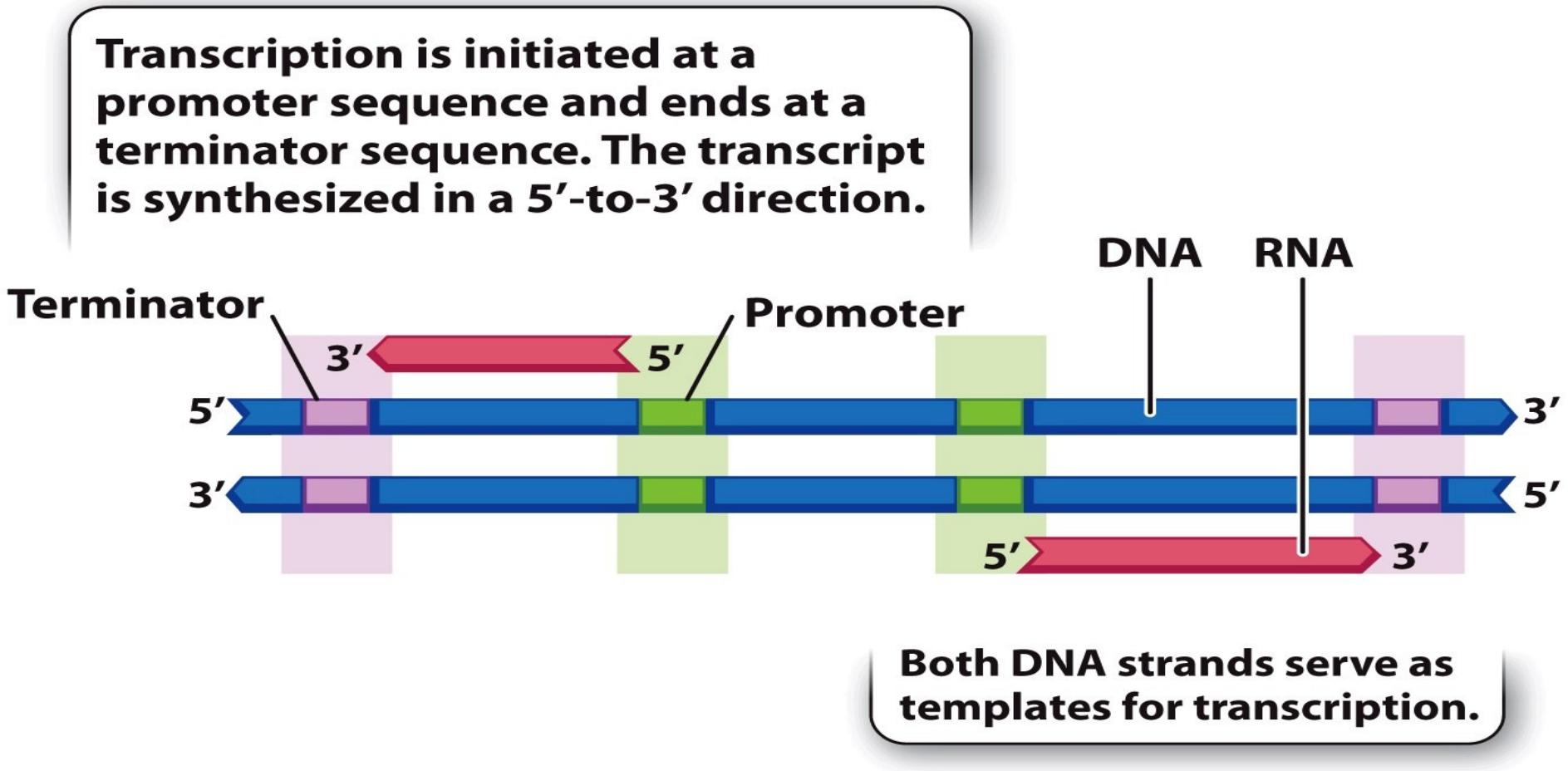
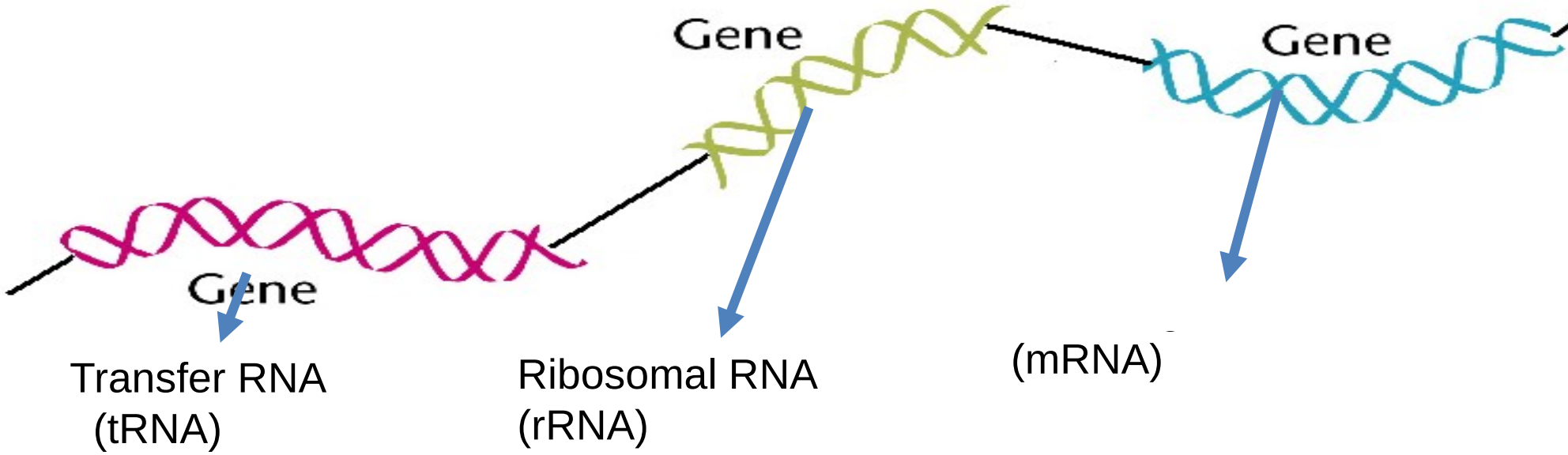


