# CMPSC 300
# Bioinformatics
# Spring 2021

## Lab 3 Assignment:
## Writing Python code to complete an Analysis of DNA

**Submit deliverables through your assignment GitHub repository and complete the Google Form.**

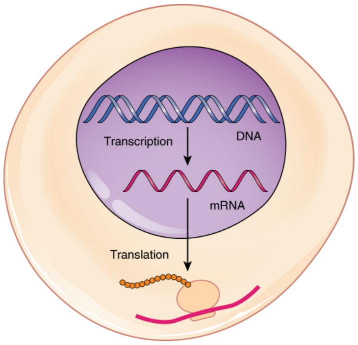
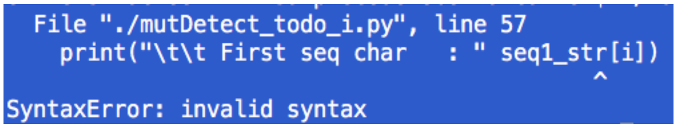


Figure 1: The central dogma of biology in a nutshell (i.e., DNA → RNA → Protein.). Note that the DNA remains safely in the nucleus. From Trascription, RNA is created and is sent out in to the cell to be translated into protein products.

## Objectives

In research in Bioinformatics, writing computer code is frequently an important part of completing an analysis. When writing code, the researcher may have to spend time to ensure that the code is working correctly by checking for bugs or typographical errors that would prevent a clean execution of the code, illustrated in Figure 2. In this lab, you are invited to spend time to check for bugs in a provided piece of code and then to implement a function to complete the analysis.

As shown in Figure 1, we note the Central Dogma of Biology. In this lab, we will be working with this central aspect of biology in our code to perform transcription and translation.



```
  File "./mutDetect_todo_i.py", line 57
    print("\t\t First seq char   : " seq1_str[i])
                                            ^
SyntaxError: invalid syntax
```

Figure 2: Finding bugs in code is a normal part of programming.

## Clone Your Assignment Repository

In this section, we will be using `Git` commands. It is suggested that the reader refer to online searches for help. For example, GitHub provides good documentation at the following link; `https:`

//git.github.io/htmldocs/git.html.

In many cases, you will be given a new repository containing assignment materials and you will save your files in this assignment repository as you continue to work on them. Copy and paste the assignment repository cloning command into your terminal to create your assignment repositories. Be sure to place your assignment repositories in a directory such as cs300/ to keep your class materials organized by class.

Today's assignment repository can be found at the below link to a GitHub Classroom repository. Here you will work on your assignment and then push your work to the cloud where the instructor will be able to view your work for grading. Often, there will be files in your assignment repositories which you are to edit before you submit them by using the below commands for git.

https://classroom.github.com/a/9xaAIGx0

To use this link, please follow the steps below.

- Click on the link and accept the assignment
- Once the importing task has completed, click on the created assignment link which will take you to your newly created GitHub repository for this lab,
- Clone this repository (bearing your name) and work locally
- As you are working on your lab, you are to commit and push regularly. The commands are the following.
  - git add -A
  - git commit -m ''Your notes about commit here''
  - git push

### Check Your Submission

After you have pushed your work to your repository, please visit the repository the GitHub website (you may have to log-in) to verify that your files were correctly sent. Importantly, please cheack that GitHub Actions has checked your submission. For this, look for an orange dot that will turn into a red check mark to indicate errors, or a green check on the top line of your repository to indicate that all checks have passed.

## Optional Reading

If you feel that you need some extra help with Python, then please read Chapters 2 and 3 in the "ThinkPython" book.

- The main page for the online free textbook is at link:
  - https://greenteapress.com/wp/think-python/
- The PDF of the book can be found at link:
  - https://greenteapress.com/thinkpython/thinkpython.pdf.

### Analysis Program

The program that you have been given in the `src/mutDetect_i.py` is supposed to compare sequences and to perform basic translations of two user-entered sequences. It is then supposed to compare the protein sequences of the two DNA sequences to find changes in product. Unfortunately, this code was written hastily and, as a result, contains TWELVE (12) basic coding bugs (i.e., typographical errors) that prevent the code from working properly. **Your task is to complete the code by fixing the errors to allow it to run and to display the output shown below.**

Your output should look like the following from your Docker container. *Note, you will have to provide a dummy parameter to run the code or the help screen will be displayed instead.*

```
bioinformaticsNumberOneFan$ python3 mutDetect.py run

 Welcome to mutDetect!
 A program to compare DNA, make protein and compare protein sequences.

 ____ INPUT DNA SEQUENCES ____
Enter a sequence :CTTCTT
 + Length of first sequence  : 6
Enter a sequence :CTTCTC
 + Length of second sequence : 6

 ____ COMPARING DNA SEQUENCES ____
 + Sequences are both same length:  True
 [-] Bases not the same at pos:  5
 First seq char   :  t
 Second seq char  :  c
 + Original DNA       : cttctt , length is : 6
 + RNA from DNA     : cuucuu
 + DNA from RNA     : cttctt
 + PROTEIN from RNA   : LL
 + protein1 sequence  : LL
 + Original DNA       : cttctc , length is : 6
 + RNA from DNA     : cuucuc
 + DNA from RNA     : cttctc
 + PROTEIN from RNA   : LL
 + protein2 sequence  : LL

 ____ COMPARING PROTEIN SEQUENCES ____
 + Sequences are both same length:  True
 The sequences are the same.

 ____ DETECTING SILENT MUTATIONS ____
 DNA seq 1:  cttctt
 DNA seq 2:  cttctc
 Protein sequence 1:  LL
 Protein sequence 2:  LL
[+] Silent mutation has been detected.
```

## Silent Mutations and Detection Function in Code

We discussed several ailments which may be the result of mutations in DNA. One more interesting mutation is called the "Silent Mutation" in which mutated DNA produces protein code with is exactly the same as non-mutant DNA. In your code, you are challenged with the task of adding a silent mutation detection function to your code in `src/mutDetect_i.py` . In addition, in the tt writing/reflections.md, you are to briefly discuss these mutations and to offer an example of a disorder where silent mutations are though to be responsible.

### Required Deliverables

- Your completed and working (bug free) code should be saved in your GitHub Classroom repository as `src/mutDetect_i.py` .
- Write a reflection of about 100 words to describe what is meant by a "Silent mutation" and to discuss any type of disorder which may be categorized as a this kind of mutation. It is expected that you will have to perform an online search to investigate a disorder of a silent mutations in more detail. Your work will be saved in `writing/reflections.md`.

## Grading

The grade that you receive for this lab assignment will be based on the following:

- 60% Complete and correctly running Python programming (free of bugs!).
- 20% The implementation of the `detectSilentMutation`() function in the code of `src/mutDetect_i.py`.
- 10% Your reflection document where you discuss silent mutations. Here you are to name and briefly discuss a particular disorder associated with these types of mutations.
- 10% Complete GitHub Actions CI build-pass corresponding to all the GatorGrader checks passing.

Please see the Technical Leaders or the instructor if you have questions about the assignment submission.