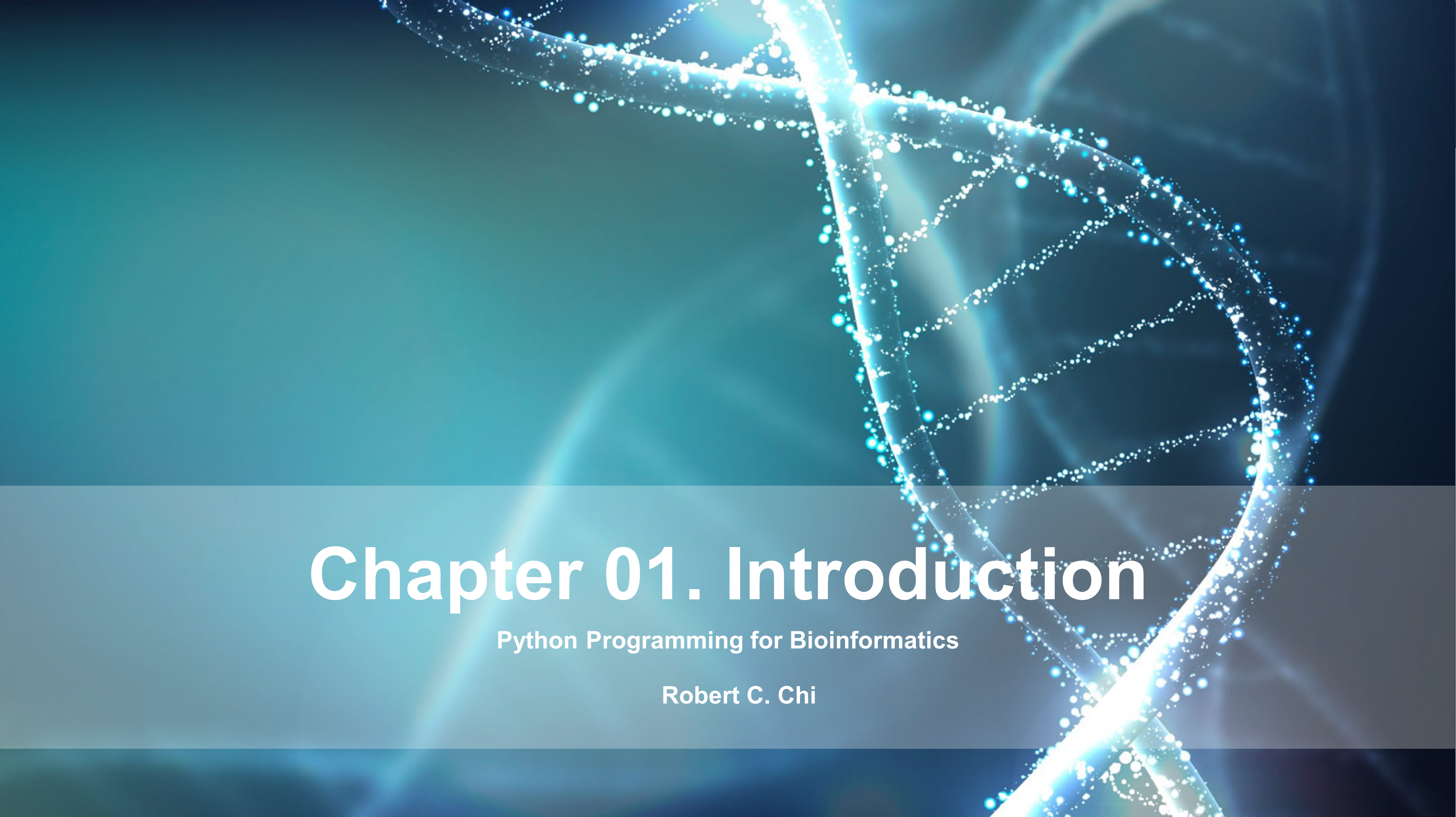


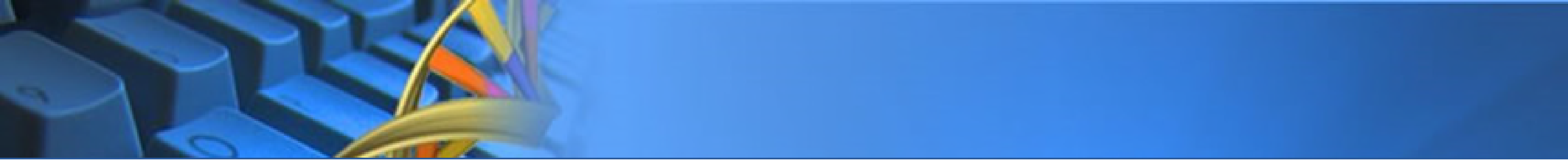
Chapter 01. Introduction

Python Programming for Bioinformatics

Robert C. Chi

Agenda

- About This Course
- Introduction to Biopython
- Installing Biopython
- A Quick Tour of Biopython



ABOUT THIS COURSE

Robert C. Chi (紀俊男)

- **Education**
  - Ph.D. Candidate / Bioinformatics
    Taiwan International Graduate Program (TIGP), 2003-2007
  - Master / Computer Sciences
    Queens College, CUNY, 1994-1996
  - Bachelor / Computer Sciences
    Fu-Jen Catholic University
- **Experience**
  - Training Director / AMI (2014-2020)
  - Founder / Hatch Information Co., Ltd. (2007-2013)
  - Research Assistant / Academia Sinica (2000-2007)
  - Manager of Tech Support / Trend Micro Co., Ltd. (1998-2000)
  - Game Developer / CG Animation Co., Ltd. (1997-1998)
- **Expertise**
  - Artificial Intelligence (AI), Embedded System, Computer Security, Game Programming.

Syllabus

- **Part I. Python (10 Hr)**
  - Python & Environments
  - Literals & Variables
  - Input & Output
  - Branch & Loop
  - String Manipulation
  - Compound Data Types
    - Tuple, List, Dictionary, Set
  - Functions
  - Data Science Packages
    - NumPy, Pandas, MatPlotLib

DONE

- **Part II. BioPython (14 Hr + 7Hr)**
  - Introduction
  - Read/Write Bioinformatic Files
    - FASTA, GenBank, SwissProt, ExPASy, KEGG…
  - Sequence Manipulation
    - Transcription, Translation, Alignment
  - Databases Handling
    - BLAST, NCBI Entrez…
  - Working with 3D Structures
  - Machine Learning
    - Data Pre-Processing
    - Classification
    - Clustering

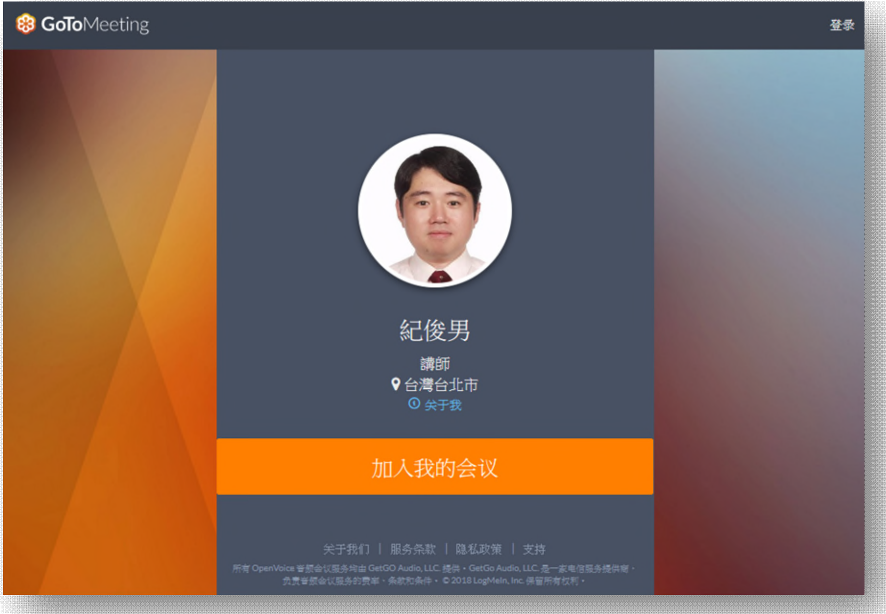
Schedule & Environment

- **Schedule**
  - Part I (5 Weeks)
    - 2021/08/06 ~ 2021/09/03
    - Fri. 15:00 ~ 17:00
  - Part II (7 Weeks)
    - 2021/10/29 ~ 2021/12/10
    - Fri. 14:00 ~ 17:00 (2Hr + 1Hr)
- **Location: Online**
  - https://www.gotomeet.me/TeacherChi/BioPython

