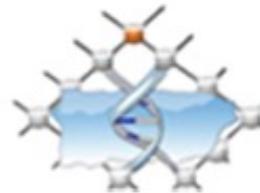




University of Puerto Rico Puerto Rico - IDeA Networks of Biomedical Research Excellence



PRINBRE
IDeA Network 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

1

June 5, 2023 at UPR-Ciencias Médicas



PennState
Huck Institutes of
the Life Sciences



PennState
Huck Institutes of
the Life Sciences

- 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).
- Unless otherwise specified, all the information contained within is Copyrighted © by University of Puerto Rico. Permission is granted for use, modify, and reproduce these materials for research and teaching purposes. A copy of the modified material should be sent to help@hpcf.upr.edu.
- Most recent versions of these presentations can be found at <http://inbre.hpcf.upr.edu/>.





Competencies

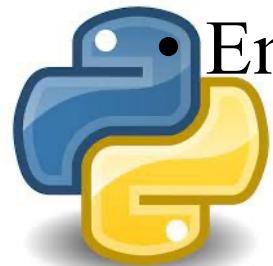
Record and write simple and common **Python scripts to deal with Bioinformatic needs for biological data analyses** using a Jupyter Notebook





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.

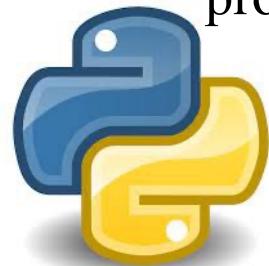




PennState
Huck Institutes of
the Life Sciences

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.





PennState
Huck Institutes of
the Life Sciences

Lesson 1.0.0

Connecting to Google

Colab to learn

common Python lines

of code

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



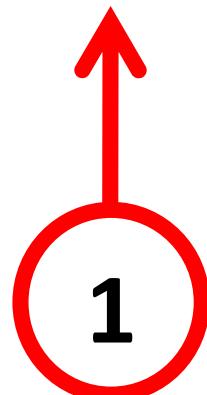


Google
colab

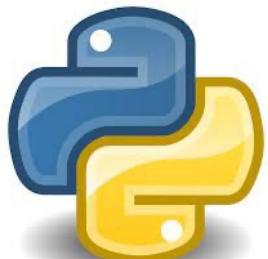


Google Colab

<https://colab.research.google.com>



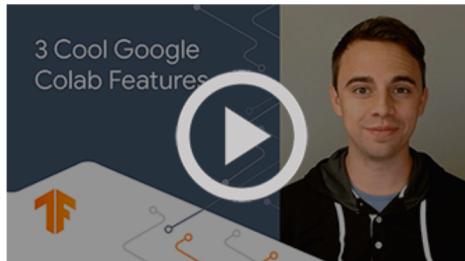
Activate the link



Activate colab.research.google.com

Screenshot of a web browser showing the Colaboratory landing page at colab.research.google.com.

The browser interface includes:

- Address bar: colab.research.google.com
- Toolbar: Back, Forward, Stop, Refresh, Download, Open, New tab, Full screen.
- Header: Welcome To Colaboratory, File, Edit, View, Insert, Runtime, Tools, Help, Share, Settings, Profile.
- Left sidebar (Table of contents):
 - Getting started
 - Data science
 - Machine learning
 - More Resources
 - Featured examples
 - Section
- Main content area:
 - 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

The screenshot shows the Google Colab interface. At the top, there's a red header with the text "Open a New notebook". Below it is a browser-like header with icons for back, forward, refresh, and download, followed by the URL "colab.research.google.com". The main content area has a title "Welcome To Colaboratory" and a "File" menu open. The "File" menu contains the following options:

- New notebook
- Open notebook ⌘/Ctrl+O
- Upload notebook
- Rename
- Save a copy in Drive
- Save a copy as a GitHub Gist
- Save a copy in GitHub
- Save ⌘/Ctrl+S
- Revision history
- Download
- Print ⌘/Ctrl+P

A large red arrow points from the top-left towards the "New notebook" option. A red circle highlights the number "1" next to the "New notebook" button, which is also highlighted with a red box. The main content area displays a "Welcome to Colab" message, a video thumbnail, and a "What is Colab?" section.

