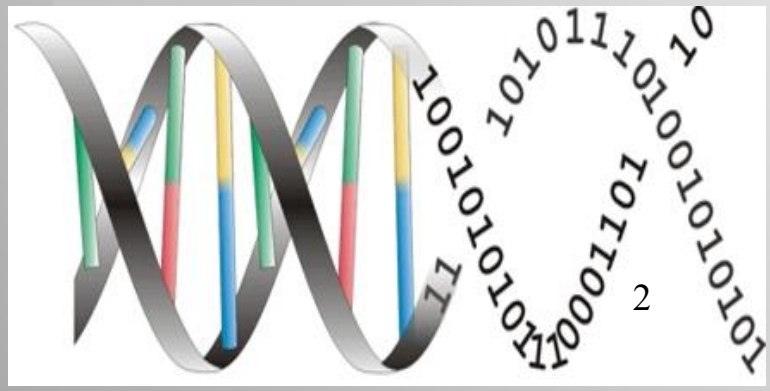
The University of Georgia

The University of Georgia

Programming for Computational & Systems Biology

Instructor: Paul Xie

Tue. & Thr. 9:35~10:50



Overview of Course

The University of Georgia

- Introduction to programming skills in computational and systems biology. Topics include real world examples, such as processing genome or proteome data, and analyzing large-scale data. The idea of “big data” will be emphasized to help students with their coding skills to discovering new knowledge in biomedical sciences and solving biomedical problems.

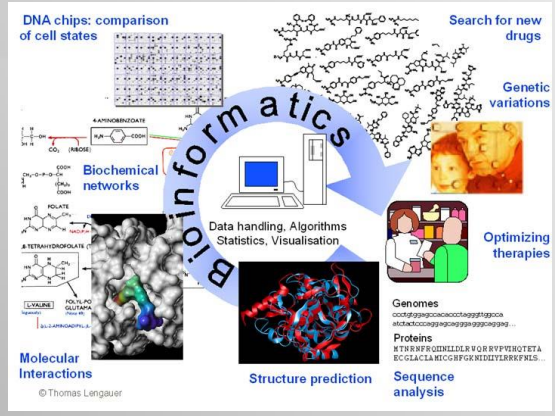
Course Format

The University of Georgia

- Computer labs + lectures (3 hours/week)
- Coding concepts (some knowledge about biological data)
- 6-8 assignments, published on eLC, (30-40%)
  - Please upload them by the due days
- Paper review (10-20%)
- 1 term paper (50-60%)

Topics

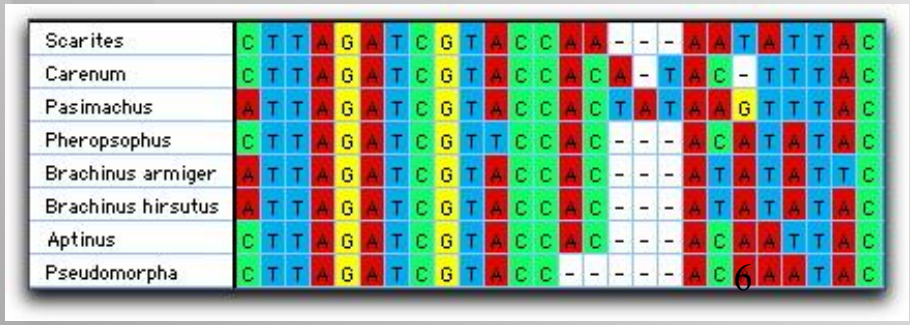
- -Omics data, e.g. Genomics, Proteomics…
- Basic Programming, e.g. I/O, variables, string, loop, regular express, array…etc.
- Data retrieval & processing
- Data analysis
- Model & building model
- Clustering and Classification
- Prediction



The University of Georgia

Data Array: List & Dictionary Transcription & Translation

Instructor: Paul Xie (4)



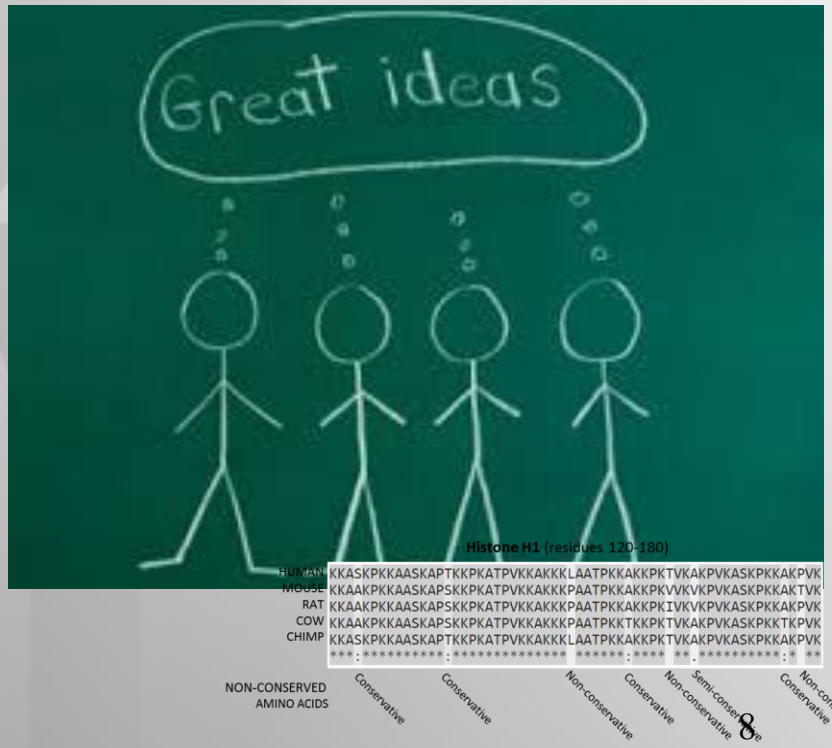
This Week

The University of Georgia

- Random number
- Data array
  - List
  - Dictionary
- Average & Std. Dev.

Group Discussion

The University of Georgia



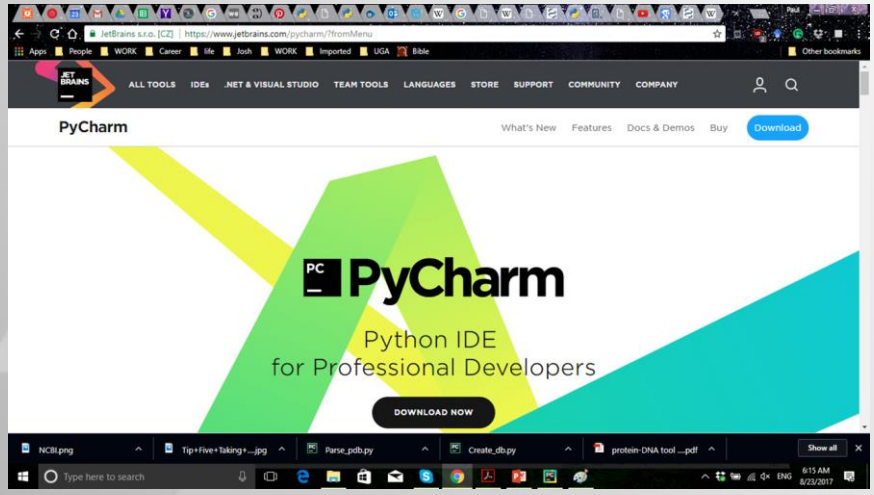
Python Time

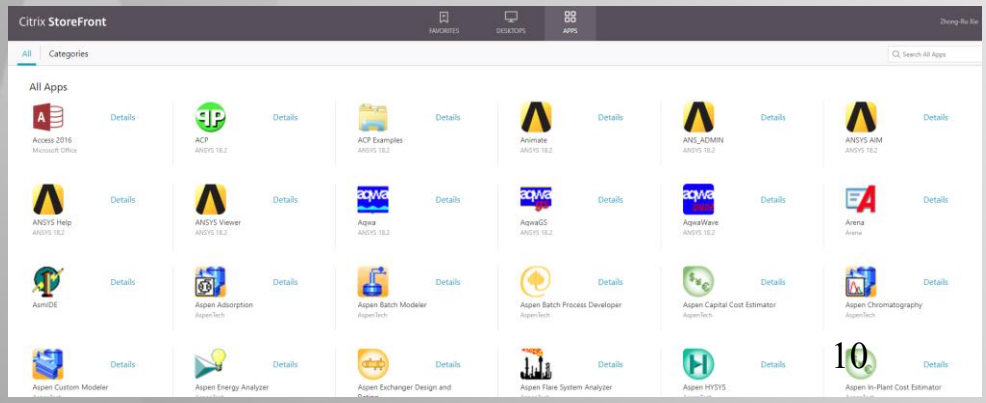
The University of Georgia

Open PyCharm

The University of Georgia

- Go to UGA CENGR Mylab
  - mylab.engr.uga.edu
- Login Citrix
  - Apps
  - PyCharm
  - New project





Syntax

The University of Georgia

- Array
  - **List** is a collection which is ordered and changeable. Allows duplicate members. []
  - **Tuple** is a collection which is ordered and unchangeable. Allows duplicate members. ()
  - **Set** is a collection which is unordered and unindexed (unchageable). No duplicate members.{}
  - **Dictionary** is a collection which is unordered, changeable and indexed. No duplicate members. {key:200}

Code

The University of Georgia

```
#array
def main():
    cars = ["Ford","Volvo","BMW"]
    for i in cars:
        print(i)

main()
```

Code

The University of Georgia

```
#array
def show_array(cars):
    print(len(cars),'cars.\nThey are:')
    for i in cars:
        print(i)
    for i in range(10):
        print('-',end='')
    print()
……..
```

Code

```
……..
def main():
    cars = ["Ford","Volvo","BMW"]
    show_array(cars)
    cars.append("Honda")  #
    show_array(cars)
    cars.pop(2)
    show_array(cars)
    cars.insert(1,"Toyota")
    show_array(cars)
    print(cars.index("Honda"))
    cars.remove("Volvo")
    show_array(cars)
    cars.sort()
    show_array(cars)
    cars.reverse()
    show_array(cars)
main()
```

Exercise

- Random number ➔ array
- Calculate max, min, mean (average)
- Check redundant
  - Double loop
- Sort
  - Bubble sort

Water Level Marker

The University of Georgia

- An array a[ ]
- A variable x to record highest (lowest) value
- (For loop)
  - Compare a[i] and x
  - Record the highest value

Average

- An array a[n]
- A variable s to record summary
- (For loop)
  - s += a[i];
- s /=n;

Practice

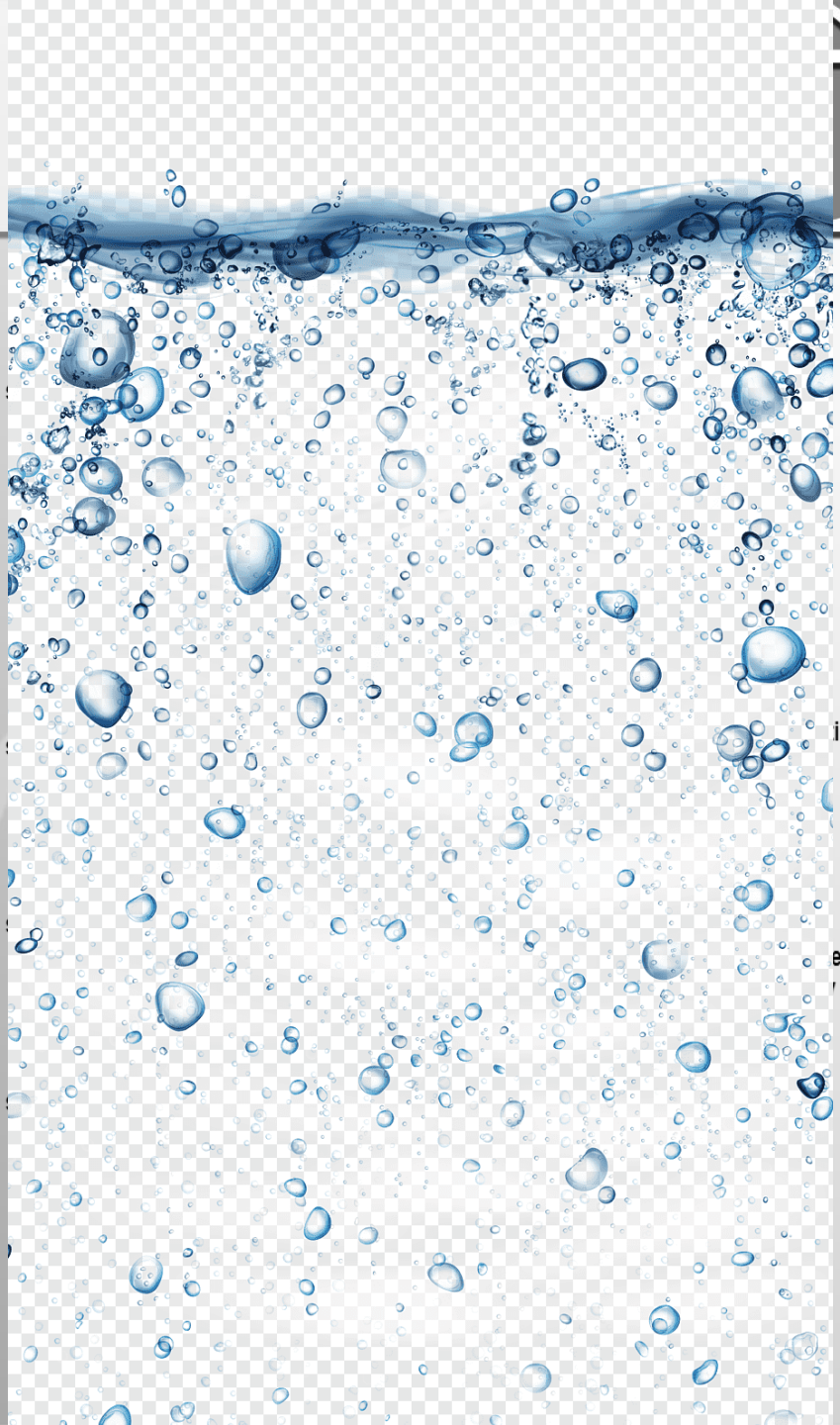
The University of Georgia

- Average and Standard deviation

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i-\overline{x})^2}=\sqrt{\frac{1}{N}\left(\sum_{i=1}^{N}x_i^2\right)-(\overline{x})^2}=\sqrt{\left(\frac{1}{N}\sum_{i=1}^{N}x_i^2\right)-\left(\frac{1}{N}\sum_{i=1}^{N}x_i\right)^2}.$$

