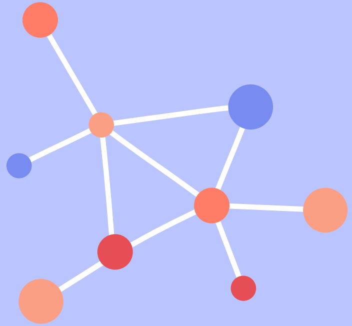
BIOHACKEANDO ADN PARA ALMACENAR BIT(UP)S

Marina Moro López

¡Hola! :D

- Ingeniera biomédica
- Futura doctora en biomedicina
- ‘Programadora’ a nivel científico



1

DEFINICIÓN DE BIOHACKING

2

TIPOS DE BIOHACKING

3

TEORÍA BÁSICA DE GENÉTICA

4

CASO PRÁCTICO

5

APLICACIONES TERAPÉUTICAS

6

RONDA DE PREGUNTAS

1

DEFINICIÓN DE BIOHACKING

Bio + hacking

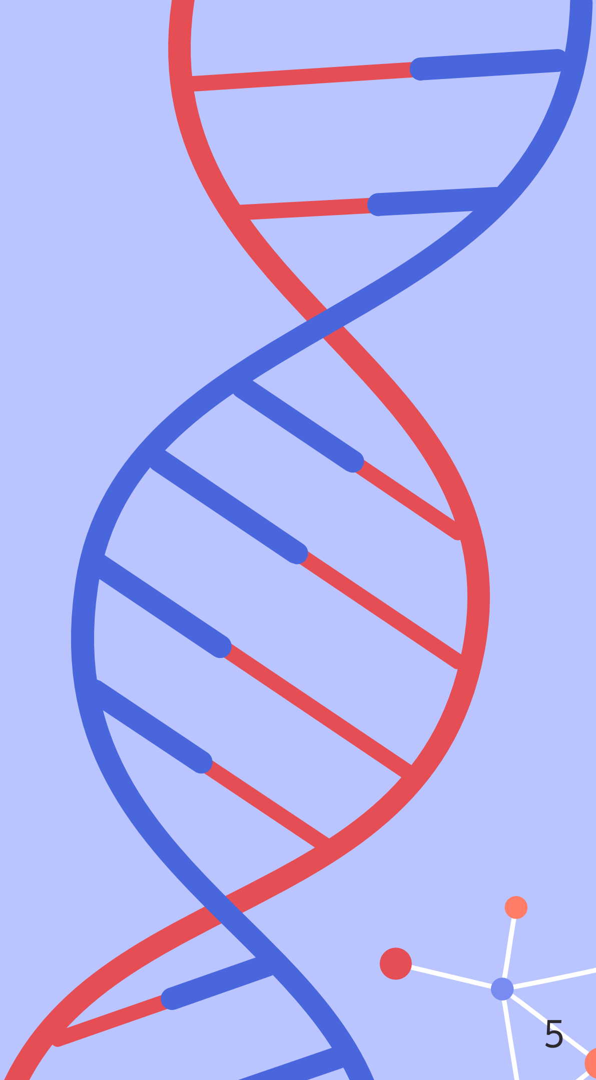
Añadir funcionalidades y resolución de problemas sociales en el ámbito bio

