

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS
ESCUELA PROFESIONAL CIENCIA DE LA COMPUTACIÓN



BIOINFORMÁTICA

Práctica N°9

UPGMA

GRUPO N°2

ALUMNOS:

APAZA CHAVEZ, MARIA LOURDES
BARRIOS CORNEJO, SELENE
GOMEZ CONTRERAS, JUNIOR VALENTIN
PILCO PANCCA, LUZ MARINA

DOCENTE:

MACHACA ARCEDA, VICENTE

AREQUIPA - PERÚ

2021

EJERCICIOS

1. Instale las librerías BioPython, scikit-bio y ETE para el análisis de árboles filogenéticos. Además describa las librerías, en un parrafo de no mas de 5 lineas.

```
pejelagarta@pejelagarta-FX503VD: ~
r, parso, jedi, pygments, IPython, Cython, hdmmedians, kiwisolver, cycier, pypar
sing, matplotlib, natsort, pandas, joblib, threadpoolctl, scipy, scikit-learn,
scikit-bio
Successfully installed CacheControl-0.12.6 Cython-0.29.23 IPython-7.25.0 backca
ll-0.2.0 cycier-0.10.0 decorator-5.0.9 hdmmedians-0.14.2 ipython-genutils-0.2.0
jedi-0.18.0 joblib-1.0.1 kiwisolver-1.3.1 matplotlib-3.4.2 matplotlib-inline-0.
1.2 msgpack-1.0.2 natsort-7.1.1 pandas-1.3.0 parso-0.8.2 pickleshare-0.7.5 prom
pt-toolkit-3.0.19 pygments-2.9.0 pyparsing-2.4.7 scikit-bio-0.5.6 scikit-learn-
0.24.2 scipy-1.7.0 threadpoolctl-2.1.0 traitlets-5.0.5 wcwidth-0.2.5
pejelagarta@pejelagarta-FX503VD:~$ pip install ETE
Collecting ETE
  Downloading ete-1.0.0-py3-none-any.whl (7.1 kB)
Collecting toml<0.11.0,>=0.10.2
  Downloading toml-0.10.2-py2.py3-none-any.whl (16 kB)
Collecting click<9.0.0,>=8.0.0
  Downloading click-8.0.1-py3-none-any.whl (97 kB)
  | 97 kB 432 kB/s
Collecting PyYAML<6.0.0,>=5.4.1
  Downloading PyYAML-5.4.1-cp38-cp38-manylinux1_x86_64.whl (662 kB)
  | 662 kB 868 kB/s
Installing collected packages: toml, click, PyYAML, ETE
Successfully installed ETE-1.0.0 PyYAML-5.4.1 click-8.0.1 toml-0.10.2
pejelagarta@pejelagarta-FX503VD:~$
```

Figura 1: Instalación de librerías

```
pejelagarta@pejelagarta-FX503VD: ~
pejelagarta@pejelagarta-FX503VD:~$ pip list
Package              Version
-----
apturl               0.5.2
asgiref              3.4.0
backcall             0.2.0
bcrypt               3.1.7
biopython            1.79
blinker              1.4
Brlapi               0.7.0
CacheControl         0.12.6
certifi              2019.11.28
chardet              3.0.4
click                8.0.1
colorama             0.4.3
command-not-found    0.3
cryptography         2.8
cupshelpers          1.0
cycier               0.10.0
Cython               0.29.23
dbus-python          1.2.16
decorator            5.0.9
defer                1.0.6
```

Figura 2: Pip List

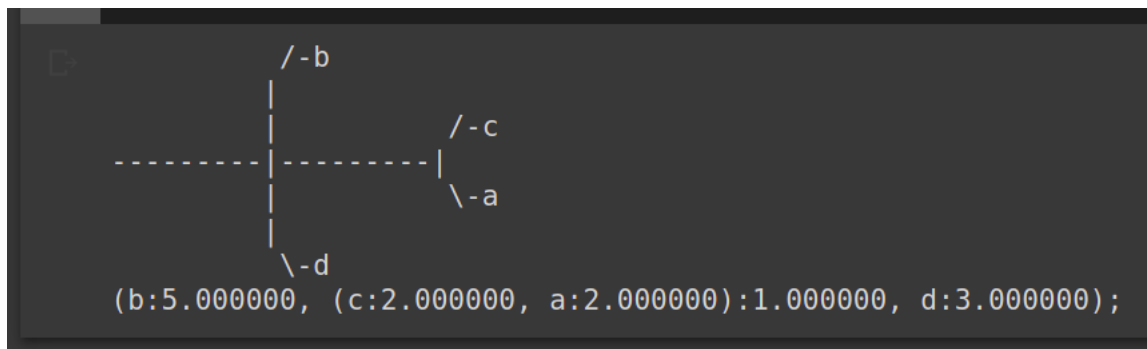
ETE permite cargar, crear, recorrer, buscar, podar o modificar estructuras de árboles filogenéticos. **Biopython** es una coleccion de modulos de Python que proporcionan funciones para manejar operaciones de secuencias de ADN, ARN y proteínas. **Scikit-bio** proporciona estructuras de datos,algoritmos y recursos educativos para bioinformática.