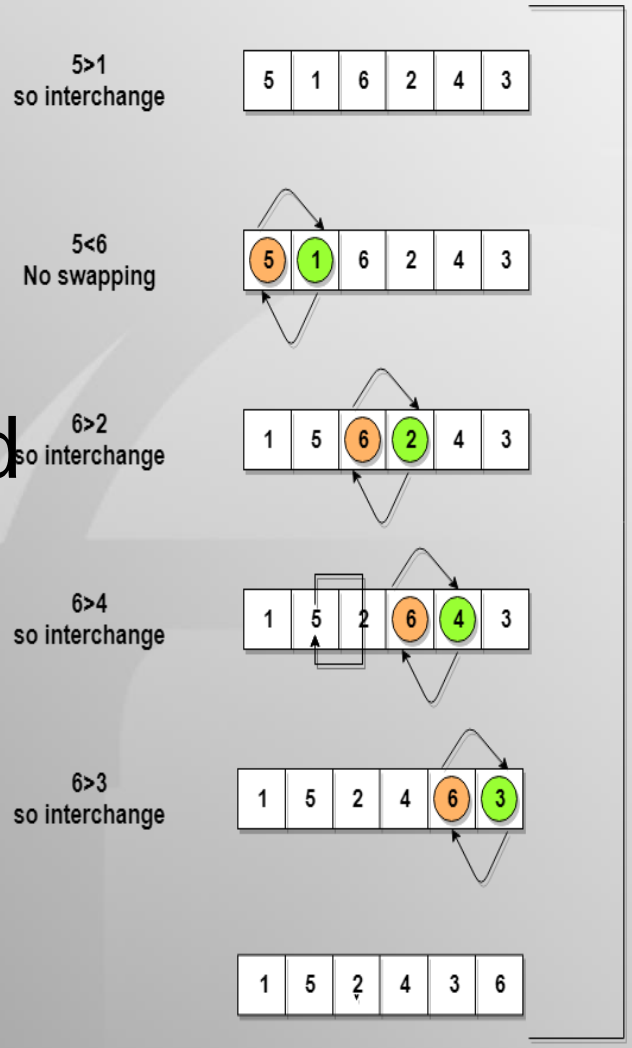
Bubble Sort

- An array a[n] contains n numbers
- How to sort them?
- Big bubbles is lighter (go up)



Bubble Sort

- An array a[ ]
- Double loops (loop #1 and #2)
- Compare numbers and switch



Non-redundant

- Lottery
- 5 or 6 non-redundant balls (numbers)
- For loop
  - Generate ball[i]
  - Compare balls

Taiwan Lottery

The University of Georgia

- Prize
  - 1st (6 matches) : 82%
  - 2nd (5+special): 7%
  - 3rd (5 matches): 6.5%
  - 4th (4+special): 4.5%
  - 5th (4 matches): $80
  - 6th (3+special): $40
  - 7th (2+1 or 3 matches): $8
- ~56% of incomes

Exercises

- sort: from big to small
- List all Prime from 1 to 100
  - For loop
  - If /else
  - Array
  - Size unknown
- Store factorial in an array, e.g. 1! ~ 10!
  - Do we need to calculate 1!, 2!, 3!, … separately?

Multiplication Table

The University of Georgia

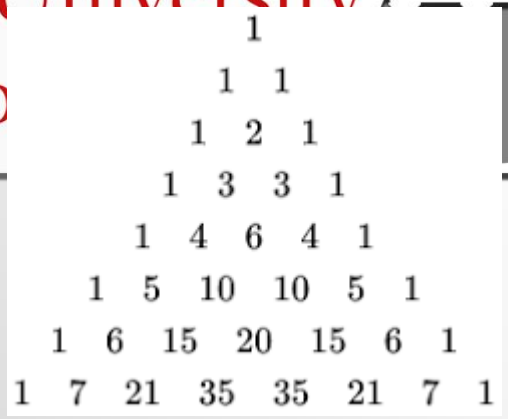
- Double Loop
- A loop inside another

```
a = []; b = []; c = []
for i in range(10):
    k = []; l = [];  m = []
    for j in range(10):
        k.append(i+1)
        l.append(j+1)
        m.append((i+1)*(j+1))
    a.append(k)
    b.append(l)
    c.append(m)
```

