University of Puerto Rico
Puerto Rico - IDeA Networks of Biomedical Research Excellence

Python3 Part 1 – Crash Course in Python3 for Future STEM Coders

Judith S. Rodriguez-Martinez, M.S.
Ph.D. Candidate
Penn State University – University Park Campus
jzr5814@psu.edu

June 5, 2023 at UPR-Ciencias Médicas

- The following material is the result of PR-INBRE research and curriculum development effort to provide a set of educational materials for research training and curriculum changes in biology programs across the island to support PR-INBRE efforts to establish a Community of Practice in Bioinformatics that offers a fruitful environment to increase computational and bioinformatics skills among traditional researchers and students (undergraduate and graduate) in the island. They have been developed as a part of the NIH funded project "**Puerto Rico IDeA Network Biomedical Research Excellence (PRINBRE)**" (Award Number 5P20GM103475).

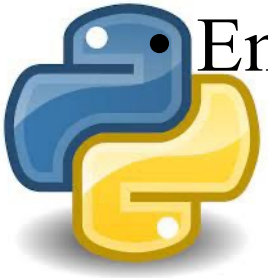
- Most recent versions of these presentations can be found at **http://inbre.hpcf.upr.edu/**.

Competencies

Record and write simple and common **Python scripts to deal with Bioinformatic needs for biological data analyses** using a Jupyter Notebook

Objectives

- Use **Google Colab as an environment** to practice and learn common **Python** lines of code.
- Formulate simple calculations using **Python**
- Identity the following **datatypes**: integer, float, and string
- Employ **variables** for different datatypes



Objectives

- Manipulate integer, float, and string datatypes
- Use **bioinformatic tools** to find a protein-coding sequence
- **Evaluate a protein-coding sequence** of interest and calculate its GC content.
- **Debug** code

Target Audience

- This training is addressed to beginners, highly motivated (eager to learn what is needed in order to be competitive without the tendency to self-limit when learning computational skills by saying that is difficult) wanting to become familiar with the **Python** programming language and become the Script Master of their bioinformatic analysis.

The importance of Python

- R is great however… eventually, you'll have to use Python language
- No matter your field, python will be a necessity (chemistry, biology, engineering, ecology, mathematics, etc)
- Knowing Python gives you an edge when job searching and navigating graduate research
- The majority bioinformatic tools are python-based and will lead you to understand under the hood code in order to successfully execute said program's purpose.

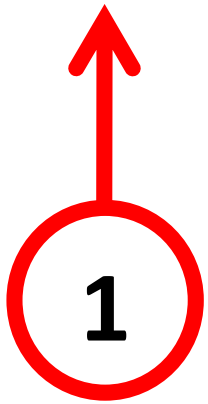
Lesson 1.0.0

Connecting to Google Colab to learn common Python lines of code

(Gmail account ready for colab.research.google.com)

Google Colab

https://colab.research.google.com



Activate the link

Activate colab.research.google.com

colab.research.google.com

Welcome To Colaboratory

File Edit View Insert Runtime Tools Help

Share

Table of contents

Getting started
Data science
Machine learning
More Resources
Featured examples
Section

+ Code + Text Copy to Drive Connect Editing

Welcome to Colab!

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view, and the command palette.

What is Colab?

Colab, or "Colaboratory", allows you to write and execute Python in your browser, with

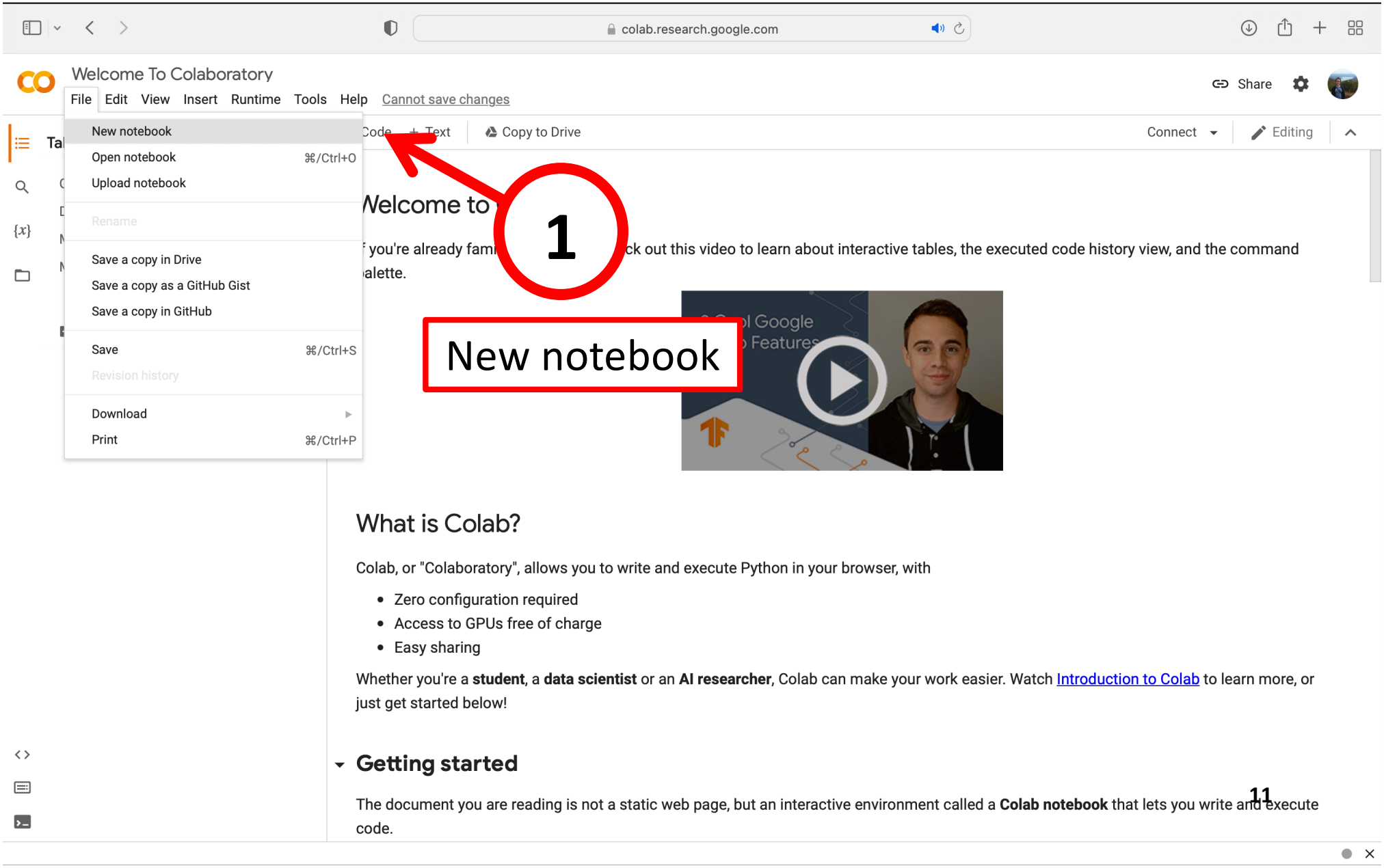
- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

Whether you're a **student**, a **data scientist** or an **AI researcher**, Colab can make your work easier. Watch Introduction to Colab to learn more, or just get started below!