- **Environments**
  - Google Colab
- **Lecturing in**
  - English
- **Teaching Style**
  - Part I: Lecturing
  - Part II: Lecturing + Practiceing

Live Broadcasting

- URL: https://www.gotomeet.me/TeacherChi/BioPython



紀俊男
講師
台灣台北市
关于我

所有 OpenVoice 音频会议服务均由 GetGO Audio, LLC. 提供。GetGo Audio, LLC. 是一家电信服务提供商，负责音频会议服务的费率、条款和条件。 © 2018 LogMeIn, Inc. 保留所有权利。

App Download

GoToMeeting

ANDROID APP ON Google play

Download on the App Store

Meeting ID： 117-684-245

Pre-requisites

Python Programming

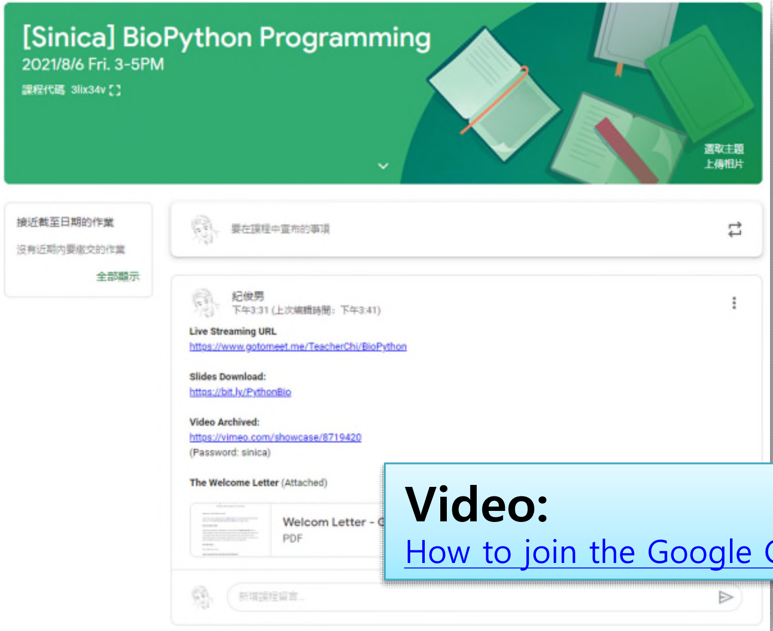
(Flow Controls, Compound Data...etc.)

Google Colab

(Development Environment)

Resources & Courses Make Up

- Google Classroom： https://bit.ly/BioPy-202108



[Sinica] BioPython Programming

2021/8/6 Fri. 3-5PM

課程代碼 3lix34v

接近截至日期的作業

沒有近期內要繳交的作業

要在課程中宣布的事項

紀俊男
下午3:31 (上次編輯時間：下午3:41)

Live Streaming URL
https://www.gotomeet.me/TeacherChi/BioPython

Slides Download:
https://bit.ly/PythonBio

Video Archived:
https://vimeo.com/showcase/8719420
(Password: sinica)

The Welcome Letter (Attached)

Welcom Letter - C
PDF

新增課程留言...

Video:
How to join the Google Classroom

App Download

ANDROID APP ON Google play

Download on the App Store

Google Classroom

Join ID: 3lix34v

INTRODUCTION TO BIOPYTHON

What is Biopython?

- The most popular **Python package** for **computational molecular biology**.

https://biopython.org/

Official Name:

Biopython ✓

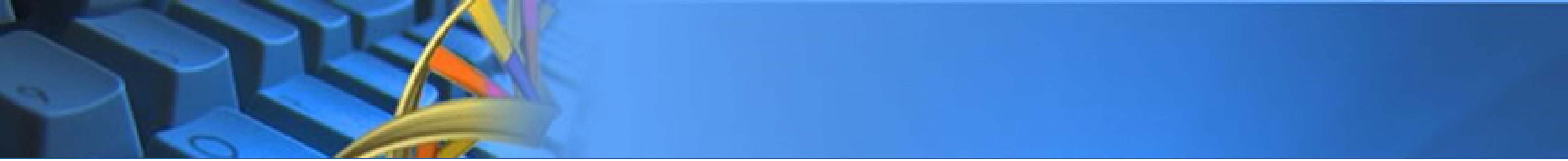
BioPython ✗

Abilities of Biopython

- **Communicating with Bioinformatic Services**
  - NCBI Services (Blast, Entrez, PubMed)
  - ExPASy Services (Swiss-Prot, Prosite entries, Prosite searches)
- **Parsing Bioinformatic Files**
  - Blast, Clustalw, FASTA, GenBank, PubMed/Medline, ExPASy (Enzyme/Prosite), SCOP (‘dom’ & ‘lin’ files), UniGene, SwissProt
- **Performing Sequence Operations**
  - Translation, transcription, weight calculations, alignments…etc.
- **Performing Classification of Data**
  - k Nearest Neighbors, Naive Bayes, Support Vector Machines…etc.
- **Integrating with BioSQL**
  - A sequence database schema also supported by the ***BioPerl*** and ***BioJava*** projects.

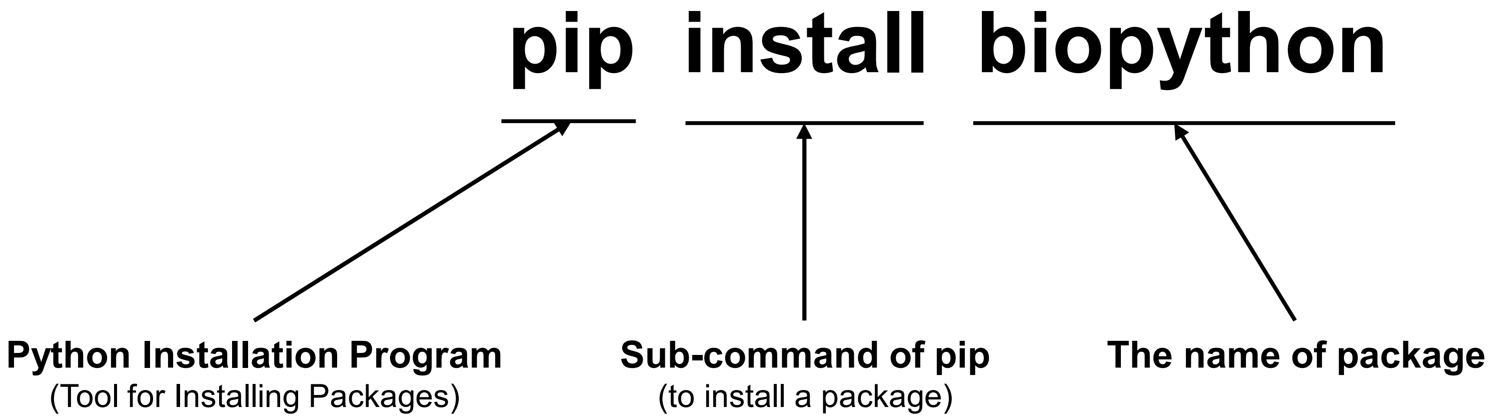
How to Cite Biopython?

