



EXAMEN PARCIAL PYTHON

GBI6-2021II: BIOINFORMÁTICA

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

03-08-2022

Melissa Purgul
GO1

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.

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 as msc
help(msc.download_pubmed)
```

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

```
import os
import re
c = msc.download_pubmed("chicungunya")
b = len(c)
print('El número artículos para keyword es:', b)
with open("Data/chicungunya.txt", "w") as txt:
    txt.write(c)
```

```
import os
import re
z = msc.download_pubmed(" Zika")
b = len(z)
print('El número artículos para keyword es:', b)
with open("Data/zika.txt", "w") as txt:
    txt.write(z)
```

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`

In [3]:

Escriba aquí su código para el ejercicio 6

```

from Bio import Entrez
from Bio import Phylo
from Bio.Phylo.TreeConstruction import DistanceCalculator
from Bio.Phylo.TreeConstruction import DistanceTreeConstructor
from Bio import AlignIO
from Bio import SeqIO
from Bio.Align.Applications import ClustalwCommandline
import os

with open("../data/sequence.seq") as file:
    text = file.read()
text = text.split('\n')
text = ' '.join(text[:103])

handle = Entrez.open(handle="nucleotide", rettype="gb", retmode="text", id=text)
print(handle.url)
records = SeqIO.parse("../data/sequence.gb", "genbank")
count = SeqIO.write(records, "sequence.fasta", "fasta")

```

Escriba aquí la interpretación del árbol

```

Clustal Align = AlignIO.read("sequence.aln", "clustal")
with open("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(distance_matrix)
rag2_tree.rooted = True
fig = Phylo.draw(rag2_tree)
fig.savefig('arbol.pdf')

```

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 []: