EXAMEN PARCIAL PYTHON

GBI6-2021II: BIOINFORMÁTICA

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

03-08-2022 Josue Chávez Martinez

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` . · extraer al pubmed

ii. `science_plots` : la función debe → guardar data.txt → ~~Exportar~~ → cargar data

- utilizar como argumento de entrada la data descargada por `download_pubmed`
- ordenar los conteos de autores por país en orden ascedente y — Contar los autores por país y ordenar por descendente
- 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). - Cambiar por País

iii *Cree un docstring para cada función.* - De ayuda frente la función

1/4

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

In [1]:

```
# Escriba aqui su código para el ejercicio 1
import miningscience_g01
import miningscience_g01 as msc

help(msc.download_pubmed)
help(msc.science_plots)
```

Ejercicio 2 [2 puntos]

Utilice dos veces la función `download_pubmed` para: — 2 Key word

- 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 articulos para KEYWORD es: XX' # Que se cargue con inserción de texto o valor que correspondea KEYWORD y XX
```

In [2]:

```
# Escriba aqui su código para el ejercicio 2
paper= msc.download_pubmed ("Eumenince")
print ("El numero de articulos para Eumenince es:", len(paper))

artic = msc.download_pubmed ("termite")
print ("El número de articulos para termite es:", len(artic))
```

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`

]:

Escriba aquí su código para el ejercicio 3

```
Eumeninae = use. science_plots (paper)
with open ("img/Eumeninae.png", "w") as jpg:
    Eumeninae

termite = use. science_plots (article)
with open ("img/termite.png", "w") as jpg:
    termite
```

Ejercicio 4 [1 punto]

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

En el primer diagrama de pastel se pueden observar los cinco países con mas publicaciones referentes a la especie Eumeninae, donde el porcentaje más alto de autores con investigaciones respecto a dichos alteres lo tiene China con 41,5%, seguido de Alemania con 29,2% y en tercer lugar Brasil con el 15,4%

Escriba la respuesta del ejercicio 5. o por su nombre científico Eumeninae,

Ejercicio 5 [2 puntos]

Nombre enzima → gluconeogenesis

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. — Solo 15
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.

In [3]:

```
# Escriba aquí su código para el ejercicio 6
with open('Data/sequence.seq') as file:
    text = file.read()
text = text.split('\n')
handle = Entrez.efetch(db="nucleotide", rettype="gb", retmode="text", id=text)
records = SeqIO.parse("Data/sequence.gb", "genbank")
clustalw_exe = r"C:/Program Files (x86)/clustalw/clustalw.exe"
clustalw_cline = ClustalwCommandline(clustalw_exe, infile="Data/sequence.fasta")
assert os.path.isfile(clustalw_exe), "Clustal W executable is missing or not found"
stdout, stderr = (clustalw_cline)
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)
tree = constructor.build_tree(alignment)
tree.rooted = True
fig = plt.figure(figsize=(10,12), dpi=200)
matplotlib.rc('font', size=12)
matplotlib.rc('xtick', labelsize=20)
matplotlib.rc('ytick', labelsize=20)
axes = fig.add_subplot(1, 1, 1)
Phylo.draw(tree, axes=axes)
fig.savefig("img/arbol.jpg")
```

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 ):
    """
    Este comando me ayudara buscar articulos en pubmed a través de una palabra
    clave. """

    Entrez.email = "gualapaco.moises@gmail.com"
    busq = Entrez.read (Entrez.esearch (db = "pubmed", term = Keyword, usehistory = "y"))

    webenv = busq ["WebEnv"]
    query_key = busq ["QueryKey"]
    """
    handle = Entrez.efetch (db = "pubmed", rettyp = "medline", retmode = "text", retstart=0,
                            retmax = 543, webenv = webenv, query_key = query_key)

    data = handle.read ()
    data exp = re.sub (r'\n\s{6}', '', data)
```

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

Ikiam
Universidad Regional Amazónica

Nombre [Apellido, Nombre]:

```
def science_plots(    doc
    """                                    ):
```

Esta función me ayuda a contar los paises de autores que produjeron articulos relacionados a la palabra clave de busqueda

```
Correc = re.sub (r'\s[\w.-%+-]+[\w.-]+\.[a-zA-Z]{2,4}', '',
        doc)
Coord = re.sub (r'\.\d.', ' ', ' ', correc)
  """
numb = re. sub (r'1..\d.', '', coord)
X = numb [1:]. split ('PMID-')

Countries_A = []
For PMID in X:
q = PMID. split ('\n')
~~for~~  For fila in q:
            w = fila. split (' ')
            if w[0] == 'AD':
                e = fila. split (',')
                Countries_A. append (e[-1])

a=0
Countries_B = [0]*len(Countries_A)
For lis in Countries_A:
    bytes (lis, encoding = "utf8")
    if lis != '':
        w = lis
        if w[0] == ' ':
            w = re.sub (r'^\s', '', w)
        if w[-1] = ' ':
            w = re.sub (r'\.$', '', w)
        w = re.sub (r'\.$', '', w)
        w = re.sub (r'\s$', '', w)
        Countries_B[a] = w
        a = a+1
    Countries_all = [todos los paises]
```

```
Countries_C = Countries_B
k = countries_all
f = len(k)
Countries Count = [0] * f
K = 0
for elen in k:
    d = 0
    for comp in Countries_C:
        if elen == str(comp):
            Countries Count [K] = d
            K = K + 1

Countries_D = []
Counter = []
O = 0
For Une in Countries Count
    If str(Une) != '0':
        Counter.append(Une)
        m = countries_all [O]
        Countries_D.append(m)
        O = O + 1

table_D = pd.DataFrame({'Country': Countries_D, 'nom_auth': counter})
Order = table_D.sort_values(by = ['nom_auth'], ascending = [false])
taken = Order.iloc [0:5]
soma = taken ['nom_auth'].sum()
iv = taken.iloc [:, 0]
su = pd.series (iv)
i = taken.iloc [:, 1]
sa = pd.series (i)
s = sa.tolist()

prom = []
for number in s:
    xa = (number / soma) * 100
    prom.append(xa)
table_B = pd.Dataframe({'Country': su, 'percent': prom})

fig 1, ax1 = plt.subplots()
ax1.pie (prom, labels = s, autopct = '%1.1f%%',
        shadow = true, startangle = 90)
ax1.axis('equal')
plt.show()
plt.savefig('img/Grafica de pie.....jpg', dpi = 500)
return (table_B)
```