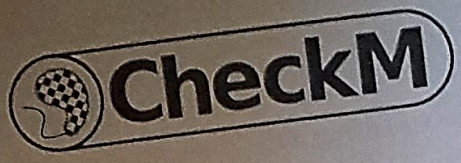

EXAMEN PARCIAL PYTHON

GBI6-2021II: BIOINFORMÁTICA

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

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 parámet
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 ascedente y
- seleccionar los cinco más abundantes. Con esta selección debe graficar un `pie_plot` . Como guía
conteo por países puede usar el ejemplo de MapOfScience (https://github.com/CSB-
/regex/solutions/MapOfScience_solution.ipynb).

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

In [1]:

```
# Escriba aquí su código para el ejercicio 1
import miningscience as msc
help(download_pubmed)
help(science_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:

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
from miningscience import download_pubmed
VR1 = download_pubmed ("VIRUELA")
VR2 = (download_pubmed ("VRLA"))
print ("El número de artículos para VIRUELA es: ", len (VR1))
print ("El número de artículos para VRLA es: ", len (VR2))
```


[4]:

```
Escriba aquí su código para el ejercicio 3
science-plots("VIRUELA")
science-plots("VRLA")
```

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.Phylo.TreeConstruction import DistanceTreeConstructor
from Bio import Entrez
import re
import os
from Bio.Align.Applications import ClustalwCommandline
with open ("data/sequence.seq") as f:
    dat = f.readlines()[0:15]
out_sequence = open("data/secuencias.fasta", "w")
```

(lo demás está en el archivo de notebook, no alcanzó la hoja)

Escriba aquí la interpretación del árbol

Construya las funciones del módulo miningscience.py

```
def download_pubmed( VIRUELA                    ):
    """
    Función que pide como input la palabra de búsqueda en tipo str del pubmed
    y como output, guardando un documento con extensión txt que contiene
    los datos de la búsqueda y se hace un llamado de librería data de
    pubmed importando desde bio pfjfthon para el gen HPV18J1 en humanos.
    El code 'efetch' Recupera registros en el formato solicitando de una lista


    """
    import Bio
    import re
    from Bio.seq import Seq
    from Bio import Entrez
    Entrez.email = "ignacio.carranco@est.ikiam.edu.ec"
    handle =Entrez.esearch (db= "pubmed",
                            term= "VIRUELA",
                            usehistory = "y")
    record = Entrez. read (handle)
    id_list = record ["IdList"]
    webenv = record ["WebEnv"]
    query_key = record ["QueryKey"]
    handle= Entrez.efetch (db= "pubmed",
                           rettype= "medline",
                           retmode = "text",
                           restart = 0
                           retmax = 543,
                           webenv = webenv,
                           query_key = query_key
out_handle = open ("data/ VIRUELA-pub.txt", "w")
data = handle .read()
(id_list)
handle.close()
out_handle.write (data)
out_handle.close()

return id_list
```

Nombre (Apellido, Nombre):

Universidad Regional Amazonica

```
def science_plots( PAISES, VIRUELA ):
    """
    Función con la cual llamaremos los datos de la pregunta 1
    y reguardara en el archivo txt de la carpeta data
    además hará el conteo de los paises.


    """
    import matplotlib.pyplot as plt
    import csv
    import re
    import pandas as pd
    from collections import Counter
    with open("data/VIRUELA-pub.txt", errors="ignore") as I:
        texto = I.read()
        texto = re.sub(r"\n\s{6}", " ", texto)
        countries_1 = re.findall(r"AD\s{2}-\s[A-Za-z].*,\s([A-Za-z]+)\.\s", texto)
        unique_countries = list(set(countries_1))
        conteo = Counter(countries_1)
        resultado = {}
        for clave in conteo:
            valor = conteo[clave]
            if valor > 1:
                resultado[clave] = valor
                ordenar = (sorted(resultado.values()))
                ordenar.sort(reverse = True)
    import operator
    pais = []
    contador = []
    reverse = sorted(resultado.items(), key = operator.itemgetter(1), reverse=True)
    for name in enumerate(reverse):
        pais.append(name[1][0])
        contador.append(resultado[name[1][0]])
    cinco_paises = pais[0:5]
    fig = plt.figure(figsize = (10,7))
    plt.pie(frecuencia_cinco, labels = cinco_paises)
    (plt.savefig("img/VIRUELA.jpg", dpi=100, bbox_inches='tight'))
    plt.show()
    return(conteo)
```