

```
# cargamos los datos
!pip install Bio
from Bio import SeqIO

# Assuming the file is in the /content/ directory
for record in SeqIO.parse("/content/gene_sars_corv2.fna", "fasta"):
    print(record.id)
    print(repr(record.seq))
    print(len(record))
```

Requirement already satisfied: Bio in /usr/local/lib/python3.11/dist-packages (1.7.1)
 Requirement already satisfied: biopython>=1.80 in /usr/local/lib/python3.11/dist-packages (
 Requirement already satisfied: gprofiler-official in /usr/local/lib/python3.11/dist-package
 Requirement already satisfied: mygene in /usr/local/lib/python3.11/dist-packages (from Bio)
 Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (from Bio)
 Requirement already satisfied: pooch in /usr/local/lib/python3.11/dist-packages (from Bio)
 Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from Bi
 Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from Bio) (
 Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (from biopy
 Requirement already satisfied: biothings-client>=0.2.6 in /usr/local/lib/python3.11/dist-pa
 Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-pac
 Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (fro
 Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (f
 Requirement already satisfied: platformdirs>=2.5.0 in /usr/local/lib/python3.11/dist-packag
 Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (
 Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-p
 Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (fro
 Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.11/dist-package
 Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.11/dist-package
 Requirement already satisfied: httpx>=0.22.0 in /usr/local/lib/python3.11/dist-packages (fr
 Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from py
 Requirement already satisfied: anyio in /usr/local/lib/python3.11/dist-packages (from httpx
 Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (fr
 Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.11/dist-packages (
 Requirement already satisfied: sniffio>=1.1 in /usr/local/lib/python3.11/dist-packages (fro
 NC_000019.10:c38930763-38915266
 Seq('GAAGTGCCGCCTTGTCCCGGTCCAAGATGGCTGCGTCCATGGCGCGGCGCTTG...AAA')
 15498
 NW_014040929.1:c340435-324938
 Seq('GAAGTGCCGCCTTGTCCCGGTCCAAGATGGCTGCGTCCATGGCGCGGCGCTTG...AAA')
 15498
 NC_060943.1:c41734732-41719237
 Seq('GAAGTGCCGCCTTGTCCCGGTCCAAGATGGCTGCGTCCATGGCGCGGCGCTTG...AAA')
 15496

```
# cuantos caracteres hay en el genoma humano

from Bio import SeqIO
# Assuming the file is in the /content/ directory
for record in SeqIO.parse("/content/gene_sars_corv2.fna", "fasta"):
    print(len(record.seq))
```

15498
 15498
 15496

 Generar

busca mutaciones

Cerrar < 1 of 1 > [Deshacer cambios](#) [Usa el código con precaución](#)

```
# vamos abuscar lo que nos interesa en busca de mutaciones

from Bio import SeqIO
from Bio.Seq import Seq

# Assuming the file is in the /content/ directory
for record in SeqIO.parse("/content/gene_sars_corv2.fna", "fasta"):
    # Get the sequence
    sequence = record.seq

    # Find mutations (example: looking for changes from 'A' to 'T')
    mutations = []
    for i in range(len(sequence)):
        if sequence[i] == 'A': # Example condition: looking for A
            if i > 0 and sequence[i-1] == 'T': # Example mutation condition: A preceded by T
                mutations.append((i, sequence[i], sequence[i-1])) # Store the position and change
            elif i < len(sequence) - 1 and sequence[i+1] == 'T': # Another mutation condition:
                mutations.append((i, sequence[i], sequence[i+1]))

    # Print the found mutations
    if mutations:
        print("Mutations found:")
        for mutation in mutations:
            print(f"Position {mutation[0]+1}: {mutation[1]} changed from {mutation[2]}") #mutation[
    else:
        print("No mutations found based on this criteria.")
```



```

Position 10377: A changed from T
Position 10399: A changed from T
Position 10424: A changed from T
Position 10510: A changed from T
Position 10555: A changed from T
Position 10566: A changed from T
Position 10588: A changed from T
Position 10609: A changed from T
Position 10647: A changed from T
Position 10665: A changed from T
Position 10668: A changed from T
Position 10749: A changed from T
Position 10766: A changed from T
Position 10774: A changed from T
Position 10779: A changed from T
Position 10783: A changed from T
Position 10790: A changed from T
Position 10838: A changed from T
Position 10846: A changed from T
Position 10866: A changed from T
Position 10929: A changed from T
Position 10932: A changed from T
Position 10981: A changed from T
Position 11005: A changed from T
Position 11015: A changed from T
Position 11017: A changed from T
Position 11042: A changed from T
Position 11073: A changed from T
Position 11082: A changed from T
Position 11088: A changed from T