Figure 3.16
How Life Works

Transcription

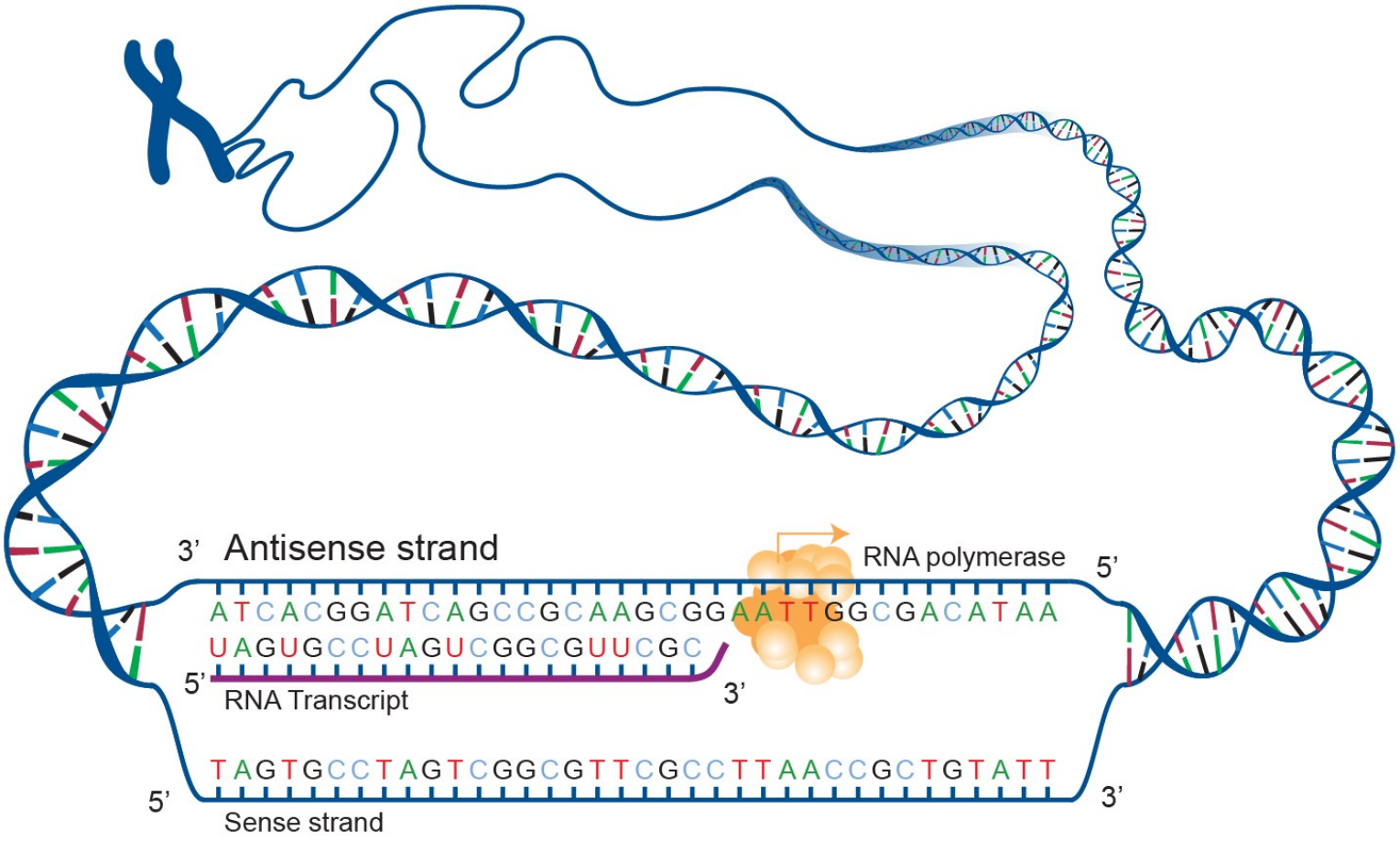
- Transcribe specific regions of DNA – genes
  - Human genome ~25,000 genes (just 1.5% of genome)
- RNA is the direct product of transcribing a gene (DNA)
  - DNA -> RNA
  - same language (nucleotides)



Transcription Video

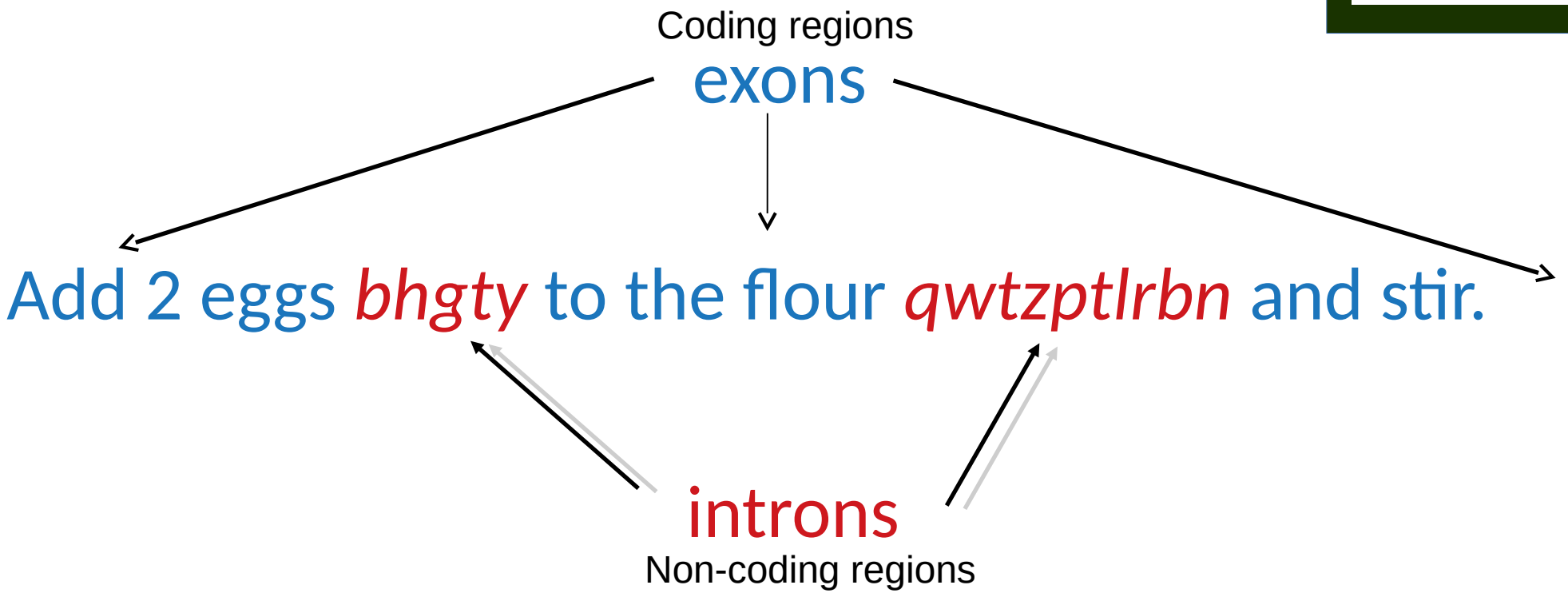
https://www.dnalc.org/resources/3d/12-transcription-basic.html

Sense and Antisense DNA



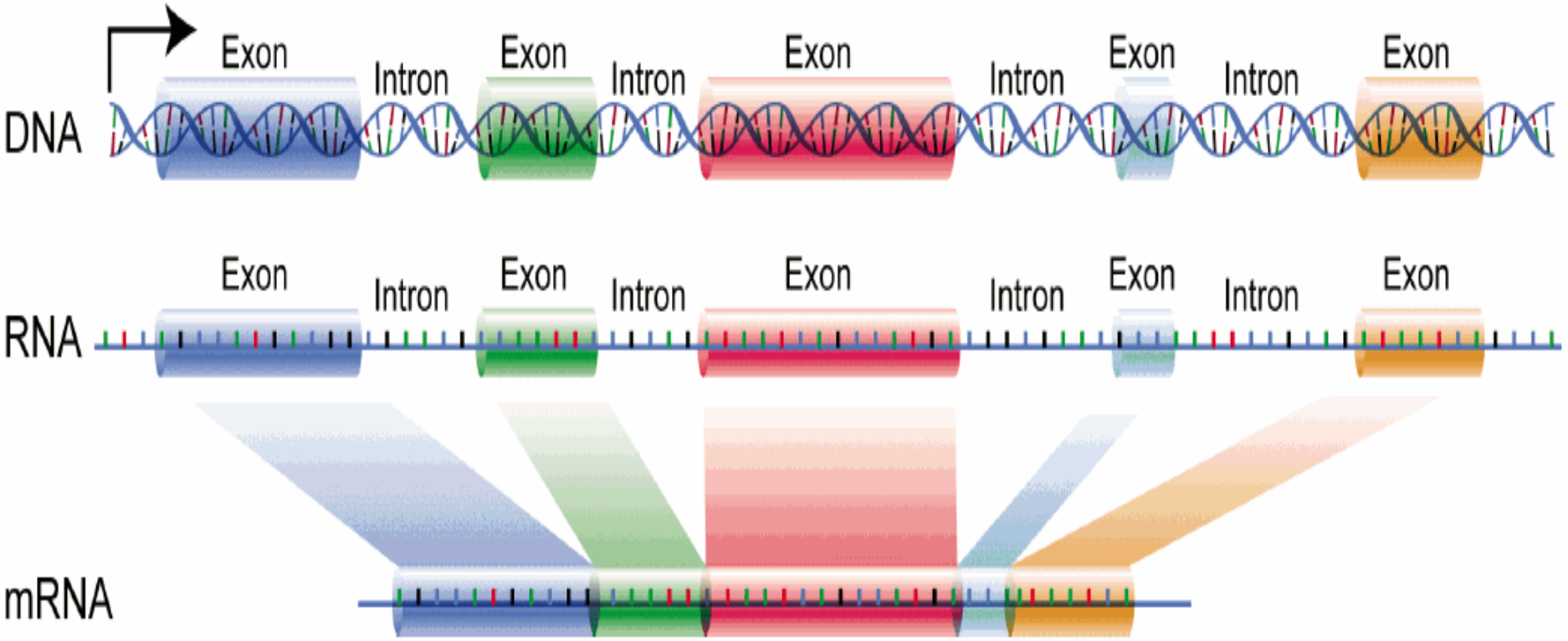
- Antisense is the non-coding DNA strand of a gene
- A cell uses antisense DNA strand as a template for producing messenger RNA (mRNA) that directs the synthesis of a protein.

Exon and Introns



- In most eukaryotic genes, coding regions (exons) are interrupted by noncoding regions (introns). Introns do not contain the message and are removed from the RNA after transcription but prior to translation.
  During the process of RNA splicing, introns are removed and exons joined to form a contiguous coding sequence.

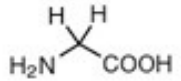
Splicing Exon Material



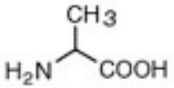
- Exons: a segment of a DNA or RNA molecule containing information coding for a protein or peptide sequence.
- Eukaryotic pre-mRNA contains exons and introns*
  - *some pre-mRNAs contain only one exon

