



Big Data for Biologists: Decoding Genomic Function

HUMBIO51

Annette Salmeen & Anshul Kundaje
TA: Anna Shcherbina



Contact Information & Office Hours

Annette Salmeen

e-mail: asalmeen@stanford.edu

Office: Main Quad Bldg 20-21F

Office Hours:

Tu 4:00-5:00PM

Fri 10:00-11:00AM

Or by appointment

Anshul Kundaje

e-mail: akundaje@stanford.edu

Office: Lane L301B

Office Hours:

By appointment

Anna Shcherbina

e-mail: annashch@stanford.edu

Office hours:

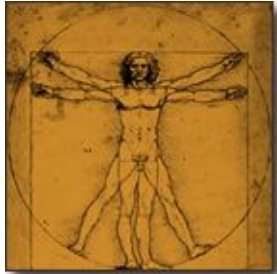
Wednesday 10:00 - 11:00 am

Agenda

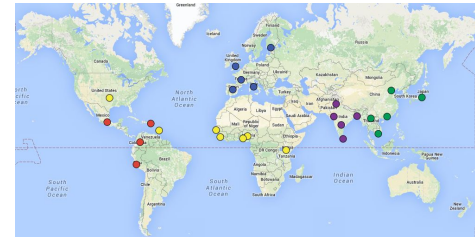
- Course introduction & Objectives
- Class introductions
- Accessing Jupyter Notebooks
- Reading a DNA sequence into a computer

Why would biologists want to collect or
analyze big data?

Why would biologists want to collect or analyze big data?



Human Genome
Project
2003

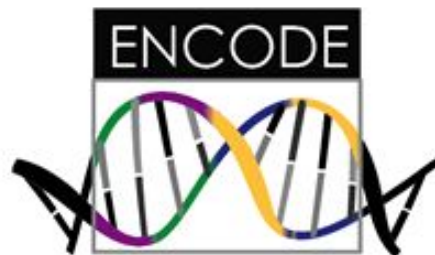


1000 Genomes
Project
(Phase 1)
2012



2015

2007



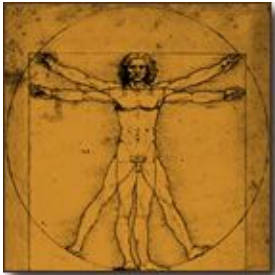
UK Biobank
500,000 Genomes
plus other health metrics

Course Objectives

- Use Unix and/or Python to work with large biological data sets
- Interpret if a variant in the genome is likely to exert its effects via a protein coding region or through regulatory elements including promoters or enhancers.
- Analyze datasets to identify sites in the genome that are likely to be relevant to a disease.
- Query a large data set and visualize the data by making or interpreting a scatter plot, barplot, histogram or heatmap.
- Conduct a collaborative programming project applying best practices for generating reproducible data analysis scripts.

Unit 1:

What are genes, DNA, RNA and proteins? Getting Started with Python



Human Genome
Project

2003

2007

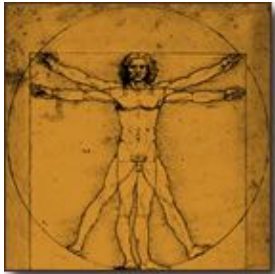
2012

2015

Unit 2:

How are cell types different?

Using Python with biological datasets

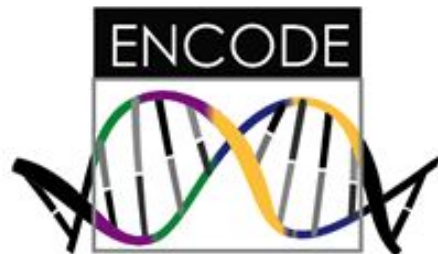


Human Genome
Project
2003

2007

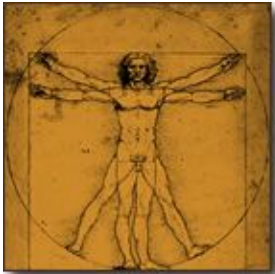
2012

2015



Unit 3:

How do genomes vary across populations? Using Python to analyze genomes



Human Genome
Project
2003

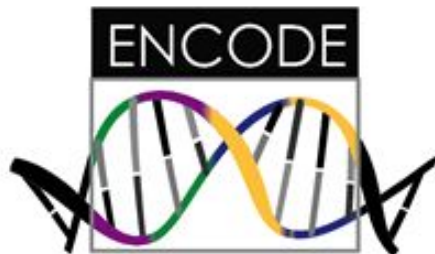


1000 Genomes
Project
(Phase 1)
2012



2015

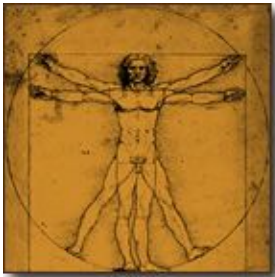
2007



UK Biobank
500,000 Genomes
plus other health metrics

Unit 4:

Introduction to Genetics and Disease Collaborative Computational Project



Human Genome
Project
2003

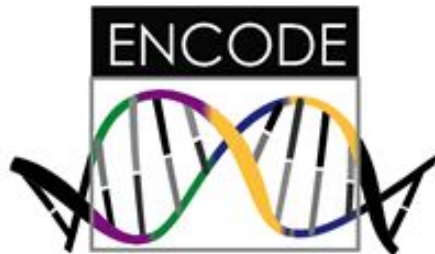


1000 Genomes
Project
(Phase 1)
2012



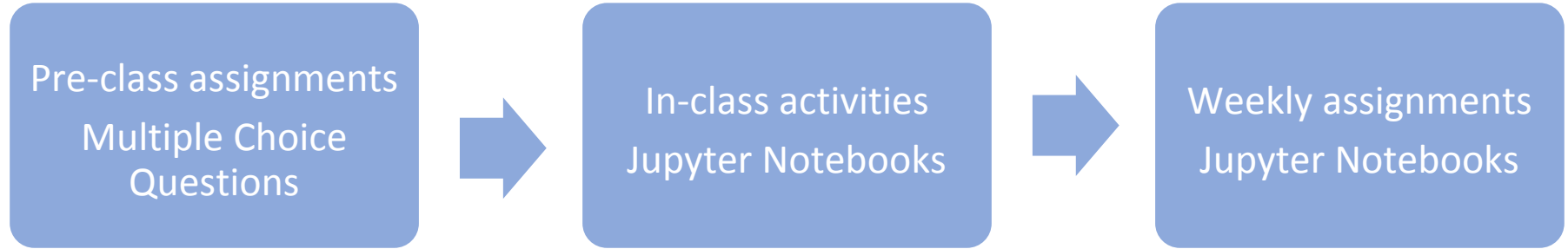
2015

2007



UK Biobank
500,000 Genomes
plus other health metrics

Course Components



Class participation	10%
Pre-class assignments	10% (due Tues or Thurs at 10AM)
In-class activities	10% (due by 8PM after class)
Weekly assignments	40% (due Tues by 10AM)
Collaborative Project	30%

Collaborative Computational Biology Project

11/14: Introduction to Jupyter Notebook for course project

Example course project presentation

Assign groups (3-4 students each)

Each student will receive a coding and a non-coding variant

11/28: Submit drafts of Jupyter Notebooks for non-coding variants

12/5: Submit drafts of Jupyter Notebooks for coding variants

12/7: Class Presentation of projects

12/8: Final projects due

A Note on the Honor Code: Citations & Collaborations

- Collaboration on in-class activities and the collaborative computational biology project is encouraged.
- Pre-class and weekly assignments should be completed on your own.
- If you consult sources outside of the class notes for the weekly assignments and course project then citations should be provided.

Accommodations & Extensions

- If you require any accommodations, letters from the Office of Accessible Education (OAE) can be submitted to asalmeen@stanford.edu.
- For questions regarding missed class or extensions on assignments due to University Sanctioned Activities, illness or personal emergency please contact asalmeen@stanford.edu.

Feedback on ways to
optimize your learning experience
is welcome!











Agenda

- Course introduction & Objectives
- Class introductions
- Accessing Jupyter Notebooks
- Reading a DNA sequence into a computer

Accessing Course Materials on Canvas & the Google Cloud Platform

- Readings can be accessed through the syllabus links on Canvas
- Pre-class assignments will be posted on Canvas under Assignments
- In-class activities and weekly programming assignments will be posted as Jupyter Notebooks on Canvas and on the Google Cloud Platform
- All Assignments need to be submitted through Canvas!

Course Materials posted on Canvas



Stanford University

Account

Dashboard

Courses

Calendar

Inbox

Commons

Help

≡ F17-HUMBIO-51-01 > Syllabus

Fall 2017

Home

Announcements

Assignments

Grades

Syllabus

Quizzes

Modules

Roster Photos

Outcomes

Conferences

Collaborations

Discussions

Pages

People

Files

Zoom

Settings

Big Data for Biologists - Decoding Genomic Function

Jump to Today 

Big Data for Biologists: Decoding Genomic Function
Fall 2017
Tuesdays and Thursdays 12:00-1:20PM
Location: STLC105

Course Description

Biology and medicine are becoming increasingly data-intensive fields. This course is designed to introduce students interested in human biology and related fields to methods for working with large biological datasets. There will be in-class activities analyzing real data that have revealed insights about the role of the genome and epigenome in health and disease. For example, we will explore data from large-scale gene expression and chromatin state studies. The course will provide an introduction to the relevant topics in biology and to fundamental computational skills such as editing text files, formatting and storing data, visualizing data and writing data analysis scripts. Students will become familiar with both UNIX and Python. This course is designed at the introductory level. Previous university-level courses in biology and programming experience are not required.


Course Objectives

Students will be able to:

1. Use Unix and/or Python to view, sort and parse large data sets such as those from genome-wide gene expression studies.
2. Use computational methods to interpret if a variant in the genome is likely to exert its effects via a protein coding region or through regulatory elements including promoters or enhancers.
3. Analyze datasets from cases and controls to identify sites in the genome that are likely to be relevant to a disease.
4. Query a large data set and visualize the data by making or interpreting a scatter plot, barplot, histogram or heatmap.
5. Conduct a collaborative programming project applying best practices for generating reproducible data analysis scripts.

Course Status


 Unpublished  Publish

 Import from Commons

 Choose Home Page

 View Course Stream

 Course Setup Checklist

 New Announcement

< September 2017 >

27	28	29	30	31	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
1	2	3	4	5	6	7

Assignments are weighted by group:

Group	Weight
Pre-class assignments	10%
In class Programming or Data Analysis Activities	10%

Two options for accessing Jupyter Notebooks

- Via the class server on the Google Cloud Platform (link posted on Canvas)

<https://console.cloud.google.com/start>

- By installing Anaconda and running Jupyter Notebooks from your computer.

<https://www.anaconda.com/download/>

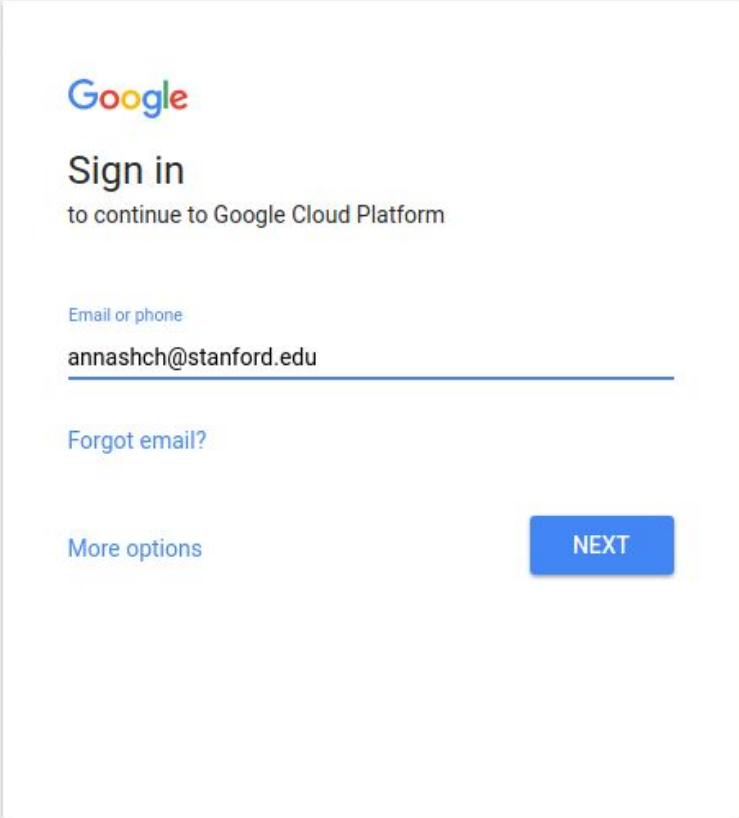
Note: this will also require you to install Python libraries and other programs