```

```
# los codones afectados son
```

```
# Assuming the file is in the /content/ directory
```

```
for record in SeqIO.parse("/content/gene_sars_corv2.fna", "fasta"):
```

```
    # Get the sequence
```

```
    sequence = record.seq
```

```
# Find codons affected by mutations (example: looking for changes from 'A' to 'T')
```

```
mutations = []
```

```
for i in range(0, len(sequence) - 2, 3): # Iterate through codons
```

```
    codon = sequence[i:i+3]
```

```
    for j in range(3): # Iterate over each nucleotide
```

```
        if codon[j] == 'A': # Example: Looking for A's
```

```
            if j > 0 and codon[j-1] == 'T': # Example mutation condition: A preceded by T
```

```
                mutations.append((i+j, codon[j], codon[j-1], codon)) #Store position, changed
```

```
            elif j < 2 and codon[j+1] == 'T': # Example mutation condition: A followed by T
```

```
                mutations.append((i+j, codon[j], codon[j+1], codon))
```

```
# Print the found mutations with codon context
```

```
if mutations:
```

```
    print("Mutations found:")
```

```
    for mutation in mutations:
```

```
        print(f"Position {mutation[0]+1}: {mutation[1]} changed from {mutation[2]} within c
```

```
else:
```

```
    print("No mutations found based on this criteria.")
```



```

Position 13484: A changed from T within codon TAA
Position 13500: A changed from T within codon GTA
Position 13527: A changed from T within codon GTA
Position 13528: A changed from T within codon ATT
Position 13549: A changed from T within codon ATG
Position 13630: A changed from T within codon ATT
Position 13641: A changed from T within codon GTA
Position 13719: A changed from T within codon CTA
Position 13732: A changed from T within codon ATC
Position 13735: A changed from T within codon ATG
Position 13782: A changed from T within codon CTA
Position 13793: A changed from T within codon AAT
Position 13825: A changed from T within codon ATC
Position 13909: A changed from T within codon ATC
Position 13913: A changed from T within codon TAT
Position 13922: A changed from T within codon AAT
Position 13924: A changed from T within codon ATA
Position 13926: A changed from T within codon ATA
Position 13930: A changed from T within codon ATC
Position 13959: A changed from T within codon GTA
Position 13960: A changed from T within codon ATC
Position 14032: A changed from T within codon ATC
Position 14109: A changed from T within codon CTA
Position 14120: A changed from T within codon AAT
Position 14128: A changed from T within codon ATG
Position 14224: A changed from T within codon ATA
Position 14226: A changed from T within codon ATA
Position 14263: A changed from T within codon ATC
Position 14297: A changed from T within codon TAG
Position 14299: A changed from T within codon ATG
Position 14312: A changed from T within codon TAT
Position 14375: A changed from T within codon CAT
Position 14384: A changed from T within codon CAT
Position 14456: A changed from T within codon GAT
Position 14458: A changed from T within codon ATG
Position 14489: A changed from T within codon TAC
Position 14503: A changed from T within codon ATT
Position 14515: A changed from T within codon ATG
Position 14602: A changed from T within codon ATC
Position 14681: A changed from T within codon CAT
Position 14684: A changed from T within codon CAT
Position 14910: A changed from T within codon GTA
Position 15050: A changed from T within codon TAC
Position 15062: A changed from T within codon GAT
Position 15067: A changed from T within codon ATC
Position 15098: A changed from T within codon TAC
Position 15100: A changed from T within codon ATC
Position 15150: A changed from T within codon GTA
Position 15155: A changed from T within codon TAA
Position 15351: A changed from T within codon TTA
Position 15392: A changed from T within codon TAC
Position 15431: A changed from T within codon CAT
Position 15455: A changed from T within codon TAC
Position 15479: A changed from T within codon AAT

```

El codón ATG desempeña un papel crucial en la síntesis de proteínas, ya que cumple dos funciones esenciales:

Codón de inicio: Señaliza el punto de partida para la traducción del ARN mensajero (ARNm), el proceso mediante el cual se construye una proteína a partir de la información genética. **Codificación de metionina:** Codifica el aminoácido metionina, que es el primer aminoácido en la mayoría de las proteínas. Por lo tanto, el codón ATG no produce proteínas específicas por sí solo, sino que es

indispensable para el inicio de la síntesis de prácticamente todas las proteínas en los organismos eucariotas y procariotas.