Proteins are made of amino acids

Small

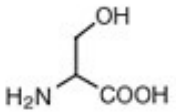


Glycine (Gly, G)
MW: 57.05

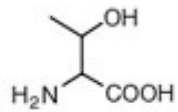


Alanine (Ala, A)
MW: 71.09

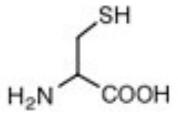
Nucleophilic



Serine (Ser, S)
MW: 87.08, pK_a ~ 16

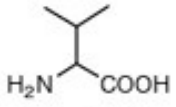


Threonine (Thr, T)
MW: 101.11, pK_a ~ 16

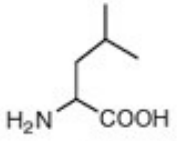


Cysteine (Cys, C)
MW: 103.15, pK_a = 8.35

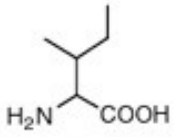
Hydrophobic



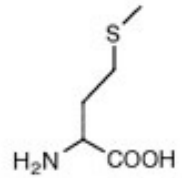
Valine (Val, V)
MW: 99.14



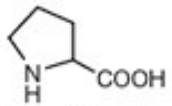
Leucine (Leu, L)
MW: 113.16



Isoleucine (Ile, I)
MW: 113.16

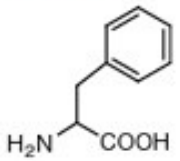


Methionine (Met, M)
MW: 131.19

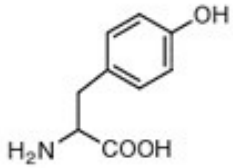


Proline (Pro, P)
MW: 97.12

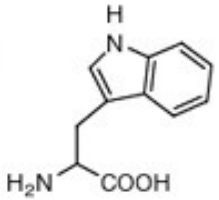
Aromatic



Phenylalanine (Phe, F)
MW: 147.18

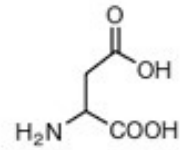


Tyrosine (Tyr, Y)
MW: 163.18

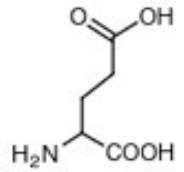


Tryptophan (Trp, W)
MW: 186.21

Acidic

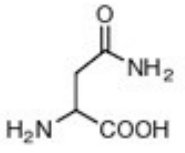


Aspartic Acid (Asp, D)
MW: 115.09, pK_a = 3.9

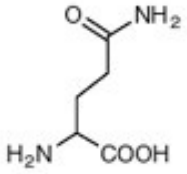


Glutamic Acid (Glu, E)
MW: 129.12, pK_a = 4.07

Amide

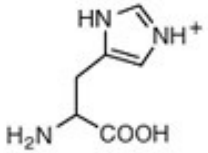


Asparagine (Asn, N)
MW: 114.11

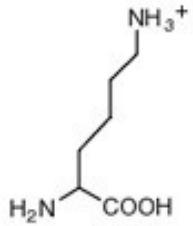


Glutamine (Gln, Q)
MW: 128.14

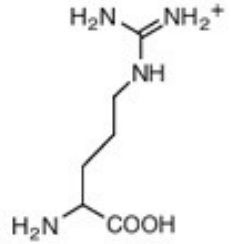
Basic



Histidine (His, H)
MW: 137.14, pK_a = 6.04



Lysine (Lys, K)
MW: 128.17, pK_a = 10.79



Arginine (Arg, R)
MW: 156.19, pK_a = 12.48

The Genetic Code: RNA into Protein

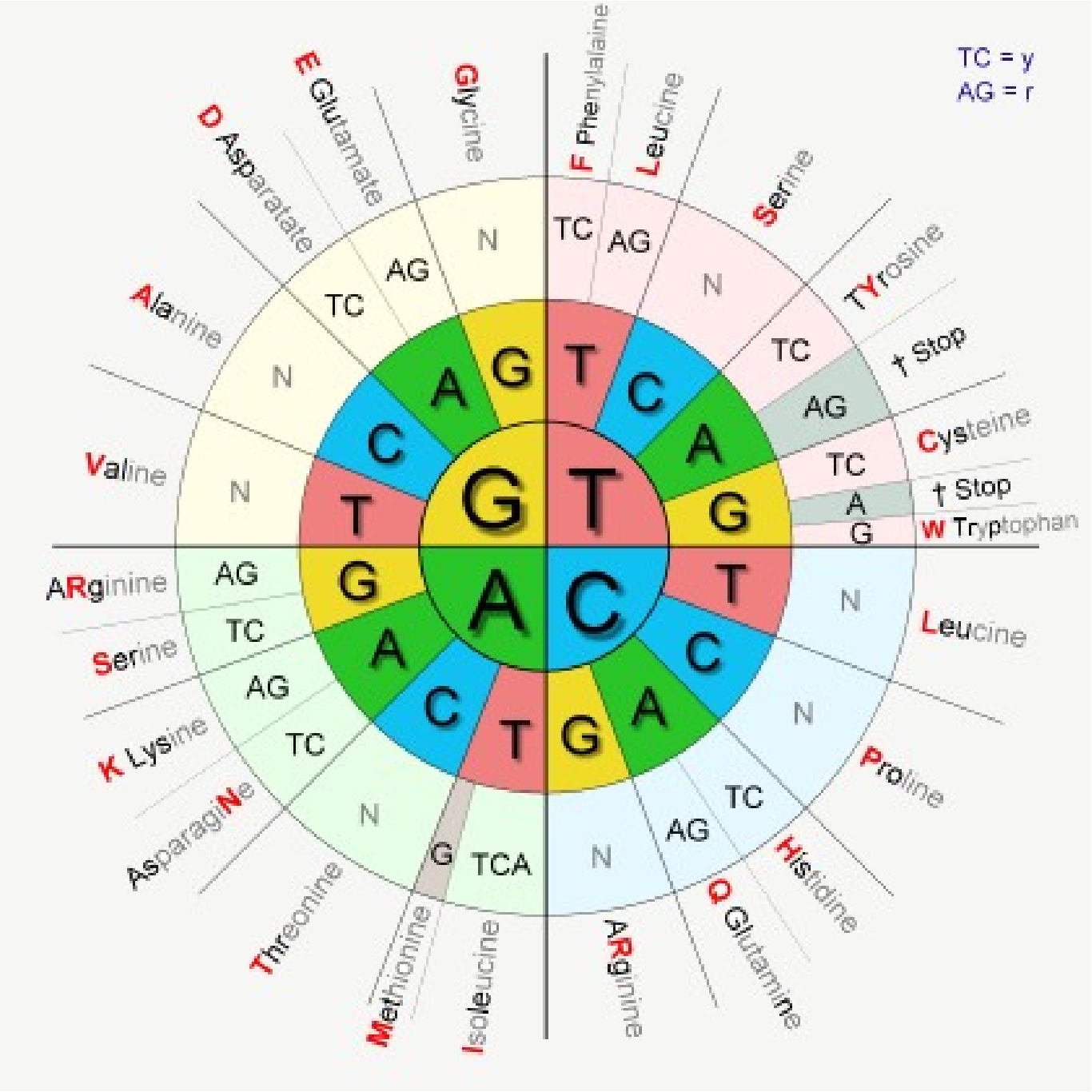
- Triplet code
  - Combinations of three nucleotides code for one amino acid
  - Three nucleotides = codon
- Redundancy
  - Sometimes >1 codon codes for same amino acid
  - 20 amino acids, 64 possible codons

- Start and Stop codons
  - First codon of many transcripts is "AUG", which codes for *methionine*
  - Codons UAA, UAG, and UGA indicate the end of the transcript

Standard genetic code

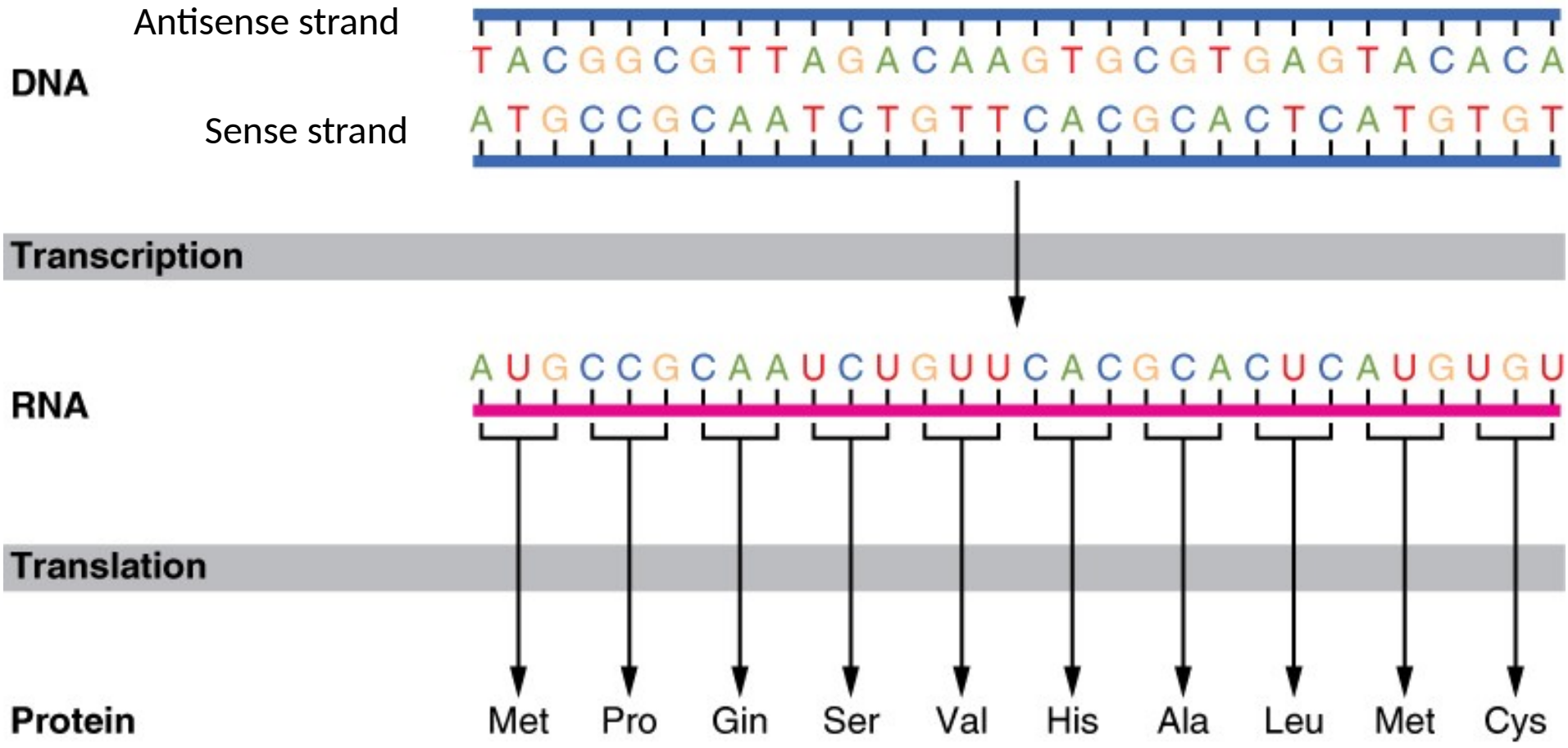
1st base	2nd base								3rd base
	T		C		A		G		
T	TTT	(Phe/F) Phenylalanine	TCT	(Ser/S) Serine	TAT	(Tyr/Y) Tyrosine	TGT	(Cys/C) Cysteine	T
	TTC		TCC		TAC		TGC		C
	TTA	(Leu/L) Leucine	TCA		TAA[B]	Stop (*Ochre*)	TGA[B]	Stop (*Opal*)	A
	TTG		TCG		TAG[B]	Stop (*Amber*)	TGG	(Trp/W) Tryptophan	G
C	CTT		CCT	(Pro/P) Proline	CAT	(His/H) Histidine	CGT	(Arg/R) Arginine	T
	CTC		CCC		CAC		CGC		C
	CTA		CCA		CAA	(Gln/Q) Glutamine	CGA		A
	CTG		CCG		CAG		CGG		G
A	ATT	(Ile/I) Isoleucine	ACT	(Thr/T) Threonine	AAT	(Asn/N) Asparagine	AGT	(Ser/S) Serine	T
	ATC		ACC		AAC		AGC		C
	ATA		ACA		AAA	(Lys/K) Lysine	AGA	(Arg/R) Arginine	A
	ATG[A]	(Met/M) Methionine	ACG		AAG		AGG		G
G	GTT	(Val/V) Valine	GCT	(Ala/A) Alanine	GAT	(Asp/D) Aspartic acid	GGT	(Gly/G) Glycine	T
	GTC		GCC		GAC		GGC		C
	GTA		GCA		GAA	(Glu/E) Glutamic acid	GGA		A
	GTG		GCG		GAG		GGG		G

