



UNIVERSIDAD SIMÓN BOLÍVAR

Departamento de Computación y Tecnología de la Información

Laboratorio de Algoritmos y Estructuras II Abr-Jul 2016

## Proyecto 1- v. 1.0

Como estudiantes del Laboratorio de Algoritmos y Estructuras II se requiere su colaboración para implementar algoritmos para emular el proceso de síntesis de proteínas. La información necesaria de todo el proceso se encuentra en los siguientes enlaces.

[https://es.wikipedia.org/wiki/C%C3%B3digo\\_gen%C3%A9tico](https://es.wikipedia.org/wiki/C%C3%B3digo_gen%C3%A9tico)

[https://es.wikipedia.org/wiki/Bios%C3%ADntesis\\_proteica](https://es.wikipedia.org/wiki/Bios%C3%ADntesis_proteica)

<http://www.monografias.com/trabajos/sinteproteinas/sinteproteinas.shtml>

<http://anaisabelromerasanchez.blogspot.com/2013/11/sintesis-de-yproteinas-transcripcion-.html>

1.- Una cadena doble de ADN consiste de dos secuencias entrelazadas y complementadas que sólo consiste de los nucleótidos Adenina, Guanina, Timina, y Citocina. Adenina y Timina se complementan, al igual que Guanina y Citosina. Una representación clásica sería así:

```
...ATGGGCAATCGGTTTGC...  
...|||||||...  
...TACCCGTTAGCCAAACG...
```

Esta información se representa por pares alternados en una cadena de caracteres

(AT) (TA) (GC) (GC) (GC) (CG) (AT) (AT) (TA) (CG) (GC) (GC) (TA) (TA) (TA) (GC) (CG)

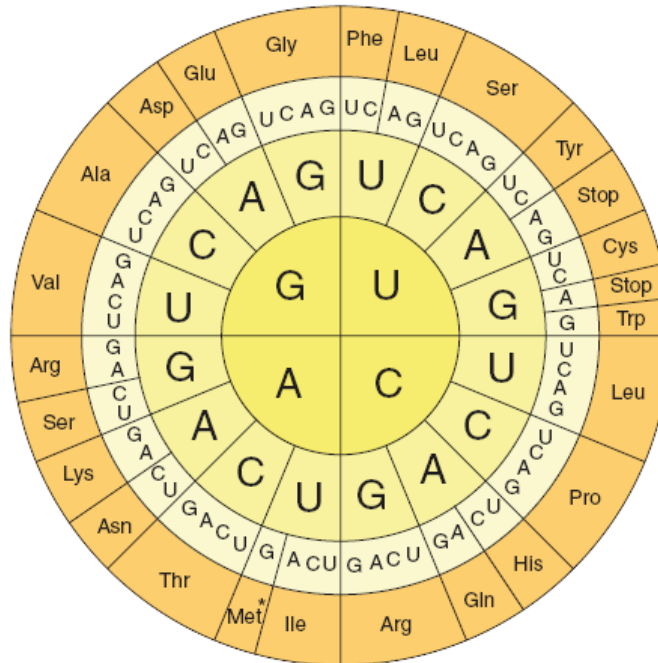
El ADN se puede separar en sus dos cadenas simples originales. Usted deberá crear las clases *ADNDoble*, *ADNSimple*, *ARNt* y *Proteína*, que deberán tener como atributo adicional la longitud de la cadena. Cada cadena o proteína será un objeto de la clase respectiva. Deberá definir las clases *CONTENEDORAS* que almacenan los objetos homogéneos respectivos y sobre las que se implementan los algoritmos de ordenamiento. Estas clases deberán tener el método *escribir (archivo)* que guarde respectivamente todas las cadenas o proteínas almacenadas.

La clase *ADNDoble* tendrá como mínimo los siguientes métodos:

- *zip (cadenasimple)*: recibe una cadena simple, la complementa y crea la cadena doble.
- *unzip ()*: Separa una cadena doble en dos cadenas simples.
- *mitosis ()*: Crea una copia de la doble cadena mediante **mitosis**.
- *buscar (subcadena)*: Recibe una subcadena *ADNSimple* y determina si existe un **match** de sus nucleótidos en una *ADNDoble*. Busca simultáneamente *subcadena* y su complemento.
- *imprimir ()*: escribe en forma de pares.

La clase **ADNSimple** tendrá además como mínimo el siguiente método:

- *complementar ()*: Dada una cadena simple, devuelve la cadena complemento.
- *transcribir ()*: Devuelve como un **ARNt** el complemento de la cadena simple, pero con la Timina sustituida por Uracilo.



\* Codón de iniciación

La clase **ARNt** tendrá además como mínimo el siguiente método:

- *traducir ()*: Complementa el **ARNt**, busca el codón de **inicio** y devolverá la proteína que resulta de encadenar los aminoácidos producto del recorrido de la secuencia según la Tabla de aminoácidos, hasta que llega a un **stop**. La traducción comienza con el codón "AUG" que es además de señal de inicio (el aminoácido **met**, *metionina*), El sentido de la traducción de codones es del centro hacia afuera en la Tabla. El número de codones posibles es 64, con 61 que codifican aminoácidos (uno sólo es el codón de señal de inicio, *met*) y los tres restantes son sitios de *parada* o **stop** (UAA, llamado *ocre*; UAG, llamado *ámbar*; UGA, llamado *ópalo*). La secuencia de codones determina la secuencia de aminoácidos en una proteína en concreto, con una estructura y una función específica. Ni el codón de **inicio** ni el codón de **stop** formarán parte de la secuencia proteica.

OJO: en una cadena de **ARNt** posea *cero o más* de una secuencia **inicio-stop**, cada una de las cuales da origen a una proteína. Es posible también que haya una cadena "rota", es decir, existe un **inicio** pero se llega al final de la cadena sin conseguir el respectivo **stop**. Si la cadena no codifica a ninguna proteína, el trozo se cataloga como **ARNBasura**, y el trozo de ADN que lo originó se almacena como **ADNBasura**. La proteína resultado es la cadena de aminoácidos en secuencia: **Ser,Leu,Arg,Pro,Leu,Leu,His**.

3.- Un procedimiento *lectura\_cadena(archivo)* lee las cadenas **simples** de ADN de un archivo, una por línea, y crea las cadenas dobles complementando y entrelazando pares adecuadamente.

Ej: **ATTTCGC** → (**AT**) (**TA**) (**TA**) (**CG**) (**GC**) (**CG**)

4.- En el proceso de traducción se va contando la frecuencia de cada codón. Cada codón con su frecuencia se guardan en un **heap** que se mantiene ordenado en orden *decreciente* de frecuencia. Al final se deberá recorrer este *heap*, imprimiendo los codones comunes y su frecuencia, y escribir la proteína que codificarían en esa secuencia.

5.- Una vez transcritas todas las proteínas, deben ordenarse usando **quicksort** por longitud de cadena de aminoácidos primero y en orden alfabético de aminoácidos para proteínas de la misma longitud.

6.- Las cadenas almacenadas en **ADNBasura** deben ordenarse por nucleótidos usando **mergesort**.

Se suministrarán casos de prueba y un programa principal de muestra.

**ENTREGA:** Cree un archivo comprimido del tipo “tgz” llamado *Proy1\_XX-XXXXXX\_YY-YYYYY.tgz*, donde XX-XXXXXX es su número de carnet y YY-YYYYY el de su compañer@. Su contenido será el programa *Proy01.py*. Debe subir el archivo en el aula virtual el día **26 de mayo de 2016** antes de las **1:30 p.m.**