2. Evalúe el código a continuación y describa su funcionamiento

```

1  from ete3 import PhyloTree, TreeStyle
2  from skbio import DistanceMatrix
3  from skbio.tree import nj
4
5  data = [[0, 8, 4, 6],
6          [8, 0, 8, 8],
7          [4, 8, 0, 6],
8          [6, 8, 6, 0]]
9  ids = list('abcd')
10 dm = DistanceMatrix(data, ids)
11 tree = nj(dm) # build a tree using neighbors joining algorithm
12 print(tree.ascii_art())
13
14 newick_str = nj(dm, result_constructor=str) # return newick format
15 print(newick_str)
16
17 t = PhyloTree(newick_str) # plot tree using ETE
18 t.show()

```



En la línea 5 se define un objeto de matriz de distancia que describa las distancias entre 4 OTUs: a, b, c y d. En la línea 10 se está construyendo el árbol de unión vecino que representa la relación entre esas OTUs. Esto se devuelve como un objeto `TreeNode`. En la línea 14 se vuelve a construir el árbol de unión vecino, pero en su lugar devuelve la cadena newick que representa el árbol, en lugar del objeto `TreeNode`. Línea 17 y 18 es para dibujar y mostrar el árbol.

3. Evalúe el código a continuación y describa su funcionamiento.

```

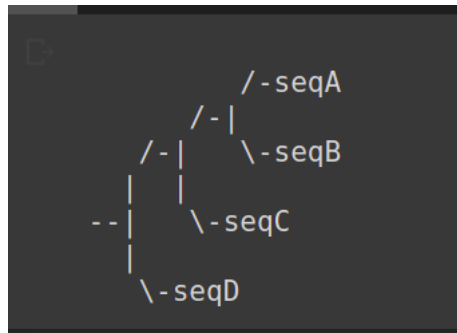
1  from ete3 import PhyloTree, TreeStyle
2  from skbio import DistanceMatrix
3  from skbio.tree import nj
4
5  fasta_txt = """
6  >seqA
7  MAEIPDETIQQFMALT---HNIQVQYLSEFGDLNEALNSYYASQTDDIKDRREEAH
8  >seqB
9  MAEIPDATIQQFMALTNVSHNIAVQY--EFGDLNEALNSYYAYQTDDQKDRREEAH
10 >seqC
11 MAEIPDATIQ---ALTNVSHNIAVQYLSEFGDLNEALNSYYASQTDDQKDRREEAH
12 >seqD
13 MAEAPDETIQQFMALTNVSHNIAVQYLSEFGDLNEAL-----REEAH

```

```

14  """
15
16  # Load a tree and link it to an alignment.
17  t = PhyloTree("((seqA,seqB),seqC),seqD);")
18  t.link_to_alignment(alignment=fasta_txt, alg_format="fasta")
19  t.show()

```



El código que se muestra sirve para la vinculación de árboles filogenéticos con alineaciones de secuencias múltiples. Las instancias permiten que las filogenias moleculares estén vinculadas a las alineaciones de secuencias múltiples. Para asociar alineaciones de secuencias múltiples con un árbol filogenético, se puede utilizar el método `t.link_to_alignment()`, se usa `alg_format` para especificar el formato.

4. Implemente el algoritmo UPGMA. Este debe tomar como entrada una matriz de distancias y debe retornar el árbol filogenético en formato newick. Luego utilice la librería *ete3* para visualizar el árbol. Puede tomar como entrada la matriz de la pregunta 2.

Solución:

5. Conclusiones

- UPGMA es un método muy sencillo.
- Al realizar los cálculos basados en la matriz de distancias y no directamente sobre las secuencias es mucho más rápido computacionalmente que los métodos basados en carácter, tiene una complejidad de $O(n^2)$.
- Las secuencias no son consideradas como tales, sino que se trabaja con la matriz de distancias lo que puede causar pérdida de información.

6. Repositorio

-