Another Triplet Table



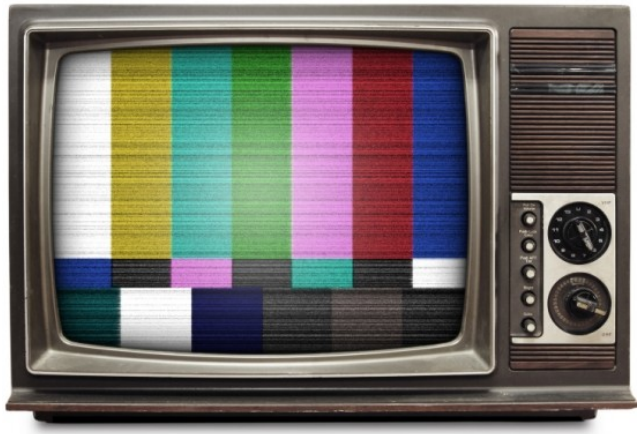
Translation

- The information from DNA is rewritten in a new language: RNA



Translation Videos

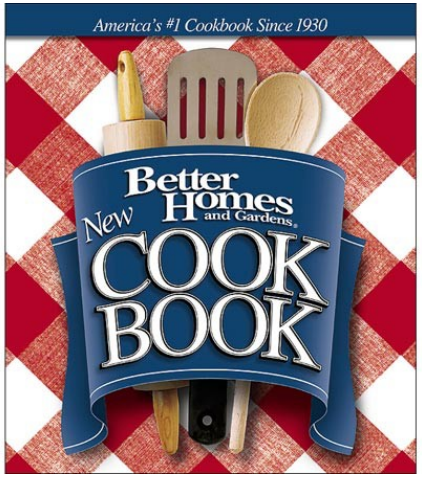
- mRNA Translation (2 mins)
  - https://www.youtube.com/watch?v=8dsTvBaUMvw
- Protein Synthesis and the Lean, Mean Ribosome Machines (7 mins)
  - https://www.youtube.com/watch?v=h5mJbP23Buo
- DNA transcription and translation (includes gene expression, 7 mins)
  - https://www.youtube.com/watch?v=2BwWavExcFI



Genes vs Gene Expression

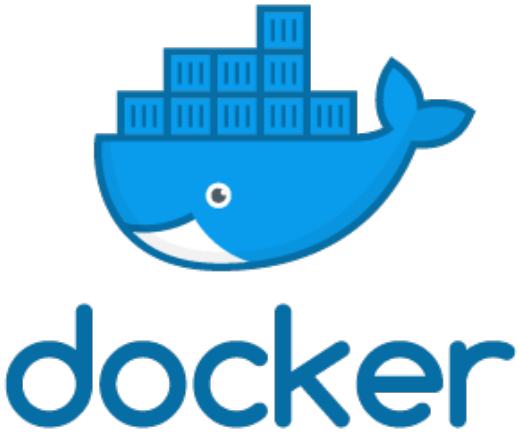
All genes are present in the genome
genes only expressed when needed

Of the many recipes in the cookbook…

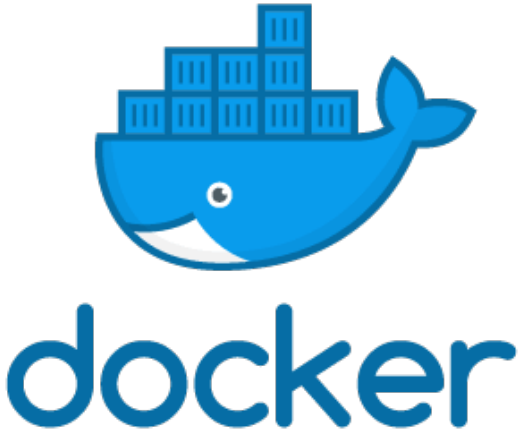


… Only transcribe and translate 4th of July recipes in **July**

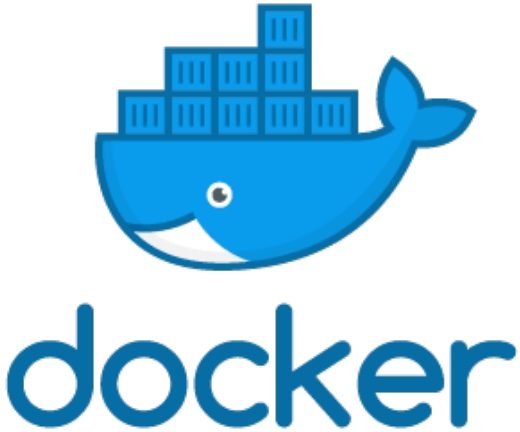
… Only transcribe and translate the Thanksgiving turkey recipe in **November**



- Prepare to run some commands for Docker …
- Or, wait for a few slides and run Python3 shell in your browser online



- Note: If you are not using ToolBox, Docker should already be working in the background
- Navigate to where you have stored your ***docker_getMeToThePython*** directory.



Mac and Windows ToolBox: find and run the “Docker QuickStart Terminal”

