



## EXAMEN PARCIAL PYTHON

### GBI6-2021II: BIOINFORMÁTICA

Apellidos, Nombres <--- CAMBIE POR LOS QUE CORRESPONDA A SUS DATOS

03-08-2022

Guillermo Kely  
Color de texto

### REQUERIMIENTOS PARA EL EXAMEN

Utilice de preferencia Jupyter de Anaconda, dado que tienen que hacer un control de cambios en cada pregunta.

Para este examen se requiere dos documentos:

1. Archivo `miningscience.py` donde tendrá dos funciones:
2. Archivo `2022I_GBI6_ExamenPython` donde se llamará las funciones y se obtendrá resultados.

### Ejercicio 0 [0.5 puntos]

Realice cambios al cuaderno de jupyter:

- Agregue el logo de la Universidad
- Coloque sus datos personales
- Escriba una **tabla** con las características de su computador

### Ejercicio 1 [2 puntos]

Cree el archivo `miningscience.py` con las siguientes dos funciones:

- i. `download_pubmed` : para descargar la data de PubMed utilizando el **ENTREZ** de Biopython. El parámetro de entrada para la función es el `keyword`.
- ii. `science_plots` : la función debe
  - utilizar como argumento de entrada la data descargada por `download_pubmed`
  - ordenar los conteos de autores por país en orden ascendente y
  - seleccionar los cinco más abundantes. Con esta selección debe graficar un `pie_plot`. Como guía para el conteo por países puede usar el ejemplo de `MapOfScience` ([https://github.com/CSB-book/CSB/blob/master/regex/solutions/MapOfScience\\_solution.ipynb](https://github.com/CSB-book/CSB/blob/master/regex/solutions/MapOfScience_solution.ipynb)).
- iii Cree un `docstring` para cada función.

8/5/22, 1:23 PM

Luego de crear las funciones, cargue el módulo `miningscience` como `msc` e imprima docstring de cada función.

In [1]:

# Escriba aquí su código para el ejercicio 1

```
Import miningscience_gol
help (miningscience_gol.download_pubmed)
help (miningscience_gol.science_prot)
```

## Ejercicio 2 [2 puntos]

Utilice dos veces la función `download_pubmed` para:

- Descargar la data, utilizando los keyword de su preferencia.
- Guardar el archivo descargado en la carpeta `data`.

Para cada corrida, imprima lo siguiente:

'El número artículos para KEYWORD es: XX' # Que se cargue con inserción de texto o valor que correspondea KEYWORD y XX

In [2]:

# Escriba aquí su código para el ejercicio 2

```
Orquideas = miningscience_gol.download_pubmed("orchids")
print ("el numero de articulo de orchids es " + str(len(Orquideas)))

Vacas = miningscience_gol.download_pubmed("cows")
print ("el numero de articulo de cows es " + str(len(Vacas)))
```

## Ejercicio 3 [1.5 puntos]

Utilice dos veces la función `science_plots` para:

- Visualizar un `pie_plot` para cada data descargada en el ejercicio 2.
- Guardar los `pie_plot` en la carpeta `img`

3 PM

[4]:

#Escriba aquí su código para el ejercicio 3

```
mimisciencgol.science_pbls(Carquideas)
```

```
mimisciencgol.science_pbls(Cvacas)
```

### Ejercicio 4 [1 punto]

Interprete los resultados de las figuras del ejercicio 3

Escriba la respuesta del ejercicio 5.

Los 5 países con más artículos son: Reino Unido, Suiza, Estados Unidos, Alemania y Holanda. Los países con más autores que hablan de vacas son: Suiza, Estados Unidos, Reino Unido, Holanda y Alemania.

### Ejercicio 5 [2 puntos]

Para algún gen de las enzimas que intervienen en la ruta metabólica de la gluconeogénesis (Lista de genes por tipología (<https://www.genome.jp/pathway/map00010+C00068>)), realice lo siguiente:

1. Una búsqueda en la página del NCBI nucleotide (<https://www.ncbi.nlm.nih.gov/nucleotide/>).
2. Descargue el Accession List de su búsqueda y guarde en la carpeta `data`.
3. Cargue el Accession List en este notebook y haga una descarga de las secuencias de los quince primeros IDs de la accesión.
4. Arme un árbol filogenético para los resultados del paso 3.
5. Guarde su árbol filogenético en la carpeta `img`.
6. Interprete el árbol.

In [3]:

```
# Escriba aquí su código para el ejercicio 6
from Bio import Entrez
from Bio import phylo
from Bio.phylo.treeConstruction import Distance Calculator
from Bio.phylo.treeConstruction import Distance Tree Constructor
from Bio.phylo.treeConstruction import AlignIO
from Bio import SeqIO
from Bio.Align.Applications import ClustalWCommandline
import matplotlib.pyplot as plt
import os with open ("data/sequence.seq") as f:
    text = f.read()
    text = text.split("\n")
    text = "\n".join(text[0:15])
```

```
handle = Entrez.efetch(db="nucleotide", rettype="gb", retmode="text"
    retmode="text", id=text)
print(handle.read())
record = SeqIO.parse("data/sequence.gb", "genbank")
cont = SeqIO.write(records, "data/sequence.fasta", "fasta")
clustalw_exe = r"C:\Users\Downloads\clustalw2\clustalw.exe"
clustalw_cli = ClustalWCommandline(clustalw_exe, infile="data/
    sequence.fasta")
assert os.path.isfile(clustalw_exe), "clustalw
    executable is missing or not found"
stdout, stderr = clustalw_cli()
print(clustalw_cli)
ClustalAlign = AlignIO.read("data/sequence.aln", "clustal")
with open("data/sequence.aln", "r") as aln:
    alignment = AlignIO.read(aln, "clustal")
calculator = DistanceCalculator("identity")
distance_matrix = calculator.get_distance(alignment)
constructor = DistanceTreeConstructor(calculator)
rag2_tree = constructor.build_tree(alignment)
rag2_tree.rooted = True
plt.savefig("img(arbol).pdf")
```



```

import matplotlib.pyplot as plt
import os
with open("data/sequence.seq") as f:
    text = f.read()
text = text.split("\n")
text = "\n".join(text[0:1000])

```

Escriba aquí la interpretación del árbol

## Ejercicio 6 [1 punto]

1. Cree en GitHub un repositorio de nombre GBI6\_ExamenPython .
2. Cree un archivo Readme.md que debe tener lo siguiente:
  - Datos personales
  - Características del computador
  - Versión de Python/Anaconda y de cada uno de los módulos/paquetes y utilizados
  - Explicación de la data utilizada
  - Un diagrama de procesos del módulo miningscience
3. Asegurarse que su repositorio tiene las carpetas data e img con los archivos que ha ido guardando en las preguntas anteriores.
4. Realice al menos 1 control de la versión (commits) por cada ejercicio (del 1 al 5), con un mensaje que inicie como:

Carlitos Alimaña ha realizado el ejercicio 1

Carlitos Alimaña ha realizado el ejercicio 2

...

In [ ]:

Nombre [Apellido, Nombre]:

Construya las funciones del módulo miningscience.py

def download\_pubmed(keyword):

""" Esta Funcion busca palabras clave en la base de datos PubMed usando el paquete Entrez de BioPython. para usarlo, se debe especificar lo siguiente download\_pubmed(keyword). La funcion devuelve un diccionario con los ID de las publicaciones en un conteo de cuantas publicaciones y demas informacion en un diccionario """

pubmed\_salida = open('data/' + keyword, 'w')

Entrez.email = 'A.N.Other@example.com'

result = Entrez.esearch(db="pubmed",

Term=keyword

retmax=1000, usehistory="g")

data = Entrez.read(result)

id\_list = data["IdList"]

webenv = data["webEnv"]

query\_key = data["query-key"]

handle = Entrez.efetch(db="pubmed",

rettype="medline",

retformat="text",

retmax=1000,

webenv=webenv

query\_key=query\_key)

texto = handle.read()

pubmed\_salida.write(texto)

pubmed\_salida.close()

texto = texto.split("1n PMID")

years = []

num\_authors = []

titles = []

nk")

=asta"  
ere"  
e:le

For articulo in TextoL["J"]:

year = re.findall(r"DPLS\s-\s(\d\d\d\d\d\d)", articulo)[0]

num-autor = len(re.findall(r"AU-", articulo))

pais = re.findall(r"PL\s-\s(\w+)", articulo)[0]

IF pais == "England":

pais = "United Kingdom"

years.append(year)

paises.append(pais)

num-autors.append(num-autor)

pmtable = pd.DataFrame({"ID": id\_list,

"years": years,

"Autors": num-autors,

"pais": paises})

return pmtable

~~def since\_plots(download\_probed):~~

~~"Se utiliza el download\_probed y ordena los contenidos co~~

handle = Entrez.efetch(Cdb="nucleotide", rettype="text",  
retmode="text", id=text)  
p=

GB16 - BIOINFORMÁTICA [2022I]  
Examen Final [Python]

**Ikiam**  
Universidad Regional Amazónica

Nombre [Apellido, Nombre]:

```
def science_plots(download PubMed):
```

se utiliza download PubMed y ordena los datos por  
autores por país en orden ascendente y selecciona  
los 8 más abundantes

```
download PubMed2 = download PubMed.groupby("país").sum()
download PubMed2 = download PubMed2.sort_values("Autores",
ascending=False).head(8)
plt.plot(download PubMed2["Autores"], labels =
download PubMed2.index)
plt.savefig("iny publicacion 2020.pdf")
```