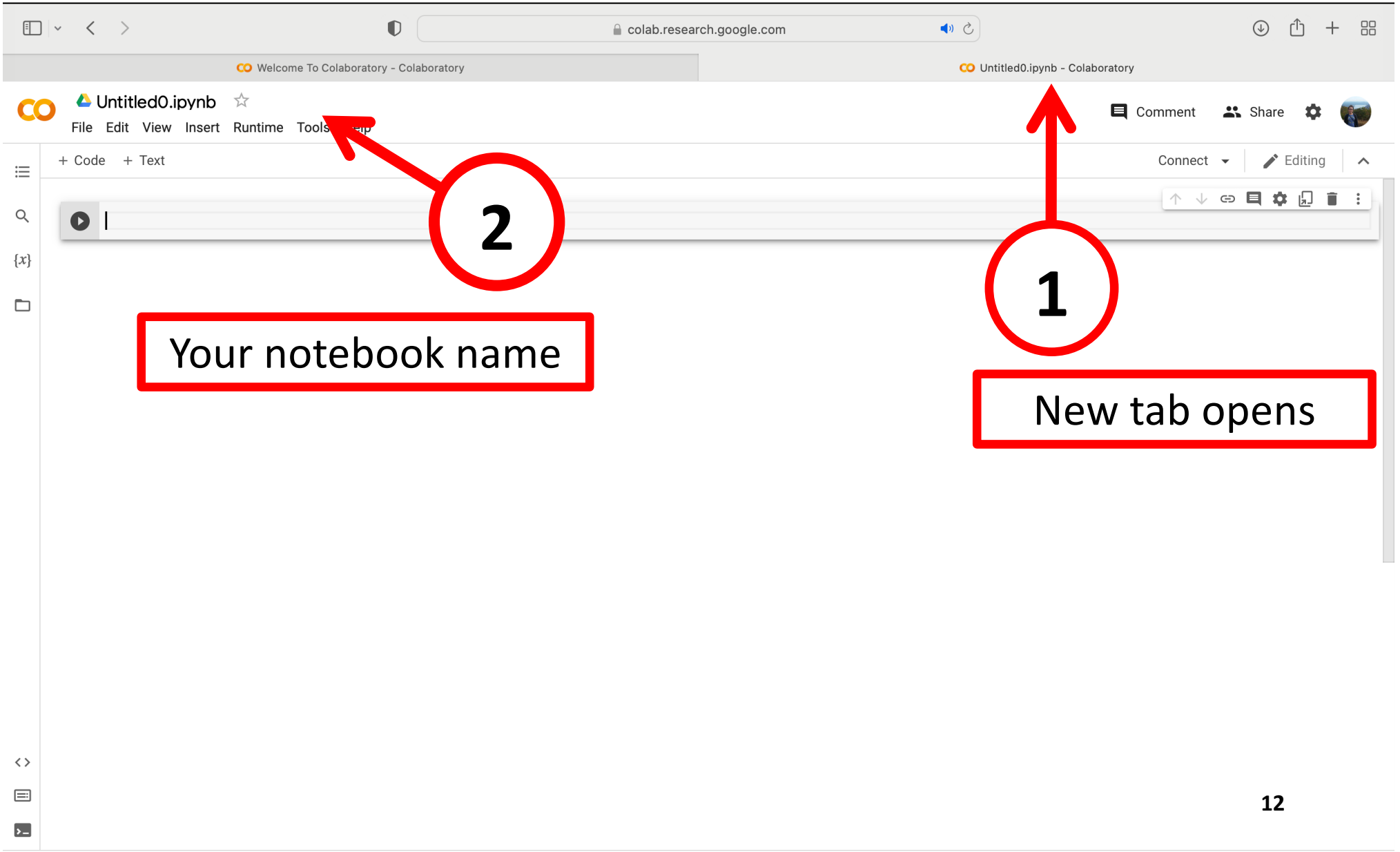
Getting started

The document you are reading is not a static web page, but an interactive environment called a **Colab notebook** that lets you write and execute code.

Open a New notebook



Welcome to your new notebook!



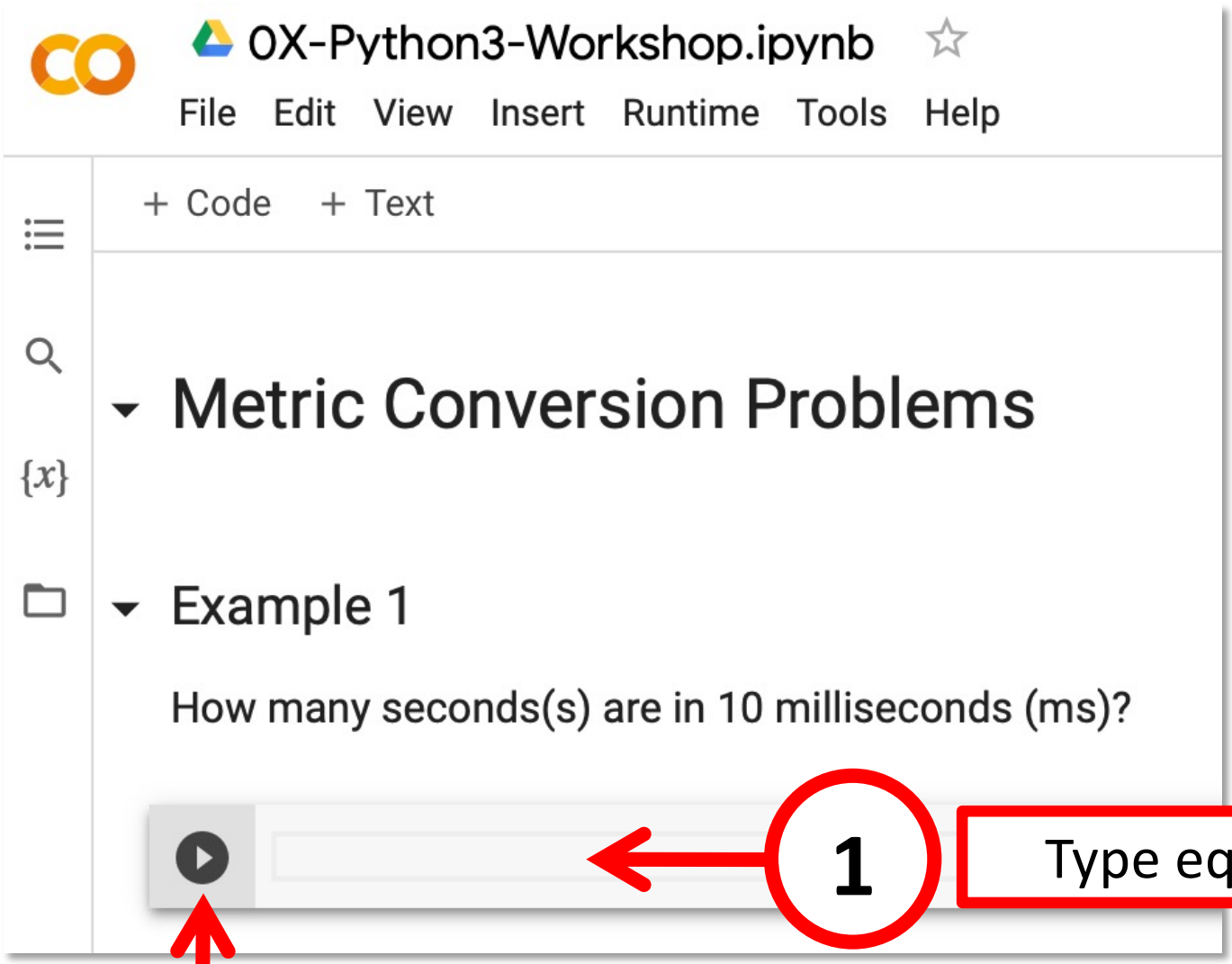
Lesson 1.1.0

Test snippets of code using the Python interpreter

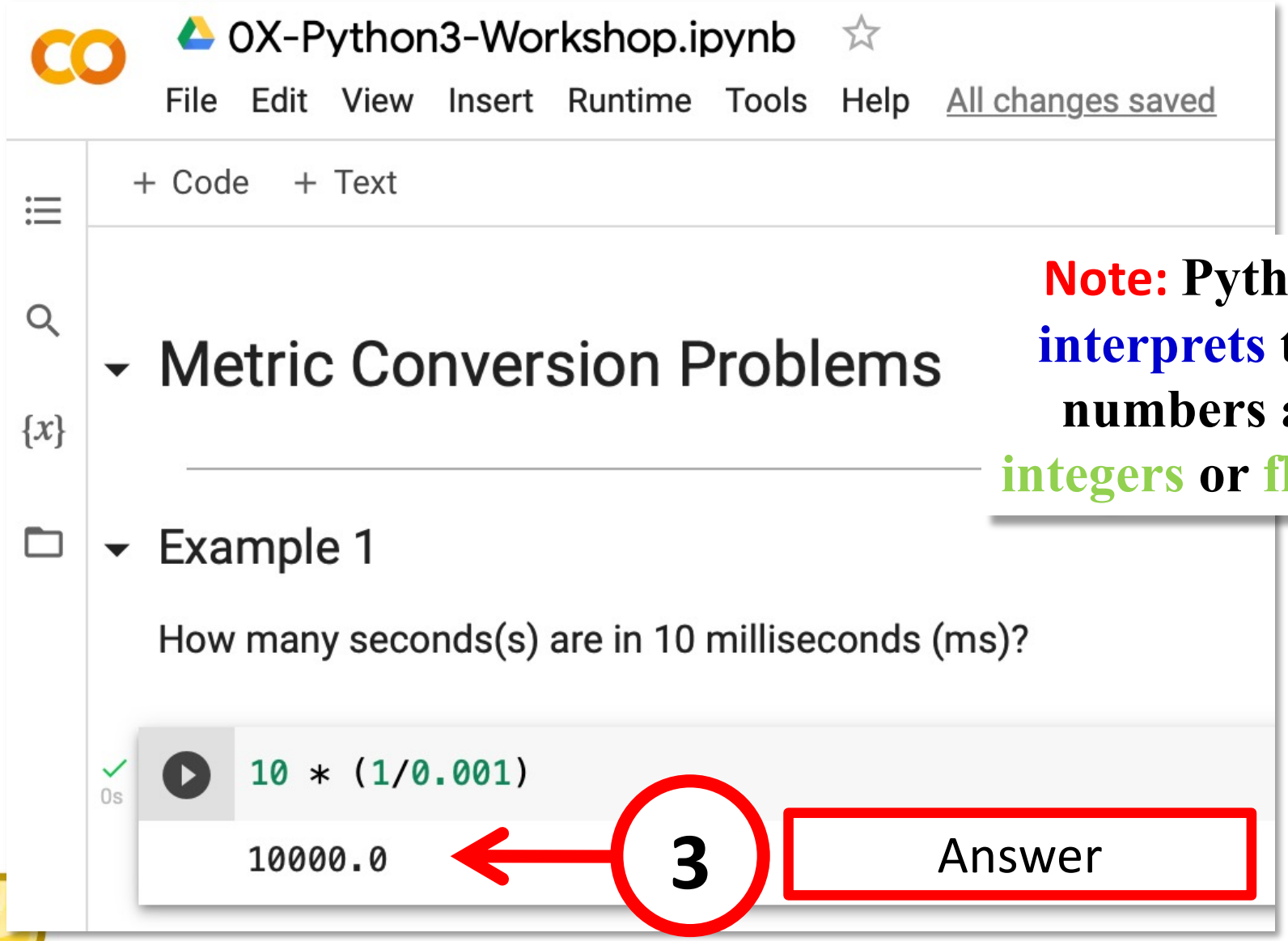
Python is your new calculator!