Aquí hay algunos puntos clave para comprender mejor el papel del codón ATG:

Universalidad: El codón ATG es altamente conservado en todos los organismos vivos, lo que subraya su importancia fundamental en la biología molecular. Variaciones: Aunque el ATG es el codón de inicio más común, existen casos raros en los que se utilizan otros codones de inicio alternativos.

Importancia en la traducción: La correcta función del codón ATG es esencial para asegurar que las proteínas se sinteticen correctamente y cumplan sus funciones biológicas. En resumen, el codón ATG es el punto de partida para la creación de cualquier proteína, y además codifica para el aminoácido metionina.

Haz doble clic (o pulsa Intro) para editar

```
# ahora sabremos el gen en el que se encuentra la mutacion Position 11422: A changed from T w

from Bio import SeqIO
from Bio.Seq import Seq

# Assuming the file is in the /content/ directory
for record in SeqIO.parse("/content/gene_sars_corv2.fna", "fasta"):
    sequence = str(record.seq)
    #Find the mutation position 11422
    try:
        mutation_position = 11421 # Adjust to 0-based indexing
        codon_start = mutation_position - (mutation_position % 3)
        codon_end = codon_start + 3

        if 0 <= mutation_position < len(sequence):
            codon = sequence[codon_start:codon_end]
            print(f"Codon at position {mutation_position +1}: {codon}")
            if codon == "ATG":
                print("The mutation is within the ATG start codon.")
                # Further analysis
                print(f"Nucleotide at position {mutation_position +1}: {sequence[mutation_positio")
            else:
                print(f"The mutation is not in an ATG start codon, it's in the codon {codon}.")
        else:
            print("Mutation position is out of bounds.")
    except IndexError:
        print(f"Error: Index {mutation_position} is out of bounds for the sequence.")
```



```
Codon at position 11422: ATG
The mutation is within the ATG start codon.
Nucleotide at position 11422: A
Codon at position 11422: ATG
The mutation is within the ATG start codon.
Nucleotide at position 11422: A
Codon at position 11422: GAG
The mutation is not in an ATG start codon, it's in the codon GAG.
```

La metionina desempeña un papel fundamental en la síntesis de proteínas, ya que es el aminoácido inicial en la mayoría de ellas. Sin embargo, no es correcto decir que "salen" proteínas de la metionina. Más bien, la metionina es el primer eslabón en la cadena de aminoácidos que forman una proteína.

✓ explicacion del este proceso:

Codón de inicio (ATG): El codón ATG en el ARN mensajero (ARNm) codifica para la metionina y también actúa como la señal de inicio para la traducción. La traducción es el proceso mediante el cual los ribosomas leen el ARNm y ensamblan una cadena de aminoácidos.

Inicio de la traducción:

Cuando un ribosoma encuentra un codón ATG, comienza a añadir aminoácidos a la cadena en crecimiento, comenzando con la metionina. Formación de la proteína: A medida que el ribosoma se desplaza a lo largo del ARNm, añade más aminoácidos a la cadena, siguiendo el código genético. Finalmente, cuando el ribosoma encuentra un codón de parada, la traducción se detiene y la proteína se libera. Metionina inicial: En muchas proteínas, la metionina inicial se elimina después de la traducción. Por lo tanto, aunque la metionina es el primer aminoácido incorporado, no siempre está presente en la proteína final. En resumen:

La metionina es esencial para iniciar la síntesis de casi todas las proteínas. Todas las proteínas en su inicio, comienzan con el aminoácido metionina. Es el primer aminoácido de toda proteína, pero no necesariamente se encuentra en la proteína ya terminada.

y este aminoácido para que proteínas crea glutamato

El glutamato es un aminoácido no esencial que desempeña una gran variedad de funciones cruciales en el organismo. No es que "cree" proteínas específicas por sí solo, sino que es un componente fundamental de muchas proteínas y participa en diversos procesos biológicos.

