Tarea 1

Equipo: Los bípedos

Francisco Contreras Ibarra , **José Ethan Ortega González** y **Alvaro Ramírez López**

---

1. Expresiones regulares

1.1. Ejercicio 1

Realiza los siguientes ejercicios sobre expresiones regulares.

1. ¿Describe con lenguaje natural la siguiente expresión regular `r'[CG]{3}(TA){4,5}(CG)'` . Da un par de ejemplos que la cumplan (El alfabeto es $\Sigma = \{A, C, G, T\}$).
2. A continuación se presentan 10 secuencias hipotéticas. Diseña una expresión regular que detecte regiones codificantes válidas y específica cuales de las secuencias la cumplen:

```
0. ATATATACATACTGGTAATGGGCGCGCGTGTGTTAAGTTCTGTTGTAGGGGTGATTAGGGGCG
1. GGCCCACACCCCACACCAATATATGTGGTGTGGGCTCCACTCTCTCGCGCTCGCGCTGGGGAT
2. ATAAGGTGTGTGGGCGCGCCCCGCGCGCGCGTTTTTTCGCGCGCCCCCGCGCGCGCGCGCGCG
3. GGCGCGGGACGCGGCGGCGGATCCCGATCCGTGCGTCAATACTATTATGGCCAGATAGAATAA
4. GTGCTGCTGCGGCGCCCACACCTATTATCTCTCTCTCTCTGCCTCTCCACCTCGGGGCTTAAT
5. GCGCTGCTGCTGGCTCGATGGGCGCGTGCGTCGTAGCTCGATGCTGGCTCGAGCTGTAATCTT
6. GGCGCTCGCTCGGATGCGCGGCCGGGCTCTCTGCTCGCGCTCGCTTCGCGCTCGTGACCGCTG
7. AATTGGTGCGCGCTCGCGCACACACAGAGAGAGGGTTTATATAGGATGATATATCCACATTGG
8. ATGCTGCTGCTGGCTCTGCTTGCGCTCTGCTCGCTGGGGTGTGTGTGCCGCGCGCTGCTGCTC
9. GCTGGGCTCGCTCGATGCGCGCGGGCGCGCGACCGCGGACGGCGTCGCTGCTAAATGGGCTTC
```

Debes entregar una lista con el índice de la línea en las que hay una región codificante válida, ejemplo, si en las líneas 0, 1 y 5 hay, entonces el resultado debería ser:

```
[0, 1, 5, ...]
```

Solución

1. Si descomponemos la expresión regular en partes, podemos describirla de la siguiente manera:
   a. `[CG]{3}` : checa que se tenga la letra *C* o la letra *G* 3 veces seguidas, en cualquier posición.
   b. `(TA){4,5}` : checa que se tenga la letra *T* y después la letra *A* de 4 a 5 veces seguidas.
   c. `(CG)` : Checa que se tenga la letra *C* y después la letra *G*. Las dos letras se encierran en un grupo.

   Podemos juntar las descripciones para obtener la expresión regular completa. Algunas cadenas que acepta son las siguientes:
   - `CGCTATATATACG`
   - `GCCTATATATATACG`

   Se puede utilizar la página regex101 para verificar que estas cadenas cumplen con la expresión regular.

Tarea 1

Entrega 11/03/2024

Números de cuenta
Francisco Contreras Ibarra
316083786
José Ethan Ortega González
316088327
Alvaro Ramírez López
316276355

Correos
Francisco Contreras Ibarra
franciscoc.ibarra@ciencias.unam.mx
José Ethan Ortega González
ethan@ciencias.unam.mx
Alvaro Ramírez López
alvaro@ciencias.unam.mx