1

New notebook

What is Colab?

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

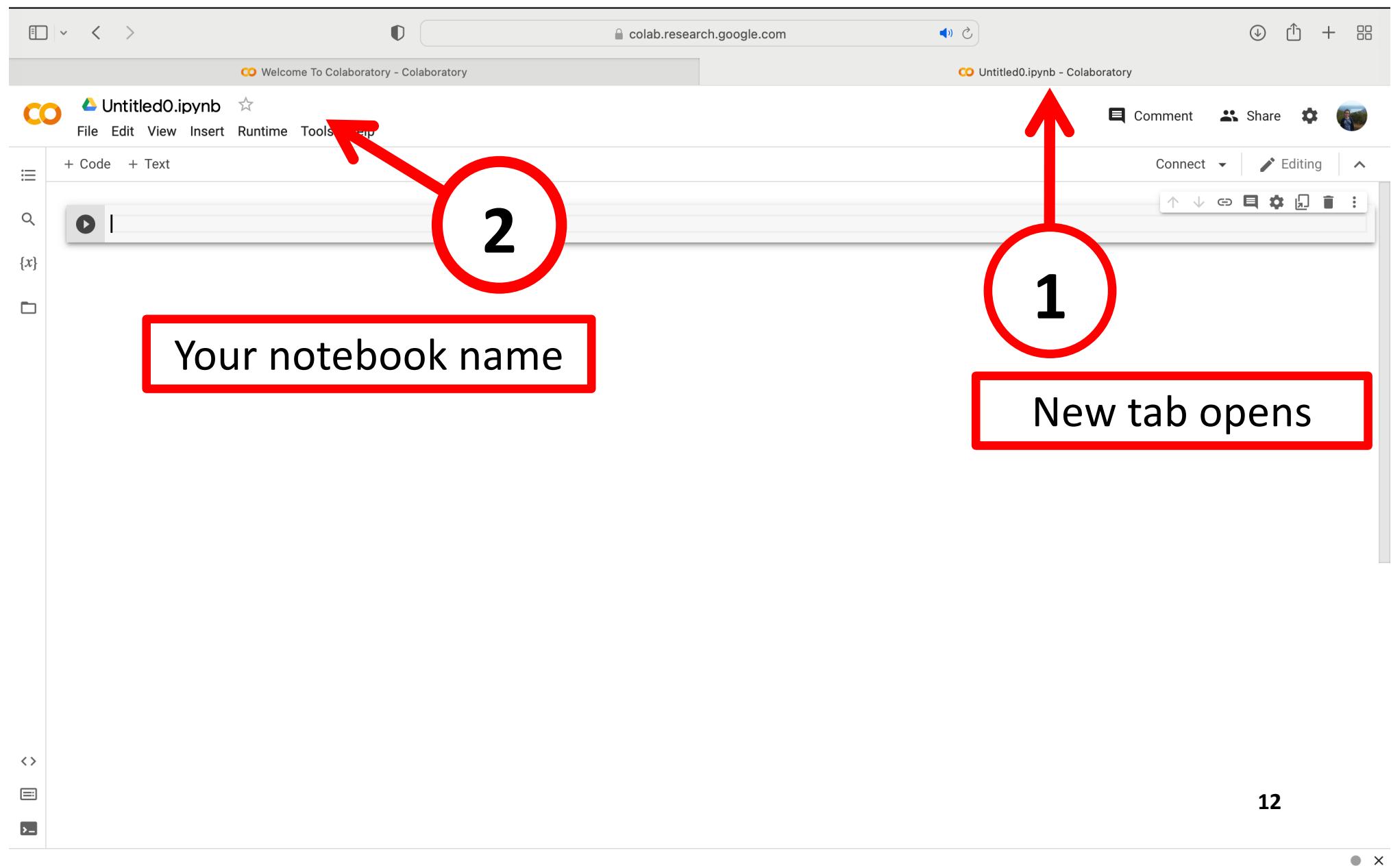
- Zero configuration required
- Access to GPUs free of charge
- Easy sharing

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.

Welcome to your new notebook!





PennState
Huck Institutes of
the Life Sciences

Lesson 1.1.0

Test snippets of code using the Python interpreter



Python is your new **calculator**!

The image shows a screenshot of a Python Colab notebook interface. On the left, there's a sidebar with icons for file operations, code, text, search, and a variable placeholder $\{x\}$. The main area has a toolbar with 'File', 'Edit', 'View', 'Code', and 'Text' buttons. A play button and a cursor are visible in the code editor. A red box highlights a list of arithmetic operators:

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

At the bottom left is the Python logo, and at the bottom right is the page number 14.

Let's solve metric conversion problems!

The image shows a Jupyter Notebook interface with the following details:

- Title Bar:** CO OX-Python3-Workshop.ipynb
- Menu Bar:** File Edit View Insert Runtime Tools Help
- Search Bar:** {x}
- Table of Contents:**
 - Metric Conversion Problems
 - Example 1
 - How many seconds(s) are in 10 milliseconds (ms)?
- Code Cell:** A cell containing a play button icon.
- Annotations:**
 - A red circle with the number "1" points to a red-bordered box containing the text "Type equation".
 - A red circle with the number "2" points to a red-bordered box containing the text "Run".
- Python Logo:** A blue and yellow Python logo is visible on the left side.

Let's solve metric conversion problems!

OX-Python3-Workshop.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

Metric Conversion Problems

Example 1

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

10 * (1/0.001)

10000.0

3

Answer

Note: Python interprets the numbers as integers or floats

16

Try it on your own!

▼ Problem 1

How many $5\mu\text{L}$ are in mL?

ANSWER



Try it on your own!

▼ Problem 1

How many $5\mu\text{L}$ 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

#Given:

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

#Solution

```
result = mass_of_h2o/molecular_mass_of_h2o  
result
```

2.7777777777777777

Note: Facilitate
your programing
employing
variables!

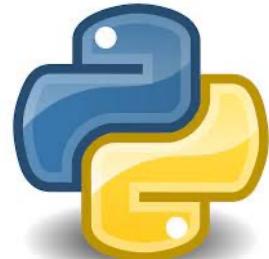




PennState
Huck Institutes of
the Life Sciences

Lesson 1.2.0

Python datatypes

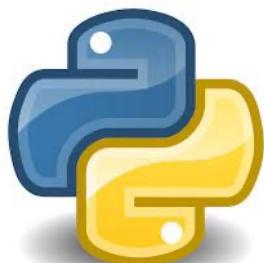


Integers and floats and strings... oh my!

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

Datatypes

<code>int()</code>	12
<code>float()</code>	12.0
<code>str()</code>	"hola"
<code>list()</code>	[1,2,3, "hola"]
<code>dictionary()</code>	{"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)
```

```
int
```



Integers and floats and strings... oh my!

- ▼ What is the datatype of molecular_mass_of_h2o?

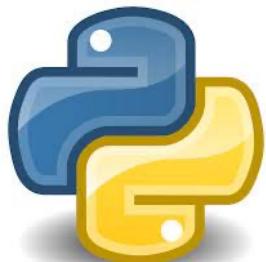
Answer



Integers and floats and strings... oh my!

- ▼ What is the datatype of molecular_mass_of_h2o?

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

10000.0

✓ [30] `seconds = 10 * (1/0.001)`
0s
 int(seconds)

10000

▼ Identify datatypes of Python objects using `type()` functions.

✓ [31] `mass_of_h2o`
0s

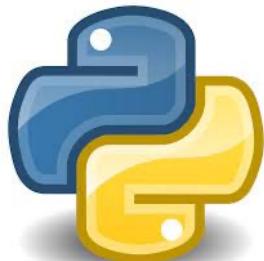
50

✓ [27] `type(mass_of_h2o)`
0s

int

✓ [29] `float(mass_of_h2o)`
0s

50.0



Integers and floats and strings... oh my!

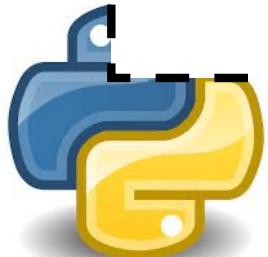
✓ [33] 'hola puerto rico'
0s