Addition	+
Subtraction	-
Multiplication	*
Division	/
Modulus	%
Floor division	//
Exponent	**

Let's solve metric conversion problems!



Let's solve metric conversion problems!



Note: Python **interprets** the **numbers as integers or floats**

Try it on your own!

▾ Problem 1

How many 5µL are in mL?

ANSWER

Try it on your own!

Problem 1

How many 5µL are in mL?

```
[ ]  5 * (0.001/1)
```

```
0.005
```

Try it on your own!

▾ Problem 2

How many kg are in 0.0034 g?

ANSWER

Try it on your own!

Problem 2

How many kg are in 0.0034 g?

```
[ ]  0.0034 * (0.001/1)
```

```
3.4e-06
```

Try it on your own!

▾ Problem 3

How many moles are there in 50g of water(H2O)?

Tip: Use variables

ANSWER

Try it on your own!

Problem 3

How many moles are there in 50g of water(H2O)?

Tip: Use variables

Note: Facilitate your programing employing **variables**!

```
#Given:
mass_of_h2o = 50
molecular_mass_of_h2o = (1)*2 + 16

#Solution
result = mass_of_h2o/molecular_mass_of_h2o
result
```

```
2.7777777777777777
```

Lesson 1.2.0
Python datatypes

Integers and floats and strings… oh my!

Tip: Knowing the function type(*object*) of objects you are working with during coding can help you get around programming obstacles!

Datatypes

int()	12
float()	12.0
str()	"hola"
list()	[1,2,3, "hola"]
dictionary()	{"blue":"azul"}

Integers and floats and strings… oh my!

- Identify datatypes of Python objects using type() functions.

Answer

Integers and floats and strings… oh my!

▾ Identify datatypes of Python objects using type() functions.

```
[15] type(mass_of_h2o)

0s

     int
```

Integers and floats and strings… oh my!

▾ What is the datatype of molecular_mass_of_h2o?

Answer

Integers and floats and strings… oh my!

▾ What is the datatype of molecular_mass_of_h2o?

```
[16] type(molecular_mass_of_h2o)
0s

     int
```

Int and Float Types can be manipulated

int(*object*)

float(*object*)

How many seconds(s) are in 10 milliseconds (ms)?

```
[23] 10 * (1/0.001)

     10000.0

[30] seconds = 10 * (1/0.001)
     int(seconds)

     10000
```

Identify datatypes of Python objects using type() functions.

```
[31] mass_of_h2o

     50

[27] type(mass_of_h2o)

     int

[29] float(mass_of_h2o)

     50.0
```

Integers and floats and strings… oh my!

```
[33] 'hola puerto rico'
```

```
'hola puerto rico'
```

▼ Assign a variable to this string and print variable

Hint: Search for python's print() function

Answer

Integers and floats and strings… oh my!

```
[33] 'hola puerto rico'
```

```
'hola puerto rico'
```

▼ Assign a variable to this string and print variable

Hint: Search for python's print() function

```
[34] greeting = 'hello puerto rico'
     print(greeting)
```

```
hello puerto rico
```

There are a multitude of functions to evaluate and manipulate your strings

len(*str*)
str.count("")
str.replace("", "")
str.capitalize()
str.split("")
"".join(*str*)

Note: These are my favorite and most used string functions!

Indexing a String

Strings have a property called indexes, which are positions in the string

```
[4]  sequence="ATGTGGTGG"
```

Indexing a String

Strings have a property called indexes, which are positions in the string

```
[4]  sequence="ATGTGGTGG"
     sequence
```

```
'ATGTGGTGG'
```

Indexing a String

Strings have a property called indexes, which are positions in the string

```
[4]  sequence="ATGTGGTGG"
     sequence
```

```
'ATGTGGTGG'
 012345678
```

Indexing a String

Strings have a property called indexes, which are positions in the string

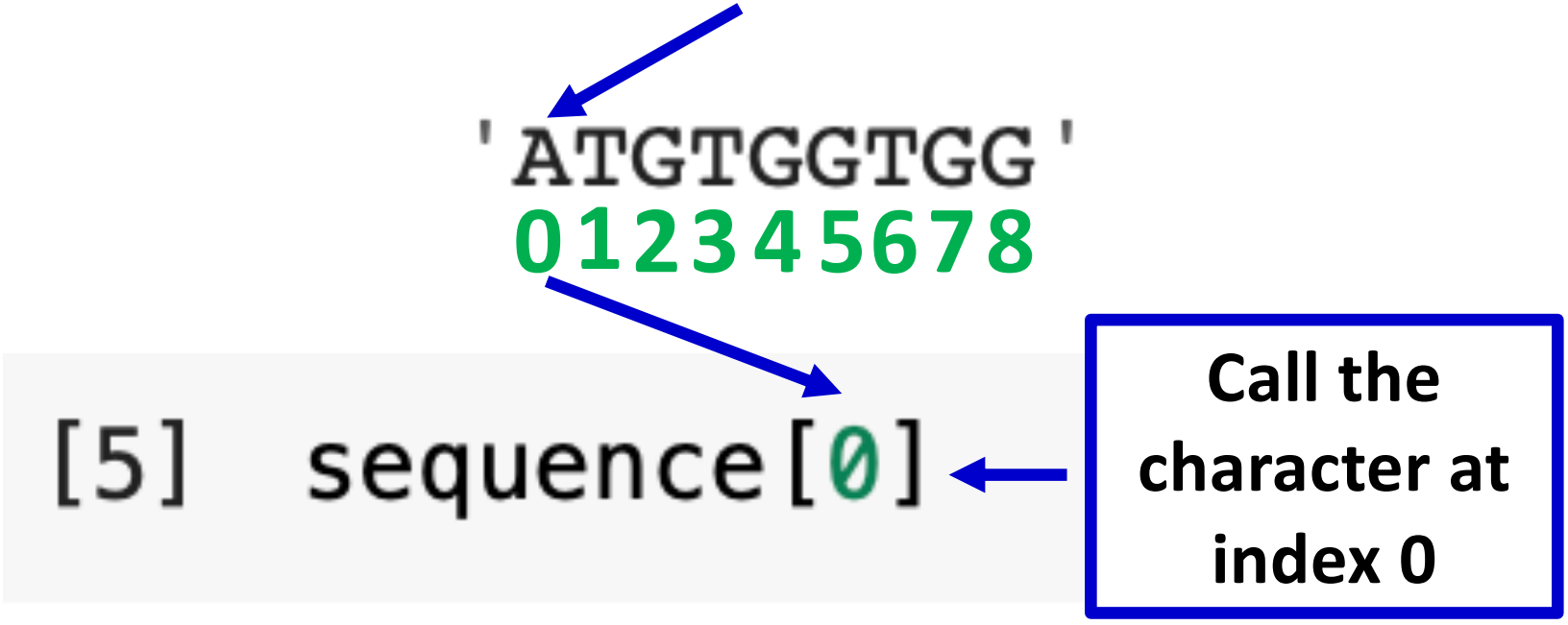
```
[4]  sequence="ATGTGGTGG"
     sequence
```

```
'ATGTGGTGG'
 012345678
```

