



EXAMEN PARCIAL PYTHON

GBI6-2021II: BIOINFORMÁTICA

Apellidos, Nombres Flores Guerrero Belsabeth Juleth

03-08-2022

Sistema Operativo	Procesador	Ram
Windows 10	Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.21 GHz	8.00 GB

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. `map_science` : para su data replique 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), donde los puntos resaltados son al menos 5 países
- iii Cree un *docstring* para cada función.

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

In [10]:

```

# Escribadeb download_pubmed (keyword):
"""
Muestras de IDs de la busqueda en pubmed
"""
from Bio import Entrez
from Bio import SeqIO
from Bio import GenBank
Entrez.email = 'juleth.flores@est.ikiam.edu.ec'
handle = Entrez.research(db='pubmed',
                        sort='relevance',
                        retmax='200',
                        retmode='xml',
                        term=keyword)
results = Entrez.read(handle)
id_list = results["IdList"]
ids = ','.join(id_list)
Entrez.email = 'juleth.flores@est.ikiam.edu.ec'
handle = Entrez.efetch(db='pubmed',
                        retmode='xml',
                        id=ids)
lista_id = ids.split(",")
return (lista_id)

import csv
import re
import pandas as pd
from collections import Counter

def map_science(tipo):
    """ Docstring map_science """
    """ Esta funcion me permite crear un MapOfScience """
    #if tipo == "AD":
    with open() as f:
        my_text = f.read(tipo)
    my_text = re.sub(r'\n\s{6}', ' ', my_text)
    zipcodes = re.findall(r'[A-Z]{2}\s(\d{5}), USA', my_text)
    unique_zipcodes = list(set(zipcodes))
    unique_zipcodes.sort()
    unique_zipcodes[:10]
    zip_coordinates = {}
    with open('CSB-master/regex/data/MapOfScience/zipcodes_coordinates.txt') as f:
        csvr = csv.DictReader(f)
        for row in csvr:
            zip_coordinates[row['ZIP']] = [float(row['LAT']),
                                           float(row['LNG'])]

    zip_code = []
    zip_long = []
    zip_lat = []
    zip_count = []
    for z in unique_zipcodes:
        # if we can find the coordinates
        if z in zip_coordinates.keys():
            zip_code.append(z)
            zip_lat.append(zip_coordinates[z][0])
            zip_long.append(zip_coordinates[z][1])
            zip_count.append(zipcodes.count(z))
    import matplotlib.pyplot as plt
    #matplotlib inline

```

```

plt.scatter(zip_long, zip_lat, s = zip_count, c= zip_count)
plt.colorbar()
# only continental us without Alaska
plt.xlim(-125,-65)
plt.ylim(23, 50)
# add a few cities for reference (optional)
ard = dict(arrowstyle="->")
plt.annotate('Tokio', xy = (-122.1381, 37.4292),
             xytext = (-112.1381, 37.4292), arrowprops= ard)
plt.annotate('Seúl', xy = (-71.1106, 42.3736),
             xytext = (-73.1106, 48.3736), arrowprops= ard)
plt.annotate('Paris', xy = (-87.6847, 41.8369),
             xytext = (-87.6847, 46.8369), arrowprops= ard)
plt.annotate('Osaka', xy = (-122.33, 47.61),
             xytext = (-116.33, 47.61), arrowprops= ard)
plt.annotate('Londres', xy = (-80.21, 25.7753),
             xytext = (-80.21, 30.7753), arrowprops= ard)
params = plt.gcf()
plSize = params.get_size_inches()
params.set_size_inches( (plSize[0] * 3, plSize[1] * 3) )
return plt.show()

```

File "C:\Users\aula\AppData\Local\Temp\ipykernel_13572\1403090939.py", line 2

```

"""
^

```

IndentationError: unexpected indent

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]:

```
?miningscience.map_science
```

In []:

```
?miningscience.download_pubmed
```

In []:

```

from Bio import Entrez
from Bio import SeqIO
from Bio import GenBank
from collections import Counter
import csv
import re
import pandas as pd

```

In []:

```
miningscience.download_pubmed("Lion")
```

In []:

```

# Contribución de Los autores
with open("data/pubmed_results.txt") as datafile:
    author_dict = {}
    for line in datafile:
        if re.match("AD", line):
            author = line.split("-", 1)[-1]# capture author
            author = author.strip()# remove leading and trailing whitespace
            author_dict[author] = 1 + author_dict.get(author, 0)# if key is present, add 1,

```

In []:

```

for author in sorted(author_dict, key = author_dict.get, reverse = True):
    print(author, ":", author_dict[author])

```

```

Health PEI (German, Lutes), Charlottetown, PEI; McGill University Health Centre : 35
Centre for Ophthalmology and Visual Science (incorporating Lions Eye Institute), : 33
Centre for Ophthalmology and Visual Science, The University of Western Australia, : 32
Centre for Eye Research Australia, Royal Victorian Eye and Ear Hospital, East : 30
Bascom Palmer Eye Institute, University of Miami Miller School of Medicine, : 18
International Centre for Eye Health, London School of Hygiene and Tropical : 18
Senior Department of Infectious Diseases, The Fifth Medical Center of Chinese PLA : 18
Centre for Ophthalmology and Visual Science (Incorporating Lions Eye Institute), : 17
Partnership for Advanced Computing in Europe 1050 Bruxelles, Belgium. : 17
Garvan-Weizmann Centre for Cellular Genomics, Garvan Institute of Medical : 15
Australian Inherited Retinal Disease Registry and DNA Bank, Department of Medical : 15
Department of Ophthalmology, Boston Children's Hospital, Harvard Medical School, : 15
Tianjin Key Laboratory of Retinal Functions and Diseases, Tianjin Branch of : 15
The LION Foundation for Dental Health (Public Interest Incorporated Foundation), : 15
Institute of Epidemiology, Disease Control and Research (IEDCR), Dhaka, : 15
The International Society of Applied Neuroimaging (ISAN), Denver, CO, United : 14
Centre for Ophthalmology and Visual Science, University of Western Australia, : 14
Laboratory of Deep Sea Microbial Cell Biology, Institute of Deep-Sea Science and : 14
Department of Ophthalmology, Saarland University Medical Center (UKS), Kirrberger : 14
Treatment and Research Center for Infectious Diseases, The Fifth Medical Center : 14
Department of Ophthalmology, B.P Koirala Lions Center for Ophthalmic Studies, : 13
Department of Ophthalmology and Visual Neurosciences, University of Minnesota, : 13
Key Laboratory of Tropical Translational Medicine of Ministry of Education, : 13
The University of Leicester Ulverscroft Eye Unit, Department of Neuroscience, : 13
Department of Anatomy, College of Osteopathic Medicine, Des Moines University, : 13
Centre for Eye Research Australia, Royal Victorian Eye and Ear Hospital, : 12
Department of Pathogen Biology, Hainan Medical University, Haikou, China. : 12
Laboratory of Infectious Diseases, College of Veterinary Medicine, Konkuk : 12
ICAR-Indian Veterinary Research Institute, Izatnagar, Bareilly-243122, Uttar : 12

