EXAMEN PARCIAL PYTHON

GBI6-2021II: BIOINFORMÁTICA

Johanna Tanguila
Grupo 1

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

03-08-2022

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

Data =
img. =

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`.

Data ← txt

ii. `science_plots` : la función debe

Pon a dataframe. Pais.

- utilizar como argumento de entrada la data descargada por `download_pubmed`
- ordenar los conteos de autores por país en orden ascedente 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).

poplot

iii *Cree un docstring para cada función.*

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

In [1]:

```
# Escriba aquí su código para el ejercicio 1
-import miningscience_g01 as msc
-help (msc.download_pubmed)
-help (msc.Gience_plots.
```

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
lista="Organismo"
palabra = msc.download_pubmed(lista)
print ('El número de artículos para', (lista), 'es:', len (palabra))
with open ("Data/organismo.txt", "w") as txt:
    txt.writelines (palabra)
lista2 = "Asthma"
palabra = msc.download_pubmed(lista2)
print ("El número artículos para', (lista2), 'es:', len (palabra))
with open ("Data/Asthma.txt", "w") as txt:
  txt.writelines (palabra)
```

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`

[4]:

Escriba aquí su código para el ejercicio 3

```
print("\n\n\tDataframes de los datos de los paises y cantidades\n\t")
df_pa_T = msc.science_plots(lista)
df_pa_T1 = df_pa_T.sort_values(by=['número de autores'], ascending=False)
df_pa_T2 = df_pa_T1.iloc[0:5]
df_pa_T2
import matplotlib.pyplot as plt
labels = 'China', 'USA', 'India', 'South Africa', 'Brazil'
sizes= [1987, 1706, 1246, 579, 521]
explode = (0, 0, 0, 0, 0)
fig1, ax1 = plt.subplots()
ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%',
        shadow=True, startangle=10)
ax1.axis('equal')
plt.savefig("img/autores Organismo.jpg")
```

Ejercicio 4 [1 punto]

Interprete los resultados de las figuras del **ejercicio 3**

Escriba la respuesta del ejercicio 5 .

Explicación ejercicio 3: Data Organismo

Se observa que el pais con mayor número de autores lo lidera China con un 32,9%, seguido de Estados Unidos con un 28,2% y finalmente el país con menor número de autores, en especial Brasil por sus porcentajes. del 8,6%

Ejercicio 5 [2 puntos]

Para algún **gen de las enzimas que intervienen en la ruta metabolica de la gluconeogenesis** (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 arbol filogénetico en la carpeta `img`
6. Interprete el árbol del paso 4.

is filogenia.

In [3]:

```
# Escriba aquí su código para el ejercicio 6
from Bio import phylo
from Bio import SeqIO
from Bio import AlignIO
from Bio.Phylo.TreeConstruction import DistanceCalculator
from Bio. import Entrez
from Bio.Seq import Seq
import csv
import re
```

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 [ ]:

GBI6 – BIOINFORMÁTICA [2022I]
Examen Final [Python]

Ikiam
Universidad Regional Amazónica

Nombre [Apellido, Nombre]:

Construya las funciones del módulo miningscience.py

```
def download_pubmed( keyword ):
    """
```

La frase de busqueda es la función input, donde el resultado de la lista id de la busqueda en pubmed """

```
from Bio import Entrez
from Bio import SeqIO
from Bio import GenBank

Entrez.email = """gA.N.Other@example.com'
handle = Entrez.esearch (db='pubmed',
                         sort='relevance',
                         retmax='200',
                         retmode='xml',
                         term=keyword)
results = Entrez.read(handle)
id_list = results["IdList"]
ids = ','.join(id_list)
Entrez.mail = 'gA.N.other@example.com'
handle = Entrez.efetch (db='pubmed',
                        retmode='xml',
                        id=ids)

lista_id = ids.split(",")
return (lista_id)
```

GBI6 – BIOINFORMÁTICA [2022I]
Examen Final [Python]

Ikiam
Universidad Regional Amazónica

Nombre [Apellido, Nombre]:

```
def science_plots(    Archivo
    """                          ):
```

descripción de la funcion

```
AD = []
pa1 = [ ]
pa2 = [ ]
pa3 = [ ]
pa4 = [ ]
pa5 = [ ]
pa6 = [ ]
pa7 = [ ]
pa8 = [ ]
pa9 = [ ]
pa10 = [ ]
pa_T = []
for line in lista.splitlines():
    if line.startswith("AD_"):
        AD.append(line[:])
for line in lista.splitlines():
  if line.startwith("AD-"):
        AD= line[:]
        p1 = re.findall(r'\,\s(\w{2,16}\.', AD)
        P2= re.findall(r'\,\s(\w{2,16}[^0-9\,]\s\w{2,16}[^0-9])\.', AD)
        P3= re.findall(r'\,\s(\w{3,16}[^0-9\,]\s\w{2,3}[^0-9\,]\s\w{3,16}[^0-9\,])\.', AD)
        P4=
        P5=
        P6=
        P7=
        P8=
        P9=
        P10=
        pa_T = pa1 + pa2 + pa3 + pa4 + pa5 + pa6 + pa7 + pa8 + pa9 + p10

pa_T = list(itertools.chain.from_iterable(pa_T))
len(pa_T)
Unique_pa_T.sort()
len(unique-pa-T)
```

```
import csv
coordenadas = {}
with open ('data/ubipais.txt') as f:
csvr = csv.DictReader(f)
for row in csvr:
        coordenadas [row['Nom']] = [row['Latitude'], row['longitude']]
Country = []
Longitude = []
Latitude = []
almacen = []
for z in unique_pa_T:
        if z in coordenadas.keys():
        country.append(z)
        latitude.append(float(coordenadas[z][0]))
        longitude.append(float(coordenadas[z][1]))
        almacen.append(pa_T.count(z))
df_pa_T = pd.Dataframe()
df_pa_T["pais"] = country
df_pa_T["Número de autores"] = almacen
return(df_pa_T)
```