

# Introducción a la Bioinformática

TP2

Enrique Alonso

- 1) RETO TI: ¿Podrías buscar un ejemplo de macromoléculas que almacenen información sobre la 'identidad' de una célula?

Los Ácidos Nucleicos son macromoléculas. El ADN es una de ellas y guarda información sobre la identidad del organismo.

- 2) RETO TI: Como vimos el objetivo de este taller es darle una vuelta de tuerca a la Biología, si bien existen muchas herramientas disponibles para analizar datos biológicos, poder construir tus propias herramientas te dará la posibilidad de comprender en profundidad los conceptos y aplicar este conocimiento a nuevas problemáticas. Ahora bien, es importante conocer nuestra herramienta de trabajo, en este caso nuestra PC o teléfono inteligente. ¿Podrías identificar qué parte de la computadora estaremos trabajando?

Mayormente en la memoria donde creo que almacenaremos y luego procesaremos datos sobre cadenas, de aminoácidos, proteínas, ADN, por dar algún ejemplo.

- 3) RETO I: ¿Podés descubrir y anotar el orden en que se ha ejecutado cada operación?

3\*5: Se ejecuta directamente (si bien primero el 3 entra en registro, luego el 5 en registro y luego se hace la multiplicación de ambos)

8/4: Se ejecuta directamente

5+30\*20: Se multiplican 30\*20 y luego se suma 5 al resultado de la primera operación.

(5+30)\*20: se suman 5 +30 y al resultado lo multiplicamos por 20

((4+5)\*2)/5: se suman 4+5, al resultado (9) se lo multiplica por 2 y luego a este resultado se lo divide en 5.

- 4) RETO II: Creá una variable llamada doble, que sea el doble de la suma entre a y b.

doble=2\*(a+b)

- 5) PARA PENSAR: ¿Qué pasa si el contenido de la variable lo\_mismo comienza con mayúsculas?

El resultado de la comparación daría Falso puesto que no es lo mismo Si que si

- 6) RETO III: Digamos que el ADN no es más que un mensaje en clave, que debe ser descifrado o interpretado para la síntesis de proteínas. El mensaje está escrito por una secuencia determinada de 4 nucleótidos distintos representados por las letras A, T, G y C. Dentro de la célula, el mensaje es transportado por otra molécula, el ARN, muy similar al ADN pero con U en vez de T. En este mensaje, cada triplete o grupo de tres letras del ARN se denomina codón, y cada aminoácido de las proteínas está codificado por uno o varios codones. Así por ejemplo el codón 'AUG' codifica para el aminoácido Metionina, el codón 'AAA' para Lisina, el codón 'CUA' para Leucina, etc. ¿Podrías escribir una cadena de ARN que codifique para el péptido (una cadena corta de aminoácidos) 'Met-Lis-Lis-Lis-Leu-Leu-Met' combinando las variables met = 'AUG', lis = 'AAA' y leu = 'CUA' utilizando operadores matemáticos?

En Python se podría hacer con el operador suma, pero porque opera como concatenador, entonces:

```
met = 'AUG'
```

```
lis = 'AAA'
```

```
leu = 'CUA'
```

```
print (met+lis+lis+lis+leu+leu+met)
```

Esta ultima línea imprimirá: AUGAAAAAAAAACUACUAAUG

- 7) RETO IV: ¿Cadenas? ¿letras? Si hablamos de cadenas y letras en Biología, lo primero que se nos viene a la cabeza son las macromoléculas. Como bien sabemos, el ADN es un mensaje en clave que guía la síntesis de proteínas. Este mensaje está escrito por una secuencia determinada de 4 nucleótidos distintos representados por las letras A, T, G y C. El contenido de C y G (es decir el porcentaje de CG) presente en el ADN de un organismo es una característica distintiva: por ejemplo las Actinobacterias tienen un contenido característicamente más alto de CG que otros organismos. Ahora, contar la cantidad de C y G en una cadena de ADN larguísima a mano puede ser un verdadero tedio ¿Podrías crear un programa que calcule el porcentaje de C y G de una cadena dada de ADN?

```
1 def porcentajeCyG(cadenaADN):
2     nucleotidos = list(cadenaADN)
3     totalNucleotidos=len(nucleotidos)
4     totalCG=0
5
6     for cadaNucleotido in nucleotidos:
7         if cadaNucleotido in ["C", "G"]:
8             totalCG=totalCG+1
9
10    return totalCG/ totalNucleotidos*100
11
12
13 cadenaADNEjemplo="AUGAAAAAAACUACUAAUG"
14 print ("El porcentaje de CyG es : ",porcentajeCyG(cadenaADNEjemplo) )
15
```

- 8) RETO V: La Asombrosa Maravillosa es nuestra valiente superheroína. Sus poderes son producto de mutaciones en un gen muy común, cuya secuencia en la mayoría de las personas es 'ATGGAAGTTCGAATCGAAGTTGGC'. A diferencia de nosotros, el gen mutado de la Asombrosa Maravillosa incluye la secuencia 'GTTTGTGGTTG' en su interior. La Asombrosa Maravillosa adquirió sus poderes al beber Jugo Vencido. El primer sorbo de esta poción prohibida causa el cambio de todas las citosinas (C) por timinas (T). El siguiente sorbo cambia todas las adeninas (A) por guaninas (G). El tercer sorbo cambia las citosinas (C) por adeninas (A). El cuarto sorbo... puede ser mortal. ¿Podés escribir un programa que nos diga cuántos sorbos de Jugo Vencido debe beber un portador del gen normal, para ganar los poderes de la Asombrosa Maravillosa

```
1 def tomasParaObtenerElGenMutado(secuenciaADN):
2     #Retorna la cantidad de tomas necesarias para obtener el gen mutado
3     #Devuelvo 9999 cuando llego a la cuarta toma porque puede ser mortal
4     genMutado= 'GTTTGTGGTTG'
5
6     if genMutado in secuenciaADN:
7         return 0
8
9     for toma in range(1, 5):
10
11         if toma==1:
12             secuenciaADN=secuenciaADN.replace("C","T")
13
14         if toma==2:
15             secuenciaADN=secuenciaADN.replace("A","G")
16
17         if toma==3:
18             secuenciaADN=secuenciaADN.replace("C","A")
19
20         if toma>=4:
21             return 9999
22
23         if genMutado in secuenciaADN:
24             return toma
25
26
27
28
29 secuenciaEjemplo='ATGGAAGTTCGAATCGAAGTTGGC'
30
31 print (tomasParaObtenerElGenMutado(secuenciaEjemplo))
```

- 9) RETO VI: ¿Se te ocurre qué operadores podrías usar para las listas?

Concatenación (append), longitud(len), quitar un elemento (remove), agregar un elemento (add), reversar (reverse), ordenar(sort) y más.

- 10) RETO VII: Ya que encontramos el espécimen de rana con pelo en Marte, nos gustaría contrastar sus características con las ranas terrestres. Sabiendo que el gen de la proteína diminuta es 'ATGGAAGTTGGAATCCAAGTTGGA' y el gen de una proteína similar de rana terrestre es 'ATGGAAGTTAATGGAAGTTGGAGGAGA' ¿podés crear un programa que compare la longitud de ambos genes y según cuál sea más grande nos imprima un mensaje informándonos el resultado?

```
1 def compararGenes(secuencia1 , secuencia2):
2     #Retorna la secuencia que es mas larga
3
4     return max(secuencia1,secuencia2)
5
6
7
8 s1= 'ATGGAAGTTGGAATCCAAGTTGGA'
9 s2= 'ATGGAAGTTAATGGAAGTTGGAGGAGA'
10
11
12 print (compararGenes(s1,s2))
13
```

- 11) RETO VIII: Si nos ponemos un poco más estrictos, y siguiendo con el tema de los clones de bacterias, el programa que creamos antes tiene algunas fallas 'numéricas': en cada vuelta de división celular binaria se generarán dos clones, no uno. ¿Podrías escribir un programa que imprima '¡Somos 2 clones nuevos!' en cada una de 20 vueltas?

```
1 for i in range(0,20):
2     print('¡Somos 2 clones nuevos!')
```

- 12) RETO IX: Si ahora queremos hacer nuestro programa un poco más estricto, por cada vuelta deberíamos sumar el total de células que tenemos e imprimir ese número en el mensaje. Entonces, por ejemplo, como en la primera vuelta tenemos dos células, imprimimos como mensaje '¡Somos 2 clones!', pero en la segunda vuelta serán en total 4 células y el mensaje a imprimir debería ser '¡Somos 4 clones!'. ¿Podrías escribir esta modificación del programa?

```
1 total=0
2 for i in range(0,20):
3     total=total+2
4     print("¡Somos ", total, " clones!")
```