Note: See file, *quickStartCommands.md*, for these commands

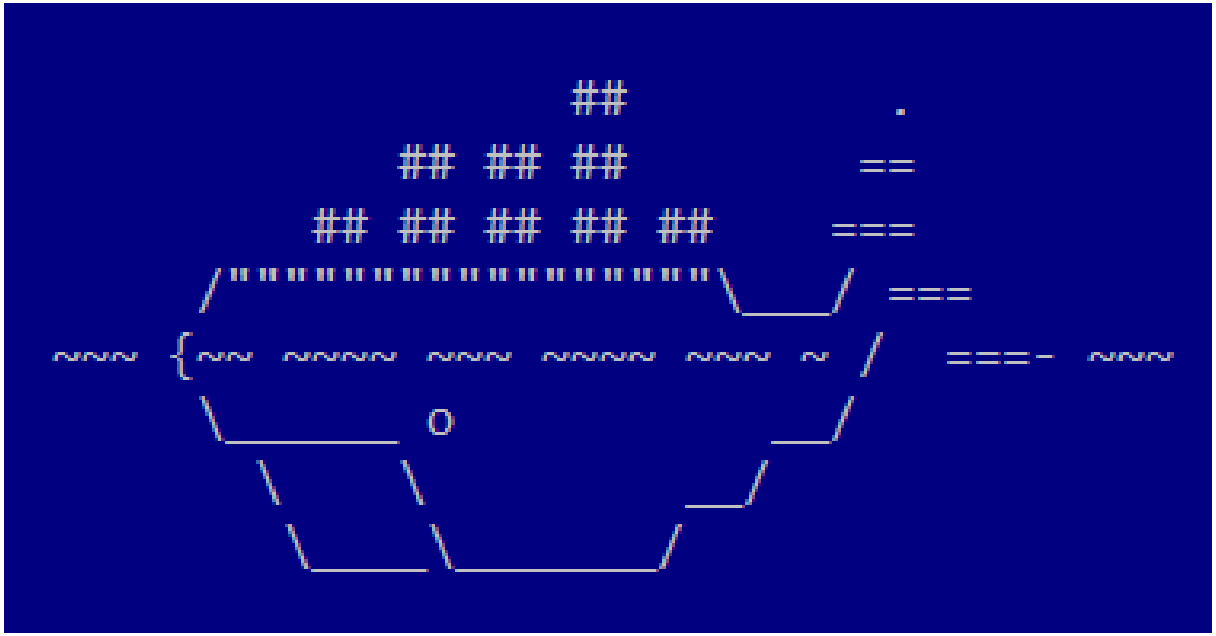
Note: The Docker ToolBox commands to initiate server

Windows Quickstart Command:

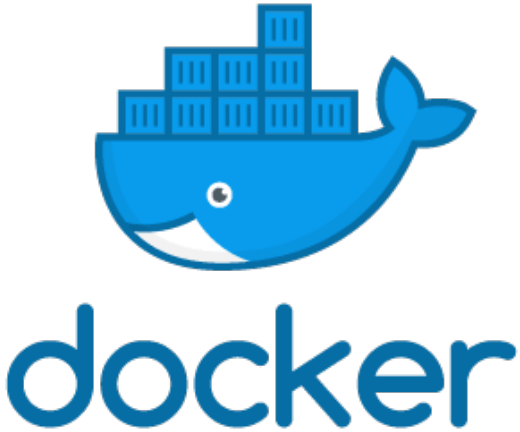
```
"C:\Program Files\Git\bin\bash.exe" --login -i "C:\Program Files\
Docker Toolbox\start.sh"
```

MacOS Quickstart Command:

```
bash --login '/Applications/Docker/Docker Quickstart Terminal.app/
Contents/Resources/Scripts/start.sh'
```

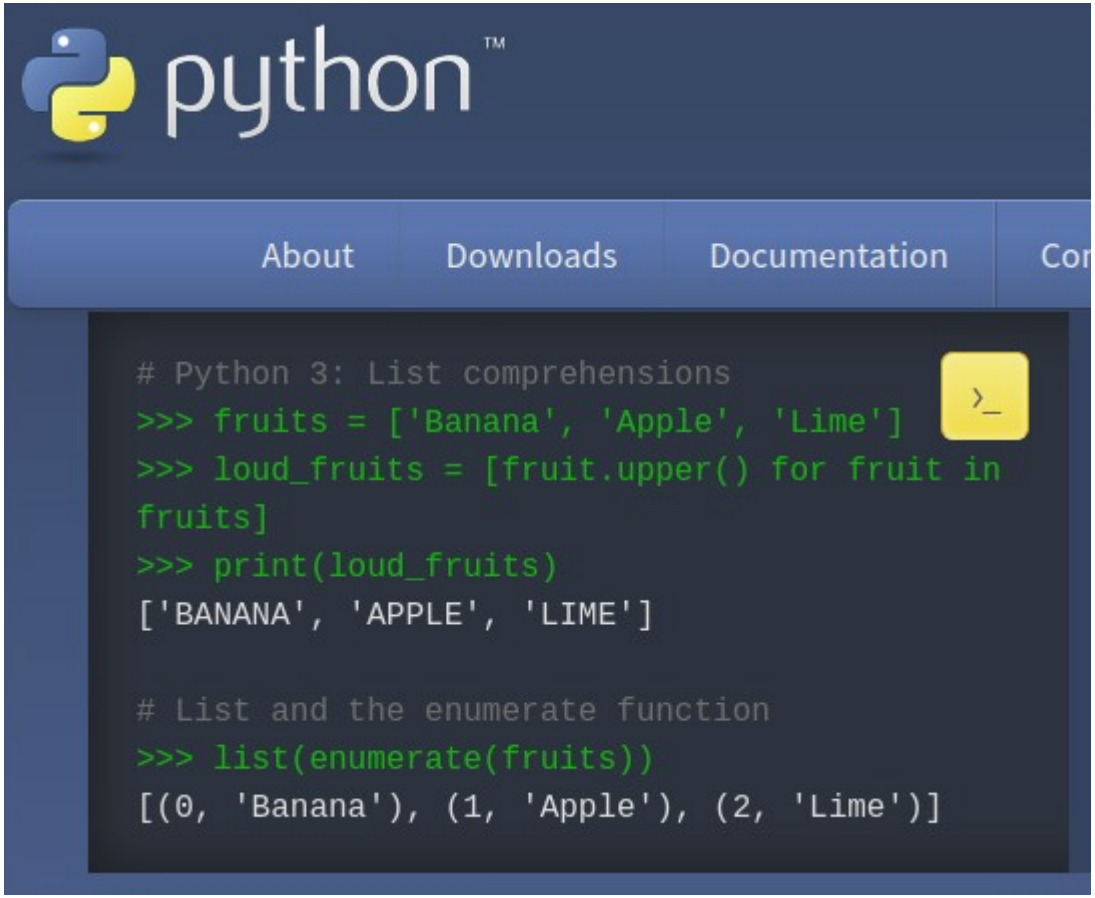


If your server was properly initialized then, you should see this cute whale.