How to use the Google Cloud Platform:

Setting up your account

1. <https://console.cloud.google.com/start>

2. Specify your stanford.edu e-mail account as the login account

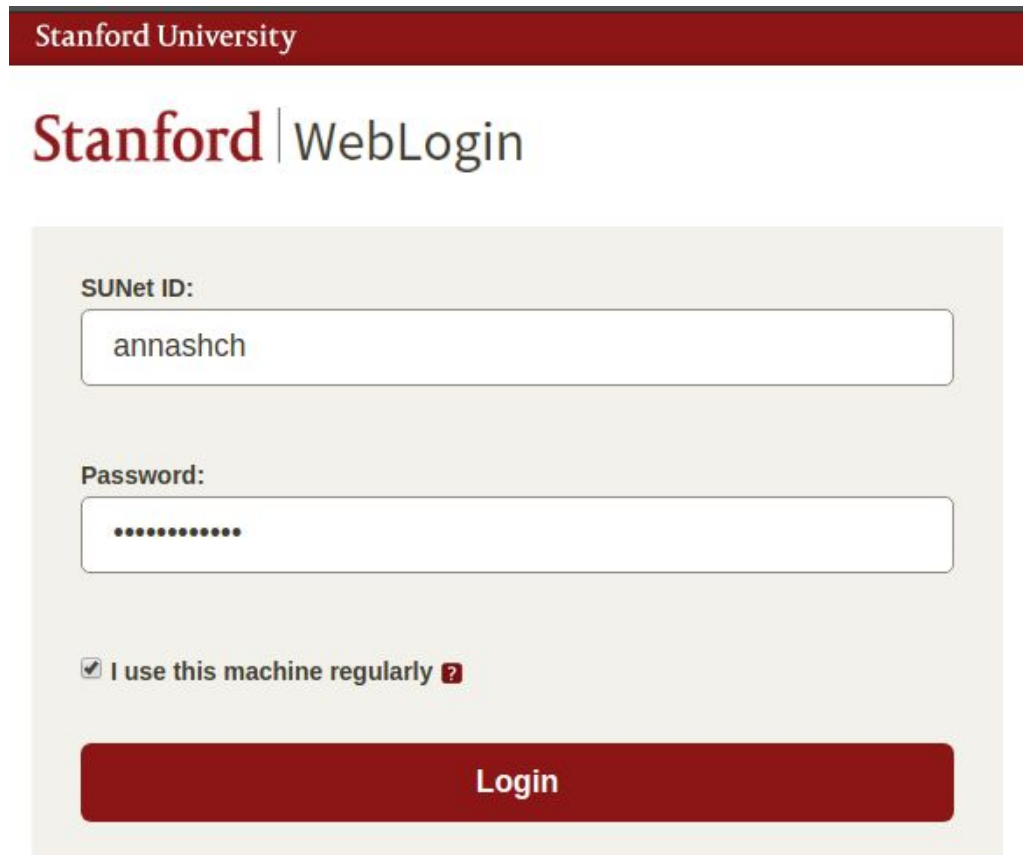


The screenshot shows the Google Cloud Platform sign-in interface. At the top is the Google logo. Below it, the text "Sign in" is followed by "to continue to Google Cloud Platform". There is a label "Email or phone" above a text input field containing "annashch@stanford.edu". Below the input field are links for "Forgot email?" and "More options". A blue "NEXT" button is positioned to the right of the "More options" link. At the bottom of the page, there is a footer with "English (United States)" and a dropdown arrow, and links for "Help", "Privacy", and "Terms".

How to use the Google Cloud Platform:

Setting up your account

3. Complete the Stanford WebLogin prompt



The image shows the Stanford University WebLogin interface. At the top is a dark red header with the text "Stanford University" in white. Below the header is the "Stanford | WebLogin" logo, where "Stanford" is in red and "WebLogin" is in grey. The main login area has a light beige background and contains the following elements: a label "SUNet ID:" followed by a text input field containing "annashch"; a label "Password:" followed by a password input field with ten dots; a checkbox that is checked, followed by the text "I use this machine regularly" and a small red question mark icon; and a large red "Login" button at the bottom.

Stanford University

Stanford | WebLogin

SUNet ID:

annashch

Password:

.....

☒ I use this machine regularly ?

Login

How to use the Google Cloud Platform:

Setting up your account

4. You will see the GCP portal

The screenshot shows the Google Cloud Platform 'Getting started' page. The header is blue with the Google Cloud Platform logo, a search bar, and user account icons. The main content area is titled 'Getting started' and features several interactive cards:

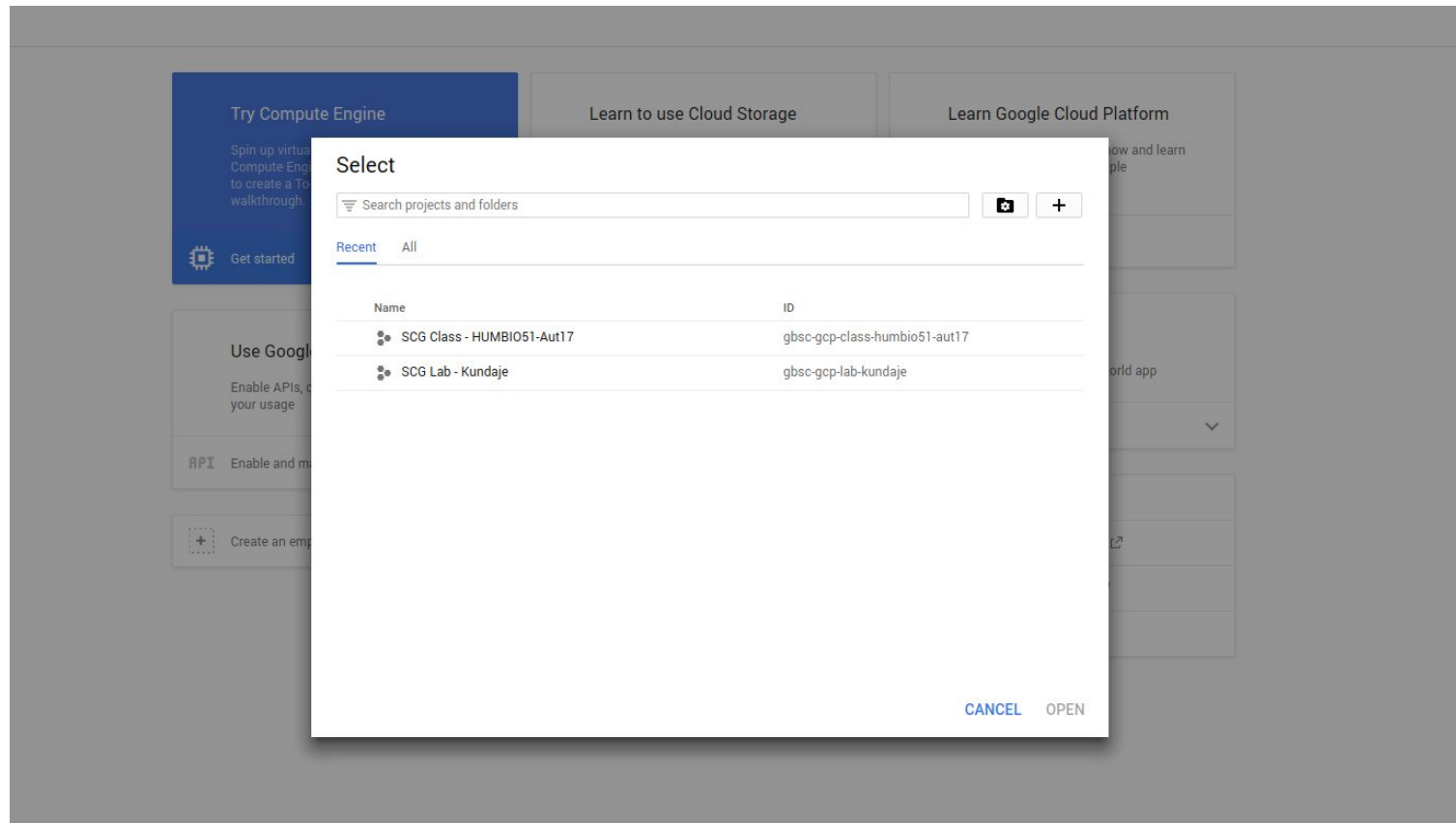
- Try Compute Engine**: A blue card with a 'Get started' button. Text: 'Spin up virtual machines using Google Compute Engine, Node.js, and MongoDB to create a To-Do app in this guided walkthrough.'
- Learn to use Cloud Storage**: A card with a 'Get started' button. Text: 'Cloud Storage is a powerful and simple storage service. In this tutorial you'll learn the basics by creating a storage bucket, and then uploading and sharing a sample file as a public URL link.'
- Learn Google Cloud Platform**: A card with a 'Get Started' button. Text: 'Take an interactive tutorial now and learn how to deploy and build simple applications.'
- Use Google APIs**: A card with an 'API Enable and manage APIs' button. Text: 'Enable APIs, create credentials, and track your usage.'
- Create a Cloud SQL instance**: A card with a 'Get started' button. Text: 'Cloud SQL is a MySQL database that runs in Google's cloud, with no installation or maintenance required.'
- Try App Engine**: A card with a 'Get started' button and a dropdown arrow. Text: 'Create and deploy a Hello World app.'
- Documentation**: A section with links to 'Learn about Compute Engine', 'Learn about Cloud Storage', and 'Learn about App Engine'.
- Create an empty project**: A card with a plus icon and a 'Create an empty project' button.

At the bottom center, there is a 'VIEW MORE' link.

How to use the Google Cloud Platform:

Setting up your account

4. In the top blue bar, where it says “Select a project”, select “SCG Class-HUMBIO51-Aut17”



How to use the Google Cloud Platform:

Setting up your account

You have now successfully set up your GCP account. Next time, to access your account, simply go to <https://console.cloud.google.com> and follow the steps on the next slides.

How to use the Google Cloud Platform

The screenshot shows the Google Cloud Platform dashboard for the project "SCG Class - HUMBIO51-A...". The left-hand navigation menu is visible, with "Compute Engine" highlighted under the "COMPUTE" section. A red "1)" is placed next to the "Compute Engine" link. A dropdown menu is open for "Compute Engine", showing options like "Instance groups", "Instance templates", "Disks", "Snapshots", "Images", "Committed use discounts", "Metadata", "Health checks", "Zones", "Operations", "Quotas", and "Settings". A red "2)" is placed next to the "VM instances" link in the dropdown menu. The main content area displays the "Project info" card, the "Compute Engine" CPU usage graph, and the "Billing" section showing estimated charges of \$0.00.

Google Cloud Platform SCG Class - HUMBIO51-A... [CUSTOMIZE](#)

Home DASHBOARD ACTIVITY

Cloud Launcher
Billing
APIs & services
Support
Getting started
IAM & admin

COMPUTE

1) **Compute Engine**

- 2) VM instances
- Instance groups
- Instance templates
- Disks
- Snapshots
- Images
- Committed use discounts
- Metadata
- Health checks
- Zones
- Operations
- Quotas
- Settings

Project info
Project name: SCG Class - HUMBIO51-Aut17
Project ID: gbsc-gcp-class-humbio51-aut17
Project number: 499612801043

Compute Engine
CPU (%)
0.1
0.08
0.06
0.04
0.02
Sep 23, 3:30 PM Sep 23, 4:27 PM
CPU: 0.075
Go to the Compute Engine dashboard

Google Cloud Platform status
All services normal
Go to Cloud status dashboard

Billing
Estimated charges: \$0.00
For the billing period Sep 1 - 23, 2017
View detailed charges

Error Reporting
No sign of any errors. Have you set up Error Reporting?
Learn how to set up Error Reporting

News
Committed use discounts for Google Compute Engine now generally available

- 1) Click where it says Compute Engine
- 2) Click VM instances

Turn your instance on if needed

VM instances [+ CREATE INSTANCE](#) [IMPORT VM](#) [REFRESH](#) [START](#) [STOP](#) [RESET](#) [DELETE](#)

4)

Filter VM instances Columns

<input type="checkbox"/> Name ^	Zone	Recommendation	Internal IP	External IP	Network	Labels	Connect
<input checked="" type="checkbox"/> <input type="radio"/> instance-1	us-west1-a		10.138.0.14	None	default		SSH ⌵ ⋮

3)

3) check the box next to your instance

4) click on "START"

How to use the Google Cloud Platform

Google Cloud Platform SCG Class - HUMBI051-A...