- **The main Biopython reference**
  - [Cock *et al.*, 2009] (URL)
- **The official project announcement**
  - [Chapman and Chang, 2000] (URL)
- **For Bio.PDB**
  - [Hamelryck and Manderick, 2003] (URL)
- **For Bio.Cluster**
  - [De Hoon *et al.*, 2004] (URL)
- **For Bio.Graphics.GenomeDiagram**
  - [Pritchard *et al.*, 2006] (URL)
- **For Bio.Phylo and Bio.Phylo.PAML**
  - [Talevich *et al.*, 2012] (URL)
- **For the FASTQ file format as supported in Biopython, BioPerl, BioRuby, BioJava, and EMBOSS**
  - [Cock *et al.*, 2010] (URL)



INSTALLING BIOPYTHON

Install Biopython



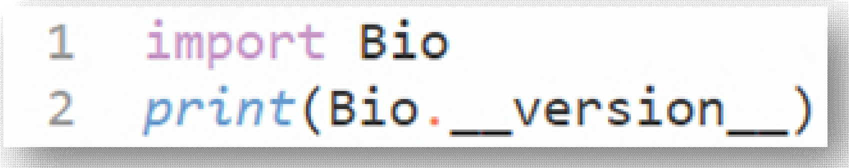
```
pip install biopython
```

Python Installation Program
(Tool for Installing Packages)

Sub-command of pip
(to install a package)

The name of package

Check If Installed Successfully

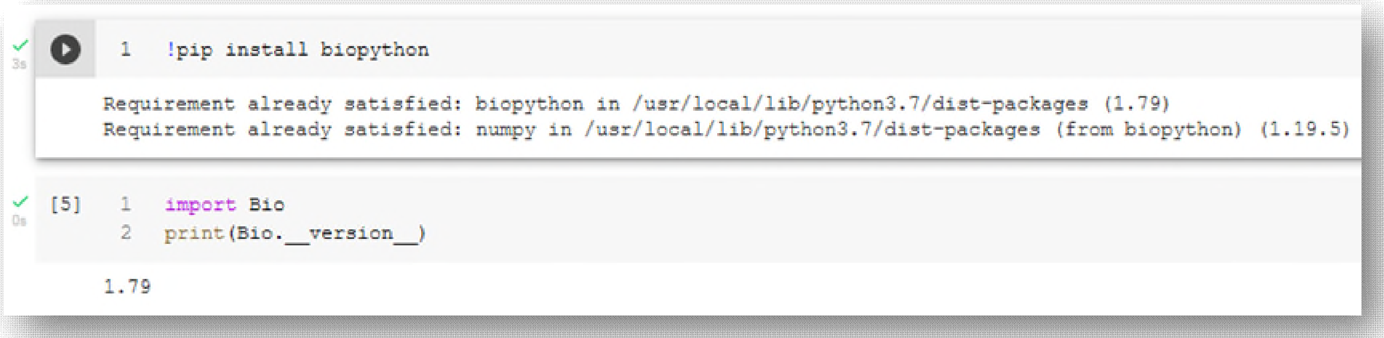


```
import Bio
print(Bio.__version__)
```

```
1.79
```

Practice

- **Install Biopython**
  - Open a Colab page called “BiopythonInstall.ipynb”.
  - Write and run the following codes:



```
!pip install biopython
```

```
Requirement already satisfied: biopython in /usr/local/lib/python3.7/dist-packages (1.79)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from biopython) (1.19.5)
```

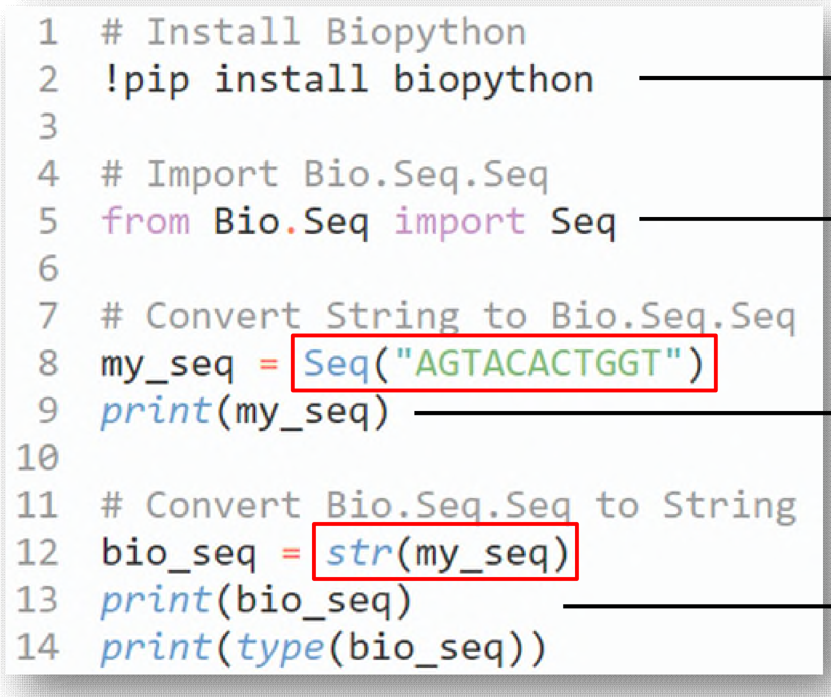
```
[5]
import Bio
print(Bio.__version__)
```

```
1.79
```

(Solution URL of this Practice)

A QUICK TOUR OF BIOPYTHON

Convert Strings to Bio.Seq.Seq



```python
# Install Biopython
!pip install biopython

# Import Bio.Seq.Seq
from Bio.Seq import Seq

# Convert String to Bio.Seq.Seq
my_seq = Seq("AGTACACTGGT")
print(my_seq)

# Convert Bio.Seq.Seq to String
bio_seq = str(my_seq)
print(bio_seq)
print(type(bio_seq))
```

Install Biopython onto Colab

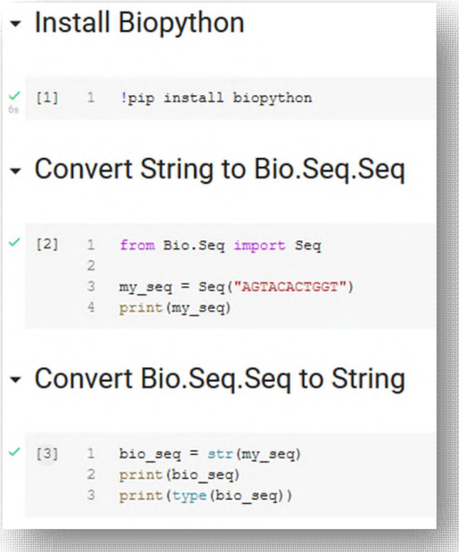
Import Bio.Seq.Seq

```
AGTACACTGGT
```

```
AGTACACTGGT
<class 'str'>
```

Practice

- **Convert Strings to Bio.Seq.Seq**
  - Write and Run the following codes on a Colab page called “QuickTour.ipynb”:



Install Biopython

```
!pip install biopython
```

Convert String to Bio.Seq.Seq

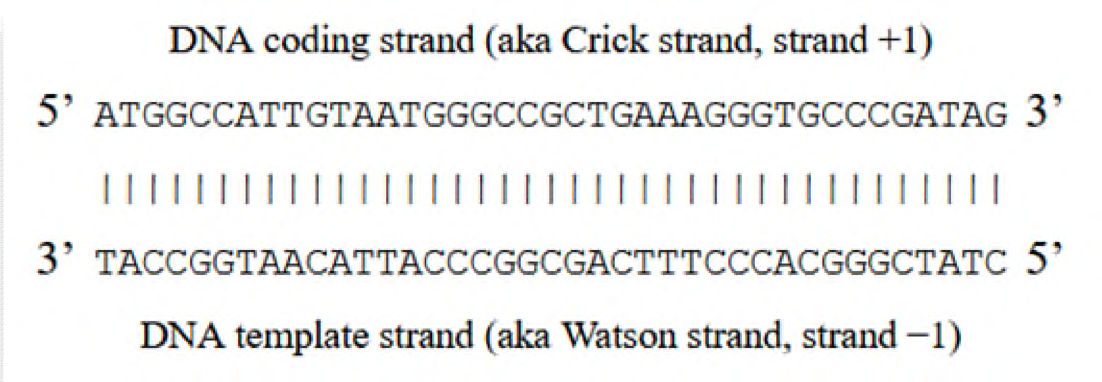
```
from Bio.Seq import Seq

my_seq = Seq("AGTACACTGGT")
print(my_seq)
```

Convert Bio.Seq.Seq to String

```
bio_seq = str(my_seq)
print(bio_seq)
print(type(bio_seq))
```

(Solution URL of this Practice)

Complement of a Sequence



DNA coding strand (aka Crick strand, strand +1)

```
5' ATGGCCATTGTAATGGGCCGCTGAAAGGGTGCCCGATAG 3'
   |||||||||||||||||||||||||||||||||||||||
3' TACCGGTAACATTACCCGGCGACTTTCCCACGGGCTATC 5'
```

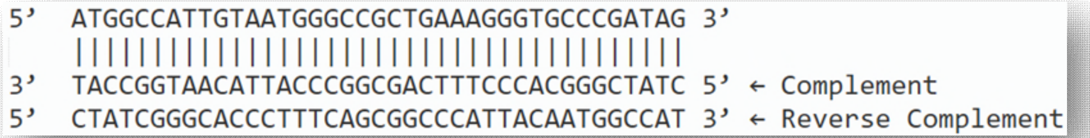
DNA template strand (aka Watson strand, strand −1)

Complement

A → T   C → G

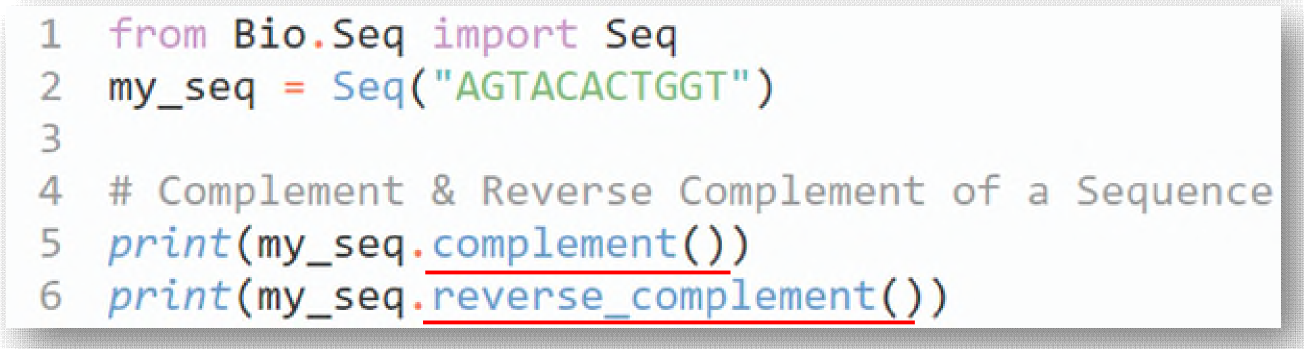
T → A   G → C

Reverse Complement



```
5'  ATGGCCATTGTAATGGGCCGCTGAAAGGGTGCCCGATAG 3'
    |||||||||||||||||||||||||||||||||||||||
3'  TACCGGTAACATTACCCGGCGACTTTCCCACGGGCTATC 5' ← Complement
5'  CTATCGGGCACCCTTTCAGCGGCCCATTACAATGGCCAT 3' ← Reverse Complement
```

Complement of a Sequence

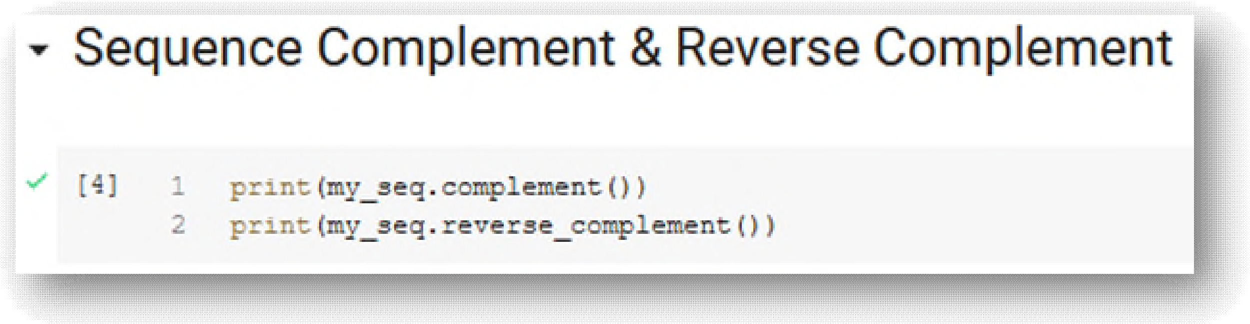


```python
from Bio.Seq import Seq
my_seq = Seq("AGTACACTGGT")

# Complement & Reverse Complement of a Sequence
print(my_seq.complement())
print(my_seq.reverse_complement())
```

```
TCATGTGACCA
ACCAGTGTACT
```

Practice

- **Complement of a Sequence**
  - Write and Run the following codes on a Colab page called “QuickTour.ipynb”:



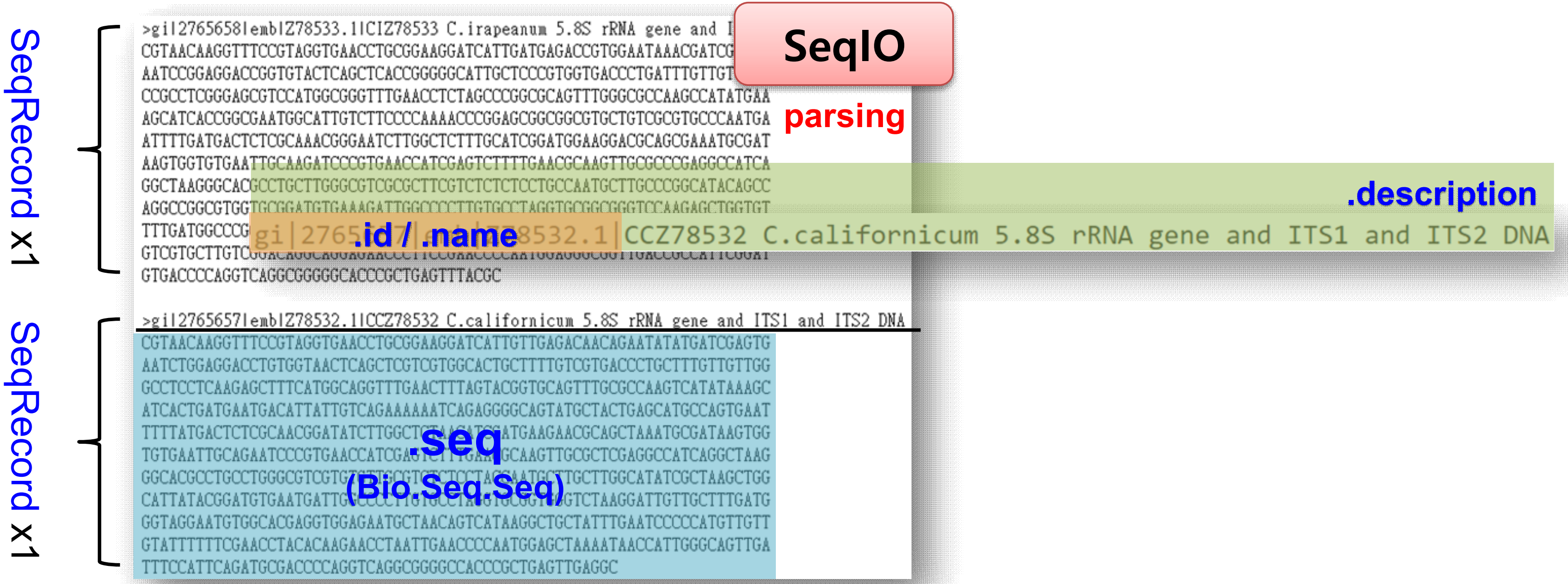
Sequence Complement & Reverse Complement

```
print(my_seq.complement())
print(my_seq.reverse_complement())
```

(Solution URL of this Practice)

Parsing FASTA Files

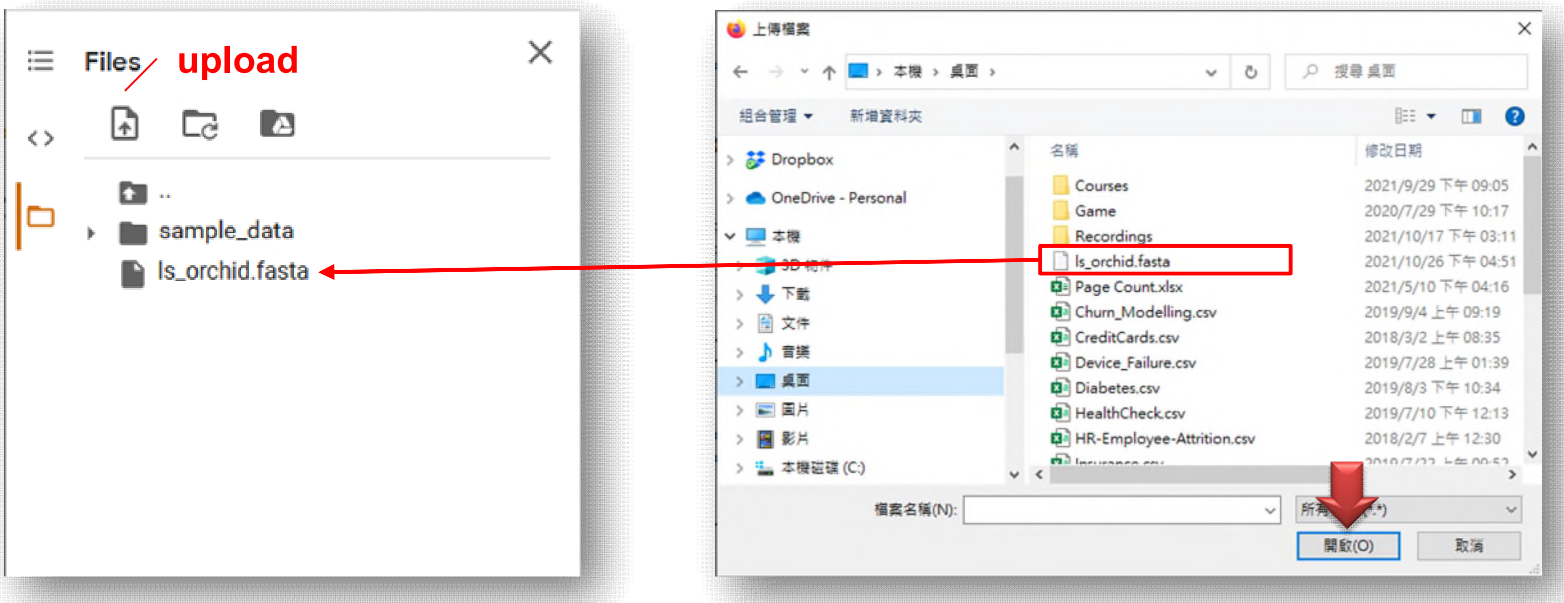
- "ls_orchid.fasta" (Download Link)



```
>gi|2765658|emb|Z78533.1|CIZ78533 C.irapeanum 5.8S rRNA gene and I
CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGGAATAAACGATCG
AATCCGGAGGACCGGTGTACTCAGCTCACCGGGGGCATTGCTCCCGTGGTGACCCTGATTTGTTGT
CCGCCTCGGGAGCGTCCATGGCGGGTTTGAACCTCTAGCCCGGCGCAGTTTGGGCGCCAAGCCATATGAA
AGCATCACCGGCGAATGGCATTGTCTTCCCCAAAACCCGGAGCGGCGGCGTGCTGTCGCGTGCCCAATGA
ATTTTGATGACTCTCGCAAACGGGAATCTTGGCTCTTTGCATCGGATGGAAGGACGCAGCGAAATGCGAT
AAGTGGTGTGAATTGCAAGATCCCGTGAACCATCGAGTCTTTTGAACGCAAGTTGCGCCCGAGGCCATCA
GGCTAAGGGCACGCCTGCTTGGGCGTCGCGCTTCGTCTCTCTCCTGCCAATGCTTGCCCGGCATACAGCC
AGGCCGGCGTGGTGCGGATGTGAAAGATTGGCCCCTTGTGCCTAGGTGCGGCGGGTCCAAGAGCTGGTGT
TTTGATGGCCCG
GTCGTGCTTGTCGGACAGGCAGGAGAACCCTTCCGAACCCCAATGGAGGGCGGTTGACCGCCATTCGGAT
GTGACCCCAGGTCAGGCGGGGGCACCCGCTGAGTTTACGC

>gi|2765657|emb|Z78532.1|CCZ78532 C.californicum 5.8S rRNA gene and ITS1 and ITS2 DNA
CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGTTGAGACAACAGAATATATGATCGAGTG
AATCTGGAGGACCTGTGGTAACTCAGCTCGTCGTGGCACTGCTTTTGTCGTGACCCTGCTTTGTTGTTGG
GCCTCCTCAAGAGCTTTCATGGCAGGTTTGAACTTTAGTACGGTGCAGTTTGCGCCAAGTCATATAAAGC
ATCACTGATGAATGACATTATTGTCAGAAAAAATCAGAGGGGCAGTATGCTACTGAGCATGCCAGTGAAT
TTTTATGACTCTCGCAACGGATATCTTGGCTC
TGTGAATTGCAGAATCCCGTGAACCATCGA
GGCACGCCTGCCTGGGCGTCGTG
CATTATACGGATGTGAATGATTG
GGTAGGAATGTGGCACGAGGTGGAGAATGCTAACAGTCATAAGGCTGCTATTTGAATCCCCCATGTTGTT
GTATTTTTTCGAACCTACACAAGAACCTAATTGAACCCCAATGGAGCTAAAATAACCATTGGGCAGTTGA
TTTCCATTCAGATGCGACCCCAGGTCAGGCGGGGCCACCCGCTGAGTTGAGGC
```

Parsing FASTA Files

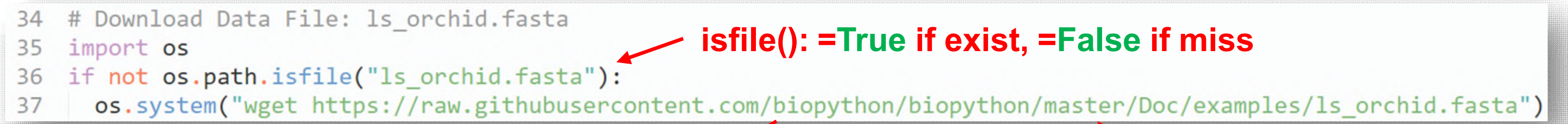
- Upload “ls_orchid.fasta” to Colab (by the Local File)



名稱	修改日期
Courses	2021/9/29 下午 09:05
Game	2020/7/29 下午 10:17
Recordings	2021/10/17 下午 03:11
ls_orchid.fasta	2021/10/26 下午 04:51
Page Count.xlsx	2021/5/10 下午 04:16
Churn_Modelling.csv	2019/9/4 上午 09:19
CreditCards.csv	2018/3/2 上午 08:35
Device_Failure.csv	2019/7/28 上午 01:39
Diabetes.csv	2019/8/3 下午 10:34
HealthCheck.csv	2019/7/10 下午 12:13
HR-Employee-Attrition.csv	2018/2/7 上午 12:30
Insurance.csv	2019/7/22 上午 09:52

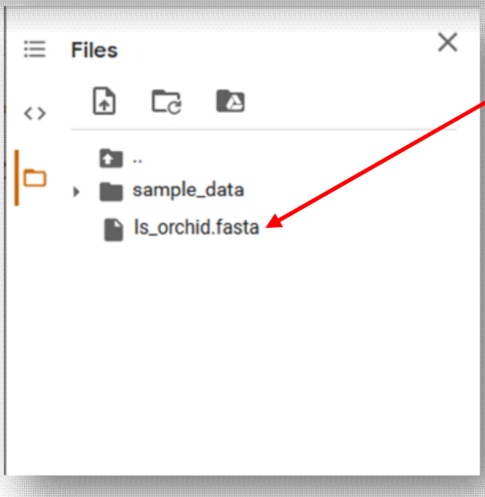
Parsing FASTA Files

- **Upload "ls_orchid.fasta" to Colab (by fetching from Internet)**



```python
# Download Data File: ls_orchid.fasta
import os
if not os.path.isfile("ls_orchid.fasta"):
    os.system("wget https://raw.githubusercontent.com/biopython/biopython/master/Doc/examples/ls_orchid.fasta")
```

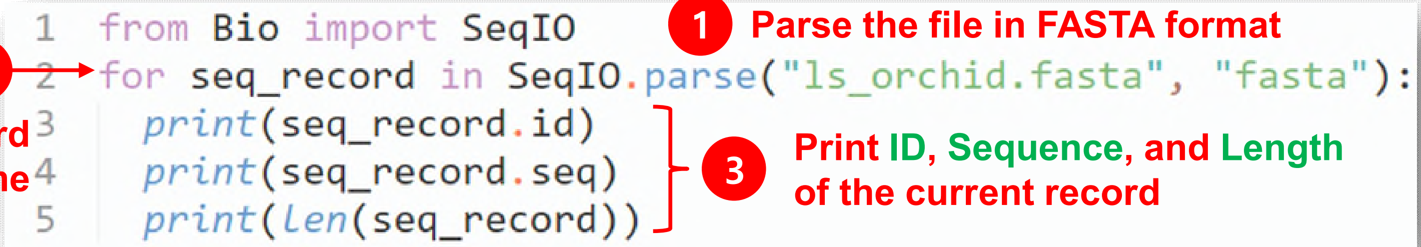
isfile(): =True if exist, =False if miss



wget: Web GET. Download file by URL.

os.system(): Send the command to OS.

Parsing FASTA Files

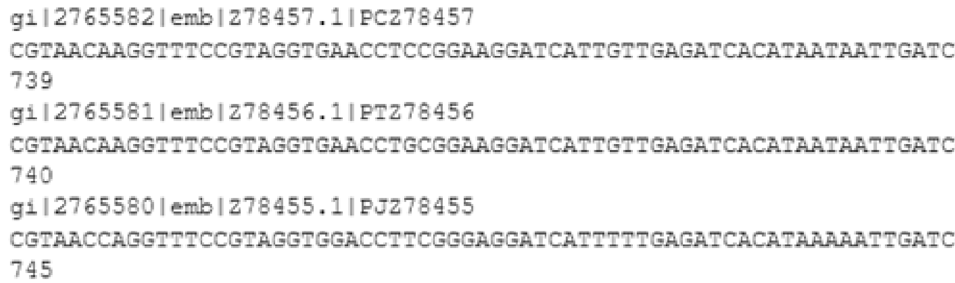


```python
from Bio import SeqIO
for seq_record in SeqIO.parse("ls_orchid.fasta", "fasta"):
    print(seq_record.id)
    print(seq_record.seq)
    print(len(seq_record))
```

1 Parse the file in FASTA format

2 Get one record a time

3 Print ID, Sequence, and Length of the current record



1st SeqRecord

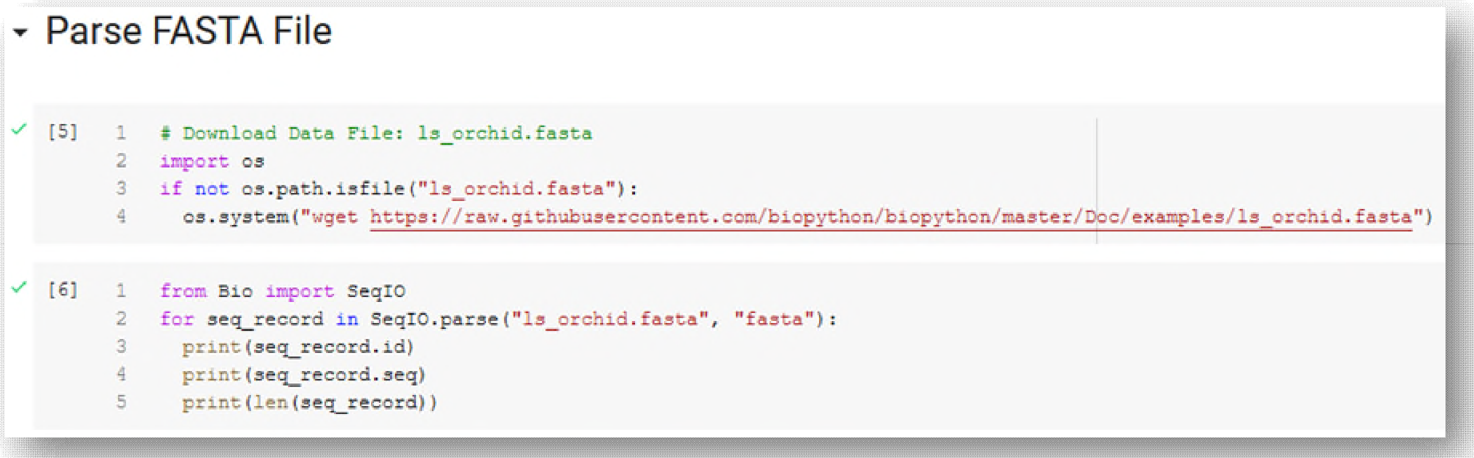
2nd SeqRecord

3rd SeqRecord

```
gi|2765582|emb|Z78457.1|PCZ78457
CGTAACAAGGTTTCCGTAGGTGAACCTCCGGAAGGATCATTGTTGAGATCACATAATAATTGATC
739
gi|2765581|emb|Z78456.1|PTZ78456
CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGTTGAGATCACATAATAATTGATC
740
gi|2765580|emb|Z78455.1|PJZ78455
CGTAACCAGGTTTCCGTAGGTGGACCTTCGGGAGGATCATTTTTGAGATCACATAAAAATTGATC
745
```