Solución

2. La solucion propuesta a este ejercicio se puede ver implementada y lista para ejecutar en el archivo **t1_losBipedos.py**, o como se puede ver a continuacion

```python
import math, random, re

# Solucion Ejercicio 2
lista_secuencias = [
    'ATATATACATACTGGTAATGGGCGCGCGTGTGTTAAGTTCTGTTGTAGGGGTGATTAGGGGCG',
    'GGCCCACACCCCACACCAATATATGTGGTGTGGGCTCCACTCTCTCGCGCTCGCGCTGGGGAT',
    'ATAAGGTGTGTGGGCGCGCCCCGCGCGCGCGTTTTTTCGCGCGCCCCCGCGCGCGCGCGCGCG',
    'GGCGCGGGACGCGGCGGCGGATCCCGATCCGTGCGTCAATACTATTATGGCCAGATAGAATAA',
    'GTGCTGCTGCGGCGCCCACACCTATTATCTCTCTCTCTCTGCCTCTCCACCTCGGGGCTTAAT',
    'GCGCTGCTGCTGGCTCGATGGGCGCGTGCGTCGTAGCTCGATGCTGGCTCGAGCTGTAATCTT',
    'GGCGCTCGCTCGGATGCGCGGCCGGGCTCTCTGCTCGCGCTCGCTTCGCGCTCGTGACCGCTG',
    'AATTGGTGCGCGCTCGCGCACACACAGAGAGAGGGTTTATATAGGATGATATATCCACATTGG',
    'ATGCTGCTGCTGGCTCTGCTTGCGCTCTGCTCGCTGGGGTGTGTGTGCCGCGCGCTGCTGCTC',
    'GCTGGGCTCGCTCGATGCGCGCGGGCGCGCGACCGCGGACGGCGTCGCTGCTAAATGGGCTTC']

def ejercicio_2(lista_secuencias):
    L = []
    r = re.compile('(ATG|TTG|GTG)([ACTG]{3})+(TAG)')
    L = [i for i, item in enumerate(lista_secuencias) if re.search(r, item)]
    return L

print(ejercicio_2(lista_secuencias))
```

Figura 1: Diseño de una expresion regular que detecta regiones codificantes.

2. Probabilidad y estadística

2.1. Ejercicio 1

En el archivo `promotores.txt` se encuentra la lista de secuencias tomadas del genoma de Vitis vinifera y cada una de las secuencias puede que tenga alguno de las diferentes formas en las que se ha encontrado el promotor GATA: Este se caracteriza por tener una región fija con la identidad GATA pero estar precedido ya sea por A o T, y seguido de A o G.

Deseamos estudiar estas regiones en función del promotor GATA y por lo tanto lo primero que deseamos es saber cuántas veces aparecen los promotores en cada región.

- Haz un script en python que lea todas las líneas en promotores.txt y cuya salida sen un archivo llamado `promotores_conteo.txt` . Este debe contener en cada línea la siguiente información: la secuencia analizada y la cantidad de promotores encontrados. Un ejemplo de la salida se muestra en la figura 1.
- Haz un boxplot con la distribución de cada uno de los promotores
- ¿Cuál es la media y desviación estándar de cada promotor?

Solución

Este es el codigo propuesto de la solucion:

```python
import re

promotores = ['AGATAG', 'TGATAG', 'AGATAA', 'TGATAA']

def leer_archivo(archivo):
    with open(archivo, 'r') as file:
        for line in file:
            yield line.strip()

def ejercicio_2_1(archivo, promotores):
    L = []
    for line in leer_archivo(archivo):
        contador = 0
        identificador = re.split(r'\s+', line)[0]
        line = re.split(r'\s+', line)[1]
        for _ in re.finditer('|'.join(promotores), line):
            contador += 1
        L.append(identificador+" : " + str(contador))
    return L

lista = ejercicio_2_1("promotores.txt", promotores)
  for item in lista:
    print(item)
```

Figura 2: Analisis del archivo **promotores.txt**

2.2. Ejercicio 2

Haz un script en python que reciba como parámetro un entero que determinará la cantidad de iteraciones para el siguiente algoritmo

Data: m iteraciones
Result: Número flotante x

$i \leftarrow 1$;
$D \leftarrow 0$;
while $i < M$ **do**
    $i \leftarrow i + 1$;
    $x \leftarrow -1 \leq uniform() \leq 1$;
    $y \leftarrow -1 \leq uniform() \leq 1$;
    $d \leftarrow \sqrt{x^2 + y^2}$;
    if $d \leq 1$ **then**
        $D \leftarrow D + 1$;
    end
$x \leftarrow 4 \cdot \frac{D}{i}$;
return x;

Algoritmo 1: Algoritmo misterio.