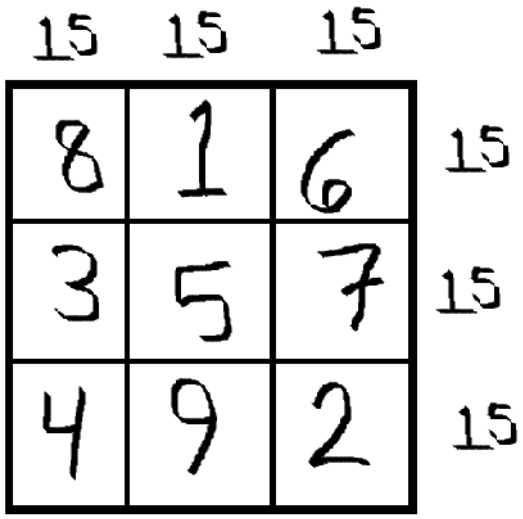
2D Array

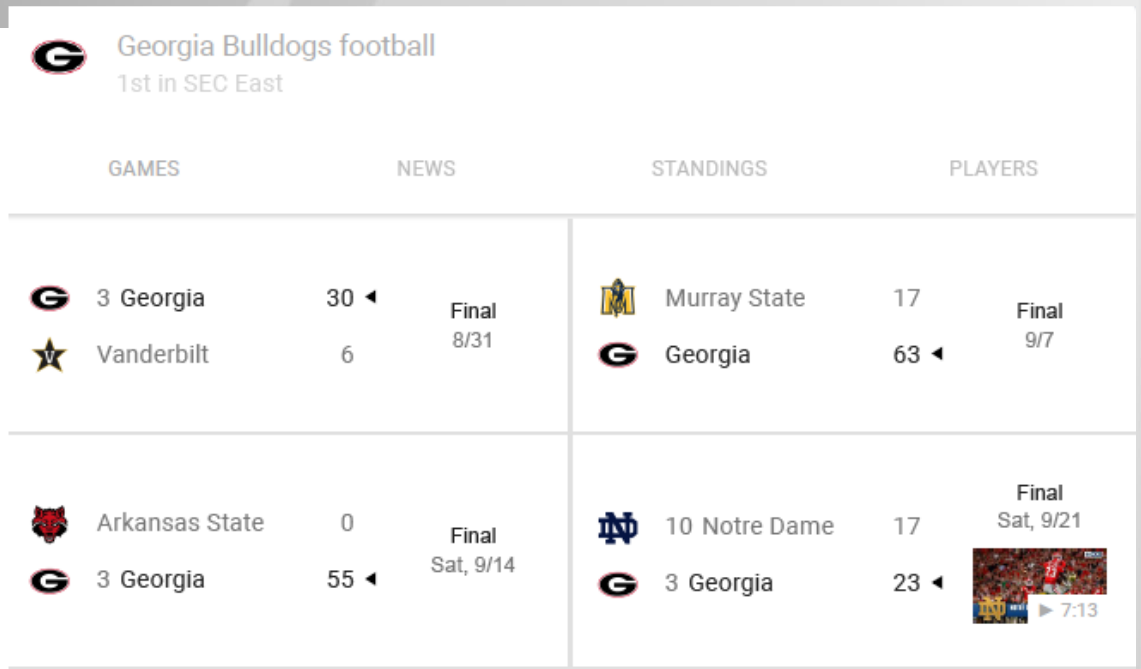
- a = [[0]*m for i in range(m)]
- Or, b = [1,2,3,4,5], c = [2,4,6,8,10],…
- a.append(b)
- a.append(c)
- …
- matrix

Exercise



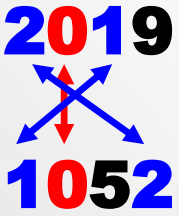
- Create a 2D array (list)
- 2d array for Dawgs score
- 2d array for your assignment scores
- Pascal triangle and Magic square





Exercise Quiz

- Please design a number guessing game called 1A2B



- Please randomly generate a 4-digit number with 4 different number (for example 2019)
- A player make a first guess, say 1024, compare two number2, 1 correct digit (0) has correct position ➔ 1A; and 2 correct digits has incorrect positions ➔ 2B
- Based on previous hints, keep guess until get it right

Open a File

- Syntax
- File_id = open(file_name, operation)
- ‘r’,’w’,’a’
- File_id.read()
- File_id.readline()
- File_id.readlines()

Syntax

- Array
- Dictionary
  - Key-value

Code


```
#dictionary
def main():
    cars =  {
        "brand": "Toyota",
        "model": "Corolla",
        "year": 2003
    }
    print(cars)
    cars["year"] = 2018
    cars["color"] = "red"
    cars["owner"] = "paul"
    cars["seller"] = "Toyota Athens"
    cars["plate"] = "Georgia"
    -------------------------
```

Code

The University of Georgia

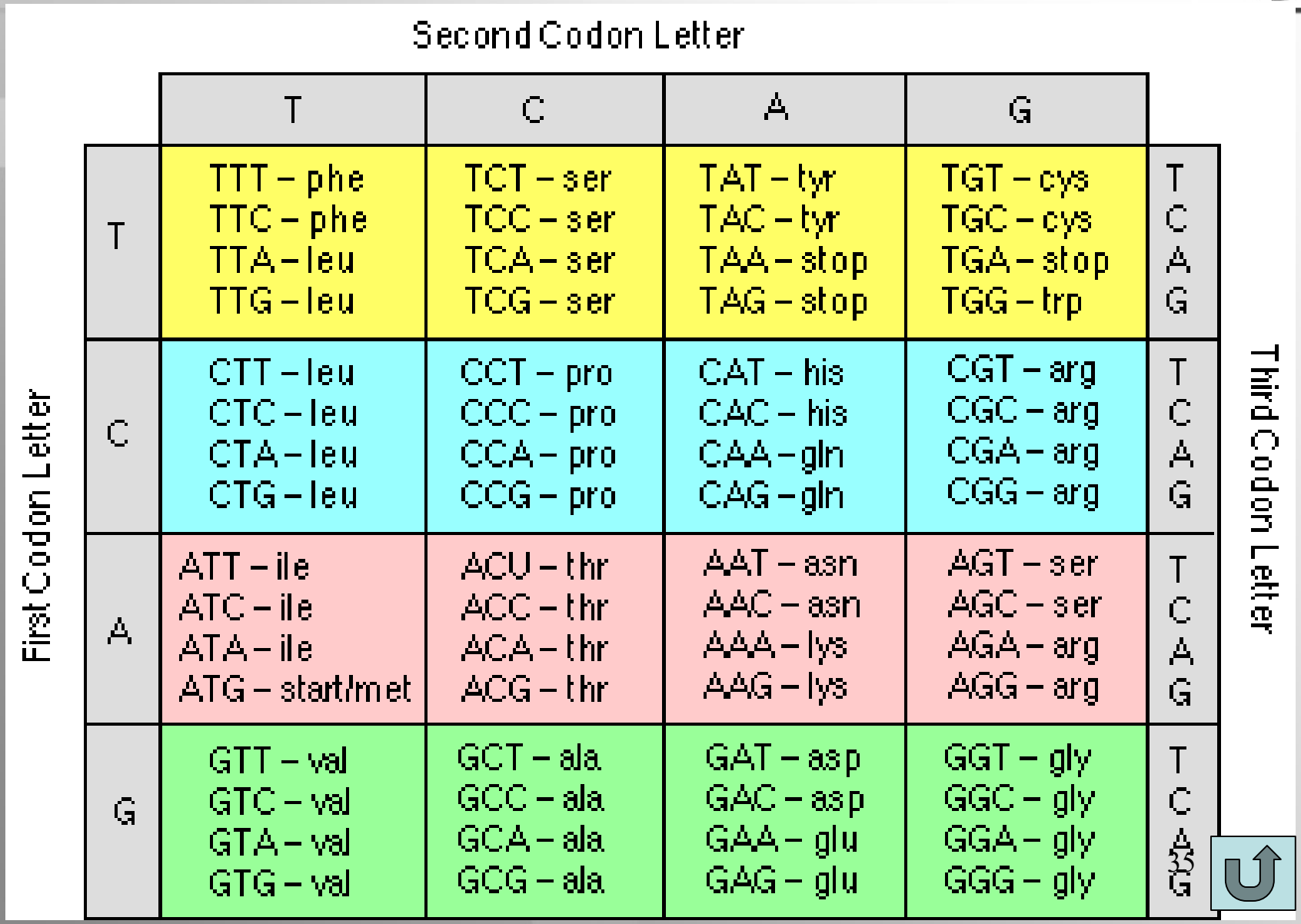
```
print(len(cars))
    for i in cars:
        print(i)
    for i in cars.keys():
        print(i)
    for i in cars.values():
        print(i)
    for x,y in cars.items():
        print(x,y)
    cars.popitem()
    cars.pop("seller")
    del cars[]
    print(cars)
```