```

Ocular Tissue Engineering Laboratory, Lions Eye Institute, Nedlands, WA 6009, : 12
 Institute of Hematology and Blood Transfusion, Prague, Czech Republic. : 11
 Lions Eye Institute, Nedlands, Western Australia, Australia. : 11
 Department of Clinical Microbiology, Infection and Immunology, Umea University, : 11
 Department of Ophthalmology, Flinders University, Flinders Medical Centre, : 11
 Department of Ophthalmology, Bascom Palmer Eye Institute, University of Miami : 11
 Department of Ophthalmology, Juntendo University Graduate School of Medicine, : 11
 Department of Digital Medicine, Juntendo University Graduate School of Medicine, : 11
 Wilmer Eye Institute, Johns Hopkins University School of Medicine, Baltimore, : 11
 Antwerp Unit for Data Analysis and Computation in Immunology and Sequencing, : 11
 Department of Immunology, Genetics and Pathology, Uppsala University, 75185 : 11
 Nanchang Key Laboratory of Animal Health and Safety Production, Jiangxi : 11
 Arthur and Sonia Labatt Brain Tumor Research Centre, Hospital for Sick Children, : 11
 College of Wildlife and Protected Area, Northeast Forestry University, Harbin, : 10
 Faculty of Science, Sydney School of Veterinary Science, The University of : 10
 Human Immunology and Immunopathology, Institut National de la Sante et de la : 10
 European Synchrotron Radiation Facility, 71 Avenue des Martyrs, 38000 Grenoble, : 10
 Laboratory of Experimental Hematology, Vaccine and Infectious Disease Institute : 10
 Hoopes Vision Research Center, Hoopes Vision, Draper, UT, USA. : 10
 Washington National Primate Research Center, University of Washington, Seattle, : 10
 IHAP, UMR1225, Universite de Toulouse, INRAE, Ecole Veterinaire de Toulouse, : 10
 Bureau de Recherches Geologiques et Minieres (BRGM - French Geological Survey), 3 : 10
 Interdisciplinary Cluster for Applied Genoproteomics, University of Liege, CHU, : 10
 Department of Internal Medicine I, Division of Hematology & Hemostaseology, : 10
 Human Immunology, Pathophysiology, Immunotherapy (HIPI), INSERM U976, Universite : 10
 John A. Moran Eye Center, University of Utah School of Medicine, Salt Lake City, : 9
 Centre for Ophthalmology and Visual Sciences (incorporating the Lions Eye : 9
 Human Immunology, Pathophysiology and Immunotherapy, INSERM U 976, University : 9
 Nektar Therapeutics, San Francisco, California, USA. : 9
 International Society of Applied Neuroimaging, Denver, CO, United States. : 9
 Zambian Carnivore Programme, PO Box 80, Mfuwe, Eastern Province, Zambia. : 9
 Sahlgrenska Center for Cancer Research, Department of Laboratory Medicine, : 9
 Department of Medical Biosciences, Pathology, Umea University, Building 6M, : 9
 Laboratory of Experimental Hematology, Vaccine & Infectious Disease Institute : 9
 Massachusetts Eye and Ear, Department of Ophthalmology, Harvard Medical School, : 9
 Colorado Division of Parks and Wildlife, 4330 Laporte Avenue, Fort Collins, : 9
 Google Health, Google LLC, Mountain View, California. : 9
 Department of Microbiology, University of Washington, Seattle, WA, United States. : 9
 Research and Development Head Quarters, LION Corporation, Odawara, Kanagawa, : 9
 Univ. Lille, CNRS, UMR 8576-UGSF-Unite de Glycobiologie Structurale et : 9
 Centre for Eye Research Australia, Royal Victorian Eye and Ear Hospital, 32 : 9
 Department of Anatomy and Physiology, The University of Melbourne, Parkville, : 8
 School of Medicine, Menzies Research Institute Tasmania, University of Tasmania, : 8
 NIHR ARC NWC, Liverpool, UK. : 8

Ejercicio 3 [1.5 puntos]

Utilice dos veces la función `map_science` para:

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

In [4]:

```
miningscience.map_science("data/pubmed_results.txt")
```

Ejercicio 4 [1 punto]

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

Escriba la respuesta del ejercicio 5

Ejercicio 5 [2 puntos]

Para algún **gen de interés** (podría usar [Lista de genes por tipología](https://www.genome.jp/kegg/pathway.html#metabolism) (<https://www.genome.jp/kegg/pathway.html#metabolism>)), realice lo siguiente:

1. Una búsqueda en la página del [NCBI nucleotide](https://www.ncbi.nlm.nih.gov/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 del paso 4.

In [3]:

```
from Bio import SeqIO
from Bio import AlignIO
from Bio import Phylo
```

In []:

```
from Bio.Align.Applications import ClustalwCommandline
import os
```

In []:

```
# cargar data multiple y crear alineamientos
clustalw_exe = r"C:\Program Files (x86)\ClustalW2\clustalw2.exe"
clustalw_cline = ClustalwCommandline(clustalw_exe, infile = "data/rag2s.fasta")
assert os.path.isfile(clustalw_exe), "Clustal_W executable is missing or not found"
stdout, stderr = clustalw_cline()
print(clustalw_cline)
```

In []:

```
ClustalAlign = AlignIO.read("data/rag2s.aln", "clustal")
print(ClustalAlign)
```

****Escriba aquí la interpretación del árbol****

Alignment with 133 rows and 2895 columns

```
----- FJ230865.1
----- FJ230858.1
----- FJ039926.1
----- FJ039990.1
----- FJ039983.1
----- FJ039976.1
----- FJ039969.1
----- FJ039962.1
----- FJ039954.1
----- FJ039947.1
----- FJ039940.1
----- FJ039933.1
----- FJ009033.1
----- FJ039997.1
----- FJ009026.1
----- FJ039919.1
----- FJ039912.1
----- FJ230872.1
...
----- FJ230875.1
```