Tip: Revisa la documentación del paquete `random`, particularmente de la biblioteca de `numpy`. ¿Qué está calculando el algoritmo 1?

Solución

El script de python se presenta a continuación:

```python
import numpy as np
import sys


def calcula_pi(iteraciones: int) -> float:
    numeros_dentro = 1

    for _ in range(iteraciones):
        x = np.random.uniform(-1, 1)
        y = np.random.uniform(-1, 1)

        if np.sqrt((x**2) + (y**2)) <= 1:
            numeros_dentro += 1

    return 4 * (numeros_dentro / iteraciones)


def main():
    if len(sys.argv) != 2:
        print("uso: python script.py <iteraciones>")
        exit(1)

    try:
        print(calcula_pi(int(sys.argv[1])))
    except ValueError:
        print("error: ingresa un número")


main()
```

Figura 3: El código equivalente del pseudocódigo escrito en python.

El Algoritmo 1 está estimando el valor de pi utilizando el *método Monte Carlo*, una técnica que utiliza números aleatorios para simular sistemas complejos. [1]

La idea es generar un punto (x, y) al azar en el rango $[-1, 1]$ y calcular la distancia del origen a este punto, si la distancia es menor a 1 (es decir, que el punto está dentro del circulo) se incremente un contador, así hasta cumplir la cantidad de iteraciones.

La razón por la que este método funciona es porque la probabilidad de que un punto caiga dentro del círculo es proporcional a su área, que es pi. Por lo tanto, si contamos la cantidad de veces que el punto cae dentro del círculo, podemos estimar el valor de pi.

2.3. Ejercicio 3

Utilizando los siguientes datos de sensibilidad (93 por ciento) y especificidad (99 por ciento) reportados para cierta prueba rápida de antígeno para detectar la infección por virus SARS-COV2 y considerando una prevalencia actual de COVID en México estimada a partir del promedio de casos nuevos observados a lo largo de 2 semanas de 16000 casos activos respecto a una población total de 120000000 de habitantes encuentra:

1. ¿Cuál es la probabilidad de que si uno de ustedes se realiza una prueba rápida de este tipo y ésta resulta positiva ustedes en realidad sean portadores del virus SARS-COV2?
2. ¿Cuál es la probabilidad de que si la prueba resulta negativa ustedes en realidad no sean portadores del virus SARS-COV2?
3. Entre marzo y junio de 2021 se tuvo un promedio de nuevos contagios semanales de alrededor de 3000 casos, por lo que a lo largo de dos semanas se tendría una prevalencia aproximada de 6000 casos activos respecto a 120000000 de habitantes.
   a. Calcula las probabilidades referidas en los dos incisos anteriores pero considerando este nuevo dato de prevalencia.

b. ¿Qué puedes concluir respecto a las probabilidades obtenidas en ambos escenarios?
c. ¿consideras que en el caso de las pruebas de detección de COVID es necesaria una mayor sensibilidad o una mayor especificidad? Justifica tu respuesta.

Solución

1. Podemos definir los siguientes eventos:
   - $A = \{\text{la prueba sale positiva}\}$
   - $N = \{\text{la prueba sale negativa}\}$
   - $S = \{\text{la persona está sana}\}$
   - $E = \{\text{la persona está enferma}\}$

   Tenemos los siguientes datos:
   - Sensibilidad: $(P(A|E)) = .93$
   - Especificidad: $(P(N|S)) = .99$
   - Habitantes $= 120000000$
   - Casos activos $= 16000$

   Dados los anteriores datos tenemos que:

$$P(A) = \frac{\text{Casos activos}}{\text{Habitantes}} = \frac{16000}{120000000} = 0.00013$$