Exercise

The University of Georgia

- Calculate the occurrence & frequency of amino acids (or nucleotides)
  - Open file(s)
  - Dictionaries
  - Loops
  - Counting
  - presentation

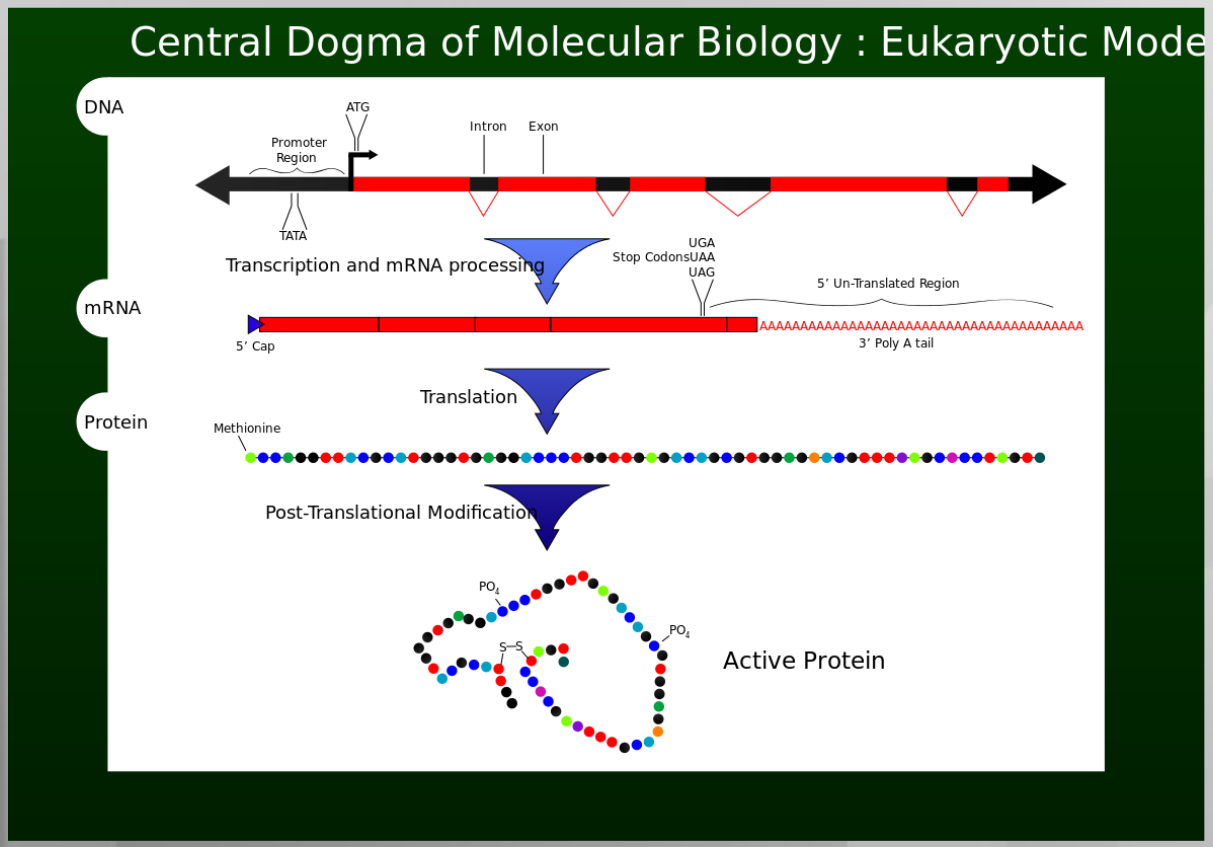
DNA → AA



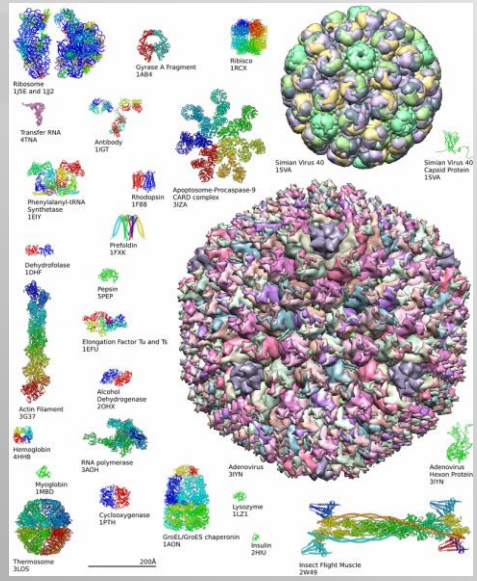
First Codon Letter	Second Codon Letter: T	Second Codon Letter: C	Second Codon Letter: A	Second Codon Letter: G	Third Codon Letter
T	TTT – phe TTC – phe TTA – leu TTG – leu	TCT – ser TCC – ser TCA – ser TCG – ser	TAT – tyr TAC – tyr TAA – stop TAG – stop	TGT – cys TGC – cys TGA – stop TGG – trp	T C A G
C	CTT – leu CTC – leu CTA – leu CTG – leu	CCT – pro CCC – pro CCA – pro CCG – pro	CAT – his CAC – his CAA – gln CAG – gln	CGT – arg CGC – arg CGA – arg CGG – arg	T C A G
A	ATT – ile ATC – ile ATA – ile ATG – start/met	ACU – thr ACC – thr ACA – thr ACG – thr	AAT – asn AAC – asn AAA – lys AAG – lys	AGT – ser AGC – ser AGA – arg AGG – arg	T C A G
G	GTT – val GTC – val GTA – val GTG – val	GCT – ala GCC – ala GCA – ala GCG – ala	GAT – asp GAC – asp GAA – glu GAG – glu	GGT – gly GGC – gly GGA – gly GGG – gly	T C A G