The index of the first position is 0

Indexing a String

The index of the first position is 0



Indexing a String

The index of the last position is can be 8 or -1


```
[7]  sequence[8]
```

Call the last character using 8

```
'G'
```

```
[8]  sequence[-1]
```

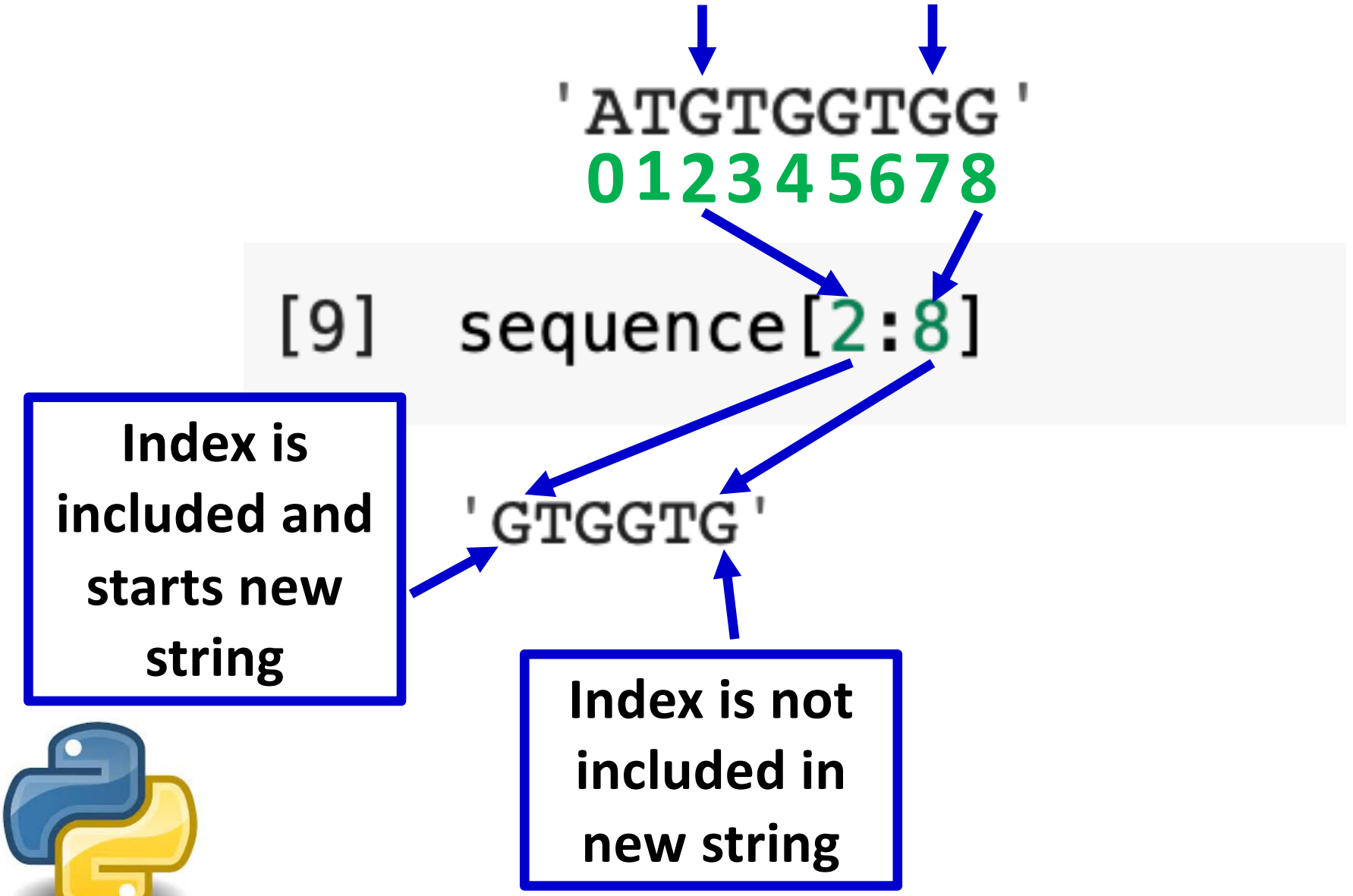
Call the last character using -1

```
'G'
```

Note: Using **-1** is useful when last index is unknown!

Indexing a String

Index a range of the sequence

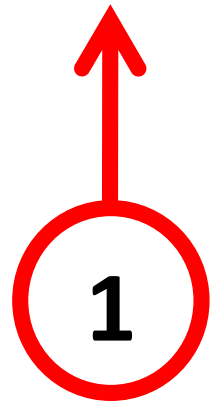


Lesson 1.3.0

Evaluate the GC content of a DNA sequence.

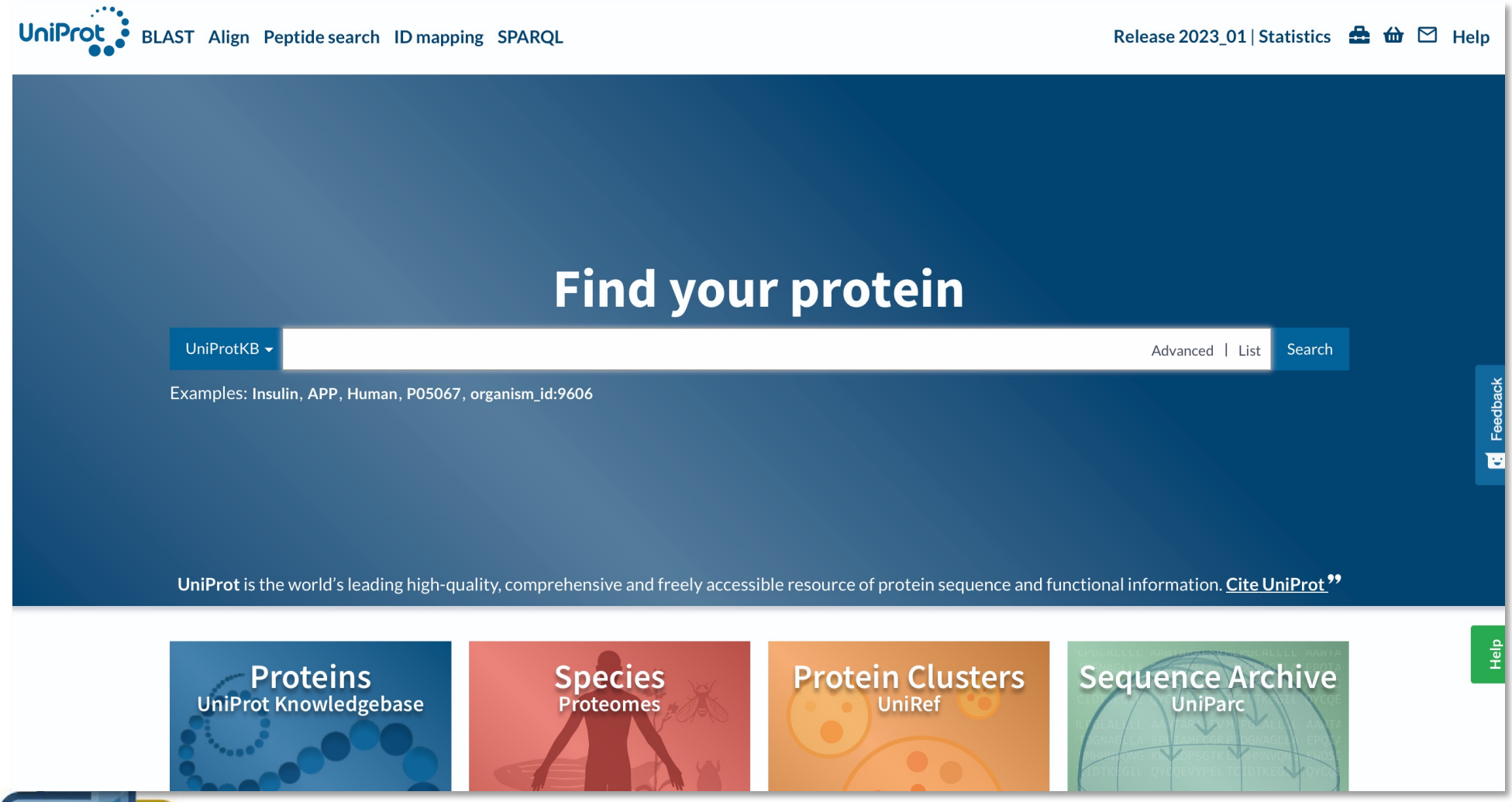
Connect to Uniprot

https://www.uniprot.org

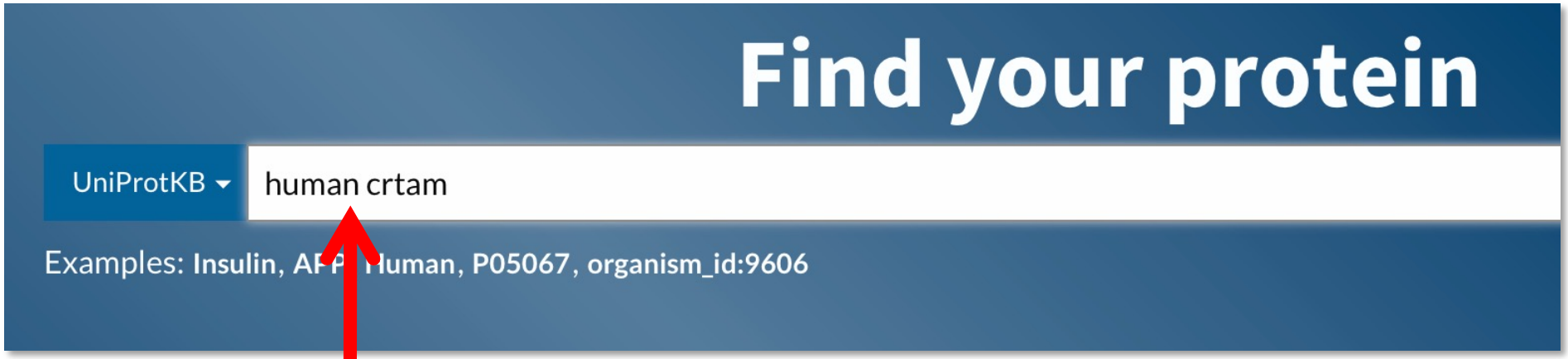


Activate link

Activate uniprot.org



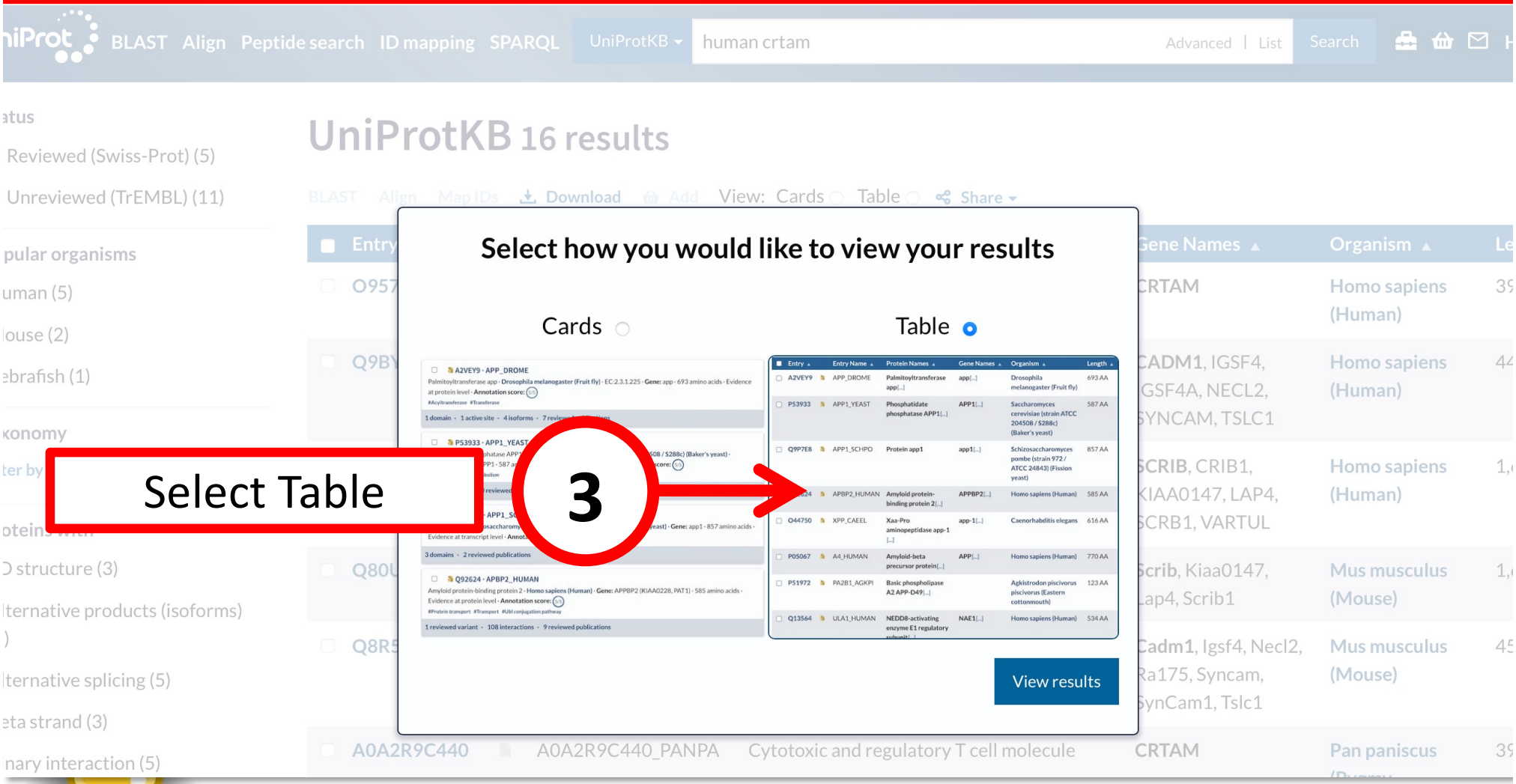
CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.



