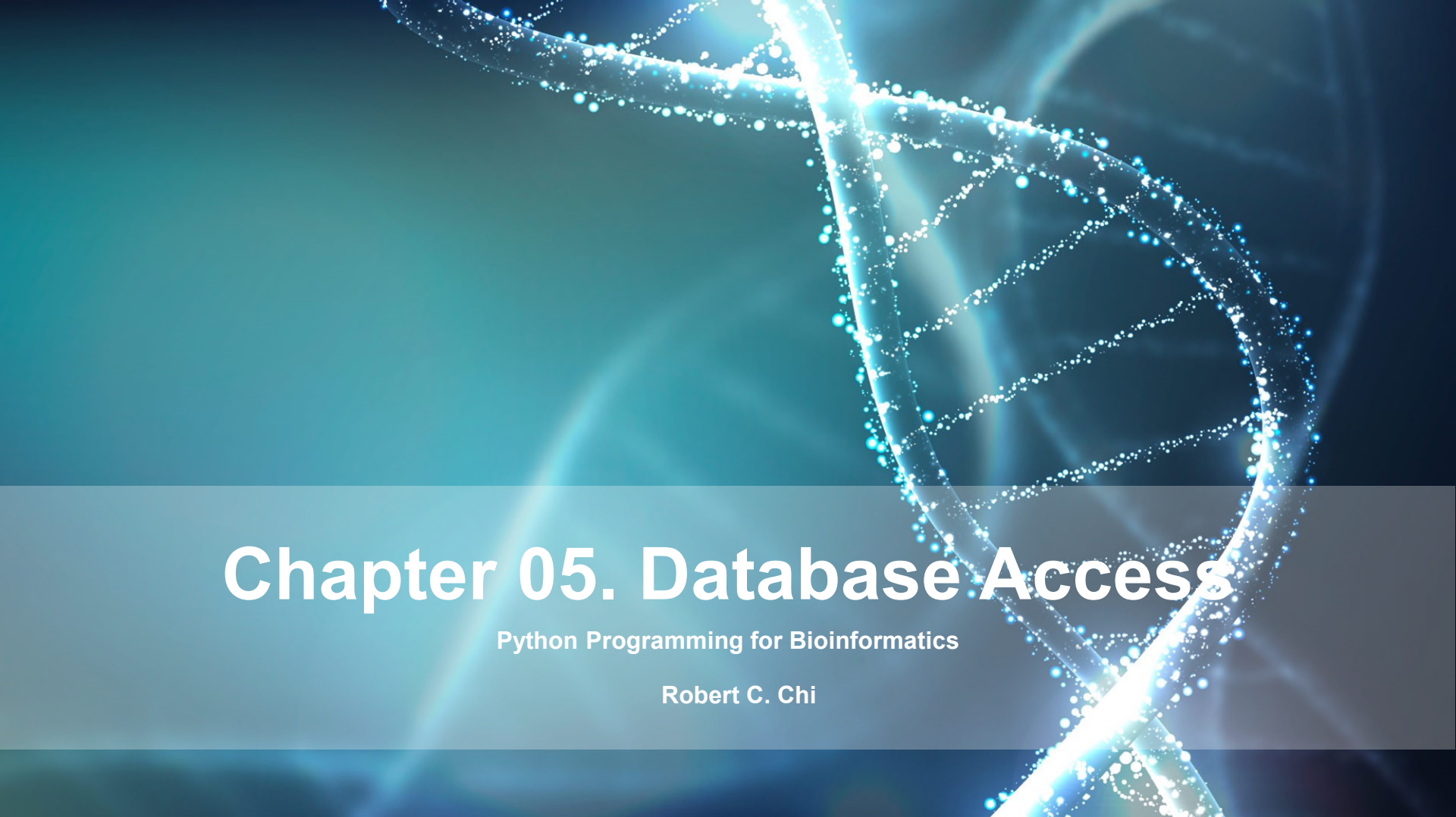


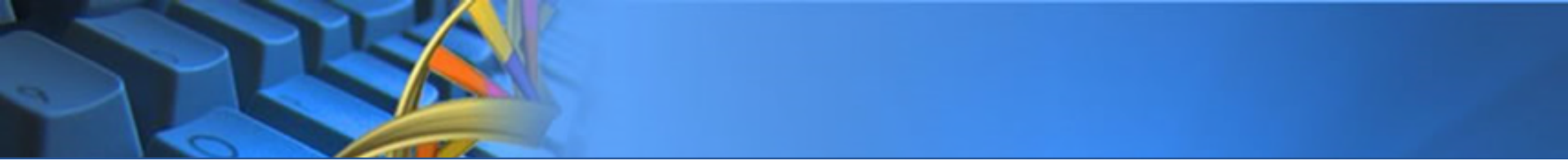
Chapter 05. Database Access

Python Programming for Bioinformatics

Robert C. Chi

Agenda

- Introduction to NCBI's Entrez
- Access Entrez
- Access PubMed & Medline
- Access Swiss-Prot / ExPASy
- Other Databases
- Summary