```
'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'
0s

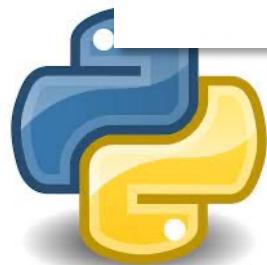
```
'hola puerto rico'
```

- ▼ Assign a variable to this string and print variable

Hint: Search for python's print() function

✓ [34] greeting = 'hello puerto rico'
0s
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"
```

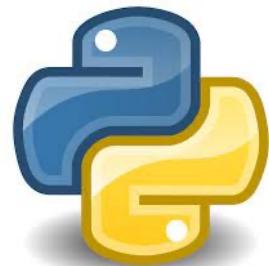


Indexing a String

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

'ATGTGGTGG'
012345678

[5] sequence[0]

Call the character at index 0

'A'



Indexing a String

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

' ATGTGGTGG '
012345678

[7] sequence[8]

Call the last character using 8

' G '

[8] sequence[-1]

Call the last character using -1

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

' G '

Indexing a String

Index a range of the sequence

' ATGTGGTGG '
012345678

[9] sequence[2:8]

Index is included and starts new string

' GTGGTG '

Index is not included in new string





PennState
Huck Institutes of
the Life Sciences

Lesson 1.3.0

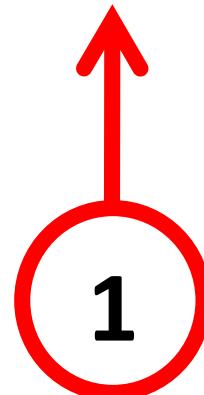
Evaluate the GC content of a DNA sequence.





Connect to Uniprot

<https://www.uniprot.org>



Activate uniprot.org

UniProt BLAST Align Peptide search ID mapping SPARQL Release 2023_01 | Statistics Help

Find your protein

UniProtKB ▾ Advanced | List Search Examples: Insulin, APP, Human, P05067, organism_id:9606 Feedback

UniProt is the world's leading high-quality, comprehensive and freely accessible resource of protein sequence and functional information. [Cite UniProt™](#)

Proteins
UniProt Knowledgebase

Species
Proteomes

Protein Clusters
UniRef

Sequence Archive
UniParc



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

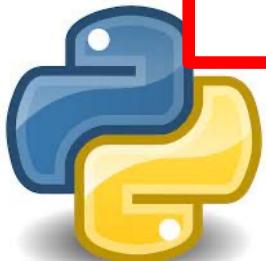
Find your protein

UniProtKB ▾ human crtam

Examples: Insulin, APP, human, P05067, organism_id:9606



Search CRTAM



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

UniProtKB 16 results

Reviewed (Swiss-Prot) (5)

Unreviewed (TrEMBL) (11)

Pular organisms

- uman (5)
- ouse (2)
- ebrafish (1)

Economy

ter by

roteins

Structure (3)

lternative products (isoforms)

lternative splicing (5)

eta strand (3)

nary interaction (5)

Entry

O957

Q9BYV

Q80UJ

Q8R59

A0A2R9C440

A0A2R9C440_PANPA

UniProtKB ▾

human crtam

Advanced | List | Search |

Select how you would like to view your results

Cards

Table

3

View results

Gene Names	Organism	Length
CRTAM	Homo sapiens (Human)	395 AA
CADM1, IGSF4, GSF4A, NECL2, SYNCAM, TSLC1	Homo sapiens (Human)	444 AA
SCRB1, CRIB1, KIAA0147, LAP4, SCRIB1, VARTUL	Homo sapiens (Human)	1,011 AA
Scrib, Kiao0147, Lap4, Scrib1	Mus musculus (Mouse)	1,011 AA
Cadm1, Igsf4, Necl2, Ra175, Syncam, SynCam1, Tslc1	Mus musculus (Mouse)	455 AA