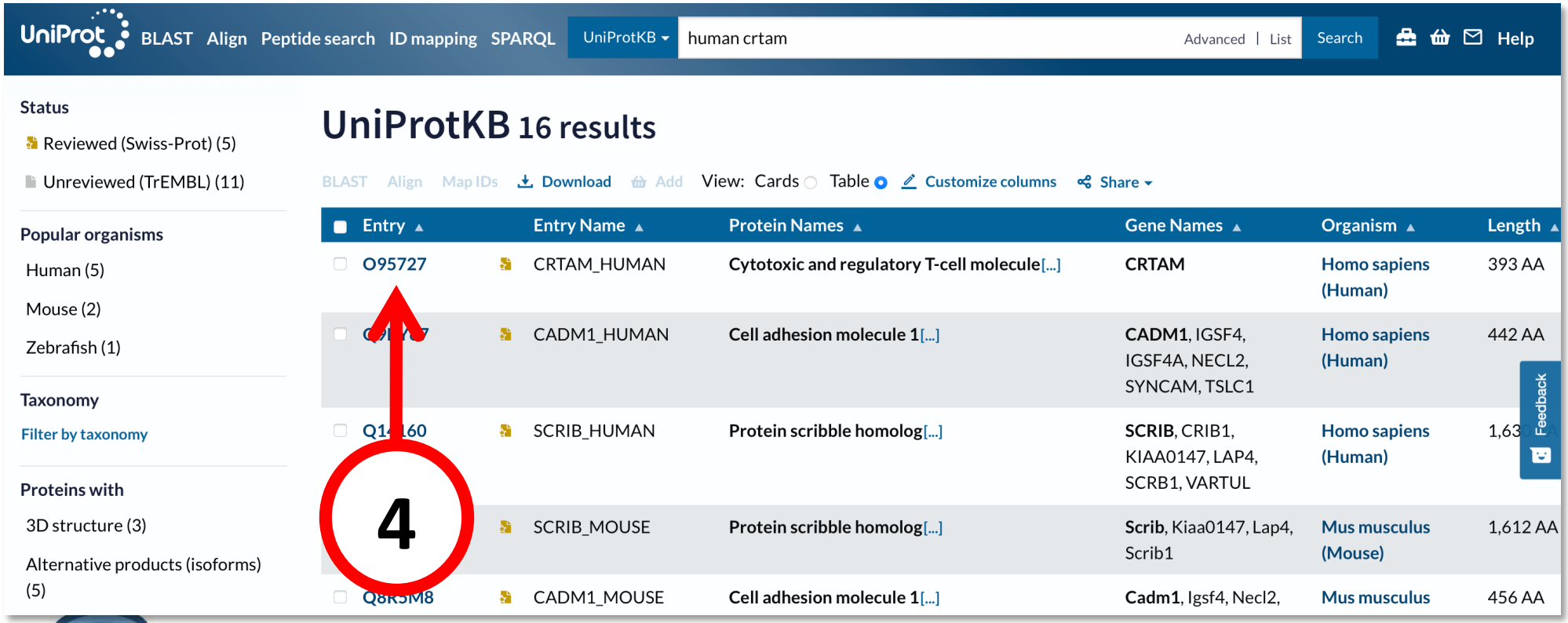
2

Search CRTAM

CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

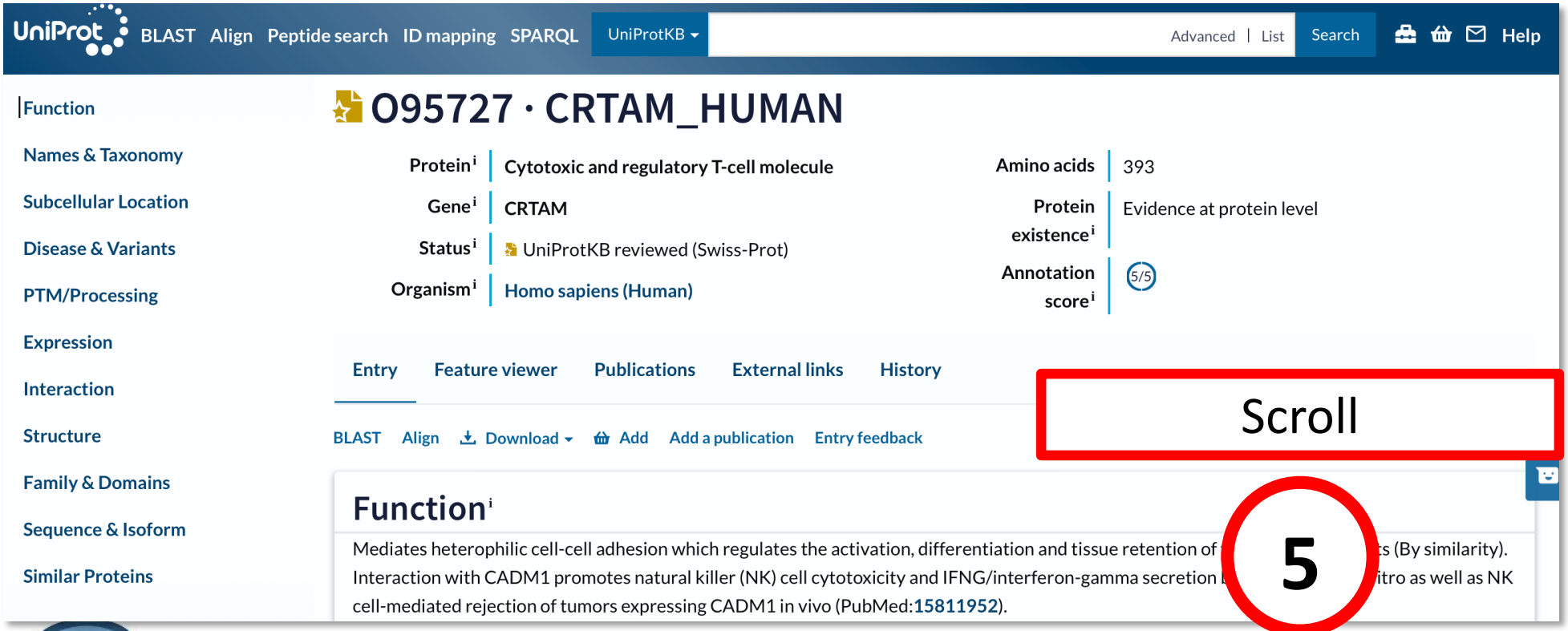


CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.



Activate link to Entry

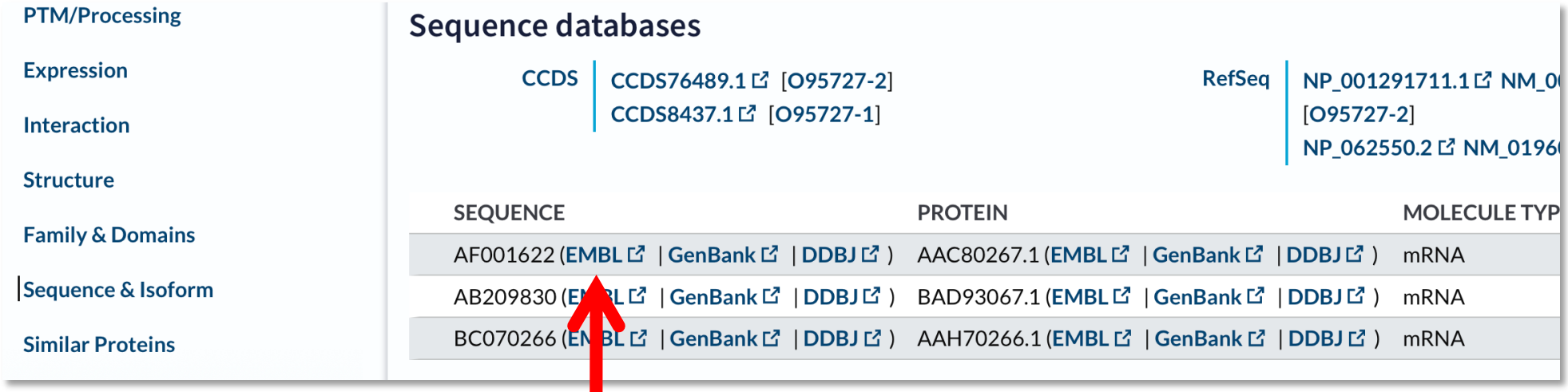
CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.



Scroll

5

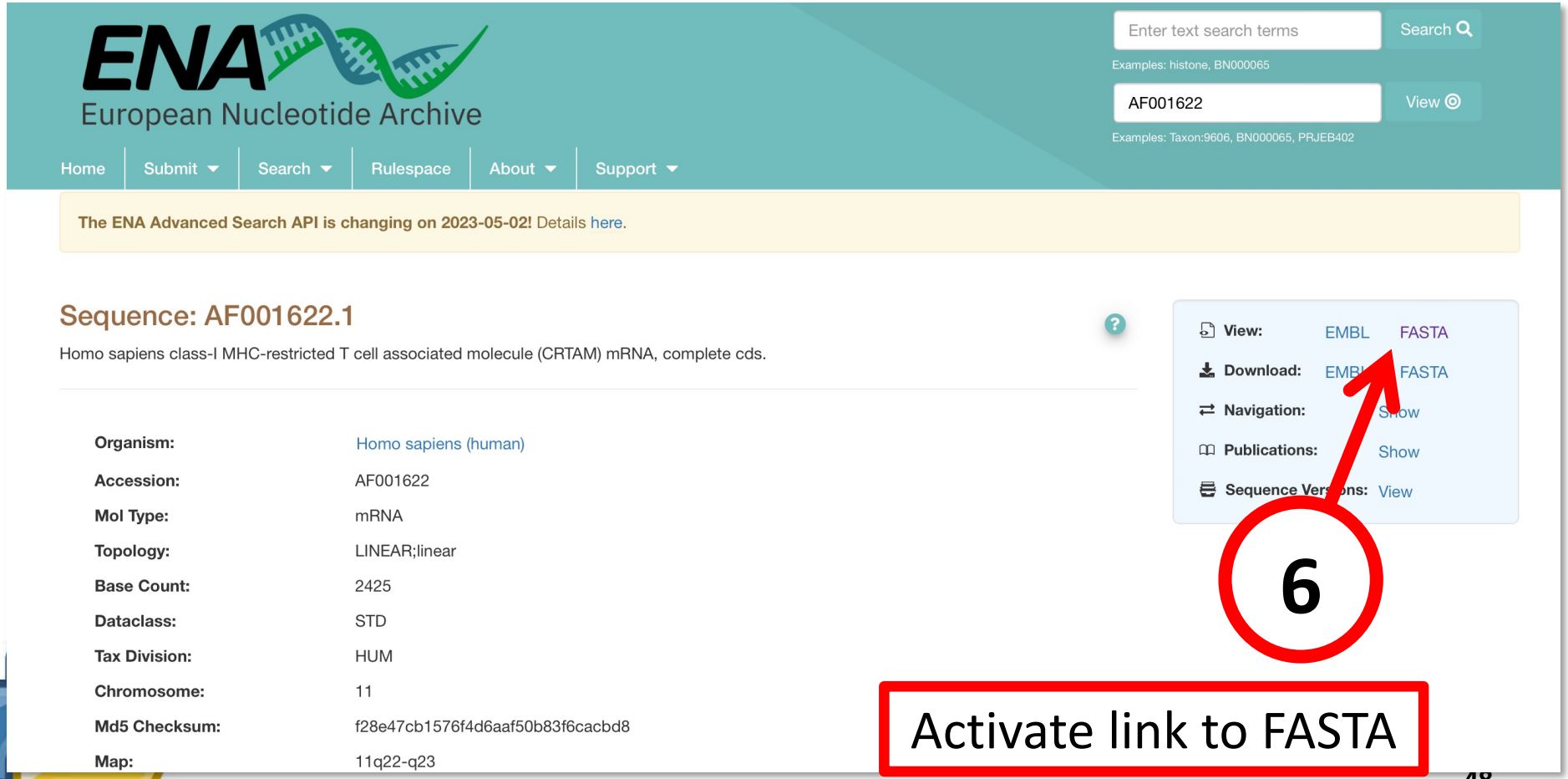
CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.



6

Activate link to EMBL

CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.



CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

>ENA|AF001622|AF001622.1 Homo sapiens class-I MHC-restricted T cell associated molecule (CRTAM) mRNA, complete cds.
ATGTGGTGGAGAGTTCTCAGCTTGCTGGCATGGTTCCCCTTGCAAGAGGCCTCTCTGACT
AACCACACAGAAACCATCACCGTGGAGGAAGGCCAGACGCTCACTCTAAAGTGTGTCACT
TCTCTGAGGAAGAACTCCTCCCTCCAGTGGCTGACCCCCTCAGGGTTCACCATTTTTTTA
AATGAGTATCCTGCTTTAAAAAATTCCAAATACCAGCTTCTTCATCACTCGGCCAATCAG
CTCTCCATCACTGTGCCTAACGTAACCCTGCAAGATGAAGGCGTGTACAAGTGCTTACAT
TACAGCGACTCTGTAAGCACAAAGGAAGTGAAAGTGATTGTGCTGGCAACTCCTTTCAAG
CCAATCCTGGAAGCTTCAGTTATCAGAAAGCAAAATGGAGAAGAACATGTTGTACTCATG
TGCTCCACCATGAGAAGCAAGCCCCCTCCGCAGATAACCTGGCTACTTGGGAATAGCATG
GAAGTGTCCGGTGGAACGCTCCATGAATTTGAAACTGATGGGAAGAAATGTAATACTACC
AGCACTCTCATAATCCACACTTATGGCAAAAATTCAACGGTGGACTGCATTATCCGACAC
AGAGGCCTGCAAGGGAGAAAACTAGTAGCACCCTTCCGGTTTGAAGATTTGGTTACTGAT
GAAGAGACAGCTTCAGATGCTCTGGAGAGAAACTCTCTATCCTCTCAAGACCCACAGCAG
CCCACCAGTACTGTCTCAGTAACGGAAGATTCTAGTACATCGGAGATTGACAAGGAAGAG
AAAGAACAAACCACTCAAGATCCTGACTTGACCACCGAAGCAAATCCTCAGTATTTAGGA
CTGGCAAGAAAGAAAAGTGGCATCCTGCTGCTCACGCTGGTGTCCTTCCTCATTTTCATA
CTCTTCATCATAGTCCAGCTCTTCATCATGAAGCTGAGGAAAGCACATGTGATATGGAAG
AGAGAAAACGAAGTTTCAGAACACACACTAGAAAGTTACAGATCAAGGTCAAATAATGAA
GAAACATCATCTGAAGAGAAAAATGGCCAATCTTCCCACCCTATGCGTTGCATGAACTAC
ATCACAAAGTTGTACTCAGAAGCAAAAACAAAGAGGAAGGAAAATGTACAACATTCAAAA
TTAGAAGAAAAGCACATCCAAGTACCAGAGAGTATTGTGTAGTGCTCTCTGCAATGGAAC
ATGTGATTTCAGGGTTGCCGCAGTGTCACCTCAGTGGACCAGCCTGGGGGAAGGAGCTTA
ATTGCTGAGACATTAATAATGACCTCTTAGTGCAATGCAAGATGGTGTCCTCGGATAATG
ATCTGCCCCGGAGCTAGGGCAGCAACATGAGGACCAAACCATGCACATAAAGCTTGTAGT
TTAAAAAAGAAAAGCAAAAAAATAATTATGCCTGACACTACTTCAGAGCAGGAGGATTCT
ACGAAGCCTTGGGGATCAGGGTCAGTGTGAGCAGCTAACATCCTACCTCAAATGGAACAG
GATTTTTTGATGCTTTGCTCTAATGGAACTGTTTTAAAAATTTTTTTTTCTTTTTAATAT
TTTCTTCTGGTCACAAAATAAAGAAATTTGGGATGCAAAGTACCTAAAGATCTCTGATCC
TAAGAAGTTACTTCTGGCCAGGCGCGGTGGCTCATGCCTGTAATCCTAGCACTTTGGGAG
GCTGAGGTAGGCAGATCACTTGAGGTCAGGAGTTGGAGACCAGCCTGGCCAACATAGTGA
AACCCCGTCTCTACTAAAAATGCAAAAATTAGCCAGGCGTAGTGGTGCGCACCTGTAGTC
TCAGATACTTGGGAGGCTGAGGGTGGAGAATCGCTTGAACCTGGGAGGTGGAGATTGCAG
TGAGTCAAGATCTCACCACTGAACTCCAGCCTGGGCGACAGAGGGAGACTCTGTCTCAAA

7

Copy Sequence

CRTAM is a biomarker found in prostate cancer and was a major part of my undergraduate research.

```
sequence = """ATGTGGTGGAGAGTTCTCAGCTTGCTGGCATGGTTCCCCTTGCAAGAGGCCTCTCTGACT
AACCACACAGAAACCATCACCGTGGAGGAAGGCCAGACGCTCACTCTAAAGTGTGTCACT
TCTCTGAGGAAGAACTCCTCCCTCCAGTGGCTGACCCCCTCAGGGTTCACCATTTTTTTA
AATGAGTATCCTGCTTTAAAAAATTCCAAATACCAGCTTCTTCATCACTCGGCCAATCAG
CTCTCCATCACTGTGCCTAACGTAACCCTGCAAGATGAAGGCGTGTACAAGTGCTTACAT
TACAGCGACTCTGTAAGCACAAAGGAAGTGAAAGTGATTGTGCTGGCAACTCCTTTCAAG
CCAATCCTGGAAGCTTCAGTTATCAGAAAGCAAAATGGAGAAGAACATGTTGTACTCATG
TGCTCCACCATGAGAAGCAAGCCCCCTCCGCAGATAACCTGGCTACTTGGGAATAGCATG
GAAGTGTCCGGTGGAACGCTCCATGAATTTGAAACTGATGGGAAGAAATGTAATACTACC
AGCACTCTCATAATCCACACTTATGGCAAAAATTCAACGGTGGACTGCATTATCCGACAC
AGAGGCCTGCAAGGGAGAAAACTAGTAGCACCCTTCCGGTTTGAAGATTTGGTTACTGAT
TTCAGATGCTCTGGAGAGAAACTCTCTATCCTCTCAAGACCCACAGCAG
TGTCTCAGTAACGGAAGATTCTAGTACATCGGAGATTGACAAGGAAGAG
CACTCAAGATCCTGACTTGACCACCGAAGCAAATCCTCAGTATTT
GAAAAGTGGCATCCTGCTGCTCACGCTGGTGTCCTTCCTCATTTT
AGTCCAGCTCTTCATCATGAAGCTGAGGAAAGCACATGTGATATG
AGTTTCAGAACACACACTAGAAAGTTACAGATCAAGGTCAAATAA
TGAAGAGAAAAATGGCCAATCTTCCCACCCTATGCGTTGCATGAA
ATCACAAAGTTGTACTCAGAAGCAAAAACAAAGAGGAAGGAAAATGTACAACATTC
TTAGAAGAAAAGCACATCCAAGTACCAGAGAGTATTGTGTAGTGCTCTCTGCAATG
ATGTGATTTCAGGGTTGCCGCAGTGTCACCTCAGTGGACCAGCCTGGGGGAAGGAG
ATTGCTGAGACATTAATAATGACCTCTTAGTGCAATGCAAGATGGTGTCCTCGGAT
ATCTGCCCCGGAGCTAGGGCAGCAACATGAGGACCAAACCATGCACATAAAGCTTGTAGT
TTAAAAAAGAAAAGCAAAAAAATAATTATGCCTGACACTACTTCAGAGCAGGAGGATTCT
ACGAAGCCTTGGGGATCAGGGTCAGTGTGAGCAGCTAACATCCTACCTCAAATGGAACAG
```

8

Assign sequence to variable as a string

Note: Use triple quotations to fit the sequence in a block of code

Using the functions that you have learned, get to know the CRTAM DNA sequence.

Answer the following questions using complete sentences:

- What is the length of the CRTAM DNA sequence?
- How many Adenines are in your sequence?

Answer

Using the functions that you have learned, get to know the CRTAM DNA sequence.

Answer the following questions using complete sentences:

- What is the length of the CRTAM DNA sequence?
- How many Adenines are in your sequence?

```
[ ]  length = len(sequence)
     print(f'The length of my DNA sequence is {length} nucleotides.')
```

```
The length of my DNA sequence is 2465 nucleotides.
```

```
[16] A_count = sequence.count('A')
     print(f'This sequence has {A_count} Adenines in total.')
```

```
This sequence has 772 Guanines in total.
```

Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines + Total Cytosines}}{\text{Total Sequence Length}} * 100$$

Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines + Total Cytosines}}{\text{Total Sequence Length}} * 100$$

What are we missing?

Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

Return the total length of DNA sequence

Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

```
[10] length = len(sequence)
     print(f'The length of my DNA sequence is {length} nucleotides.')

     The length of my DNA sequence is 2465 nucleotides.
```

Tip: Using string formatting eases mixing different datatypes together

Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

Return total Guanines and Cytosines of the DNA sequence

Let's find the GC content of the CRTAM DNA sequence.

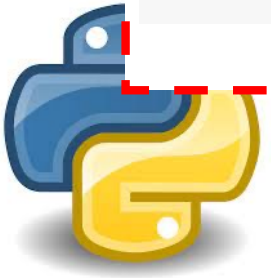
Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

```
[14] G_count = sequence.count('G')
     C_count = sequence.count('C')
     print(f'This sequence has {G_count} Guanines in total.')
     print(f'This sequence has {C_count} Cytosines in total.')
```



Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

Return GC Content

Let's find the GC content of the CRTAM DNA sequence.

Knowledge Check

GC content

- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

$$\frac{\text{Total Guanines} + \text{Total Cytosines}}{\text{Total Sequence Length}} * 100$$

```
[12] GC_content = (G_count + C_count)/length*100
     print(f'The GC content of my sequence is {GC_content}')

     The GC content of my sequence is 42.799188640973625
```

Let's find the GC content of the CRTAM DNA sequence.

Overall Code

```
[13] length = len(sequence)
     print(f'The length of my DNA sequence is {length} nucleotides.')

     The length of my DNA sequence is 2465 nucleotides.
```

```
[14] G_count = sequence.count('G')
     C_count = sequence.count('C')
     print(f'This sequence has {G_count} Guanines in total.')
     print(f'This sequence has {C_count} Cytosines in total.')

     This sequence has 537 Guanines in total.
     This sequence has 518 Cytosines in total.
```

```
[15] GC_content = (G_count + C_count)/length*100
     print(f'The GC content of my sequence is {GC_content}')

     The GC content of my sequence is 42.799188640973625
```

Lesson 1.3.1

Define a function that returns the GC content of a DNA sequence.

Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
        length = len(sequence)
        G_count = sequence.count('G')
        C_count = sequence.count('C')
        GC_content = (G_count + C_count)/length*100
        return(GC_content)
```

Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
```

In the first line, a function is initiated by using def

Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
```

Name the function as you see fit

Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
```

Identify if the function takes input. Here, our function just needs one input. The function needed as input is a DNA sequence in order to return the GC content.

DON'T FORGET PARENTHESES!

Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
```

End it with a colon

Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
        length = len(sequence)
        G_count = sequence.count('G')
        C_count = sequence.count('C')
        GC_content = (G_count + C_count)/length*100
```

Body of the function defined dictates what the function is doing

Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
        length = len(sequence)
        G_count = sequence.count('G')
        C_count = sequence.count('C')
        GC_content = (G_count + C_count)/length*100
        return(GC_content)
```

Using the return function will give you the result of the function when you run it

DON'T FORGET PARENTHESES!

Define a function that calculates GC content

```
[21] def find_GC_content(a_sequence):
       length = len(sequence)
       G_count = sequence.count('G')
       C_count = sequence.count('C')
       GC_content = (G_count + C_count)/length*100
       return(GC_content)
```

```
[22] find_GC_content(sequence)
```

```
     42.799188640973625
```

Tip: Functions are a great way to control code when running a repeated block of code more than once on different datatypes!

Lesson 1.4.0
Running into errors

Uh oh!
Ran into an error!

```
[44] length = len(sequence)
     G_count = sequence.count('G')
     C_count = sequence.count('C')
     GC_content = (G_count + C_count)/length

     print("My sequence is "+str(length)+" nucleotides long.")
     print("There are "+str(G_count)+" guanines and "+C_count+" cytocines.")
     print("The GC content of my sequence is "+str(GC_content)+".")
```

```
My sequence is 2508 nucleotides long.
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-44-b05e980949a1> in <module>
      5
      6 print("My sequence is "+str(length)+" nucleotides long.")
----> 7 print("There are "+str(G_count)+" guanines and "+C_count+" cytocines.")
      8 print("The GC content of my sequence is "+str(GC_content)+".")

TypeError: can only concatenate str (not "int") to str
```

SEARCH STACK OVERFLOW

Where is my error?

Debug and rerun program

```
[44] length = len(sequence)
     G_count = sequence.count('G')
     C_count = sequence.count('C')
     GC_content = (G_count + C_count)/length

     print("My sequence is "+str(length)+" nucleotides long.")
     print("There are "+str(G_count)+" guanines and "+C_count+" cytocines.")
     print("The GC content of my sequence is "+str(GC_content)+".")
```

```
My sequence is 2508 nucleotides long.
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-44-b05e980949a1> in <module>
      5
      6 print("My sequence is "+str(length)+" nucleotides long.")
----> 7 print("There are "+str(G_count)+" guanines and "+C_count+" cytocines.")
      8 print("The GC content of my sequence is "+str(GC_content)+".")

TypeError: can only concatenate str (not "int") to str
```

SEARCH STACK OVERFLOW

Where is my error?

Error Corrected

PennState Huck Institutes of the Life Sciences

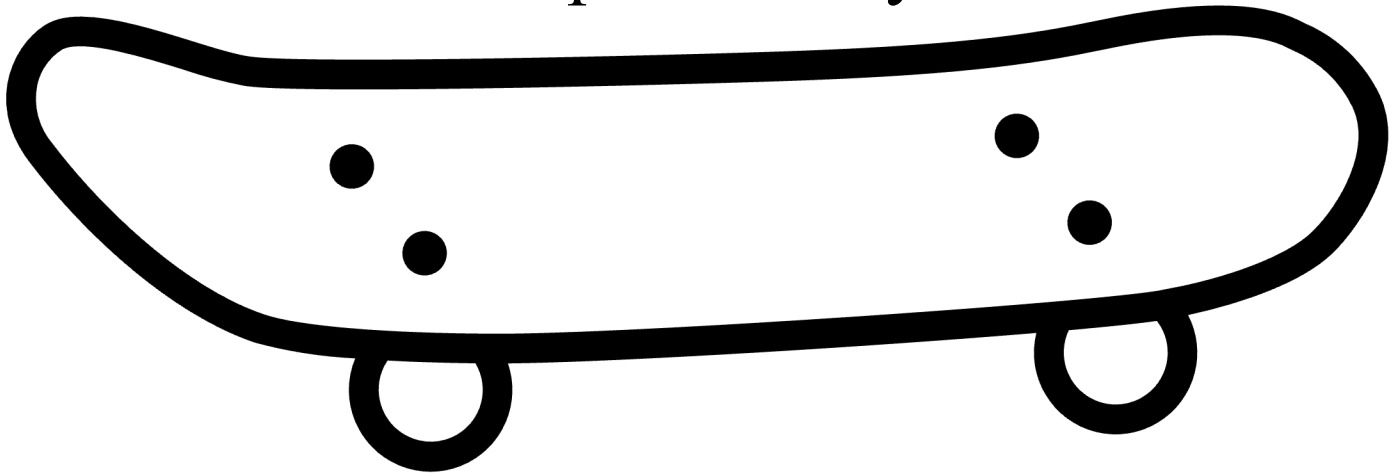
```
[45] length = len(sequence)
     G_count = sequence.count('G')
     C_count = sequence.count('C')
     GC_content = (G_count + C_count)/length

     print("My sequence is "+str(length)+" nucleotides long.")
     print("There are "+str(G_count)+" guanines and "+str(C_count)+" cytocines.")
     print("The GC content of my sequence is "+str(GC_content)+".")
```

```
My sequence is 2508 nucleotides long.
There are 549 guanines and 528 cytocines.
The GC content of my sequence is 0.42942583732057416.
```

Proficiency Assessment

Show off your new skills using Google COLAB with a proficiency assessment



Develop a block of code

1. Find a DNA sequence of your interest using Uniprot and cross referencing a database. Assign your sequence a variable.

2. Define and return the following functions:
    1. Total number of Guanines
    2. Total number of Cytosines
    3. Total number of Adenines
    4. Total number of Uridines
    5. A+T/G+C ratio