Central Dogma

The University of Georgia

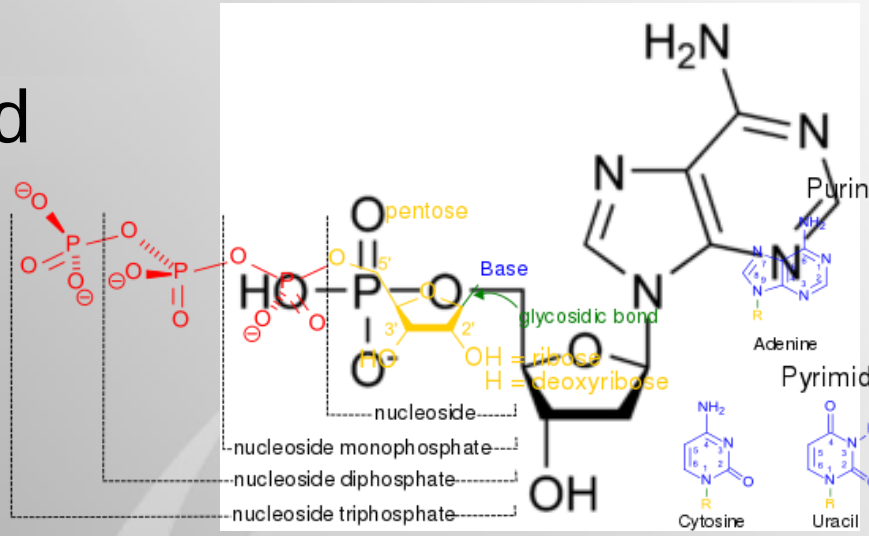


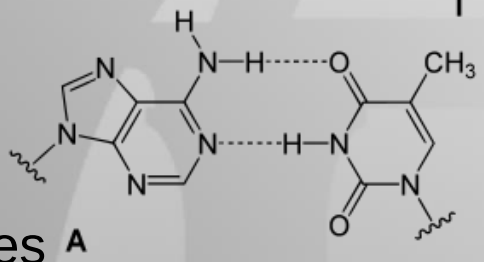
- DNA→RNA→ Protein
- Transcription & Translation (&Replication)

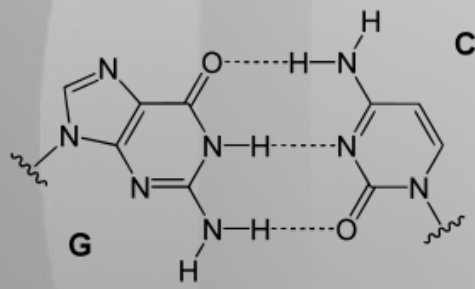


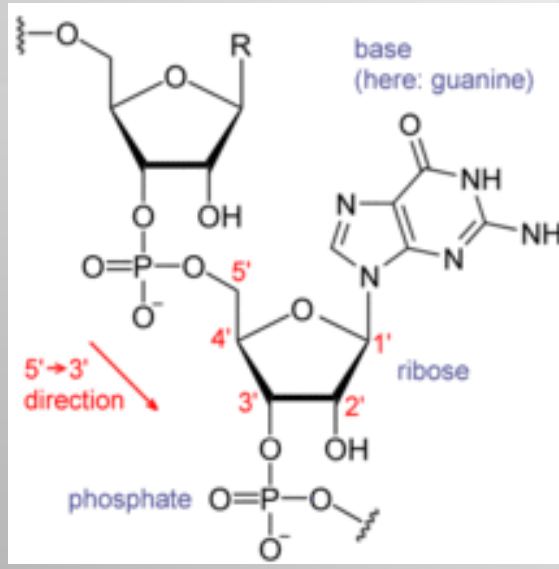
DNA & RNA

- Deoxyribonucleic acid
- Ribonucleic acid
  - Base: A, T/U, C, G
    - Adenine
    - Thymine
    - Uracil
    - Guanine
    - Cytosine
    - Purines & Pyrimidines
  - Ribose
  - Phosphate





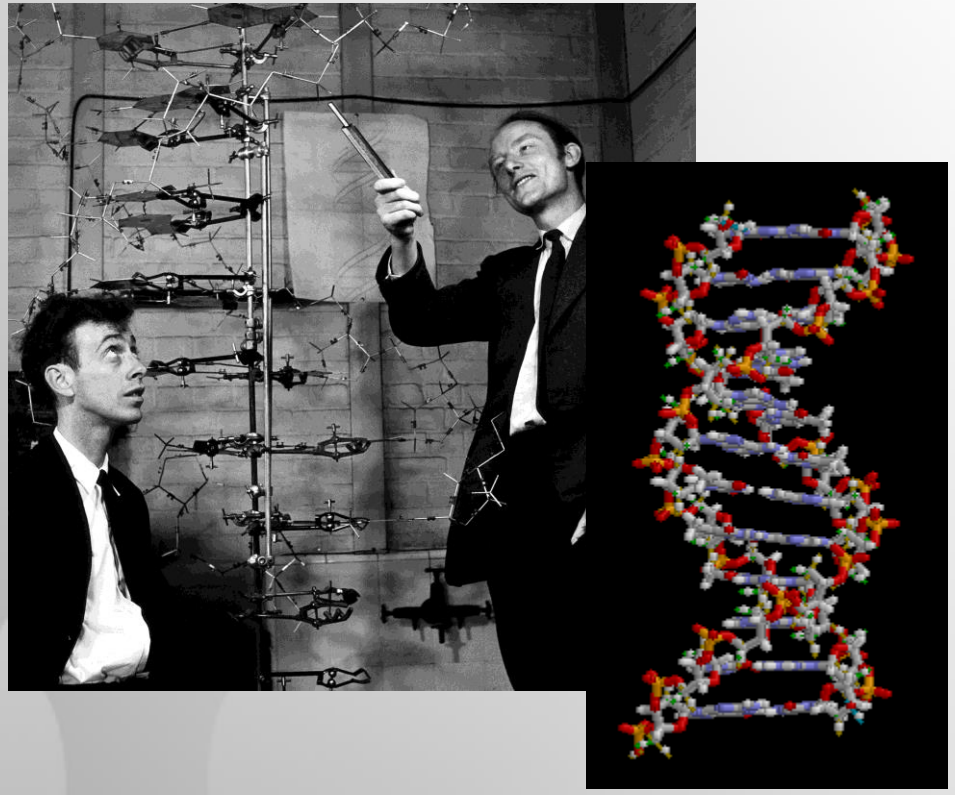


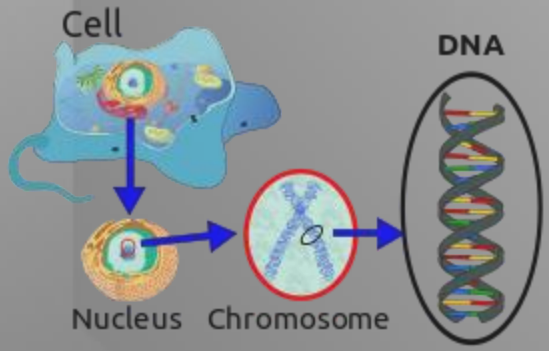


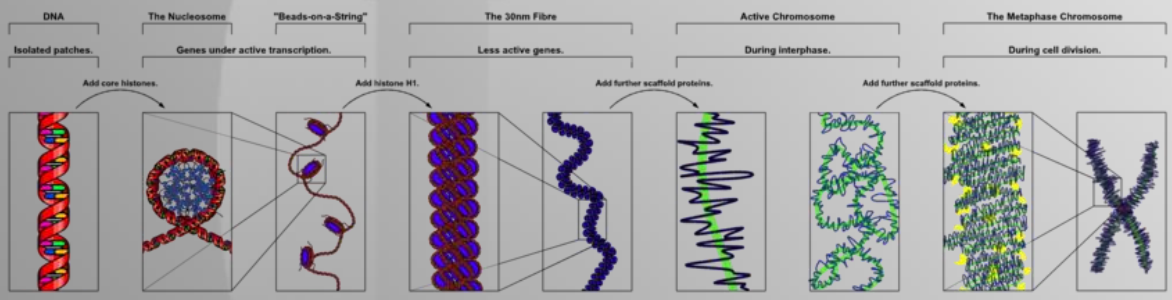
Double Helix!