Compute Engine

VM instances

CREATE INSTANCE IMPORT VM REFRESH START STOP RESET DELETE HIDE INFO PANEL

Filter VM instances

Name	Zone	Recommendation	Internal IP	External IP	Connect
instance-master	us-west1-a		10.138.0.2	35.197.55.121	SSH

Select an instance

LABELS MONITORING

Labels help organize your resources (e.g., cost_center:sales or env:prod).

No instances selected.

5) Click on the external IP address.

How to use the Google Cloud Platform

6) The password for your GCP instance is **humbio**

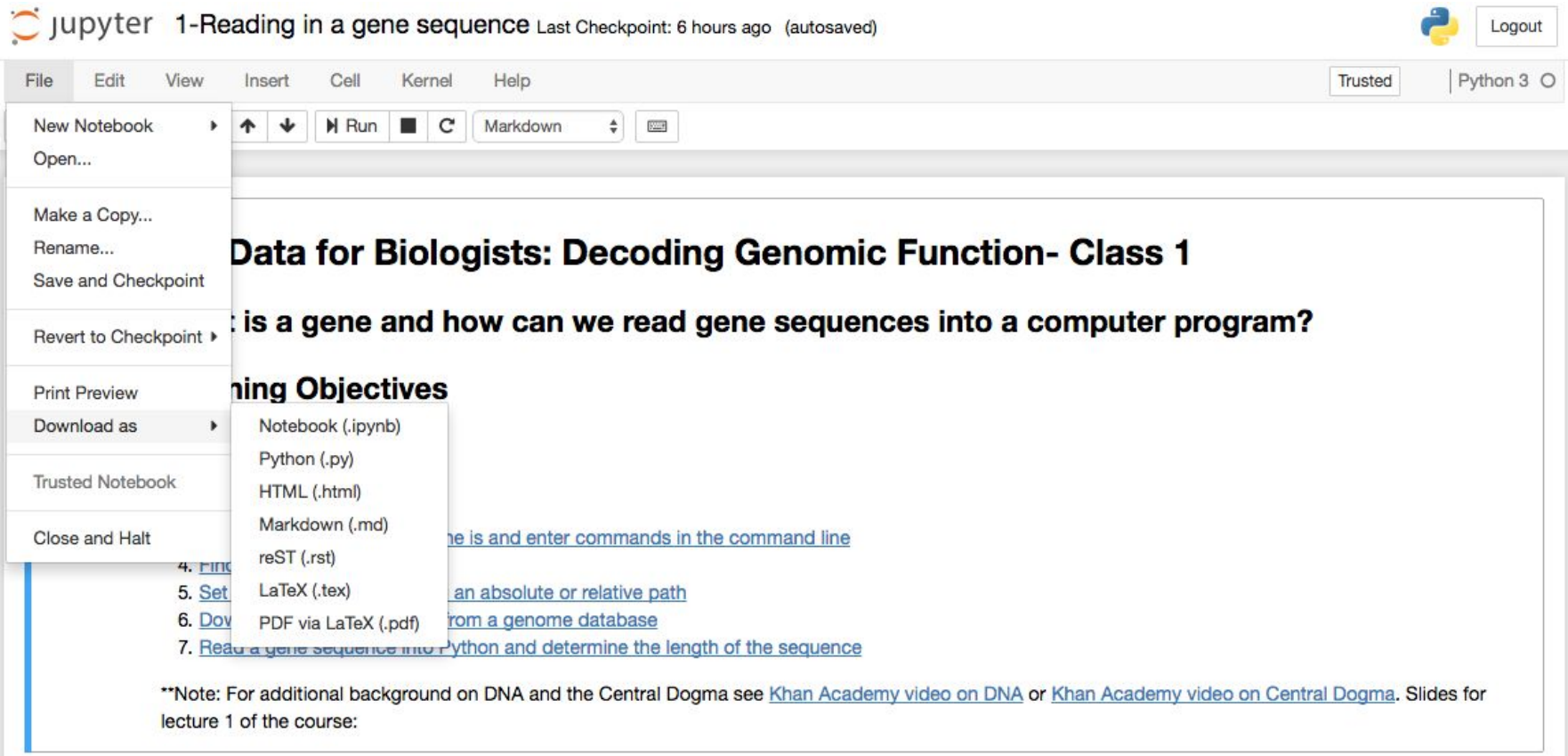


6)

Password:

Log in

How to save an assignment from the Google Cloud Platform



The screenshot shows a Jupyter Notebook interface. At the top, the title bar reads "jupyter 1-Reading in a gene sequence Last Checkpoint: 6 hours ago (autosaved)". On the right, there is a "Logout" button. Below the title bar is a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", and "Help". To the right of the menu bar are "Trusted" and "Python 3" buttons. The "File" menu is open, showing options: "New Notebook", "Open...", "Make a Copy...", "Rename...", "Save and Checkpoint", "Revert to Checkpoint", "Print Preview", "Download as", "Trusted Notebook", and "Close and Halt". The "Download as" sub-menu is open, listing file formats: "Notebook (.ipynb)", "Python (.py)", "HTML (.html)", "Markdown (.md)", "reST (.rst)", "LaTeX (.tex)", and "PDF via LaTeX (.pdf)". The notebook content includes the title "Data for Biologists: Decoding Genomic Function- Class 1", a question "What is a gene and how can we read gene sequences into a computer program?", and a section "Learning Objectives" with a list of tasks. A note at the bottom provides additional background on DNA and the Central Dogma.

jupyter 1-Reading in a gene sequence Last Checkpoint: 6 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Help Trusted Python 3

New Notebook
Open...
Make a Copy...
Rename...
Save and Checkpoint
Revert to Checkpoint
Print Preview
Download as
Trusted Notebook
Close and Halt

Notebook (.ipynb)
Python (.py)
HTML (.html)
Markdown (.md)
reST (.rst)
LaTeX (.tex)
PDF via LaTeX (.pdf)

Data for Biologists: Decoding Genomic Function- Class 1

What is a gene and how can we read gene sequences into a computer program?

Learning Objectives

1. [Define a gene](#) and enter commands in the command line
2. [Set an absolute or relative path](#)
3. [Download a gene sequence from a genome database](#)
4. [Read a gene sequence into Python and determine the length of the sequence](#)

****Note:** For additional background on DNA and the Central Dogma see [Khan Academy video on DNA](#) or [Khan Academy video on Central Dogma](#). Slides for lecture 1 of the course:

Assignments need to be submitted on Canvas!

Don't forget to turn your instance off when you are not using it!

Google Cloud Platform SCG Class - HUMBI051-A...

Compute Engine VM instances

CREATE INSTANCE IMPORT VM REFRESH START STOP RESET DELETE HIDE INFO PANEL

6)

Filter VM instances Columns

<input type="checkbox"/> Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input type="checkbox"/> instance-master	us-west1-a		10.138.0.2	35.197.55.121	SSH

Select an instance

LABELS MONITORING

Labels help organize your resources (e.g., cost_center:sales or env:prod).

No instances selected.

6) Click on “STOP”.

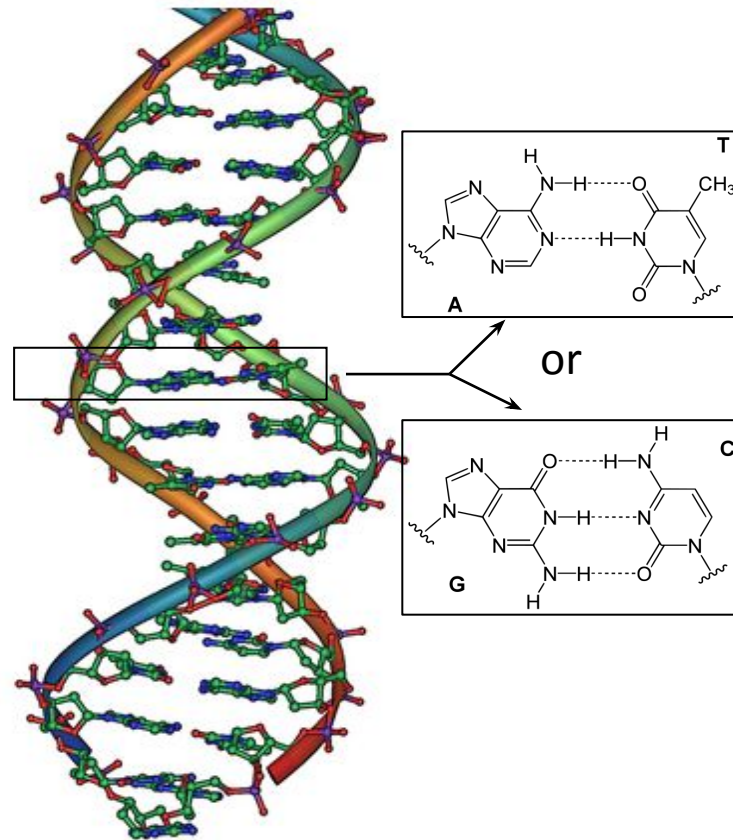
If your instance has been running for more than 12 hours, you will receive a reminder from Google to turn it off.

Learning Objectives

Students should be able to:

- Access Jupyter Notebooks
- Explain what a gene is
- Describe what a command line is and enter commands in the command line
- Find a working directory
- Set a working directory using an absolute or relative path
- Download a gene sequence from a genome database
- Read a gene sequence into Python and determine the length of the sequence

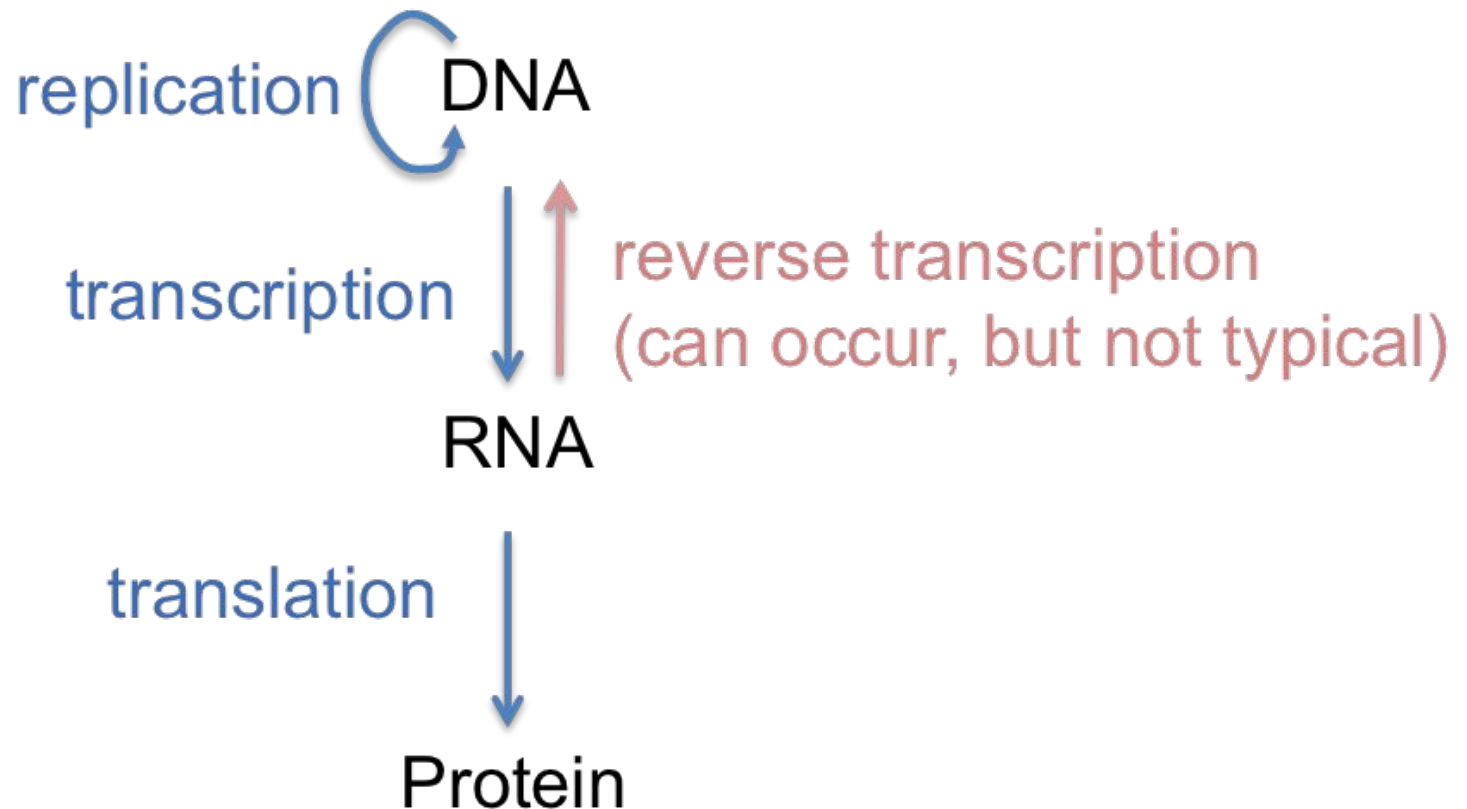
What is a gene?