INTRODUCTION TO NCBI'S ENTREZ

What is NCBI Entrez

- **A Cross-Database Search System of the NCBI databases:**

Nucleotide (GenBank)
Sequence Database

Protein
Protein Sequence Database

UniGene
Transcriptome Database

dbSNP
SNP Database

PubMed & Medline
Biomedical Literatures

Structure
3D Macromolecular Structures

Taxonomy
Organisms in GenBank Taxonomy

OMIM
Online Mendelian Inheritance in Man

Guideline of Accessing Entrez

- Access https://eutils.ncbi.nlm.nih.gov/ instead of the standard NCBI Web address (Biopython already does this).
- Always provide your E-mail before Access.
- Provide the "Tool Name" used to access the database
- Make at most 10 queries per second (Biopython will follow this rule).
- For any 100 consecutive requests, do it in the USA off-peak hours.
- For large downloads, consider using FTP rather than online access.

APIs to Access Entrez

- **EInfo**
  - Obtaining information about the Entrez databases
- **ESearch**
  - Searching the Entrez databases
- **ESummary**
  - Retrieving summaries from primary IDs
- **EFetch**
  - Downloading full records from Entrez
- **ELink**
  - Searching for related items in NCBI Entrez

ACCESS ENTREZ

EInfo

- **Obtain a list of available databases**

```
1  from Bio import Entrez
2
3  # Always provide your e-mail before access
4  Entrez.email = "cnchi1025@gmail.com"
5
6  # Get the list of available databases by Entrez EInfo
7  handle = Entrez.einfo()
8  record = Entrez.read(handle)
9  print(record)
10
11 # Print out the list of databases
12 print("Available Databases from NCBI Entrez")
13 for dbName in record['DbList']:
14   print(dbName)
```

```
{'DbList': ['pubmed', 'protein', 'nuccore', ... 'gtr']}
```

```
Available Databases from NCBI Entrez
pubmed
protein
nuccore
...
gtr
```

Practice

- **EInfo: Get the List of Available Databases**
  - Write and Run the following codes on a Colab page called "NCBIEntrez.ipynb":

```
1  from Bio import Entrez
2
3  # Always provide your e-mail before access
4  Entrez.email = "cnchi1025@gmail.com"
5
6  # Get the list of available databases by Entrez EInfo
7  handle = Entrez.einfo()
8  record = Entrez.read(handle)
9  print(record)
10
11 # Print out the list of databases
12 print("Available Databases from NCBI Entrez")
13 for dbName in record['DbList']:
14   print(dbName)
```

(Solution URL of this Practice)

EInfo

- **Get Details of the Specific Database**

```
1  from Bio import Entrez
2
3  # Always provide your e-mail before access
4  Entrez.email = "cnchi1025@gmail.com"
5
6  # Get the list of available databases by Entrez EInfo
7  handle = Entrez.einfo(db="pubmed")
8  record = Entrez.read(handle)
9  print(record)
10
11 # List all available information of PubMed
12 print(record['DbInfo'].keys())
```

{'DbInfo': {'DbName': 'pubmed', 'MenuName': 'PubMed', 'FieldList': [...]}}

dict_keys(['DbName', 'MenuName', 'Description', 'DbBuild', 'Count', 'LastUpdate', 'FieldList', 'LinkList'])

EInfo

- Get **Details** of the Specific Database

```
1  # Attributes of PubMed
2  print("Database Name:", record['DbInfo']['MenuName'])
3  print("Description:", record['DbInfo']['Description'])
4  print("Total Records:", record['DbInfo']['Count'])
5  print("Last Update:", record['DbInfo']['LastUpdate'])
```

```
Database Name: PubMed
Description: PubMed bibliographic record
Total Records: 33356059
Last Update: 2021/11/24 14:34
```