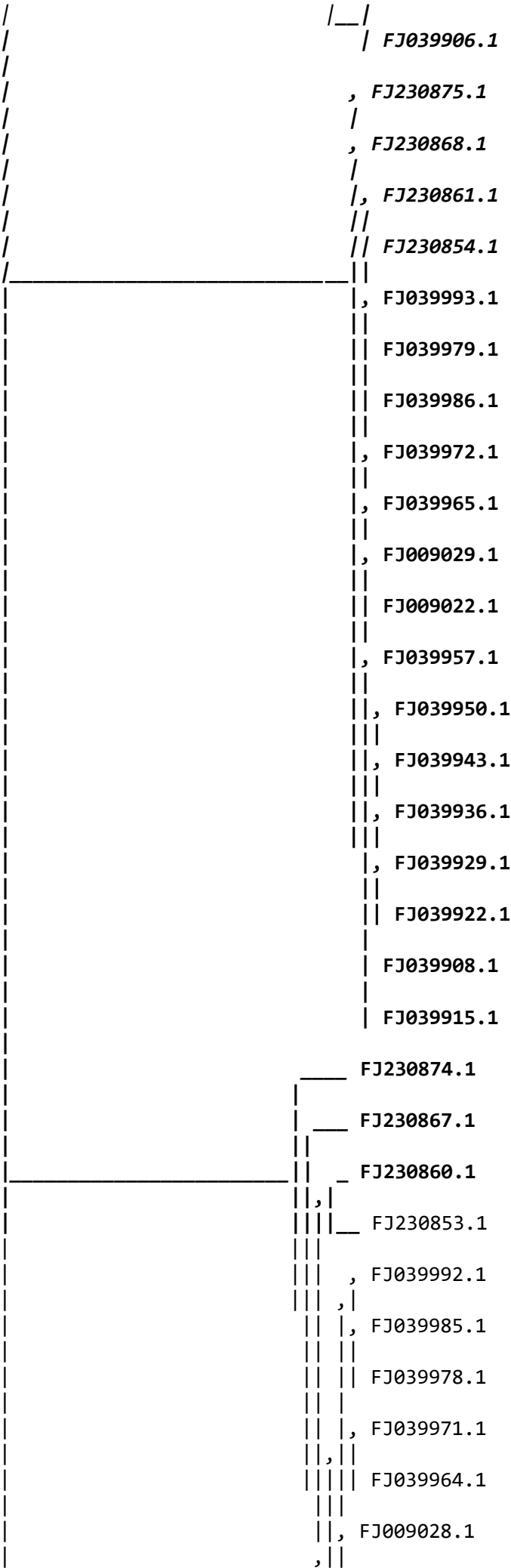
In []:

```
# Generar Dendogram (Tree)
from Bio import Phylo
tree = Phylo.read("data/rag2s.dnd", "newick")
Phylo.draw_ascii(tree)
```

```
_ FJ230879.1
|
|_ FJ230872.1
|   , FJ230865.1
|   , FJ230858.1
|   FJ039926.1
|   , FJ039997.1
```



```
, FJ039990.1
|
| FJ039983.1
|
| , FJ039976.1
|
| FJ039969.1
|
| FJ039962.1
|
| ,
| , FJ039954.1
|
| FJ039947.1
|
| , FJ039940.1
|
| FJ039933.1
|
| FJ009033.1
|
| FJ009026.1
|
| , FJ039919.1
|
| FJ039912.1
|
|
| FJ230873.1
|
| FJ230866.1
|
| FJ230859.1
|
| ,
| FJ230852.1
|
| , FJ039991.1
|
| ,
| , FJ039984.1
|
| FJ039977.1
|
|
| , FJ039970.1
|
| ,
|
| FJ039963.1
|
|
| , FJ009027.1
|
|
| FJ009020.1
|
|
| , FJ039955.1
|
|
| , FJ039948.1
|
|
| FJ039941.1
|
|
| , FJ039934.1
|
| ,
| , FJ039927.1
|
|
| FJ039920.1
|
|
| , FJ039913.1
```



1

		FJ039994.1
		FJ039987.1
		FJ039980.1
		FJ039973.1
		FJ039966.1
	,,	FJ009030.1
		FJ009023.1
	,	FJ039951.1
		FJ039923.1
		FJ039937.1
		FJ039944.1
		FJ039930.1
		FJ039958.1
	,	FJ039916.1
		FJ039909.1

```
| FJ039938.1
|
_| FJ039931.1
|
, FJ039952.1
|
| FJ039945.1
|
| FJ039924.1
```

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:

```
``sh
Carlitos Alimaña ha realizado el ejercicio 1
``
``sh
Carlitos Alimaña ha realizado el ejercicio 2
``
``sh
...
``
```

In []: