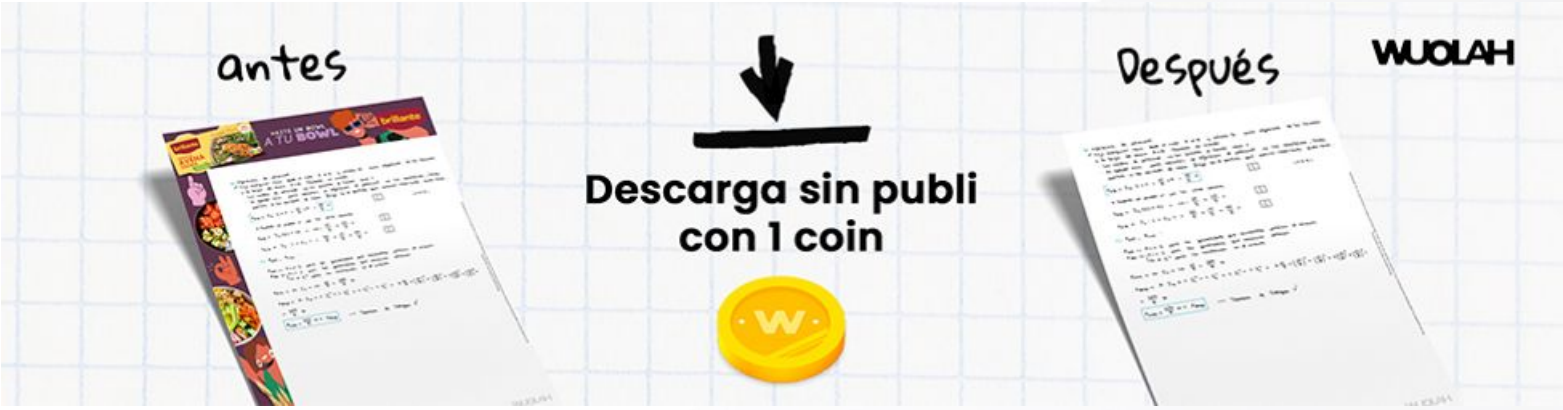
RELACION-EJERCICIOS-ADN.pdf

vaaluna

Fundamentos de Informática Para Biología

1° Grado en Biología

Facultad de Ciencias
Universidad de Granada



Valentina Otero Fernández

RELACIÓN EJERCICIOS ADN:

```
genoma = "CGATATATCCATAG"
kmer = "ATA"

'''1. Escribe una sentencia "if" para comprobar si una subcadena está en un texto mayor, y en
caso afirmativo que sume 1 a un contador. En concreto, dados kmer = "ATA" y
genoma = "CGATATATCCATAG" crea una orden que pueda comparar a partir de cualquier posición
i-ésima del genoma, es decir, no uses valores absolutos si no relativos (variables).

if genoma[i:i+len(kmer)] == kmer:
        c = c+1

2. Escribe en un comentario de código en qué posición deberemos parar la búsqueda de una
ventana de 10 nucleótidos en un genoma de longitud 1000. Esto te indicará dónde empieza el
último "kmer" que busques en tu cadena de ADN original. Pista: fíjate en el gráfico anterior.

# En el ejemplo el genoma tiene longitud 14, la ventana 3 y la última posición de búsqueda es 11
# En un genoma de longitud 1000, la búsqueda de una ventana de 10 nucleótidos debe parar
# en la posición 1000-10 = 990; con carácter general la última posición de búsqueda es
# la longitud del genoma menos la longitud de la ventana

3. Combina el ejercicio 1 dentro de un bucle que funcione como una ventana deslizante.
Prueba con genoma GCGCG y el kmer GCG; la salida debería ser igual a 2.

genoma = "GCGCG"
kmer = "GCG"
c = 0
for i in range(len(genoma)-len(kmer)+1):
        if genoma[i:i+len(kmer)] == kmer:
                c = c+1
print(c)        # el resultado es 2

4. Comprueba el funcionamiento del ejercicio con los siguientes valores reales (Vibrio Cholerae).
Cuidado con los posibles "saltos de línea" al copiar la cadena que aparece abajo.


Genoma =
"ATCAATGATCAACGTAAGCTTCTAAGCATGATCAAGGTGCTCACACAGTTTATCCACAACCTGAGTGGATGACATCAAGATAGGTCGTTGTATCTCCTTCCTCTCGTACTCTCATGACCACGGAAAGATGATCAAGAGAGGATGAT
TTCTTGGCCATATCGCAATGAATACTTGTGACTTGTGCTTCCAATTGACATCTTCAGCGCCATATTGCGCTGGCCAAGGTGACGGAGCGGGATTACGAAAGCATGATCATGGCTGTTGTTCTGTTTATCTTGTTTTGACTGAGACTT
GTTAGGATAGACGGTTTTTCATCACTGACTAGCCAAAGCCTTACTCTGCCTGACATCGACCGTAAATTGATAATGAATTTACATGCTTCCGCGACGATTTACCTCTTGATCATCGATCCGATTGAAGATCTTCAATTGTTAATTCTC
TTGCCTCGACTCATAGCCATGATGAGCTCTTGATCATGTTTCCTTAACCCTCTATTTTTTACGGAAGAATGATCAAGCTGCTGCTCTTGATCATCGTTTC"
kmer = "TGATCA"
c = 0
for i in range(len(Genoma)-len(kmer)+1):
        if Genoma[i:i+len(kmer)] == kmer:
                c = c+1
print(c)        # el resultado es 8

5. Escribe en un comentario del código cuál es el 2-mer más frecuente de la cadena
GATCCAGATCCCCATAC

# el 2-mer más frecuente de la cadena GATCCAGATCCCCATAC es CC con 4 ocurrencias
# GA-->2, AT-->3, TC-->2, CC-->4, CA-->2, AG-->1, TA-->1, AC-->1

6. Implementa un código que resuelva el problema de generar el mapa de frecuencia para un k
determinado. No desesperes, te guiaremos en la tarea.
        a. En primer lugar, crea un diccionario vacío: frecuencias = {}
        b. Recorre todo el texto haciendo "slicing" para el valor k y guarda como clave cada kmer
                y como valor un 0.
        c. Ahora actualiza la asignación directa para el valor 0 que hiciste anteriormente. Como
                pista, piensa que si el kmer está ya en tu diccionario de frecuencias debes sumarle
                uno, en lugar de asignarle un 0. Nota: para saber si un elemento está en un diccionario
                utiliza la orden "if" como "clave in diccionario"
        d. Prueba tu programa con los valores que indicamos antes: CGATATATCCATAG, k = 3

frecuencias = {}
genoma = "CGATATATCCATAG"
k = 3


for i in range(len(genoma)-k+1):
        clave = genoma[i:i+k]
        if clave in frecuencias:
                frecuencias[clave] = frecuencias[clave]+1
        else:
                frecuencias[clave] = 1
print(frecuencias)
```

```
7. Última vuelta de tuerca: queremos dar como salida solamente los kmers más frecuentes
calculados en el paso anterior. Para ello, debemos seguir los siguientes pasos:
        a. Saber cuál es el valor que más se repite. Utiliza la función "max" integrada en Python
        sobre los valores del diccionario (nota: frecuencias.values())
        b. Crea una lista vacía llamada "palabras". Ahora, recorre todas las claves (kmers) de tu
        diccionario y comprueba si su valor es igual al máximo (apartado anterior 7.a). En caso
        afirmativo, añade el kmer a la lista "palabras".

frecuencias = {}
genoma = "CGATATATCCATAG"
k = 3

for i in range(len(genoma)-k+1):
        clave = genoma[i:i+k]
        if clave in frecuencias:
                frecuencias[clave] = frecuencias[clave]+1
        else:
                frecuencias[clave] = 1
print(frecuencias)
M = max(frecuencias.values())
print("El k-mer que más se repite lo hace", M, "veces")
palabras = []
for kmer in frecuencias:
        if frecuencias[kmer] == M:
                palabras.append(kmer)
print("Los k-mer que más se repiten son:", palabras)

8. Recupera el genoma de Vibrio Cholerae y calcula los kmers desde k=2 hasta k=9.
¿Te sorprenden los resultados que has obtenido?
¿Hay alguna relación especial entre los k-mers de "orden" 9?
¡¡Parece que finalmente has encontrado un "DNA-box"!!
'''

genoma =
"ATCAATGATCAACGTAAGCTTCTAAGCATGATCAAGGTGCTCACACAGTTTATCCACAACCTGAGTGGATGACATCAAGATAGGTCGTTGTATCTCCTTCCTCTCGTACTCTCATGACCACGGAAAGATGATCAAGAGAGGATGAT
TTCTTGGCCATATCGCAATGAATACTTGTGACTTGTGCTTCCAATTGACATCTTCAGCGCCATATTGCGCTGGCCAAGGTGACGGAGCGGGATTACGAAAGCATGATCATGGCTGTTGTTCTGTTTATCTTGTTTTGACTGAGACTT
GTTAGGATAGACGGTTTTTCATCACTGACTAGCCAAAGCCTTACTCTGCCTGACATCGACCGTAAATTGATAATGAATTTACATGCTTCCGCGACGATTTACCTCTTGATCATCGATCCGATTGAAGATCTTCAATTGTTAATTCTC
TTGCCTCGACTCATAGCCATGATGAGCTCTTGATCATGTTTCCTTAACCCTCTATTTTTTACGGAAGAATGATCAAGCTGCTGCTCTTGATCATCGTTTC"
frecuencias = []        # construye una lista de 8 diccionarios (k=2 ... k=9)

for i in range(8):
        frecuencias.append({})          # inicia el diccionario para cada k
        k = i+2
        for j in range(len(genoma)-k+1):
                clave = genoma[j:j+k]
                if clave in frecuencias[i]:
                        frecuencias[i][clave] = frecuencias[i][clave]+1
                else:
                        frecuencias[i][clave] = 1
        print("")
        print("La distribución de "+str(k)+"-mer es:")
        print(frecuencias[i])
        print("Hay "+str(len(frecuencias[i]))+" "+str(k)+"-mer distintos en este genoma")
        M = max(frecuencias[i].values())
        print("El "+str(k)+"-mer que más se repite lo hace", M, "veces")
        palabras = []
        for kmer in frecuencias[i]:
                if frecuencias[i][kmer] == M:
                        palabras.append(kmer)
        print("Los "+str(k)+"-mer que más se repiten son:", palabras)
```