Proyecto BioCorp

Edgar Alejandro Santana Hernández

Ingeniería Civil Informática, Universidad de Aysén

Gabriel Núñez V.

6 de noviembre de 2022

**Desafío**

Crear un programa que lea una secuencia de nucleótidos de un gen para obtener la secuencia de la proteína correspondiente, que genere estadísticas de la secuencia y que genere un diagrama de relación de aa

**Análisis de solución**

Primero creare 2 diccionarios ambos usaran los codones como key y lo que cambiara será el contenido de cada key en una se almacenara las siglas de la proteína y en otra solo su letra luego creare un menú con un ciclo while con 3 opciones.

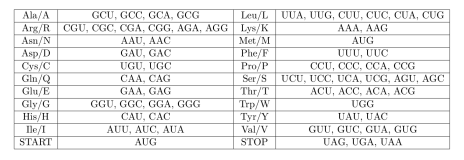
la primera opción la trabajare en una función en donde la información que me proporcionen mediante archivos fasta la meteré en una lista para modificarla lo primero que hare es reemplazar las T por U para consiguiente eliminar lo que hay entre la i y la F de los archivos una vez terminado esto que busque el codón de inicio posición por posición y cuando lo encuentre que almacene de 3 posiciones en 3 posiciones ya que esta serán los codones de la proteína cada codón que encuentre lo almacenara en una lista hasta que encuentre el codón de fin o llego directamente a su final, finalmente usando la lista y el diccionario que almacena las letras correspondientes a cada codón comparara cada key del diccionario con lo que hay en la lista luego esto será almacenado en otra que lista que finalmente será transformada en un string que se almacenara en un archivos para el usuario.

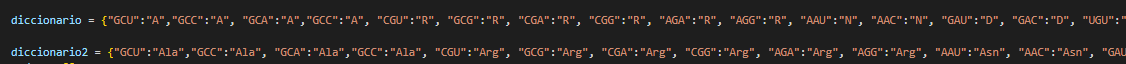
La segunda opción la trabajare en el mismo ciclo while del menú ya que no será tan larga lo primero que hará será mostrar el numero y porcentaje de nucleótidos (A, C, U, G) por secuencia valida, esto lo hare con un ciclo while y que contará la cantidad de nucleótidos que existe en una secuencia y luego imprimirá la cantidad de cada nucleótido junto a su porcentaje respecto al total de nucleótidos existentes. Usando otro ciclo while contare la cantidad de aminoácidos por secuencia valida (polares sin carga, con carga positiva, con carga negativa y apolares) que se almacenaran en variables que imprimirá al final.

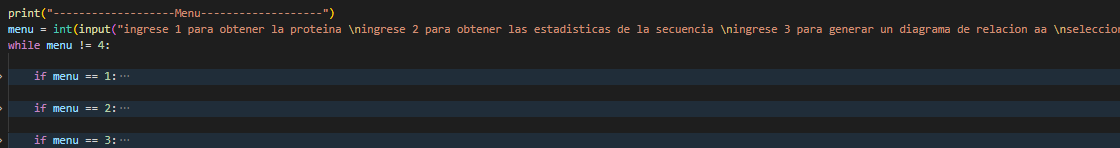
En la última opción usando la librería graphviz, la lista que almaceno los codones y el diccionario que almaceno las siglas correspondientes a cada codón generare un diagrama que de relación de aminoácido.

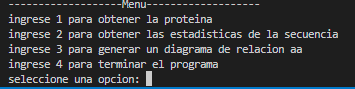
**Descripción código**

Importo la librería de pandas y de graphviz

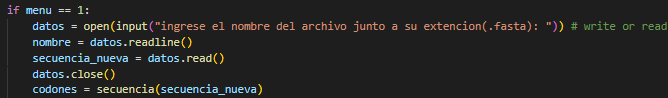
Creo 2 diccionarios ambos almacenan los codones como key pero uno almacena el resultado de esa key como su letra correspondiente y la otra su sigla



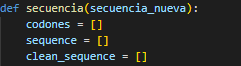
Lo primero que se muestra al usuario es un input que muestra las opciones a las que puede acceder este input trabaja junto a un ciclo while