DNA is the molecule in cells that enables the transmission of genetic information from one generation to the next.

Genes are segments of DNA that code for RNA and proteins, which are critical molecules for carrying out cellular function.

Flow of information in biological systems: The Central Dogma



What is the DNA between genes called?

Goal for the day: Read a gene sequence
into Python and calculate the length of
the sequence

What is the command line?

Command line is a location where you can enter code to give a computer program an instruction.

For example:

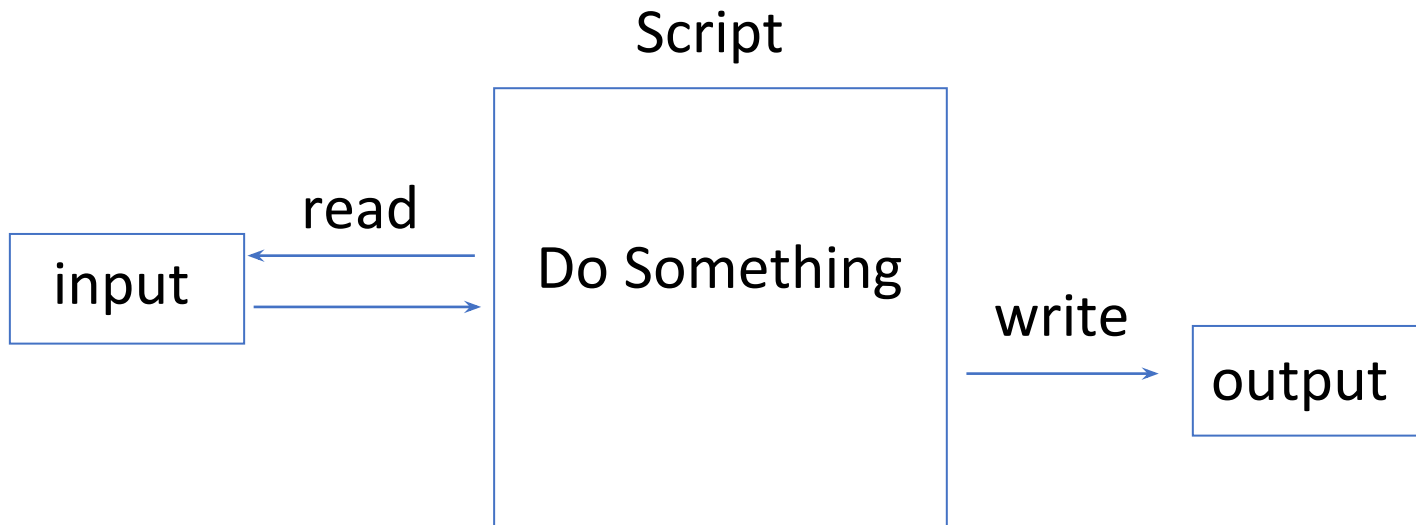
```
In [ ]: print ('Hello World')
```

Now try:

```
In [ ]: print(DNA makes RNA makes Protein)
```

Why didn't that work?

Overview of a computer program



How does the program know where to find the input
and where to put the output?

The location of the input and output can be defined
relative to your working directory

How do I find my working directory?

In Python:

```
In [ ]: #import os allows the Jupyter notebook to interact with the operating system.  
  
import os  
  
#gets the current working directory  
os.getcwd()
```

In Unix:

```
In [ ]: !pwd
```



The ! would not be necessary in a Unix terminal, but is needed to run Unix commands from a Python shell in Jupyter notebooks

How can I set my working directory using an absolute or relative path?

Relative path example:

Absolute path example:

How can I set my working directory using an absolute or relative path?

Relative path example:

```
In [ ]: import os

#changes directory back to the parent directory, two periods('.') stands for the parent directory
os.chdir('..')

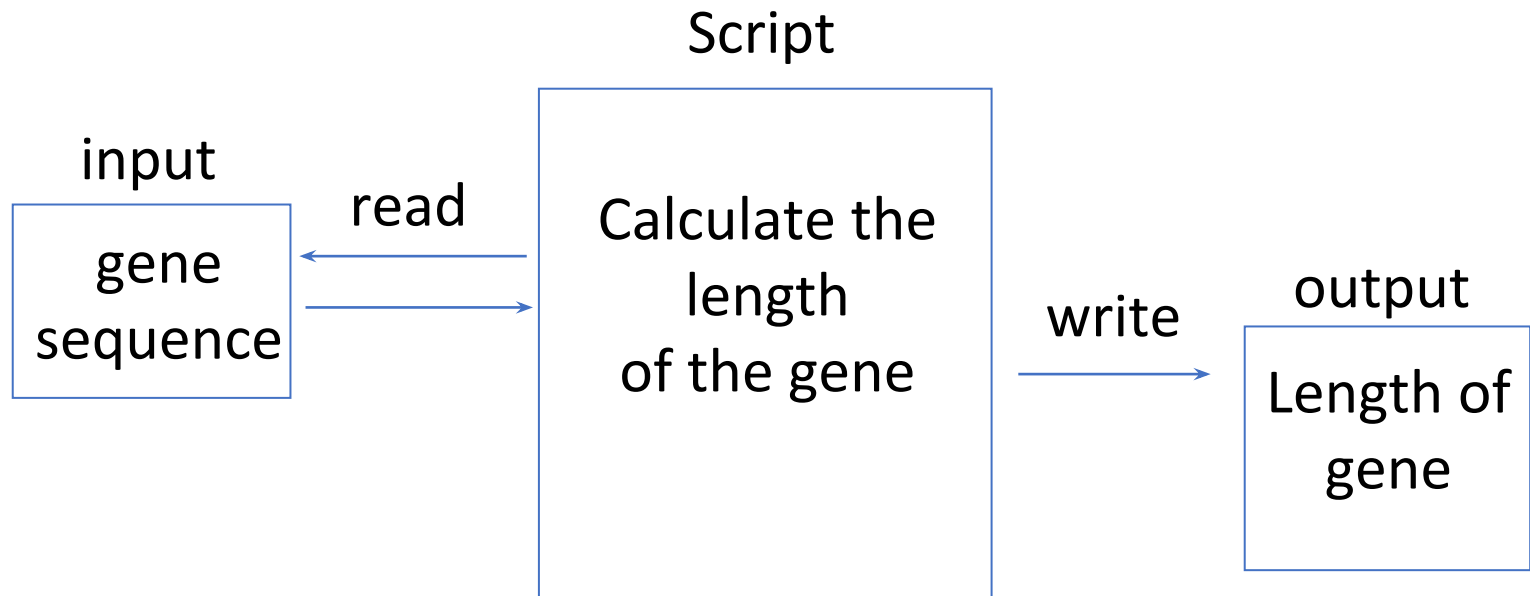
print(os.getcwd())
```