CRTAM	Pan paniscus (Dwarf)	395 AA

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

UniProt BLAST Align Peptide search ID mapping SPARQL UniProtKB ▾ human crtam Advanced | List Search Help

Status
Reviewed (Swiss-Prot) (5)
Unreviewed (TrEMBL) (11)

Popular organisms
Human (5)
Mouse (2)
Zebrafish (1)

Taxonomy
Filter by taxonomy

Proteins with
3D structure (3)
Alternative products (isoforms) (5)

UniProtKB 16 results

BLAST Align Map IDs Download Add View: Cards Table Customize columns Share

Entry	Entry Name	Protein Names	Gene Names	Organism	Length
O95727	CRTAM_HUMAN	Cytotoxic and regulatory T-cell molecule [...]	CRTAM	Homo sapiens (Human)	393 AA
Q14160	CADM1_HUMAN	Cell adhesion molecule 1 [...]	CADM1, IGSF4, IGSF4A, NECL2, SYNCAM, TSCL1	Homo sapiens (Human)	442 AA
Q14160	SCRIB_HUMAN	Protein scribble homolog [...]	SCRIB, CRIB1, KIAA0147, LAP4, SCRIB1, VARTUL	Homo sapiens (Human)	1,630 AA
Q8R5M8	SCRIB_MOUSE	Protein scribble homolog [...]	Scrib, Kiaa0147, Lap4, Scrib1	Mus musculus (Mouse)	1,612 AA
	CADM1_MOUSE	Cell adhesion molecule 1 [...]	Cadm1, Igsf4, Necl2,	Mus musculus	456 AA

Activate link to Entry



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

UniProt BLAST Align Peptide search ID mapping SPARQL UniProtKB Advanced | List Search Help

|Function

★ O95727 · CRTAM_HUMAN

Names & Taxonomy Subcellular Location Disease & Variants PTM/Processing Expression Interaction Structure Family & Domains Sequence & Isoform Similar Proteins

Protein ⁱ	Cytotoxic and regulatory T-cell molecule	Amino acids	393
Gene ⁱ	CRTAM	Protein existence ⁱ	Evidence at protein level
Status ⁱ	UniProtKB reviewed (Swiss-Prot)	Annotation score ⁱ	5/5
Organism ⁱ	Homo sapiens (Human)		

Entry Feature viewer Publications External links History

BLAST Align Download Add Add a publication Entry feedback

Functionⁱ

Mediates heterophilic cell-cell adhesion which regulates the activation, differentiation and tissue retention of T cells (By similarity). Interaction with CADM1 promotes natural killer (NK) cell cytotoxicity and IFNG/interferon-gamma secretion by NK cells (By similarity). In vitro as well as NK cell-mediated rejection of tumors expressing CADM1 in vivo (PubMed:[15811952](#)).

Scroll



5



46

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

PTM/Processing

Expression

Interaction

Structure

Family & Domains

Sequence & Isoform

Similar Proteins

Sequence databases

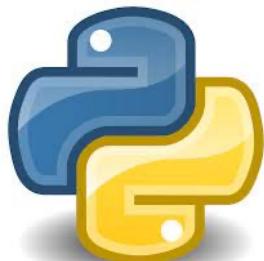
CCDS | CCDS76489.1 ↗ [O95727-2]
CCDS8437.1 ↗ [O95727-1]

RefSeq | NP_001291711.1 ↗ NM_0[O95727-2]
NP_062550.2 ↗ NM_0196

	SEQUENCE	PROTEIN	MOLECULE TYP
	AF001622 (EMBL ↗ GenBank ↗ DDBJ ↗)	AAC80267.1 (EMBL ↗ GenBank ↗ DDBJ ↗)	mRNA
	AB209830 (EMBL ↗ GenBank ↗ DDBJ ↗)	BAD93067.1 (EMBL ↗ GenBank ↗ DDBJ ↗)	mRNA
	BC070266 (EMBL ↗ GenBank ↗ DDBJ ↗)	AAH70266.1 (EMBL ↗ GenBank ↗ DDBJ ↗)	mRNA

6

Activate link to EMBL



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

The ENA Advanced Search API is changing on 2023-05-02! Details [here](#).

Sequence: AF001622.1

Homo sapiens class-I MHC-restricted T cell associated molecule (CRTAM) mRNA, complete cds.

Organism:	Homo sapiens (human)
Accession:	AF001622
Mol Type:	mRNA
Topology:	LINEAR;linear
Base Count:	2425
Dataclass:	STD
Tax Division:	HUM
Chromosome:	11
Md5 Checksum:	f28e47cb1576f4d6aaaf50b83f6cacbd8
Map:	11q22-q23

View: EMBL FASTA
Download: EMBL FASTA
Navigation: Show
Publications: Show
Sequence Versions: View

6

Activate link to FASTA

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.
ATGTGGTGGAGAGTTCTCAGCTTGCATGGTCCCTTGCAAGAGGCCCTCTGACT
AACCAACAGAAACCATCACCGTGGAGGAAGGCCAGACGCTCACTCTAAAGTGTGTCAC
TCTCTGAGGAAGAACTCCTCCCTCAGGGCTGACCCCTCAGGGTTCACCATTNTTA
AATGAGTATCCTGCTTAAAAAATTCAAATACCAGCTTCTTCATCACTCGGCCAATCAG
CTCTCCATCACTGTGCTAACGTAACCCCTGCAAGATGAAGGCCGTGTACAAGTGTACAT
TACAGCGACTCTGTAAGCACAAAGGAAGTGAAGTGTGATTGTGCTGGCAACTCCTTCAAG
CCAATCTGGAAGCTTCAGTTATCAGAAAAGCAAATGGAGAAGAACATGTTGACTCATG
TGCTCCACCATGAGAAGCAAGCCCCCTCCGAGATAACCTGGCTACTTGGGAATAGCATG
GAAGTGTCCCGTGGAACGCTCCATGAATTGAAACTGATGGGAAGAAATGTAATACTACC
AGCACTCTCATAAATCCACACTATGGCAAAAATTCAACGGTGGACTGCATTATCCGACAC
AGAGGCCCTGCAAGGGAGAAAATAGTAGCACCCCTCCGGTTGAAGATTTGTTACTGAT
GAAGAGACAGCTTCAGATGCTGGAGAGAAAATCTCTATCCTCTCAAGACCCACAGCAG
CCCACCACTACTGTCAGTAACGGAAGATTCTAGTACATCGGAGATTGACAAGGAAGAG
AAAGAACAAACCCTCAAGATCCTGACTTGACCACCGAAGCAAATCCTCAGTATTGAGGA
CTGGCAAGAAAGAAAATGGCATTCTGCTGCTCACGCTGGTGTCTCTCATTTCATA
CTCTTCATCATAGTCCAGCTTCATCATGAAGCTGAGGAAAGCACATGTGATATGGAAG
AGAGAAAACGAAGTTCAAGAACACACACTAGAAAGTACAGATCAAGGTCAAATAATGAA
GAAACATCATCTGAAGAGAAAAATGCCAATCTTCCCACCCATGCGTTGCATGAACATAC
ATCACAAAGTTGTACTCAGAAGCAAAAACAAAGAGGAAGGAAAATGTACAACATTCAAA
TTAGAAGAAAAGCACATCCAAGTACCAAGAGTATTGTGAGTGTGCTCTGCAATGGAAC
ATGTGATTCAGGGTTGCCGCAGTGTACCTCAGTGACCAGCCTGGGGAGGAGCTTA
ATTGCTGAGACATTAATAATGACCTTCTAGTGCAATGCAAGATGGTGTCTCGGATAATG
ATCTGCCCGGAGCTAGGGCAGCAACATGAGGACCAACCATGCACATAAAGCTTGAGT
TTAAAAAAAGCAAAAAATAATTATGCCCTGACACTACTTCAGAGCAGGAGGATTCT
ACGAAGCCTGGGATCAGGTCAAGTGTGAGCAGCTAACATCCTACCTCAAATGGAACAG
GATTTTTGATGCTTGTCTAAAGAACATGTTAAAAATTTTTTCTTTAATAT
TTTCTCTGGTCACAAAATAAGAAATTGGGATGCAAAGTACCTAAAGATCTGATCC
TAAGAAGTTACTCTGGCCAGGCAGGCTCATGCCGTAACTTAGCAGGAGGATTCT
GCTGAGGTAGGCAGATCACTTGAGGTCAAGGAGTTGGAGACCAGCCTGGCCAACATAGTGA
AACCCCGCTCTACTAAAAATGCCAGGCTAGTGGTGCACCTGTAGTC
TCAGATACTGGGAGGCTGAGGGTGGAGAATCGCTGAACCTGGGAGGTGGAGATTGCA
TGAGTCAGATCTACCAACTGAACCTCCAGCCTGGGCCAGAGGGAGACTCTGCTCAAA

7

Copy Sequence



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

8

Assign sequence to variable as a string