Democratización

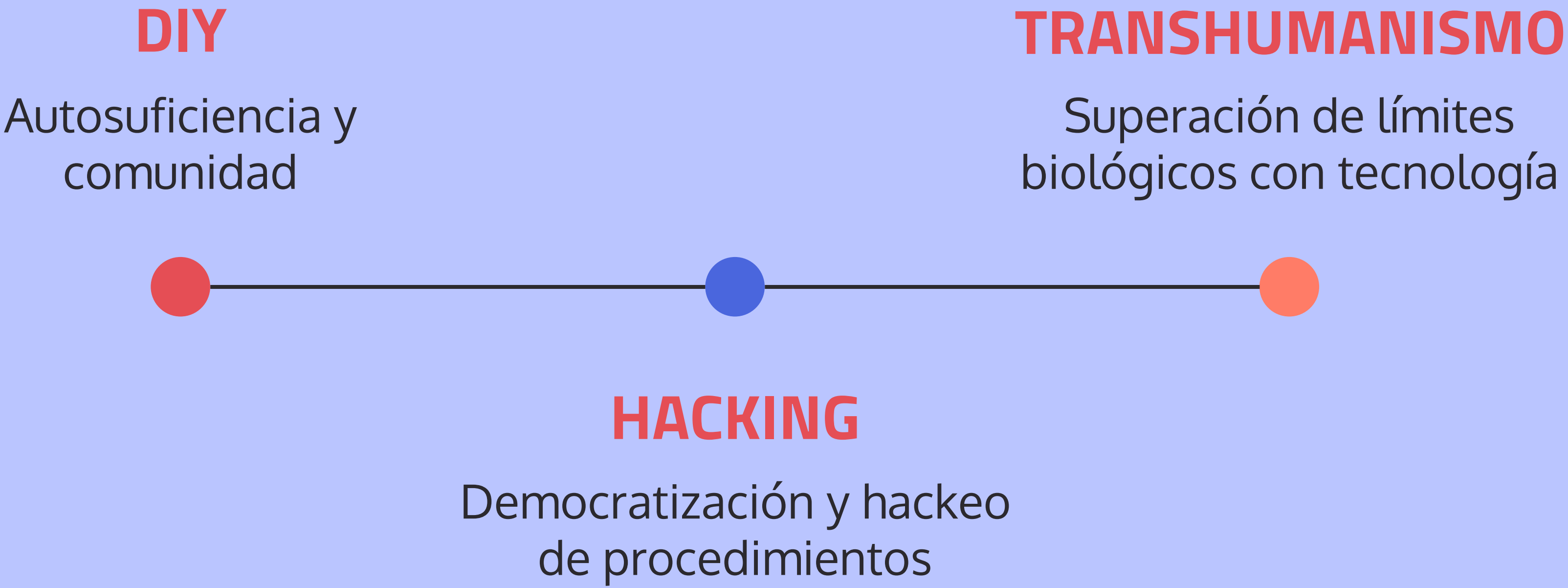
Herramientas biológicas e información fuera del ámbito institucional

Código ético

Transparencia, seguridad, educación, compromiso y responsabilidad



ORIGEN DEL BIOHACKING



DIY

Autosuficiencia y comunidad

HACKING

Democratización y hackeo de procedimientos

TRANSHUMANISMO

Superación de límites biológicos con tecnología

¿Y QUÉ SE HACE?

Proyectos de salud, medioambiente y bioarte usando tecnologías de genética, bioquímica, bioingeniería, biología sintética, electrónica...

- Desarrollo de equipos low-cost
- Producción de medicamentos
- Talleres y conferencias
- Start-ups con los productos desarrollados
- Autoexperimentación y modificaciones corporales

2

TIPOS DE BIOHACKING

BIOHACKING FISIOLÓGICO

Hacking del propio organismo con dietas, ingesta de suplementos y hábitos de vida

Ejemplos: ayuno intermitente, exposición a infrarrojos, uso de nootrópicos

BIOLOGÍA DIY

Manipulación de la biología a través de técnicas innovadoras por parte de la ciudadanía

Ejemplos: biohacking genético, neurohacking, terapia celular, producción de medicamentos y de equipo

GRINDER

Manipulación corporal con visión transhumanista (body hacking)

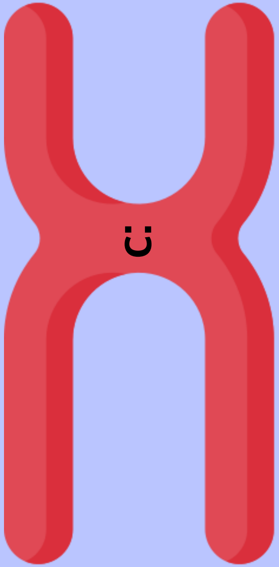
Ejemplos: implantación de hardware, modificación de implantes, edición genética y biohacking *in vivo*

3

TEORÍA BÁSICA DE GENÉTICA

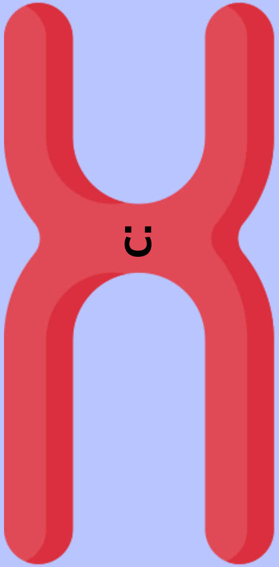
CROMOSOMA BACTERIANO – ADN

Estructura que contiene
todos los genes…

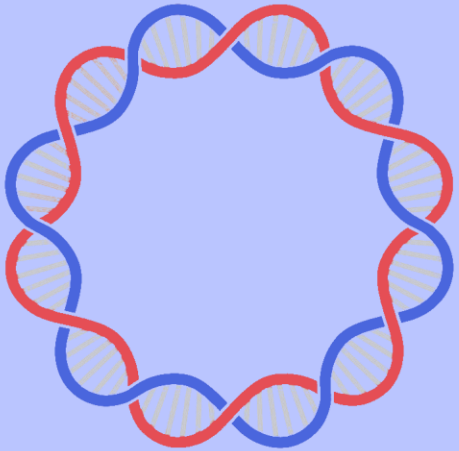


CROMOSOMA BACTERIANO – ADN

Estructura que contiene todos los genes…

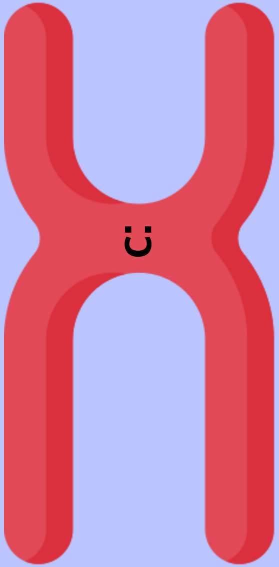


… y que es circular en bacterias

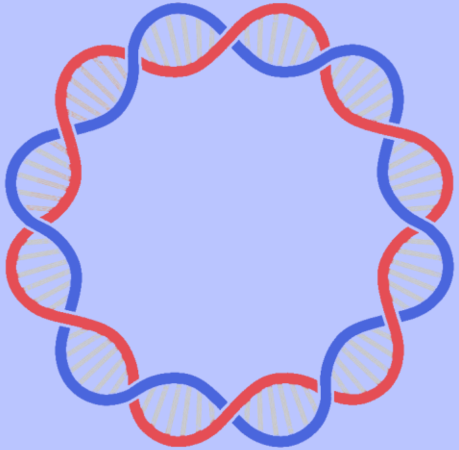


CROMOSOMA BACTERIANO – ADN

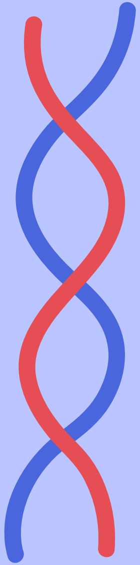
Estructura que contiene todos los genes…

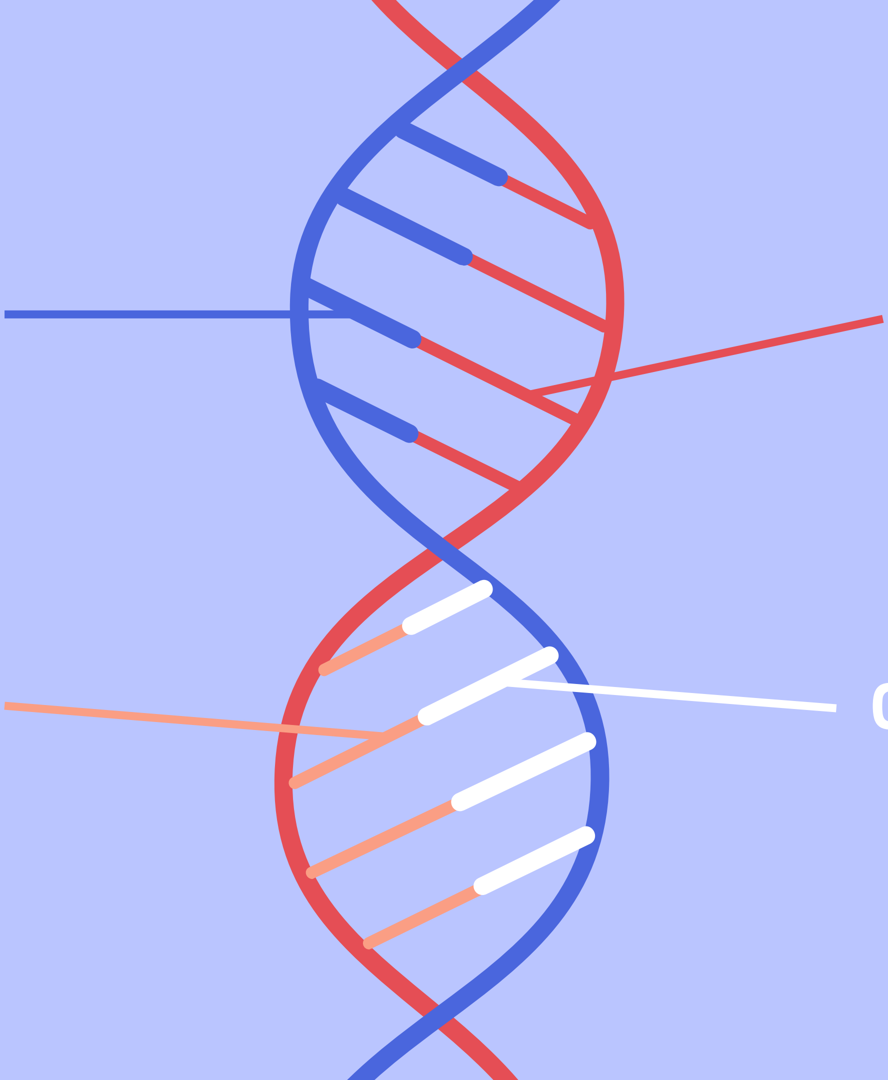


… y que es circular en bacterias



Doble hélice formada por bases





ADENINA (A)

TIMINA (T)

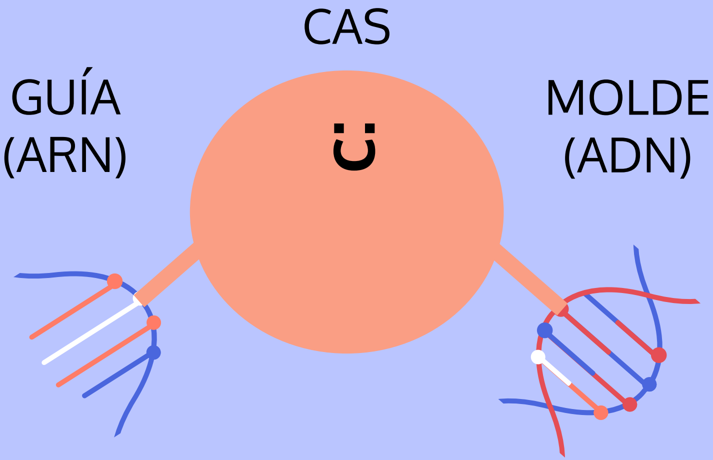
CITOSINA (C)

GUANINA (G)

CRISPR

Corta y pega de secuencias de ADN (edición genética)