Absolute path example:

```
In [ ]: import os

#changes directory back to the parent directory, two periods('.') stands for the parent directory
os.chdir('/opt/vptl-course/')
|
print(os.getcwd())
```

Applying what we learned to an example from biology



How can I download a gene sequence
from a genome database?

How can I download a gene sequence from a genome database?

Example of a gene sequence (human insulin)

Homo sapiens insulin (INS), RefSeqGene on chromosome 11

NCBI Reference Sequence: NG_007114.1

[GenBank](#) [Graphics](#)

```
>NG_007114.1:4986-6416 Homo sapiens insulin (INS), RefSeqGene on chromosome 11
AGCCCTCCAGGACAGGCTGCATCAGAAGAGGCCATCAAGCAGGTCTGTCCAAGGGCCTTTGCCGTCAGGT
GGGCTCAGGATTCCAGGGTGGCTGGACCCAGGCCCCAGCTCTGCAGCAGGGAGGACGTGGCTGGGCTCG
TGAAGCATGTGGGGGTGAGCCCAGGGGCCCCAAGGCAGGGCACCTGGCCTTCAGCCTGCCCTCAGCCCTGC
CTGTCTCCCAGATCACTGTCCTTCTGCCATGGCCCTGTGGATGCGCCTCCTGCCCCTGCTGGCGCTGCTG
GCCCTCTGGGGACCTGACCCAGCCGACGCTTTGTGAACCAACACCTGTGCGGCTCACACCTGGTGAAG
CTCTCTACCTAGTGTGCGGGGAACGAGGCTTCTTCTACACACCCAAGACCCGCCGGGAGGCAGAGACCT
GCAGGGTGAGCCAACCTGCCCATTTGCTGCCCTTGGCCGCCCCAGCCACCCCTGCTCCTGGCGCTCCAC
CCAGCATGGGCAGAAAGGGGCGAGGAGCTGCCACCCAGCAGGGGTCAGGTGCACTTTTTTAAAAAGAAG
TTCTCTTGGTCACGTCCTAAAAAGTGACCAGCTCCCTGTGGCCAGTCAGAATCTCAGCCTGAGGACGGTG
TTGGCTTGGCAGCCCCGAGATACATCAGAGGGTGGGCACGCTCCTCCCTCCACTCGCCCCCTAAACAAA
TGCCCCGCAGCCCATTTCTCCACCCTCATTTGATGACCGCAGATTCAAGTGTTTTGTAAAGTAAAGTCCT
GGGTGACCTGGGGTCACAGGGTGCCCCACGCTGCCTGCCTCTGGGCGAACACCCCATCACGCCCGGAGGA
GGGCGTGGCTGCCCTGAGTGGGCCAGACCCCTGTGCGCAGGCCCTCACGGCAGCTCCATAGTCAGGAG
ATGGGGAAGATGCTGGGGACAGGCCCTGGGGAGAAGTACTGGGATCACCCTGTTGAGGCTCCCACTGTGAC
GCTGCCCGGGGGGGGGAAGGAGGTGGGACATGTGGGCGTTGGGGCTGTAGGTCCACACCCAGTGTGG
GTGACCCTCCCTCTAACCTGGGTCCAGCCCGGCTGGAGATGGGTGGGAGTGCACCTAGGGCTGGCGGGC
AGGCGGGCACTGTGTCTCCCTGACTGTGTCCTCCTGTGTCCTCTGCCTCGCCGCTGTTCCGGAACCTGC
TCTGCGCGGCACGTCCTGGCAGTGGGGCAGGTGGAGCTGGGCGGGGGCCCTGGTGCAGGCAGCCTGCAGC
CCTTGGCCCTGGAGGGGTCCCTGCAGAAGCGTGGCATTGTGGAACAATGCTGTACCAGCATCTGCTCCCT
CTACCAGCTGGAGAACTACTGCAACTAGACGCAGCCCGCAGGCAGCCCCACACCCGCCGCTCCTGCACC
GAGAGAGATGGAATAAAGCCCTTGAACCAGC
```

How can I read a gene sequence into Python and determine the length of the sequence?

```
In [ ]: #Open a file and create a file object  
#The 'r' means that the file is readable. 'w' would mean the file is writable.  
FASTAgenessequence=open('data/Human-Insulin-NG_007114.1.txt','r')  
  
#Read the sequence file contents and print them  
print(FASTAgenessequence.read())
```


What else do we need to do
before determining the length of
the sequence?

How can I remove the first line?

```
In [ ]: #Open a file and create a file object  
FASTAgenesecquence=open('data/Human-Insulin-NG_007114.1.txt','r')  
  
#Read the lines of the sequence and trim the first line  
#The numbering of lines or characters in Python starts with 0, so the first line is line 0.  
  
genesecquence=(FASTAgenesecquence.readlines()[1:])  
  
print(genesecquence)
```

How can I read a gene sequence into Python and determine the length of the sequence?

```
In [1]: #Read in the sequence and trim the first line
FASTAgenesquence=open('data/Human-Insulin-NG_007114.1.txt','r')
genesquence=(FASTAgenesquence.readlines()[1:])

#joins the lines in genesquence into a single string
genesquence=''.join(genesquence)

#removes the linebreaks
genesquence=genesquence.replace('\n','')

#calculates the length of the genesquence
print(len(genesquence))
```