The University of Georgia

- Double helix
  - Chromosomes
- Replication
  - DNA-polymerase
- Transcription
- Translation

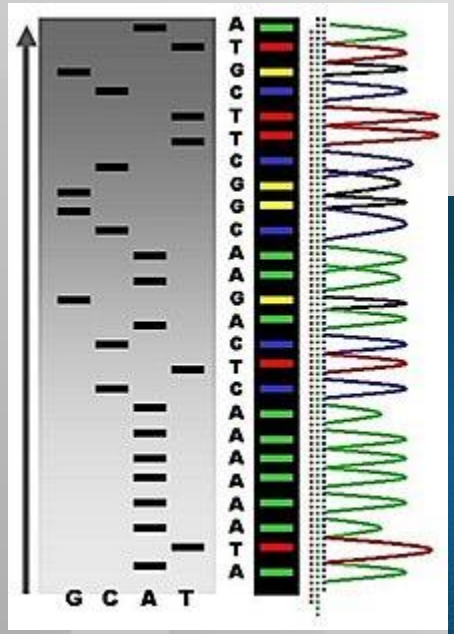


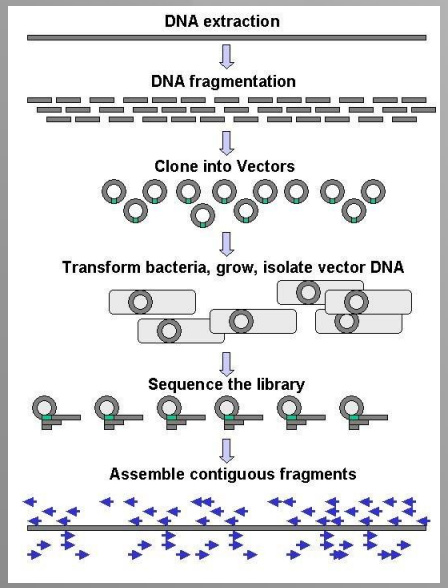


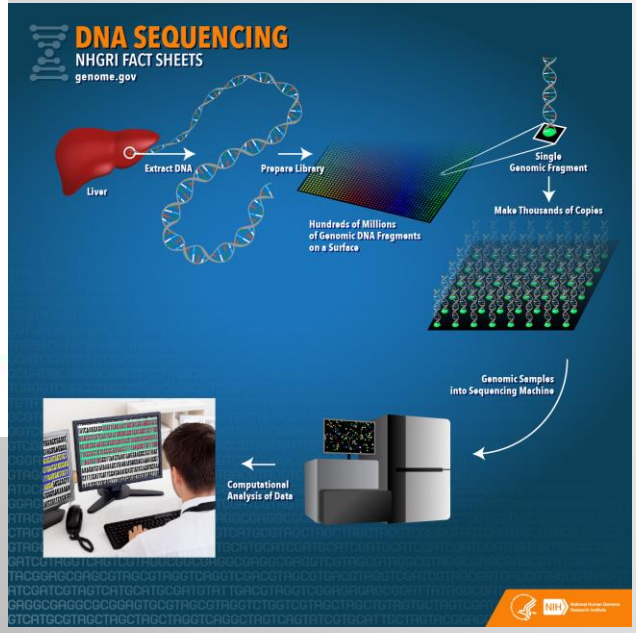


Sequencing

- Restriction Enzyme
- Gel & Electrophoresis

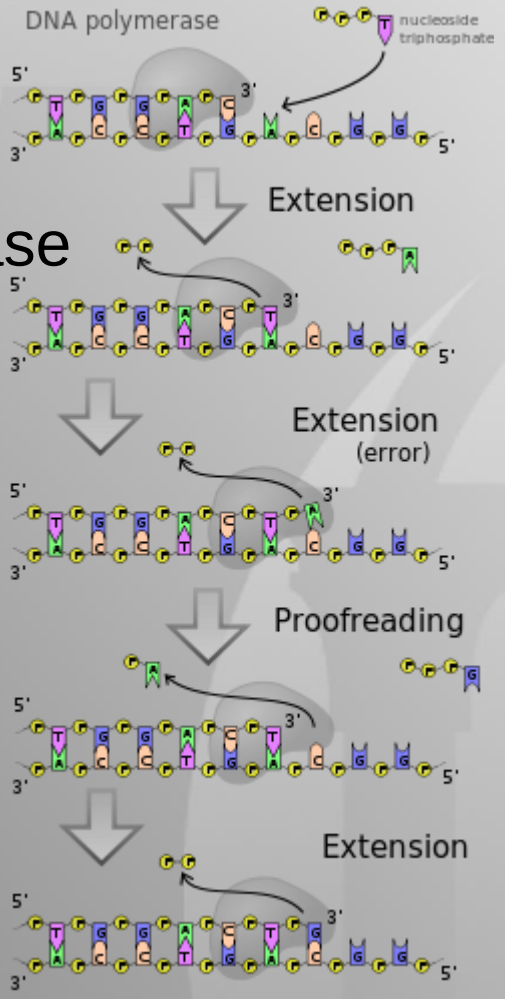


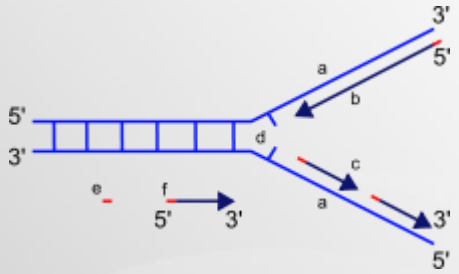


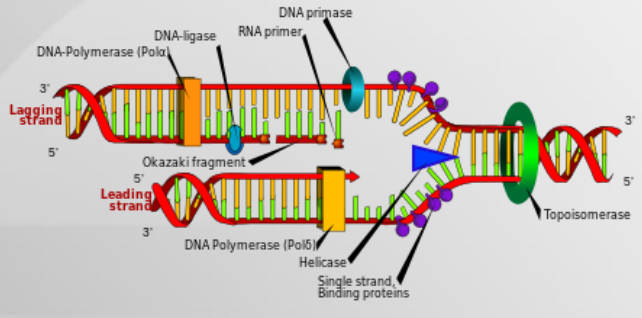


Replication

- Double helix
- Replication
  - DNA-polymerase
- Transcription
- Translation

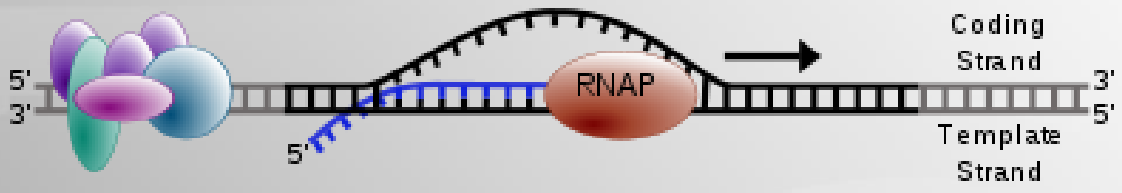


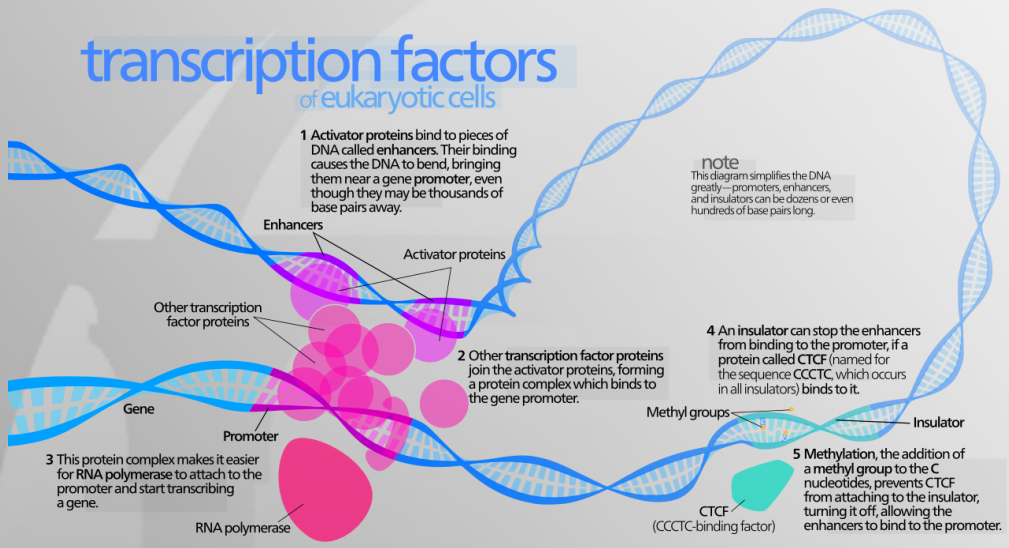


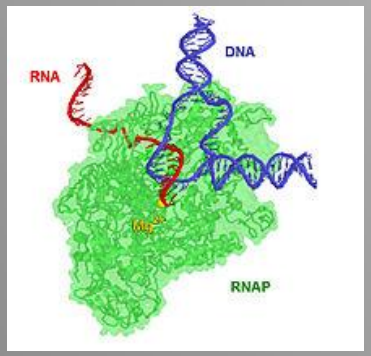


Transcription

- Double helix
- Replication
- Transcription
  - RNA Polymerase
  - Transcription Factors
  - Expression control
- Translation



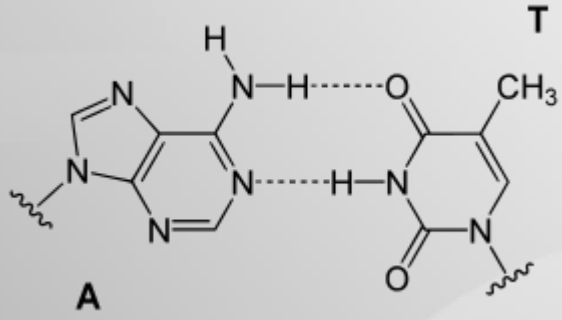


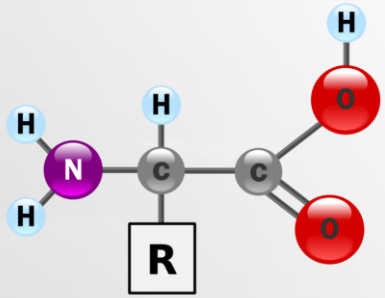


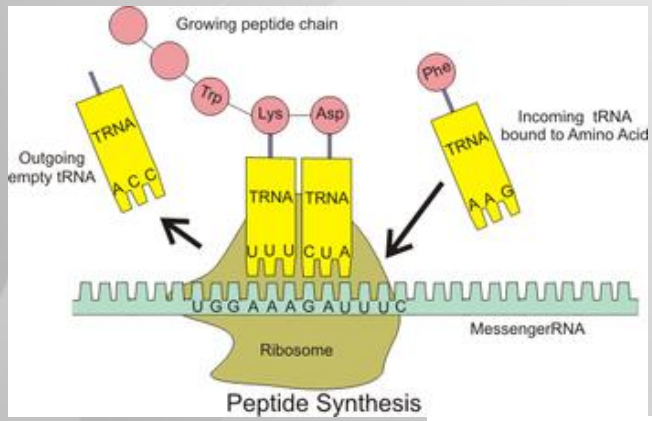
Translation

The University of Georgia

- Double helix
- Replication
- Transcription
- Translation
  - Ribosome
  - mRNA
  - tRNA
  - Nucleic acid to Amino acid







Second Codon Letter

First Codon Letter	T	C	A	G	Third Codon Letter
T	TTT – phe TTC – phe TTA – leu TTG – leu	TCT – ser TCC – ser TCA – ser TCG – ser	TAT – tyr TAC – tyr TAA – stop TAG – stop	TGT – cys TGC – cys TGA – stop TGG – trp	T C A G