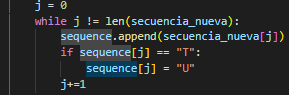
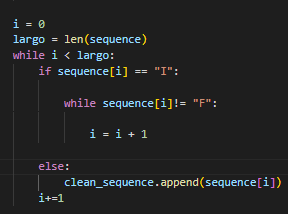
la primera opción le pide al usuario un archivo de tipo fasta para abrirlo luego con **datos.readline** lee la primera línea donde esta el nombre de la secuencia y luego con datos.read lee toda la secuencia con datos.close cierra el archivo y ya con codones = secuencia(secuencia\_nueva) llama a esa función.

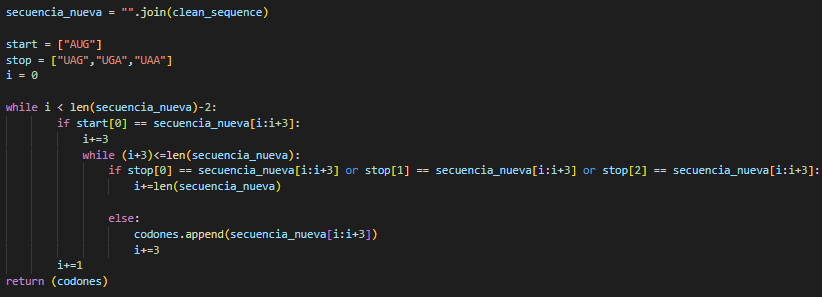


Una vez en la función se ven 3 listas que se usaran mas adelante

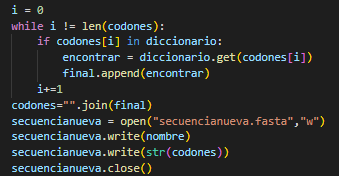


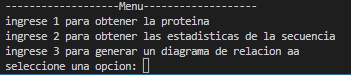
El primer ciclo while agrega a la lista sequence cada letra que hay en secuencia\_nueva que es la información que se le dio a la función una vez agregado a la lista se consulta si la letra que se agrego es una T si es el caso la reemplaza por una U si no, no hace nada luego a J se le suma 1 hasta que J sea igual al largo de la información dada a la función.

En este ciclo while consulta si en la lista sequence en la posición I (I comienza en 0) es igual a I en caso de ser así a I le suma 1 hasta encontrar una F si no se da el se agrega sequence en la posición i a una lista llamada clean\_sequence todo esto hasta que i sea igual a largo que es el largo de la lista sequence.

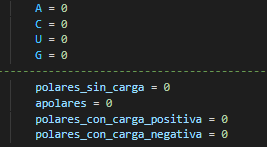
En el siguiente ciclo while antes de entrar al ciclo transforma la lista clean\_sequence a un string juntando toda la lista luego existen 2 listas una llamada start que almacena el único codón por el cual se comienza y otra lista llamada stop que almacena los 3 codones en los que se termina la secuencia. Ahora cuando entre al ciclo while el cual dura hasta que i sea mayor al largo del string (secuencia\_nueva)-2 lo primero que hará es consultar si secuencia nueva en la posición i hasta i +3 ([0:0+3] es igual a 0 , 1 , 2) es igual a lo codón de inicio de la lista start en caso de no ser así a i le suma 1 y hace la misma consulta en el caso de sean iguales entra siguiente ciclo while a i le suma 3 y almacena esas 3 posiciones en una lista llamada codones hasta que secuencia\_nueva en la posición [i:i+3] sea igual a algún codón de la lista stop o hasta que i+3 sea igual al largo de secuencia nueva, finalmente retorna la lista codones.

luego de trabajar en la función nos devuelve a la opción 1 del ciclo while donde transforma la lista codones a un string que se almacena en la variable porcentaje y almacena la lista codones en una variable llamada codones2 que usaremos para las otras opciones.