Practice

- **Parsing FASTA Files**
  - Write and Run the following codes on a Colab page called "QuickTour.ipynb":



Parse FASTA File

✓ [5]

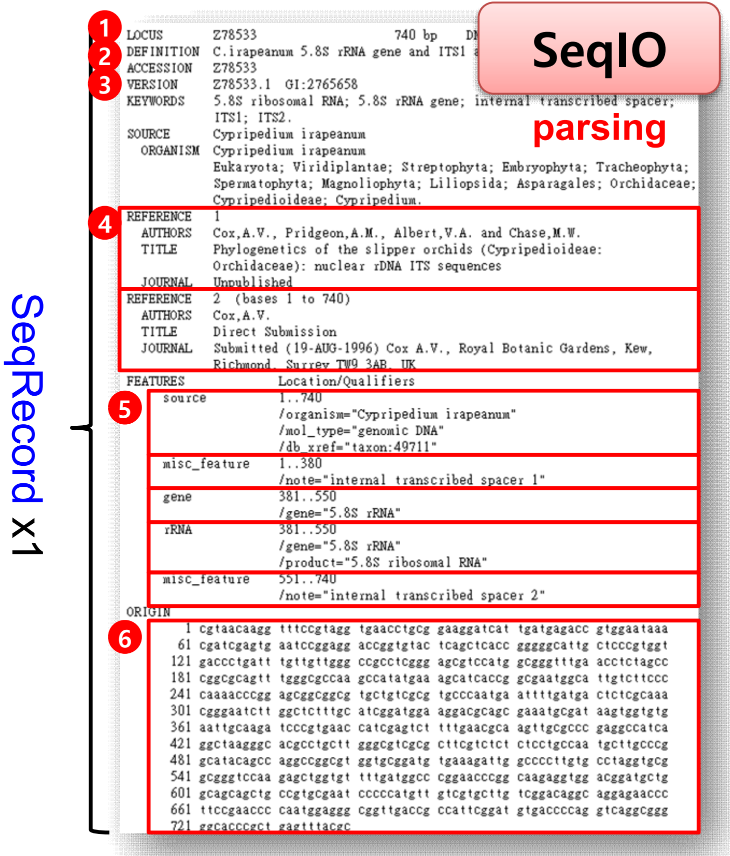
```
# Download Data File: ls_orchid.fasta
import os
if not os.path.isfile("ls_orchid.fasta"):
  os.system("wget https://raw.githubusercontent.com/biopython/biopython/master/Doc/examples/ls_orchid.fasta")
```

✓ [6]

```
from Bio import SeqIO
for seq_record in SeqIO.parse("ls_orchid.fasta", "fasta"):
  print(seq_record.id)
  print(seq_record.seq)
  print(len(seq_record))
```

(Solution URL of this Practice)

Parsing GenBank Files

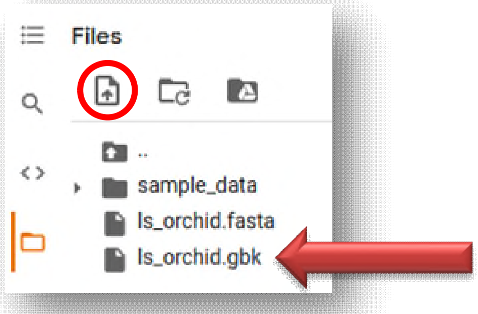


```
LOCUS       Z78533                  740 bp    DN
DEFINITION  C.irapeanum 5.8S rRNA gene and ITS1 a
ACCESSION   Z78533
VERSION     Z78533.1  GI:2765658
KEYWORDS    5.8S ribosomal RNA; 5.8S rRNA gene; internal transcribed spacer;
            ITS1; ITS2.
SOURCE      Cypripedium irapeanum
  ORGANISM  Cypripedium irapeanum
            Eukaryota; Viridiplantae; Streptophyta; Embryophyta; Tracheophyta;
            Spermatophyta; Magnoliophyta; Liliopsida; Asparagales; Orchidaceae;
            Cypripedioideae; Cypripedium.
REFERENCE   1
  AUTHORS   Cox,A.V., Pridgeon,A.M., Albert,V.A. and Chase,M.W.
  TITLE     Phylogenetics of the slipper orchids (Cypripedioideae:
            Orchidaceae): nuclear rDNA ITS sequences
  JOURNAL   Unpublished
REFERENCE   2  (bases 1 to 740)
  AUTHORS   Cox,A.V.
  TITLE     Direct Submission
  JOURNAL   Submitted (19-AUG-1996) Cox A.V., Royal Botanic Gardens, Kew,
            Richmond, Surrey TW9 3AB, UK
FEATURES             Location/Qualifiers
     source          1..740
                     /organism="Cypripedium irapeanum"
                     /mol_type="genomic DNA"
                     /db_xref="taxon:49711"
     misc_feature    1..380
                     /note="internal transcribed spacer 1"
     gene            381..550
                     /gene="5.8S rRNA"
     rRNA            381..550
                     /gene="5.8S rRNA"
                     /product="5.8S ribosomal RNA"
     misc_feature    551..740
                     /note="internal transcribed spacer 2"
ORIGIN
        1 cgtaacaagg tttccgtagg tgaacctgcg gaaggatcat tgatgagacc gtggaataaa
       61 cgatcgagtg aatccggagg accggtgtac tcagctcacc gggggcattg ctcccgtggt
      121 gaccctgatt tgttgttggg ccgcctcggg agcgtccatg gcgggtttga acctctagcc
      181 cggcgcagtt tgggcgccaa gccatatgaa agcatcaccg gcgaatggca ttgtcttccc
      241 caaaacccgg agcggcggcg tgctgtcgcg tgcccaatga attttgatga ctctcgcaaa
      301 cgggaatctt ggctctttgc atcggatgga aggacgcagc gaaatgcgat aagtggtgtg
      361 aattgcaaga tcccgtgaac catcgagtct ttgaacgca agttgcgccc gaggccatca
      421 ggctaagggc acgcctgctt gggcgtcgcg cttcgtctct ctcctgccaa tgcttgcccg
      481 gcatacagcc aggccggcgt ggtgcggatg tgaaagattg gccccttgtg cctaggtgcg
      541 gcgggtccaa gagctggtgt tttgatggcc cggaacccgg caagaggtgg acggatgctg
      601 gcagcagctg ccgtgcgaat cccccatgtt gtcgtgcttg tcggacaggc aggagaaccc
      661 ttccgaaccc caatggaggg cggttgaccg ccattcggat gtgaccccag gtcaggcggg
      721 ggcacccgct gagtttacgc
```

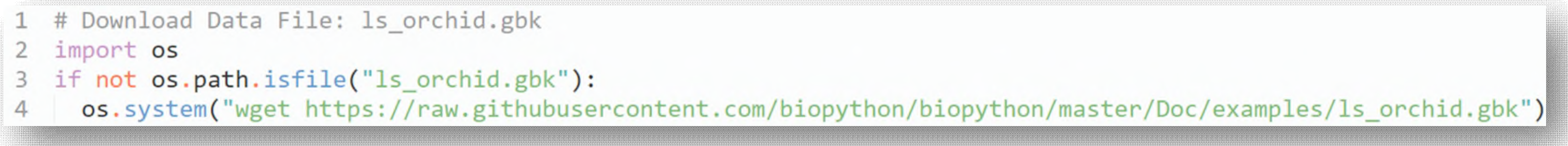
- "ls_orchid.gbk" (Download Link)
  1. .name → LOCUS
  2. .description → DEFINITION
  3. .id → VERSION
  4. .annotations['references']
     → REFERENCE xN
  5. .features → FEATURES xN
  6. .seq → ORIGIN
  7. .annotations
     → All the other information

Parsing GenBank Files

- Upload “ls_orchid.gbk” to Colab (by the Local File)



- Upload “ls_orchid.gbk” to Colab (by fetching from Internet)

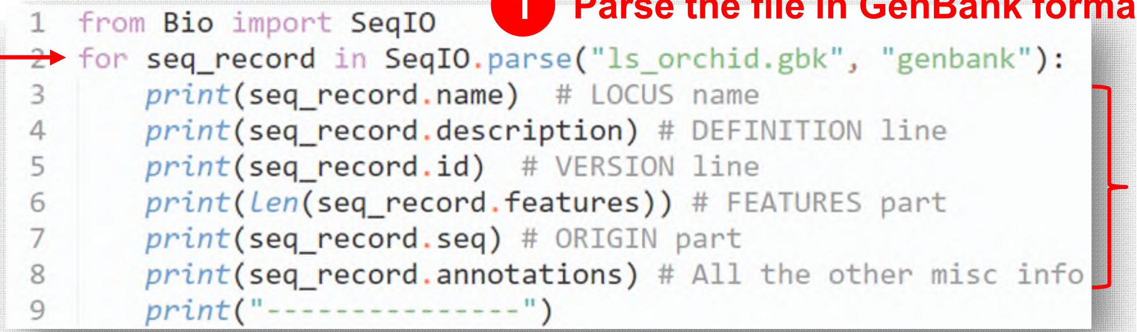


```
# Download Data File: ls_orchid.gbk
import os
if not os.path.isfile("ls_orchid.gbk"):
    os.system("wget https://raw.githubusercontent.com/biopython/biopython/master/Doc/examples/ls_orchid.gbk")
```

Parsing GenBank Files

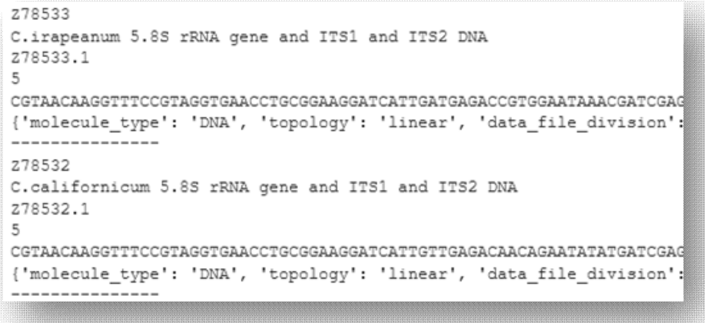
1 Parse the file in GenBank format

2 Get one record a time



```python
from Bio import SeqIO
for seq_record in SeqIO.parse("ls_orchid.gbk", "genbank"):
    print(seq_record.name)  # LOCUS name
    print(seq_record.description) # DEFINITION line
    print(seq_record.id)  # VERSION line
    print(len(seq_record.features)) # FEATURES part
    print(seq_record.seq) # ORIGIN part
    print(seq_record.annotations) # All the other misc info
    print("---------------")
```

3 Print Information of the current record



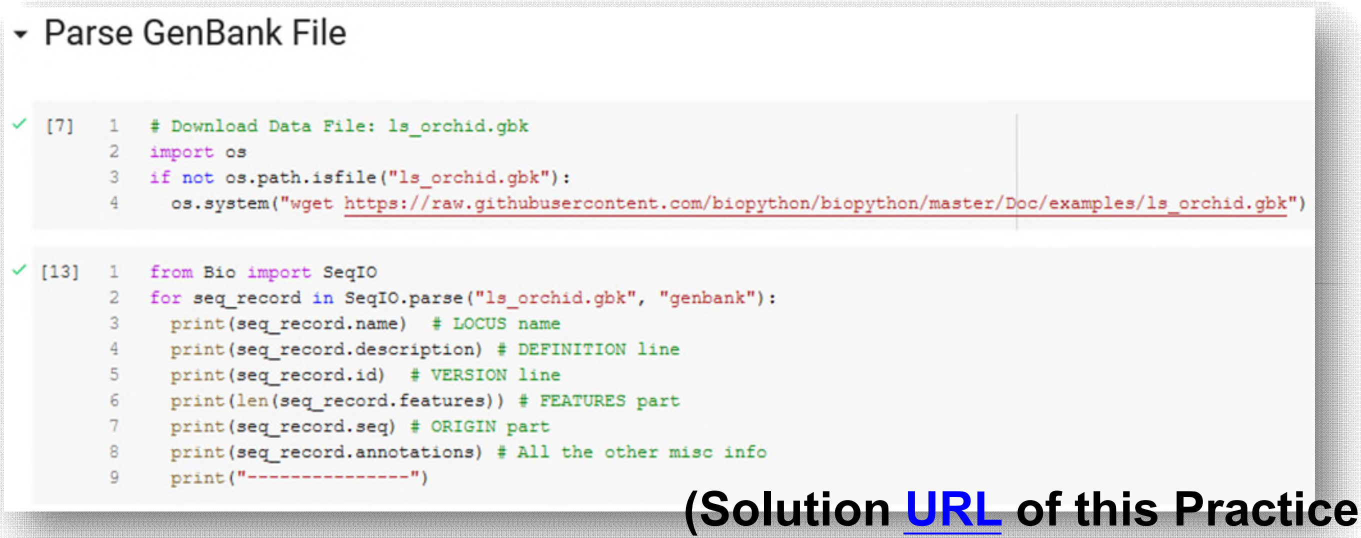
```
Z78533
C.irapeanum 5.8S rRNA gene and ITS1 and ITS2 DNA
Z78533.1
5
CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGATGAGACCGTGGAATAAACGATCGAG
{'molecule_type': 'DNA', 'topology': 'linear', 'data_file_division':
---------------
Z78532
C.californicum 5.8S rRNA gene and ITS1 and ITS2 DNA
Z78532.1
5
CGTAACAAGGTTTCCGTAGGTGAACCTGCGGAAGGATCATTGTTGAGACAACAGAATATATGATCGAG
{'molecule_type': 'DNA', 'topology': 'linear', 'data_file_division':
---------------
```

1st SeqRecord

2nd SeqRecord

Practice

- **Parsing GenBank Files**
  - Write and Run the following codes on a Colab page called “QuickTour.ipynb”:



✓ [7]

```
# Download Data File: ls_orchid.gbk
import os
if not os.path.isfile("ls_orchid.gbk"):
  os.system("wget https://raw.githubusercontent.com/biopython/biopython/master/Doc/examples/ls_orchid.gbk")
```

✓ [13]

```
from Bio import SeqIO
for seq_record in SeqIO.parse("ls_orchid.gbk", "genbank"):
  print(seq_record.name)   # LOCUS name
  print(seq_record.description) # DEFINITION line
  print(seq_record.id)   # VERSION line
  print(len(seq_record.features)) # FEATURES part
  print(seq_record.seq) # ORIGIN part
  print(seq_record.annotations) # All the other misc info
  print("---------------")
```

(Solution URL of this Practice)

Summary

- **Install Biopython**
  - pip install biopython
- **Import Biopython**
  - import Bio
- **Check the Version of Biopython**
  - print(Bio.__version__)
- **Convert Strings to Bio.Seq.Seq**
  - my_seq = Seq("AGTACACTGGT")
- **Convert Bio.Seq.Seq to Strings**
  - bio_seq = str(my_seq)
- **Complement & Reverse Complement**
  - my_seq.complement()
  - my_seq.reverse_complement()
- **Parse FASTA / GenBank**
  - SeqIO.parse("ls_orchid.fasta", "fasta")
  - SeqIO.parse("ls_orchid.gbk", "genbank")