Aquí te explico algunas de las funciones más importantes del glutamato y las proteínas en las que participa:

Funciones principales del glutamato:

Neurotransmisor:

El glutamato es el neurotransmisor excitador más abundante en el sistema nervioso central. Desempeña un papel fundamental en la transmisión de señales entre las neuronas, la plasticidad sináptica, el aprendizaje y la memoria. Participa en la activación de receptores específicos en las neuronas, lo que desencadena respuestas celulares.

Metabolismo:

El glutamato participa en diversas vías metabólicas, incluyendo el ciclo de Krebs y la síntesis de otros aminoácidos. Es un precursor del glutatión, un antioxidante importante. Sabor umami: El glutamato es responsable del sabor umami, uno de los cinco sabores básicos. Proteínas que contienen glutamato:

Prácticamente todas las proteínas contienen glutamato, ya que es uno de los 20 aminoácidos estándar que forman las proteínas. Su presencia es especialmente relevante en proteínas que participan en:

La señalización neuronal.

El metabolismo celular.

La estructura de los tejidos. Consideraciones importantes:

El exceso de glutamato puede ser tóxico para las neuronas, un fenómeno conocido como excitotoxicidad.

El glutamato monosódico (MSG) es un aditivo alimentario que realza el sabor umami. En resumen, el glutamato es un aminoácido esencial para numerosas funciones biológicas y es un componente fundamental de la mayoría de las proteínas.

El fenómeno que se puede producir al sustituir la metionina de inicio por glutamato es una alteración significativa en el proceso de síntesis de proteínas, con consecuencias potencialmente graves. Aquí se detallan los posibles escenarios:

1. Pérdida de la función del codón de inicio:

El codón ATG (metionina) es esencial para que el ribosoma inicie la traducción. Si este codón se muta a GAG (glutamato), el ribosoma podría no reconocer el sitio de inicio, lo que resultaría en la imposibilidad de sintetizar la proteína.

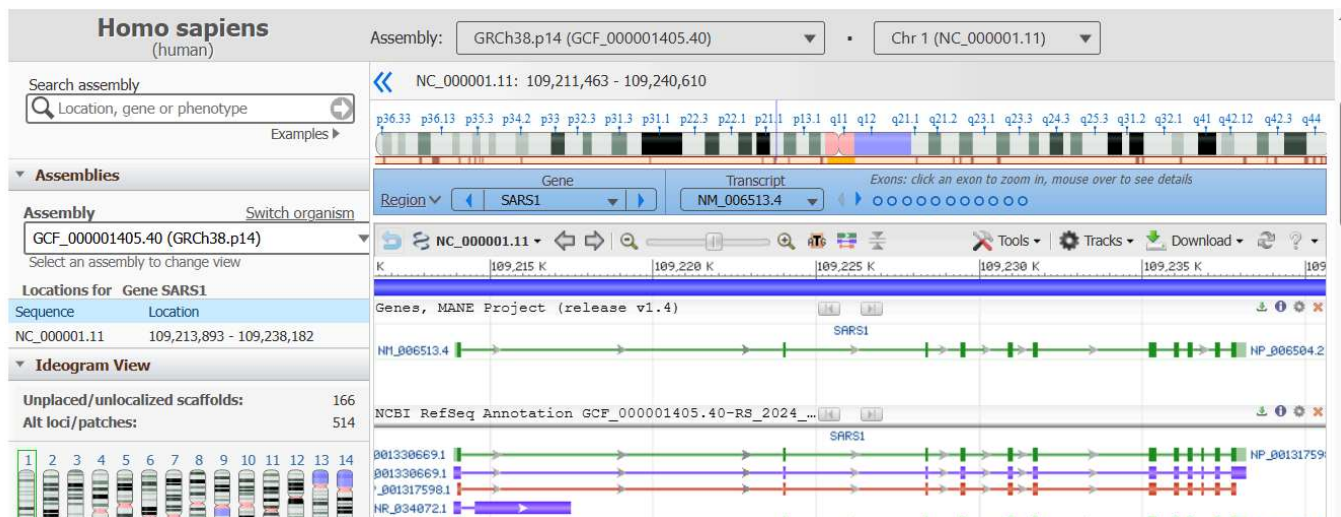
2. Inicio de la traducción en sitios alternativos:

En algunos casos, el ribosoma podría intentar iniciar la traducción en un codón ATG alternativo ubicado más adelante en el ARNm. Esto podría resultar en: Una proteína truncada: La proteína resultante sería más corta de lo normal, lo que podría afectar su función o hacerla completamente no funcional. Una proteína con una secuencia alterada: La secuencia de aminoácidos de la proteína sería diferente, lo que podría cambiar su estructura tridimensional y su actividad biológica.

3. Proteína con glutamato en el extremo N-terminal:

Si la traducción lograra iniciar en el codón mutado, la proteína resultante tendría glutamato en lugar de metionina en su extremo N-terminal. Esto podría afectar: El plegamiento de la proteína: La secuencia de aminoácidos determina la forma tridimensional de una proteína. Un cambio en el extremo N-terminal podría alterar el plegamiento correcto. Las interacciones con otras moléculas: El extremo N-terminal de una proteína puede ser importante para sus interacciones con otras proteínas, ADN u otras moléculas. Consecuencias generales:

La sustitución de la metionina de inicio por glutamato puede llevar a la pérdida de la función de la proteína, lo que puede tener efectos adversos en la célula o el organismo. Las consecuencias exactas dependen del gen afectado y de la importancia de la proteína en cuestión. En resumen, la mutación de un codón de inicio es un evento significativo que puede interferir gravemente con la síntesis de proteínas.



Aquí está un análisis detallado:

Información General:

Organismo: Homo sapiens (humano). Ensamblaje: GRCh38.p14 (GCF_000001405.40). Esta es la versión de referencia del genoma humano. Cromosoma: Chr 1 (NC_000001.11). Esto indica que el gen SARS1 se encuentra en el cromosoma 1. Rango de ubicación: NC_000001.11: 109,211,463-109,240,610. Esto especifica la posición del gen dentro del cromosoma 1. Visualización del Cromosoma:

Hay un ideograma del cromosoma 1 que muestra la ubicación del gen SARS1. También se muestra un gráfico detallado de la región del cromosoma donde se encuentra el gen, con anotaciones de exones y transcritos. Información del Gen SARS1:

Se muestra información sobre el gen SARS1, incluyendo su ubicación y transcritos asociados (NM_006513.4). Se visualizan los exones del gen, lo que permite observar la estructura del gen. Se pueden observar diferentes representaciones del gen, con distintos colores, esto se debe a las diferentes formas en las que se ha reportado este gen. Otras Características:

Hay información sobre scaffolds no colocados/no localizados y loci/patches alternativos, lo que indica la complejidad del genoma. Se proporcionan herramientas y opciones para descargar datos y ver diferentes pistas (tracks) de información. En la parte inferior se observa la representación gráfica de todos los cromosomas humanos. Interpretación:

Esta interfaz proporciona una visión detallada de la ubicación y estructura del gen SARS1 en el cromosoma 1 humano. La información presentada es útil para investigadores que estudian la genética humana, la genómica y la biología molecular. Se puede observar que el gen SARS1 tiene diferentes transcripciones, lo que implica que se pueden generar distintas proteínas a partir de él. En resumen, la imagen muestra una herramienta bioinformática que permite explorar y analizar información genética de manera visual y detallada.

Aquí hay algunos puntos clave para reforzar esta conexión:

Cromosoma 1: La imagen muestra claramente "Chr 1 (NC_000001.11)", lo que confirma que el gen SARS1 se encuentra en el cromosoma 1. Gen SARS1: La imagen identifica específicamente el "Gene

SARS1", mostrando su ubicación y estructura dentro del cromosoma 1. Información Genómica: La interfaz proporciona detalles sobre la ubicación precisa del gen, sus transcritos y otras anotaciones genómicas relevantes. Por lo tanto, la imagen es una representación visual de la información genómica que hemos estado discutiendo. Esta herramienta permite a los investigadores explorar y analizar la ubicación y estructura del gen SARS1 en el contexto del cromosoma 1.

algoritmo creado por Alejandro Cuevas V

<https://www.kaggle.com/pollicio/code>