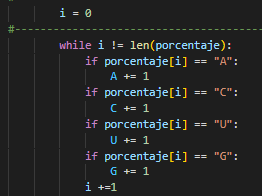
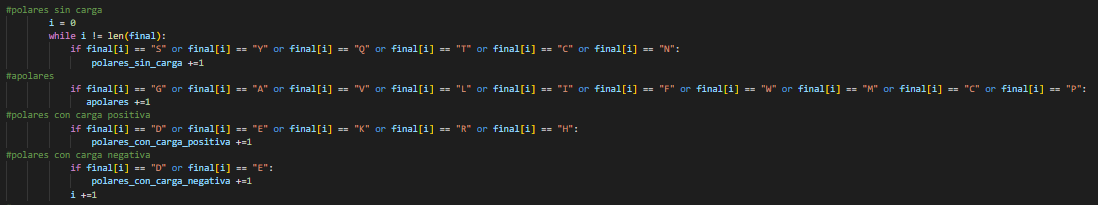
Este ultimo ciclo while lo que hace es que si codones en la posición i esta en el diccionario va a encontrar la letra que corresponda al codón y la guardara en una lista llamada final luego pasara esta lista a un string, a continuación, abrirá un archivo en caso de no existir lo creara y ahí guardara en la primera línea el nombre de la secuencia y en la segunda la proteína y cerrara el archivo.

finalmente te devolverá a las opciones del menú.

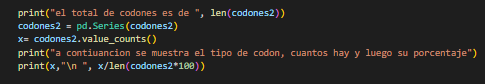
En la segunda opción estas variables que están establecidas en 0 cada una y se usaran para sacar la cantidad y el porcentaje de cada una.



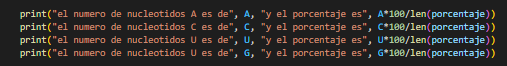
el primer ciclo while de la opción 2 se usará la variable almacenada en la opción llamada porcentaje para contar la cantidad de letras que hay en la secuencia y dependiendo la letra se contara 1 a su correspondiente variable.

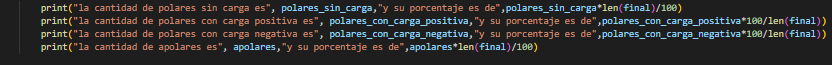
terminado lo anterior pasara al siguiente ciclo while el cual preguntara si la letra almacenada en la lista final es polar sin carga, polar con carga positiva, polar con carga negativa o apolar hasta que i sea igual al largo de la lista final

la primera información que le mostrara al usuario será la cantidad de codones en la secuencia luego usando la librería de pandas con el arreglo **pd.series** almacenara la cantidad total de codones en una variable luego en la variable x se almacenara la cantidad de codones de cada tipo que haya en la secuencia, finalmente imprimirá al usuario la cantidad de codones existentes en la secuencia junto a su porcentaje en relación a total de codones.



los siguientes print imprimirán la cantidad de existente de cada nucleótido junto a su porcentaje en relación al total de nucleótidos

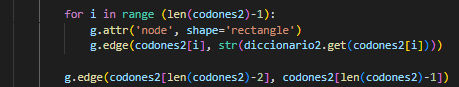


los últimos print mostraran al usuario la cantidad de aminoácidos por secuencia valida junto a su respectivo porcentaje.

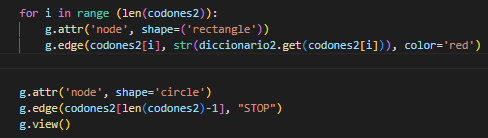
En la opción 3 se crea una variable con nombre g que es donde se crea el grafico y dentro de este se crea un nodo en forma de circulo que junta el codón de inicio AUG con el primer codón de la lista codones2 .



En el primer ciclo for anida con una flecha codones2 en la posición i hacia codones2 en la posición i+1 hasta que i sea igual al largo de la lista (codones2)-1 finalmente toma las ultimas 2 posiciones de la lista codones y las junta



El último ciclo for hace exactamente lo mismo solo que anida cada codón a su respectivo aminoácido con una flecha roja, luego el ultimo codon de la secuencia lo anida a un stop y luego abre un archivo con el diagrama



Finalmente te devuelve al menú

Para terminar el programa se debe ingresar 4