Ejercicio de programación en Python

Como punto de partida de este ejercicio tenemos el genoma de *Xylella fastidiosa*, que se encuentra en el fichero ***genoma.fasta*** en formato FASTA y un fichero llamado ***codigo.txt*** con los códigos de traducción de cada triplete de nucleótidos a aminoácidos. Se trata de un código genético directo entre la secuencia con sentido (*sense*) del DNA y la de proteína. Esto nos evita tener que transcribir a RNA antes de traducir. Los distintos codigos genéticos pueden encontrarse en NCBI:

http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi

El fichero *codigo.txt* contiene 5 líneas. La primera indica la traducción a aminoácidos, la segunda enumera los inicios de traducción (es decir, los tripletes traducibles a metionina; esta línea se puede ignorar en este ejercicio) y las 3 últimas indican, para cada aminoácido, las 3 bases que lo codifican. A continuación se muestra un extracto del archivo.

```
AAs     = FFLLSSSSYY**CC*WLLLLPPPPHHQQRRRRIIIMTTTTNNKKSSRRVVVVAAAADD
Starts = ---M---------------M------------MMMM---------------M---
Base1  = TTTTTTTTTTTTTTTTCCCCCCCCCCCCCCCCAAAAAAAAAAAAAAAAGGGGGGGGGG
Base2  = TTTTCCCCAAAAGGGGTTTTCCCCAAAAGGGGTTTTCCCCAAAAGGGGTTTTCCCCAA
Base3  = TCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTCAGTC
```

En rojo se representa el aminoácido F (felinalanina) que se codifica con el triplete (en azul): TTT

También contaréis con un fichero llamado ***lista_genes.txt*** que contiene un listado con todos los genes conocidos de esta bacteria. Cada entrada de esta lista contiene información sobre un gen. En particular, un identificador de la proteína que codifica, con la etiqueta *protein_id*, y la localización del gen en el genoma, indicada con *location* y con las posiciones de inicio y fin separadas por doble punto. Cuando el gen está codificado en la hebra negativa, este hecho se señala con la palabra *complement* y entonces las posiciones aparecen entre paréntesis. Por ejemplo, [location=1723677..1724258] es un gen codificado en la hebra positiva y [location=complement(645379..647148)] otro codificado en la negativa.

El ejercicio consiste en lo siguiente:

1. Leer el genoma y almacenar la información en una única cadena (1punto). Esta cadena se utilizará más abajo.

2. Crear un diccionario a partir del fichero *codigo.txt* en el que las claves sean los tripletes y los valores, los aminoácidos. (2 puntos)

3. Construir un fichero multifasta con todos los genes, extrayendo las secuencias del genoma, a partir de los datos de localización del listado.
   Para ello:

   a. Obtener la localización de cada gen a partir de las líneas del fichero *lista_genes.txt* y extraer la subsecuencia correspondiente del genoma

(2puntos)

b. Cuando el gen se encuentra en la hebra negativa, convertir la secuencia a su complementaria e invertir su orden (2 puntos). Utilizar un diccionario para hacer la conversión.

c. Escribir la secuencia a un fichero multifasta. Reutilizar como cabecera de cada entrada FASTA la línea correspondiente del fichero *genes.txt* y trocear la secuencia del gen en líneas de 70 caracteres (1 punto)

4. Construir un fichero en el que se muestre el nombre de cada proteína en una línea, precedido por un "mayor que" (como en la cabecera de un FASTA), y en las siguientes la secuencia de nucleótidos y la secuencia de aminoácidos (habrá, por tanto, que traducir usando el código genético) para cada uno de los genes, haciendo coincidir el primer nucleótido de cada triplete con el aminoácido correspondiente. La secuencia de cada gen debe ir precedida por su nombre (*gene* en las líneas de *lista_genes.txt*) y la de la proteína por su código de acceso, *protein_id*. Las dos secuencias deben estar perfectamente alineadas. (2 puntos)

   Ejemplo del contenido del fichero para dos genes:

```
> dimethyladenosine transferase
ksgA:       ATGATGCAACCTCCCATTAACGTGCATACGCCACACTCCAGC
ADN62042.1: M  M  Q  P  P  I  N  V  H  T  P  H  S  S

> ApaG
apaG:       ATGGAAAACAATCCCAGTTCAAAGATTGAGGTCGCTGTCTCG
ADN62043.1: M  E  N  N  P  S  S  K  I  E  V  A  V  S
```

NOTA 1: La máxima puntuación se dará a ejercicios que sólo lean cada fichero una vez, almacenen la mínima cantidad posible de información en cada paso, extraigan información de cada gen con una única expresión regular y cierren los ficheros que abran.

NOTA 2: El código debe estar ABUNDANTEMENTE comentado.

NOTA 3: Como respuesta al ejercicio, se debe subir al Campus Virtual un fichero comprimido, a ser posible en formato zip, que contenga tres ficheros: el código en python y los ficheros construidos en los puntos 2 y 3.

NOTA 4: Este es un ejercicio individual. Se descontarán al menos 1.5 puntos del total de la nota si se descubre "código compartido". Se recuerda que el plagio no está permitido.