   Mediante el Teorema de Bayes tenemos que:

$$\begin{aligned} P(E|A) &= \frac{P(A|E) \cdot P(A)}{P(A|E) \cdot P(A) + (1 - P(N|S)) \cdot (1 - P(A))} \\ &= \frac{0.93 \cdot 0.00013}{(0.93 \cdot 0.00013) + ((1 - 0.99) \cdot (1 - 0.00013))} \\ &= \frac{0.0001209}{0.0001209 + (0.01 \cdot 0.99987)} \\ &= \frac{0.0001209}{0.0001209 + 0.0099987} \\ &= \frac{0.0001209}{0.0101196} \\ &= 0.01194711253 \end{aligned}$$

   Entonces la probabilidad de que en realidad estemos enfermos es de 0.01194711253.
2. Teniendo los mismos datos que en el inciso anterior mediante el Teorema de Bayes tenemos que:

$$\begin{aligned} P(S|N) &= \frac{P(N|S)(1 - P(N|S))}{P(N|S)(1 - P(N|S)) + (1 - P(A|E))P(A)} \\ &= \frac{0.99 \cdot (1 - 0.00013)}{0.99 \cdot (1 - .00013) + (1 - 0.93) \cdot 0.00013} \\ &= \frac{0.99 \cdot 0.99987}{0.99 \cdot 0.99987 + 0.07 \cdot 0.00013} \\ &= \frac{0.9898713}{0.9898804} \\ &= 0.999990807 \end{aligned}$$

   Entonces la probabilidad de que en realidad no estemos enfermos es de 0.999990807.
3. Las respuestas son las siguientes

   a. A continuación, calculamos de nuevo los dos incisos anteriores.

- ¿Cuál es la probabilidad de que si uno de ustedes se realiza una prueba rápida de este tipo y ésta resulta positiva ustedes en realidad sean portadores del virus SARS-COV2?

$$P(A) = \frac{\text{Casos activos}}{\text{Habitantes}} = \frac{6000}{120000000} = 0.00005$$

Mediante el Teorema de Bayes tenemos que:

$$\begin{aligned} P(E|A) &= \frac{0.93 \cdot 0.00005}{(0.93 \cdot 0.00005) + (1 - 0.99) \cdot (1 - 0.00005)} \\ &= \frac{0.0000465}{0.0000465 + 0.0099995} \\ &= \frac{0.0000465}{0.010046} \\ &= 0.004628707943 \end{aligned}$$

- ¿Cuál es la probabilidad de que si la prueba resulta negativa ustedes en realidad no sean portadores del virus SARS-COV2?

Mediante el Teorema de Bayes tenemos que:

$$\begin{aligned} P(S|N) &= \frac{0.99 \cdot (1 - 0.00005)}{0.99 \cdot (1 - 0.00005) + 0.00005 \cdot (1 - 0.93)} \\ &= \frac{0.9899505}{0.989954} \\ &= 0.9999964645 \end{aligned}$$

b. Como podemos ver, al disminuir los casos activos de 16000 a 6000 hizo que el resultado de la prevalencia se redujera, lo cual provocó que la probabilidad de realizar una prueba rápida donde el resultado sea positivo en realidad sea portador del virus SARS-COV2 se redujera y la probabilidad de que si la prueba resulta negativa y en realidad no sean portadores del virus SARS-COV2 se aumentara.

Por lo tanto, es más probable que no seamos portador del virus cuando la cantidad de casos activos disminuye y la cantidad de habitantes se mantiene igual.

c. Con la sensibilidad se tiene la probabilidad de que, al realizar una prueba, el resultado sea positivo cuando sí se tenga la enfermedad, por lo que entre mayor sea la sensibilidad más enfermos serán diagnosticados adecuadamente.

Esto provoca que los falsos negativos (el resultado de una prueba que indica que una persona no tiene cierta enfermedad o afección cuando en realidad la tiene) se reduzcan.

Por otro lado, la especificidad es la probabilidad de que, al realizar una prueba, el resultado sea negativo donde no se tenga la enfermedad, por lo que entre mayor sea la especificidad más sanos serán diagnosticados adecuadamente.

