

```
# Función para analizar la secuencia de ADN
def analizar_secuencia(secuencia, tamaño_ventana=10000):
    """
    Divide la secuencia en ventanas de un tamaño dado y calcula los porcentajes de GC y
    AT para cada ventana.
    """
    regiones = [] # Lista para almacenar los resultados del análisis de ventanas
    for inicio in range(0, len(secuencia) - tamaño_ventana + 1, tamaño_ventana):
        ventana = secuencia[inicio:inicio + tamaño_ventana]
        conteo_g = ventana.count("G")
        conteo_c = ventana.count("C")
        conteo_a = ventana.count("A")
        conteo_t = ventana.count("T")
        contenido_gc = (conteo_g + conteo_c) / len(ventana) * 100
        contenido_at = (conteo_a + conteo_t) / len(ventana) * 100
        regiones.append({"GC%": contenido_gc, "AT%": contenido_at})
    return regiones
```

Esta función analiza una secuencia de ADN en fragmentos (o "ventanas") de un tamaño específico y calcula los porcentajes de bases nitrogenadas GC y AT en cada fragmento.

## 1. Definición de la función

```
def analizar_secuencia(secuencia, tamaño_ventana=10000):
```

- **secuencia:** Entrada que contiene la secuencia de ADN (cadena de caracteres formada por las letras "A", "T", "G", "C").
- **tamaño\_ventana:** Tamaño de cada ventana o fragmento en el que se dividirá la secuencia. Tiene un valor por defecto de 10,000 nucleótidos (que puede ser ajustado dependiendo del genoma a analizar).

## 2. Inicialización

```
regiones = [] # Lista para almacenar los resultados del análisis de ventanas
```

- Se crea una lista vacía `regiones` para almacenar los resultados (porcentaje de GC y AT) de cada ventana.

## 3. División en ventanas

```
for inicio in range(0, len(secuencia) - tamaño_ventana + 1, tamaño_ventana):
```

- `range(0, len(secuencia) - tamaño_ventana + 1, tamaño_ventana):` Itera a lo largo de la secuencia de ADN en pasos del tamaño de ventana.
  - `inicio`: Índice donde comienza cada ventana.
  - La secuencia se procesa hasta que quede un fragmento del tamaño de la ventana.

## 4. Extracción de la ventana

```
ventana = secuencia[inicio:inicio + tamaño_ventana]
```

- Extrae un fragmento de la secuencia desde el índice `inicio` hasta `inicio + tamaño_ventana`.

## 5. Conteo de bases

```
conteo_g = ventana.count("G")
conteo_c = ventana.count("C")
conteo_a = ventana.count("A")
conteo_t = ventana.count("T")
```

- `ventana.count("X")`: Cuenta cuántas veces aparece una base específica ("G", "C", "A", "T") en la ventana.

## 6. Cálculo de porcentajes

```
contenido_gc = (conteo_g + conteo_c) / len(ventana) * 100
contenido_at = (conteo_a + conteo_t) / len(ventana) * 100
```

- **GC%**: Proporción de "G" y "C" en la ventana.
- **AT%**: Proporción de "A" y "T" en la ventana.
- Ambos valores están normalizados como porcentajes ( $\times 100$ ).

## 7. Almacenamiento de resultados

```
regiones.append({"GC%": contenido_gc, "AT%": contenido_at})
```

- Se guarda un diccionario con los valores de GC% y AT% en la lista `regiones`.

## 8. Devolución de resultados

```
return regiones
```

- La función devuelve una lista donde cada elemento corresponde a los porcentajes calculados para una ventana de la secuencia.

## Ejemplo de uso

Supongamos que tenemos esta secuencia:

```
secuencia = "ATGCGCGCGCGTATATATATATATATATGC" * 500 # Secuencia de 15,000 nucleótidos
resultados = analizar_secuencia(secuencia, tamaño_ventana=5000)
print(resultados)
```

Salida esperada:

```
[{'GC%': 40.0, 'AT%': 60.0}, {'GC%': 50.0, 'AT%': 50.0}, {'GC%': 45.0, 'AT%': 55.0}]
```