```
sequence = """ATGTGGTGGAGAGTTCTCAGCTGCTGGCATGGTCCCTTGCAAGAGGCCTCTGACT  
AACCACACAGAAACCATACCGTGGAGGAAGGCCAGACGCTCACTCTAAAGTGTGTC  
ACTCTCTGAGGAAGAACTCCTCCCTCCAGTGGCTGACCCCTCAGGGTTACCAATT  
AATGAGTATCCTGCTTAAAAAAATTCCAATACCAGCTTCTTCATCACTCGGCCAAT  
CTCTCCATCACTGTGCCTAACGTAACCTGCAAGATGAAGGCGTGTACAAGTGCTTAC  
TACAGCGACTCTGTAAGCACAAAGGAAGTGAAGTGTGCTGGCAACTCCTTCAAG  
CCAATCCTGGAAGCTTCAGTTACAGAAAGCAAATGGAGAAGAACATGTTGACTCAT  
TGCTCCACCATGAGAAGCAAGCCCCCTCGCAGATAACCTGGCTACTTGGGAATAG  
GAAGTGTCCGGTGGAACGCTCCATGAATTGAAACTGATGGGAAGAAATGTAATA  
AGCACTCTCATAATCCACACTTATGGCAAAAATTCAACGGTGGACTGCATTATCC  
GACACAGAGGCCTGCAAGGGAGAAAAGTAGTAGCACCCTCGGTTGAAGATTGTT  
ACTGATTCAGATGCTCTGGAGAGAAAACTCTATCCTCTCAAGACCCACAGCAG  
TGTCTCAGTAACGGAAGATTCTAGTACATCGGAGATTGACAAGGAAGAG  
CACTCAAGATCCTGACTTGACCACCGAAGCAAATCCTCAGTATT  
GAAAAGTGGCATCCTGCTCAGCCTGGTGTCCCTCCTCATT  
AGTCCAGCTTCATCATGAAGCTGAGGAAAGCACATGTGATATC  
AGTTTCAGAACACACACTAGAAAGTTACAGATCAAGGTCAAATA  
TGAAGAGAAAAATGGCAATCTCCCACCCATGCGTTGATGA  
ATCACAAAGTTGACTCAGAAGCAAAACAAAGAGGAAGGAAAATGTACA  
TTAGAAGAAAAGCACATCCAAGTACCAAGAGAGTATTGTG  
TAGTGTCTCTGCAATG  
ATGTGATTCAGGGTTGCCCGAGTGTACCTCAGTGGACCAGCCTGGGG  
AGAGGAG  
ATTGCTGAGACATTAATAATGACCTCTAGTGA  
ATGCAATGCAAGATGGTGTCTCGGA  
ATCTGCCCGAGCTAGGGCAGCAACATGAGGACCA  
AAACCATGCACATAAGCTT  
AGT  
TTAAAAAAAGAAAAGCAAAAAATAATTATGCCTGACACTACT  
TCAGAGCAGGAGATTCT  
ACGAAGCCTGGGATCAGGGTCAGTGTGAGCAGCTAACAT  
CCTACCTCAAATGGAACAG
```

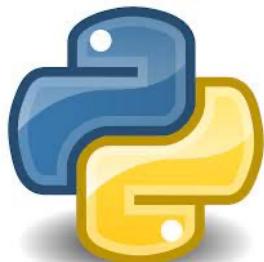
Note: Use triple quotations to fit the sequence in a block of code

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.

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



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.

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

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

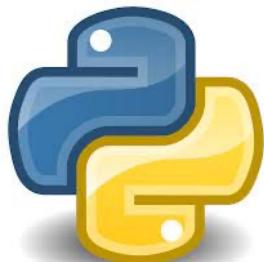
- The percentage of the proportion of guanines and cytosines to the total length of the DNA sequence.
- Provides an idea of stable a sequence is.

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

```
[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.

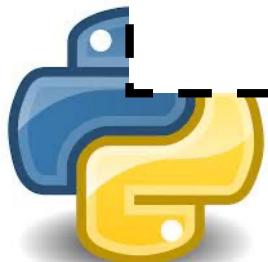
Knowledge Check

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.

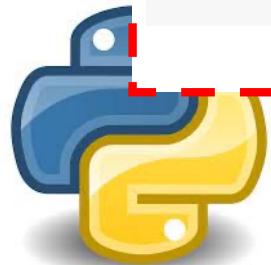
Knowledge Check

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)
0s     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')
0s     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
0s     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!





PennState
Huck Institutes of
the Life Sciences

Lesson 1.4.0

Running into errors





Uh oh!

Ran into an error!



PennState
Huck Institutes of
the Life Sciences

```
[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



PennState
Huck Institutes of
the Life Sciences

```
[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.

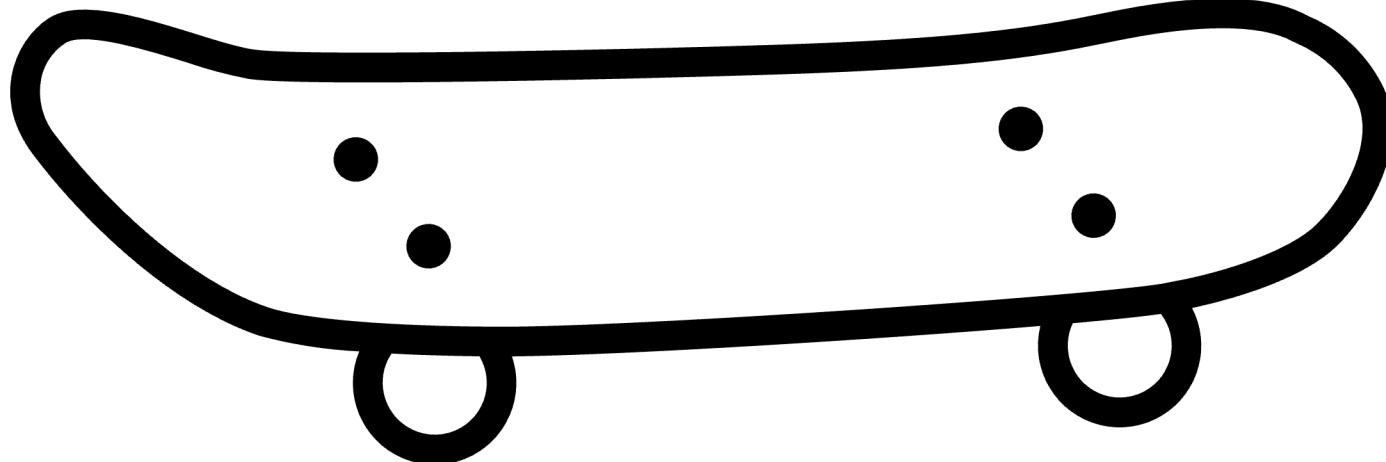




PennState
Huck Institutes of
the Life Sciences

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