Esto provoca que los falsos positivos (el resultado de una prueba que indica que una persona está afectada o que tiene cierta mutación génetica cuando verdaderamente no está afectada o no tiene la mutación) se reduzcan.

Por lo anterior, consideramos que es mejor una mayor sensibilidad ya que permite que reduzcan los falsos negativos lo cual permitirá a la persona infectada poder tomar el tratamiento o medidas necesarias lo antes posible.

2.4. Ejercicio 4

Descarga del GenBank la secuencia del virus del SARS-Cov2 y encuentra:

1. El número de ORF's dentro de la misma, sin importar su longitud, pero debe de tratarse de ORF viables.
2. El número de ORF's cuya longitud nos permita afirmar con una probabilidad menor a 0.05 que se trata de ORF no espurios.
3. El número de ORF's cuya longitud nos permita afirmar con una probabilidad menor a 0.01 que se trata de ORF no espurios.
4. Investiga cuál es el número de ORF real y contrástalo con los números encontrados en las dos preguntas anteriores, ¿cuál de los dos números es más cercano al número real?
5. Tu respuesta debe incluir, el código utilizado para responder la pregunta.

Si quieren leer *La Biblioteca de Babel* lo pueden hacer en esta liga

Solución

1. El numero de ORF's viables sin importar su tamaño es de 19.
2. El número de ORFs cuya longitud es mayor a 62 nt y nos permiten decir con probabilidad de error menor a 0.05 que son no espurios son solamente 11.
3. El número de ORFs cuya longitud es mayor a 62 nt y nos permiten decir con probabilidad de error menor a 0.01 que son no espurios son solamente 9.
4. El número real de ORF presentes en el coronavirus es de 11, por lo cual el número predicho con 0.05 probabilidad de error es el más acertado, en este caso exactamente igual al número de ORFs reportados en la literatura.

   El GenBank usado esta en las referencias.

```python
# Importamos las bibliotecas necesarias
from Bio import SeqIO
import re, math

# cargando genoma de SARS-CoV2
genome = SeqIO.read("sequence.fasta", "fasta")
sequence = str(genome.seq)
r_sequence = str(genome.seq.reverse_complement())

# Numero de ORFs
patron = r'ATG(?:(?!TAA|TAG|TGA)...)*(?:TAA|TAG|TGA)'
orfs = re.findall(patron, sequence)
print(str(len(orfs)))

# ORFs con probabilidad < 0.05 de ser no espurios
k = math.log(0.05, 61/64)
orfs_5 = []
for orf in orfs:
    if len(orf) > int(k):
        orfs_5.append(orf)
print(str(len(orfs_5)))

# ORFs con probabilidad < 0.01 de no ser espurios
k = math.log(0.01, 61/64)
orfs_1 = []
for orf in orfs:
    if len(orf) > int(k):
        orfs_1.append(orf)
print(str(len(orfs_1)))
```

Bibliografía

[1] V. Sharma, «Estimating PI using Monte Carlo methods - the modern scientist - medium», *Medium*, abr. 2023, [En línea]. Disponible en: https://medium.com/the-modern-scientist/estimating-pi-using-monte-carlo-methods-dbdf26c888d6

[2] «Severe acute respiratory syndrome coronavirus 2 genome assembly ASM985889v3». [En línea]. Disponible en: https://www.ncbi.nlm.nih.gov/datasets/genome/GCF_009858895.2/

[3] C. Sac, «¿Qué son sensibilidad y especificidad?». [En línea]. Disponible en: https://www.sac.org.ar/cuestion-de-metodo/que-son-sensibilidad-y-especificidad/

[4] [En línea]. Disponible en: https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-genetica/def/resultado-positivo-falso

[5] [En línea]. Disponible en: https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/resultado-negativo-falso-de-una-prueba

[6] [En línea]. Disponible en: https://www.cigna.com/es-us/knowledge-center/hw/sensibilidad-y-especificidad-sts14487