- Two ways to run Python3 in DockerQS or bash shell
- See file, *commands.md,* for these notes
  - `docker run -t python3`
  - **Or, build the container and run Python3 there**
    - `docker build -t py_play .`
  - **Mount a drive and then use bash to run Python3**
    - `docker run -it --mount type=bind,source=$PWD,target=/home/py_play py_play`

- *Or, try Python3 programming using an interactive shell from repl.it*
- Link: https://repl.it/languages/python3



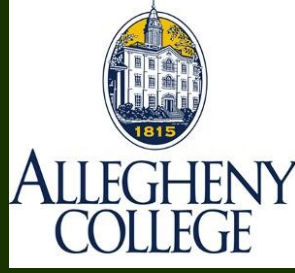
- Some trouble to make Python3 work with Docker ToolBox
- Install and use Python3 however you want!
- Get Python3 from the Python Software Foundation
- `Login` `http://www.python.org/downloads`

- Download and install the version of Python3 for your OS being sure to add the PATH to the environmental variables (check the path option!)
- Ask questions if you have trouble installing the program
- Check with the installation material to learn how to launch
- Python3 from your machine.

Python3

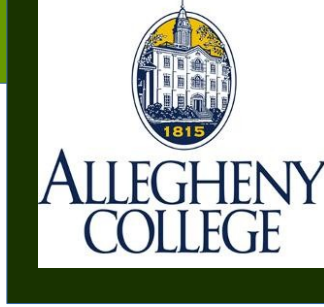
```
#Calculating values
3 / 4
2 * 6
3.1415 – 2.718
x = 1
y = 2
print(x+y)
result = x + y
print(“The result is :”,result)
```

Python3



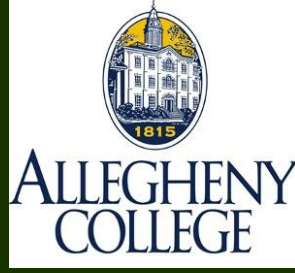

```
# Integers, counting numbers
   num_int = 1

# Floats, decimals
   num_float = 3.1415

# Strings
   s_str = ” Hello World”

# Combining variables in print statements
   x_int = 1
   print(" The integer variable is :", x_int)

   num_float = 3.14
   print(" The float variable is :", num_float)

   s_str =("Hello World'')
   print(" The integer is equal to", s_str)
```

Calculate




```
3 + 4 # Addition
3 – 4 # Subtraction
3 * 4 # Multiplication
3 / 4 # Division of 3 by 4

# Modulus; Returns the remainder from the
division 3 * *4
3%4

# Powers; raise three to the power of four
= 3*3*3*3
= 3^4
= pow(3,4)
```

Strings



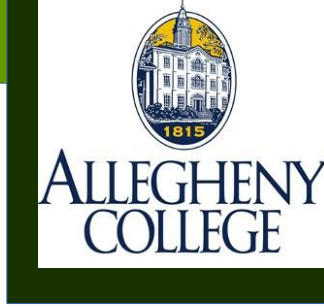

```
# Remember each char of a string has own
position
s_str = "ABC"
s_str[0] = 'A'
s_str[1] = 'B'
s_str[2] = 'C'
s_str[200] = ??

# Another way to iterate
# through a string using its length
for i_int in range(len(s_str)):
  print(s_str[i_int])
```

Counting and Finding

```
# Getting input from a user
resp_str = input(“Enter your name :”)
print(" Hello",resp_str,"!")

# Determine number of chars in a string
lengthOfName_int = len(resp_str)

# Find a subset-string in the string
resp_str.find("M")
resp_str.find("A")
resp_str.find("R")
resp_str.find("K")
resp_str.find("ARK")
```

```
>>> resp_str = input("Enter your name :")
Enter your name :Mark
>>> print(" Hello",resp_str,"!")
 Hello Mark !
>>> resp_str.find("M")
0
>>> resp_str.find("A")
-1
>>> resp_str.find("a")
1
```

Counting and Finding



Find char occurrence in a string

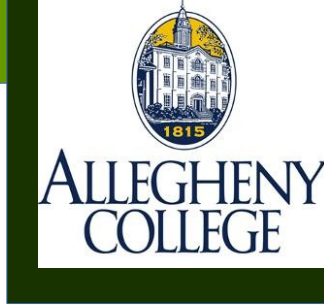
```
resp_str = “Hello!!”
resp_str.count("H")
resp_str.count("l")
```

Find number of specific triples in string.

```
resp_str = “Hellollollo!!”
resp_str.count("llo")
resp_str.find("llo")
```

```
>>> resp_str = "Hellollollo!!"
>>> resp_str.count("l")
6
>>> resp_str.count("llo")
3
```

Getting Input



Find char occurrence in a string

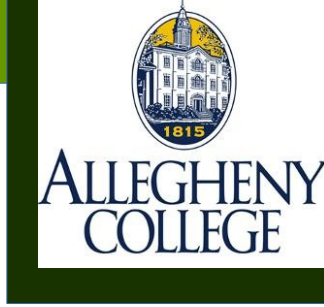
```
resp_str = “Hello!!”
resp_str.count("H")
resp_str.count("l")
```

Find number of specific triples in string.

```
resp_str = “Hellollollo!!”
resp_str.count("llo")
```

```
>>> resp_str = "Hellollollo!!"
>>> resp_str.count("l")
6
>>> resp_str.count("llo")
3
```

A Short Program
Watch for tabs that define code blocks




```
print("Welcome to the program!")
prmpt_str = " Please enter your name :"
# place the string above into input statement
name_str = input(prmpt_str)
print(" Your name is :",name_str)
print(" And is <<",len(name_str),">> chars long!")
# print the chars on lines
print(" What are the characters in the string? ")
for i in range(len(name_str)):
    # note the tabs for this block!
    # we iterate through the positions in string
    print("  + char :",name_str[i])
# findout how many a's are in the name
numChar_int = name_str.count("a")
print(" The number of a's in your name :",numChar_int)
```

For this code, see file, pythonDemo.py, in your sandbox.

Consider This ...

- Group work
- Write a short program in Python3 that …
- Accepts a short sequence of DNA (that you type in) and counts the occurrences of:
  - A's
  - T's
  - G's
  - C's
  - "AT", "TA"
  - "GC" and "CG"

Then Consider This ...

- Now, go get a real piece of DNA and try out your program
- Link for organism: *Gordonia phage Orchid*, complete genome,
  - https://www.ncbi.nlm.nih.gov/nuccore/NC_030915.1?report=fasta
- What results did you find in terms of the pairs of AT's, TA's, GC's and CG's?
- Are the numbers of pairs similar or dissimilar?