EInfo

- Get **Details** of the Specific Database

```
1  # Available Fields for Search
2  print("Available Fields for Search -----------")
3  for field in record['DbInfo']['FieldList']:
4    print("{:4s}: {}. {}".format(field['Name'], field['FullName'], field['Description']))
```

```
Available Fields for Search ----------
ALL : All Fields. All terms from all searchable fields
UID : UID. Unique number assigned to publication
TITL: Title. Words in title of publication
AUTH: Author. Author(s) of publication
JOUR: Journal. Journal abbreviation of publication
...
```

How to use these fields:

- "Biopython[TITL]":
  Search "Biopython" in PubMed Titles.
- "Jones[AUTH]":
  Search "Jones" in PubMed Authors.

EInfo

- Get **Details** of the Specific Database

```
1  # Other Related Databases
2  print("Other Related Databases -----------")
3  for db in record['DbInfo']['LinkList']:
4    print("{}: {}. {}".format(db['DbTo'], db['Menu'], db['Description']))
```

```
Other Related Databases ----------
assembly: Assembly. Assembly
bioproject: Project Links. Related Projects
biosample: BioSample Links. BioSample links
...
taxonomy: Taxonomy via GenBank. Related Taxonomy entry computed using other Entrez links
```

Practice

- **EInfo: Get the Detail Information of 'PubMed' Database**
  - Write and Run the following codes on a Colab page called "NCBIEntrez.ipynb":

```
from Bio import Entrez

# Always provide your e-mail before access
Entrez.email = "cnchi1025@gmail.com"

# Get the list of available databases by Entrez EInfo
handle = Entrez.einfo(db="pubmed")
record = Entrez.read(handle)
print(record)

# List all available information of PubMed
print(record['DbInfo'].keys())
```

```
# Attributes of PubMed
print("Database Name:", record['DbInfo']['MenuName'])
print("Description:", record['DbInfo']['Description'])
print("Total Records:", record['DbInfo']['Count'])
print("Last Update:", record['DbInfo']['LastUpdate'])

# Available Fields for Search
print("Available Fields for Search ----------")
for field in record['DbInfo']['FieldList']:
  print("{:4s}: {}. {}".format(field['Name'], field['FullName'], field['Description']))

# Other Related Databases
print("Other Related Databases ----------")
for db in record['DbInfo']['LinkList']:
  print("{}: {}. {}".format(db['DbTo'], db['Menu'], db['Description']))
```

(Solution URL of this Practice)

ESearch

- **Search for GenBank Accession ID of matK gene in Cypripedioideae orchids**

```
1  from Bio import Entrez
2
3  # Always provide your e-mail before access
4  Entrez.email = "cnchi1025@gmail.com"
5
6  # Search for GenBank Accession ID of matK gene in Cypripedioideae orchids
7  handle = Entrez.esearch(db="nucleotide", term="Cypripedioideae[ORGN] AND matK[GENE]", idtype="acc", retmax="1000")
8  record = Entrez.read(handle)
9
10 # Print the useful information
11 print(record)
12 print(record['Count'])
13 print(record['IdList'])
```

GenBank

Search Conditions

Accession ID

Return Max Records

```
{'Count': '585', 'RetMax': '585', 'RetStart': '0', 'IdList': ['NC_058834.1', ]...}
```

```
585
```

```
['NC_058834.1', 'NC_058833.1', 'NC_058832.1', ... , 'AJ581441.1']
```

Practice

- **ESearch: Search GenBank Accession IDs by Keywords**
  - Write and Run the following codes on a Colab page called “NCBIEntrez.ipynb”:

```
from Bio import Entrez

# Always provide your e-mail before access
Entrez.email = "cnchi1025@gmail.com"

# Search for GenBank Accession ID of matK gene in Cypripedioideae orchids
handle = Entrez.esearch(db="nucleotide", term="Cypripedioideae[ORGN] AND matK[GENE]", idtype="acc", retmax="1000")
record = Entrez.read(handle)

# Print the useful information
print(record)
print(record['Count'])
print(record['IdList'])
```

(Solution URL of this Practice)

ESummary

- **Get the Summary by the ESearch Result:**

```python
from Bio import Entrez

# Always provide your e-mail before access
Entrez.email = "cnchi1025@gmail.com"

# Compile the ID list for Query
id_list = ",".join(record['IdList'])

# Fetch the Summary with the Given IDs
handle = Entrez.esummary(db="nucleotide", id=id_list)
records = Entrez.read(handle)

# Print the desired information
for item in records:
    print("GenBank ID (GI):", item['Id'])
    print("Accession ID:", item['AccessionVersion'])
    print("Title:", item['Title'])
    print("FASTA Description:", item['Extra'])
    print("Create Date:", item['CreateDate'])
    print("Update Date:", item['UpdateDate'])
    print()
```

['NC_058834.1', 'NC_058833.1', 'NC_058832.1', ...'AJ581441.1']

"NC_058834.1,NC_058833.1,NC_058832.1,...,AJ581441.1"

GenBank ID (GI): 2127881387
Accession ID: NC_058834.1
Title: Paphiopedilum henryanum chloroplast, complete genome
FASTA Description: gi|2127881387|ref|NC_058834.1|[2127881387]
Create Date: 2021/11/03
Update Date: 2021/11/04

Practice

- **ESummary: Get the Summary by the Given IDs**
  - Write and Run the following codes on a Colab page called “NCBIEntrez.ipynb”:

```
from Bio import Entrez

# Always provide your e-mail before access
Entrez.email = "cnchi1025@gmail.com"

# Compile the ID list for Query
id_list = ",".join(record['IdList'])

# Fetch the Summary with the Given IDs
handle = Entrez.esummary(db="nucleotide", id=id_list)
records = Entrez.read(handle)

# Print the desired information
for item in records:
  print("GenBank ID (GI):", item['Id'])
  print("Accession ID:", item['AccessionVersion'])
  print("Title:", item['Title'])
  print("FASTA Description:", item['Extra'])
  print("Create Date:", item['CreateDate'])
  print("Update Date:", item['UpdateDate'])
  print()
```

(Solution URL of this Practice)

EFetch

- Get the GenBank File by the Given ID

```
1  from Bio import Entrez
2
3  # Always provide your e-mail before access
4  Entrez.email = "cnchi1025@gmail.com"
5
6  # Get only the first ID to fetch
7  theID = record['IdList'][0]
8
9  # Get the GenBank File by the Given ID
10 handle = Entrez.efetch(db="nucleotide", id=theID, rettype="gb", retmode="text")
11
12 # Parse the GenBank Files
13 from Bio import SeqIO
14 seq_records = list(SeqIO.parse(handle, "genbank"))
```

The Trick for Converting Parsed Results as a List

EFetch

- Get the GenBank File by the Given ID

```
16  # Show the parsed result
17  rec = seq_records[0]
18
19  # Show the attributes of SeqRecord
20  print("Locus Name:", rec.name)
21  print("Sequence Type:", rec.annotations['molecule_type'])
22  print("Sequence Topology:", rec.annotations['topology'])
23  print("Published Date:", rec.annotations['date'])
24  print("Sequence Definition:", rec.description)
25  print("Sequence Version:", rec.annotations['sequence_version'])
26  print("Organism:", rec.annotations['organism'])
27  print("Number of Features:", len(rec.features))
28  print("Sequence:", rec.seq)
```

```
Locus Name: NC_058834
Sequence Type: DNA
Sequence Topology: circular
Published Date: 04-NOV-2021
Sequence Definition: Paphiopedilum henryanum chloroplast, complete genome
Sequence Version: 1
Organism: Paphiopedilum henryanum
Number of Features: 255
Sequence: CATTCATTGG...CAAAAG
```

Practice

- **EFetch: Fetch GenBank File by the Given ID**
  - Write and Run the following codes on a Colab page called “NCBIEntrez.ipynb”:

```python
from Bio import Entrez

# Always provide your e-mail before access
Entrez.email = "cnchi1025@gmail.com"

# Get only the first ID to fetch
theID = record['IdList'][0]

# Get the GenBank File by the Given ID
handle = Entrez.efetch(db="nucleotide", id=theID, rettype="gb", retmode="text")

# Parse the GenBank Files
from Bio import SeqIO
seq_records = list(SeqIO.parse(handle, "genbank"))
```

```python
# Show the parsed result
rec = seq_records[0]

# Show the attributes of SeqRecord
print("Locus Name:", rec.name)
print("Sequence Type:", rec.annotations['molecule_type'])
print("Sequence Topology:", rec.annotations['topology'])
print("Published Date:", rec.annotations['date'])
print("Sequence Definition:", rec.description)
print("Sequence Version:", rec.annotations['sequence_version'])
print("Organism:", rec.annotations['organism'])
print("Number of Features:", len(rec.features))
print("Sequence:", rec.seq)
```

(Solution URL of this Practice)

ELink

- **Find related items by the given ID**

```python
from Bio import Entrez

# Always provide your e-mail before access
Entrez.email = "cnchi1025@gmail.com"

# Get only the first ID to fetch
theID = record['IdList'][0]

# Get the related items to the Given ID
handle = Entrez.elink(dbfrom="nucleotide", id=theID)
record = Entrez.read(handle)

# Iterate each related IDs
rec = record[0]
for link in rec['LinkSetDb']:
  for id in link['Link']:
    print(id)
```

```
{'Id': '34610149'}
{'Id': '34395889'}
{'Id': '34151006'}
...
{'Id': '1282235'}
```

Practice

- **ELink: Get related items in Entrez by the Given ID**
  - Write and Run the following codes on a Colab page called “NCBIEntrez.ipynb”:

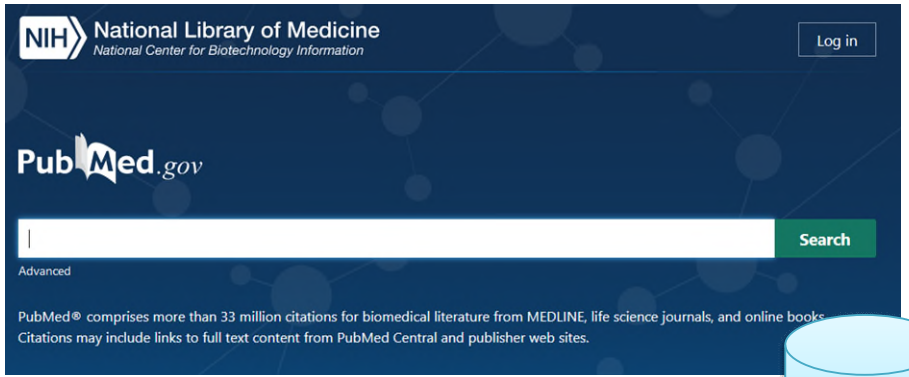
```
from Bio import Entrez

# Always provide your e-mail before access
Entrez.email = "cnchi1025@gmail.com"

# Get only the first ID to fetch
theID = record['IdList'][0]

# Get the related items to the Given ID
handle = Entrez.elink(dbfrom="nucleotide", id=theID)
record = Entrez.read(handle)

# Iterate each related IDs
rec = record[0]
for link in rec['LinkSetDb']:
  for id in link['Link']:
    print(id)
```

(Solution URL of this Practice)

ACCESS PUBMED & MEDLINE

Introduction

PubMed (Platform)



33+ million Citations
for Biomedical Literature

Medline (Database)

Medline File Format

```
PMID - 3806354
JT - Journal of Personality and Social Psychology
TA - J Pers Soc Psychol
DP - 1986
VI - 51
IP - 6
IS - 0022-3514
TI - The moderator-mediator variable distinction
     in social psychological research: conceptual,
     strategic, and statistical considerations.
PG - 1173-1182
AU - Baron,RM
AU - Kenny,DA
PT - JOURNAL ARTICLE
DO - DOI:10.1037/0022-3514.51.6.1173
SO - J Pers Soc Psychol. 1986;51(6):1173-1182.
```

- **PMID**: PubMed ID
- **JT**: Journal Title
- **TA**: Journal Title Abbreviation
- **DP**: Date of Publication
- …

The Full List of Medline Fields:

- https://bit.ly/3cP4EN7

Access PubMed/Medline

- Get PubMed IDs related to “orchid”

```
1  from Bio import Entrez
2
3  # Always provide your e-mail before access
4  Entrez.email = "cnchi1025@gmail.com"
5
6  # Get PubMed IDs related to "orchid"
7  handle = Entrez.esearch(db="pubmed", term="orchid", retmax=463)
8  record = Entrez.read(handle)
9  handle.close()
10
11 idlist = record["IdList"]
12 print("Total Records:", len(idlist))
13 print(idlist)
```

```
Total Records: 463
['34816066', '34816060', '34816059', ... '32349139']
```

Access PubMed/Medline

- Get PubMed Records by the Given IDs

```
from Bio import Medline

handle = Entrez.efetch(db="pubmed", id=idlist, rettype="medline", retmode="text")
records = list(Medline.parse(handle))

print("Total Records:", len(records))
for rec in records:
  print("Title:", rec.get("TI", "N/A"))
  print("Authors:", rec.get("AU", "N/A"))
  print("Sources:", rec.get("SO", "N/A"))
  print()
```

```
Total Records: 463
Title: Morphological and genomic evidence for a new species of Corallorhiza (Orchidaceae: Epidendroideae) from SW China.
Authors: ['Yang JX', 'Peng S', 'Wang JJ', 'Ding SX', 'Wang Y', 'Tian J', 'Yang H', 'Hu GW', 'Wang QF']
Sources: Plant Divers. 2021 Jan 23;43(5):409-419. doi: 10.1016/j.pld.2021.01.002. eCollection 2021 Oct.
```

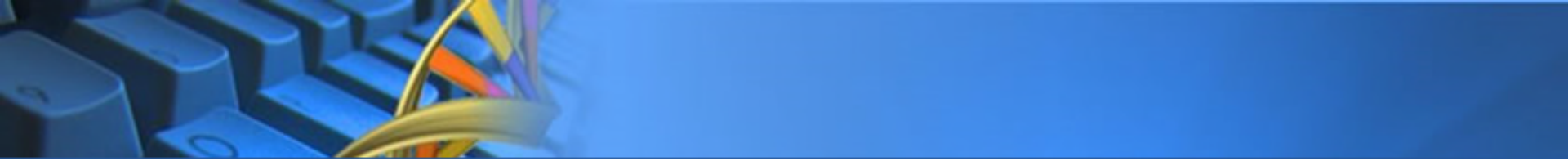
Practice

- **PubMed: Get related reference by the Given PubMed ID**
  - Write and Run the following codes on a Colab page called "NCBIEntrez.ipynb":

```python
from Bio import Entrez

# Always provide your e-mail before access
Entrez.email = "cnchi1025@gmail.com"

# Get PubMed IDs related to "orchid"
handle = Entrez.esearch(db="pubmed", term="orchid", retmax=463)
record = Entrez.read(handle)
handle.close()

idlist = record["IdList"]
print("Total Records:", len(idlist))
print(idlist)
```

```python
from Bio import Medline

handle = Entrez.efetch(db="pubmed", id=idlist, rettype="medline", retmode="text")
records = list(Medline.parse(handle))

print("Total Records:", len(records))
for rec in records:
  print("Title:", rec.get("TI", "N/A"))
  print("Authors:", rec.get("AU", "N/A"))
  print("Sources:", rec.get("SO", "N/A"))
  print()
```

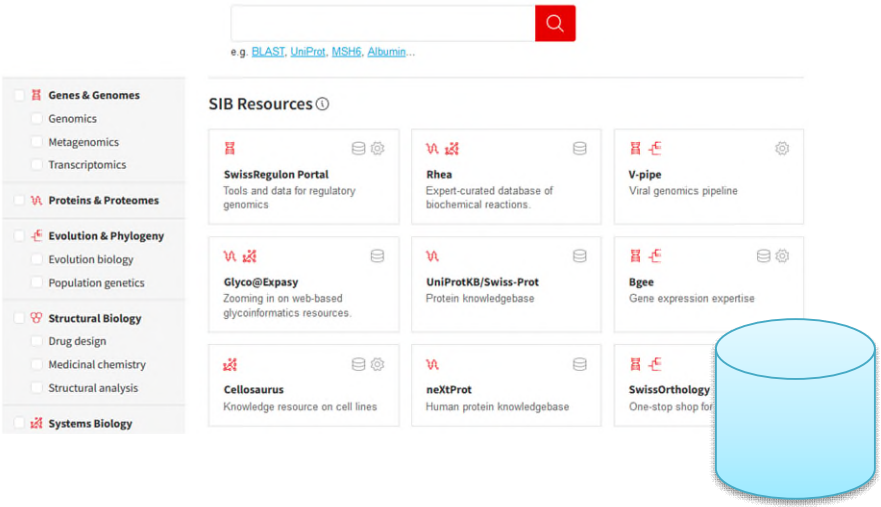
(Solution URL of this Practice)



ACCESS SWISS-PROT / EXPASY

Introduction

ExPASy (Portal)
by Swiss Institute of Bioinformatics (SIB)



Manually Annotated & Reviewed
Protein Knowledge Database

Swiss-Prot (Database)

Swiss-Prot File Format

```
ID   GRAA_HUMAN              Reviewed;         262 AA.
AC   P12544; A4PHN1; Q6IB36;
DT   01-OCT-1989, integrated into UniProtKB/Swiss-Prot.
DT   11-JAN-2011, sequence version 2.
DT   11-DEC-2019, entry version 199.
DE   RecName: Full=Granzyme A;
DE            EC=3.4.21.78;
DE   AltName: Full=CTL tryptase;
DE   AltName: Full=Cytotoxic T-lymphocyte proteinase 1;
DE   AltName: Full=Fragmentin-1;
DE   AltName: Full=Granzyme-1;
DE   AltName: Full=Hanukkah factor;
DE            Short=H factor;
DE            Short=HF;
DE   Flags: Precursor;
GN   Name=GZMA; Synonyms=CTLA3, HFSP;
OS   Homo sapiens (Human).
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia;
OC   Eutheria; Euarchontoglires; Primates; Haplorrhini; Catarrhini; Hominidae;
```

```
ID      - Identification.
AC      - Accession number(s).
DT      - Date.
DE      - Description.
GN      - Gene name(s).
OS      - Organism species.
OG      - Organelle.
OC      - Organism classification.
RN      - Reference number.
RP      - Reference position.
RC      - Reference comments.
RX      - Reference cross-references.
RA      - Reference authors.
RL      - Reference location.
CC      - Comments or notes.
DR      - Database cross-references.
KW      - Keywords.
FT      - Feature table data.
SQ      - Sequence header.
        - (blanks) sequence data.
//      - Termination line.
```

Access Swiss-Prot

```python
from Bio import ExPASy
from Bio import SeqIO

# Get Swiss-Prot Record by ID=O23729
handle = ExPASy.get_sprot_raw("O23729")
prot_records = list(SeqIO.parse(handle, "swiss"))
handle.close()

# Show the desired information
prot = prot_records[0]

print("Swiss-Prot ID:", prot.id)
print("Name:", prot.name)
print("Description:", prot.description)
print("Sequence:", prot.seq)
print("Database Cross Reference:")
for ref in prot.dbxrefs:
  print(ref)
```

```
Swiss-Prot ID: O23729
Name: CHS3_BROFI
Description:
                RecName: Full=Chalcone synthase 3;
                EC=2.3.1.74;
                AltName: Full=Naringenin-chalcone synthase 3;
Sequence: MAPAMEEIR...SVPIAGAE
Database Cross Reference:
EMBL:AF007097
SMR:O23729
...
SUPFAM:SSF53901
PROSITE:PS00441
```

Practice

- **Swiss-Prot: Get Swiss-Prot Information by the Given ID**
  - Write and Run the following codes on a Colab page called “NCBIEntrez.ipynb”:

```
from Bio import ExPASy
from Bio import SeqIO

# Get Swiss-Prot Record by ID=O23729
handle = ExPASy.get_sprot_raw("O23729")
prot_records = list(SeqIO.parse(handle, "swiss"))
handle.close()

# Show the desired information
prot = prot_records[0]

print("Swiss-Prot ID:", prot.id)
print("Name:", prot.name)
print("Description:", prot.description)
print("Sequence:", prot.seq)
print("Database Cross Reference:")
for ref in prot.dbxrefs:
  print(ref)
```

(Solution URL of this Practice)

Access Other Databases

- **Gene Expression Omnibus (GEO)**
  - https://bit.ly/3nS04nF
- **UniGene**
  - https://bit.ly/3DUFWH8
- **Swiss-Prot (in Details)**
  - https://bit.ly/3p41hrk
- **PDB (Protein Database)**
  - https://bit.ly/30VwuVR
- **KEGG**
  - https://bit.ly/3l9f7HS

Summary

- **Access NCBI Entrez Databases**
  - Bio.Entrez.einfo: Get Info of Available Databases
  - Bio.Entrez.esearch: Get IDs by Keywords
  - Bio.Entrez.esummary: Get Summary by IDs
  - Bio.Entrez.efetch: Get Contents by IDs
  - Bio.Entrez.elink: Get Related Items by IDs
- **Access Swiss-Prot / ExPASy**
  - Bio.ExPASy.get_sprot_raw()
- **Access Other Databases**
  - Gene Expression Omnibus (GEO)
  - UniGene
  - PDB (Protein Database)
  - KEGG