C	CTT – leu CTC – leu CTA – leu CTG – leu	CCT – pro CCC – pro CCA – pro CCG – pro	CAT – his CAC – his CAA – gln CAG – gln	CGT – arg CGC – arg CGA – arg CGG – arg	T C A G
A	ATT – ile ATC – ile ATA – ile ATG – start/met	ACU – thr ACC – thr ACA – thr ACG – thr	AAT – asn AAC – asn AAA – lys AAG – lys	AGT – ser AGC – ser AGA – arg AGG – arg	T C A G
G	GTT – val GTC – val GTA – val GTG – val	GCT – ala GCC – ala GCA – ala GCG – ala	GAT – asp GAC – asp GAA – glu GAG – glu	GGT – gly GGC – gly GGA – gly GGG – gly	T C A G

Code

The University of Georgia

```
#dictionary
def main():
    f1 = open("M:\\tubulin_a.txt","r")
    seq1 = f1.readline()
    f2 = open("M:\\tubulin_b.txt", "r")
    seq2 = f2.readline()
    aa1 = {}
    aa2 = {}
-----------------------------------
```

Code

The University of Georgia

```
-------------------------------------
    for i in range(len(seq1)):
        if seq1[i] in aa1:
            aa1[seq1[i]]+=1
        else:
            aa1[seq1[i]]=1
    for i in range(len(seq2)):
        if seq2[i] in aa2:
            aa2[seq2[i]] += 1
        else:
            aa2[seq2[i]] = 1
    for i in aa1:
        print(i,aa1[i],aa1[i]/len(seq1),aa2[i],aa2[i]/len(seq2))
```

2D Array

- List = [[0]*n, for I in range(n)]


Thank You


TIME
FOR A
BREAK