CRISPR

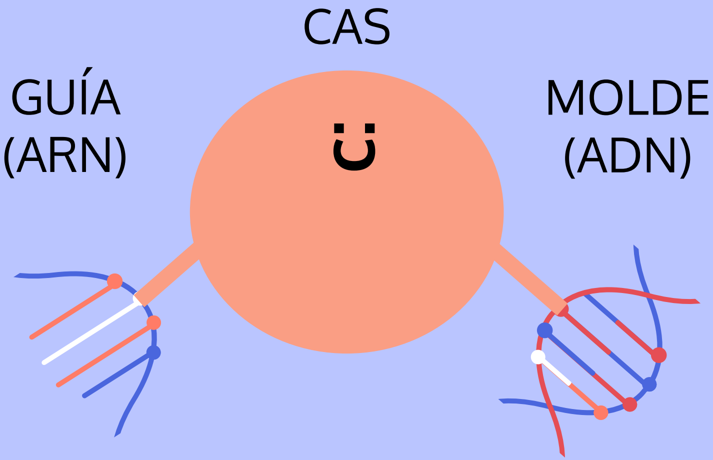


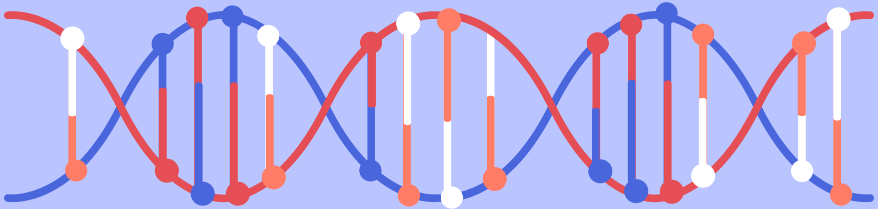
CAS

GUÍA
(ARN)

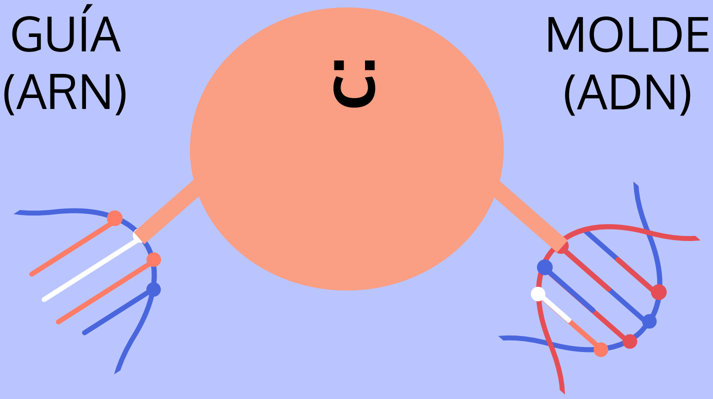
MOLDE
(ADN)

CRISPR





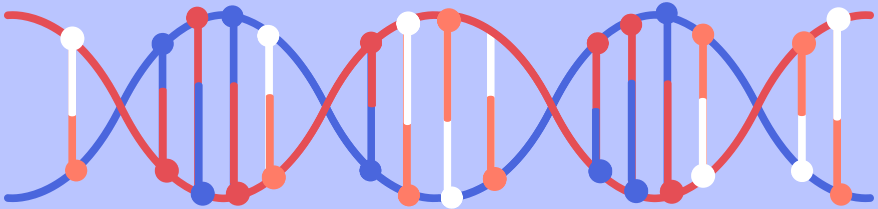
CRISPR



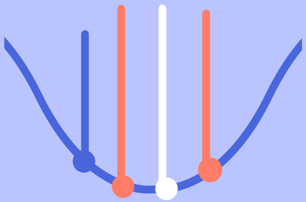
CAS

GUÍA
(ARN)

MOLDE
(ADN)

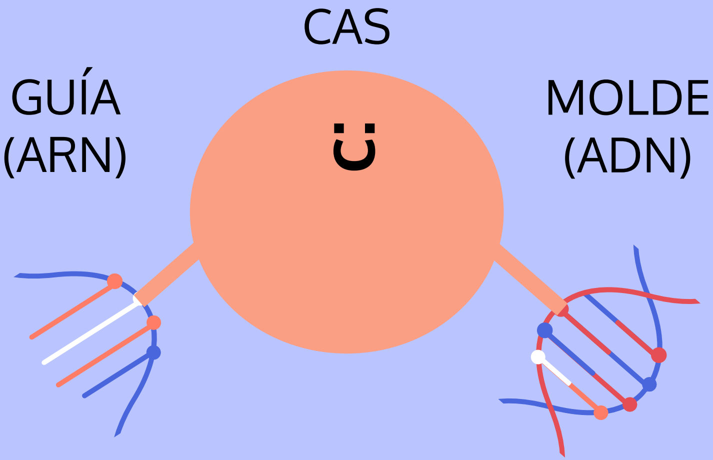


ADN
ORIGINAL



GUÍA
(ARN)

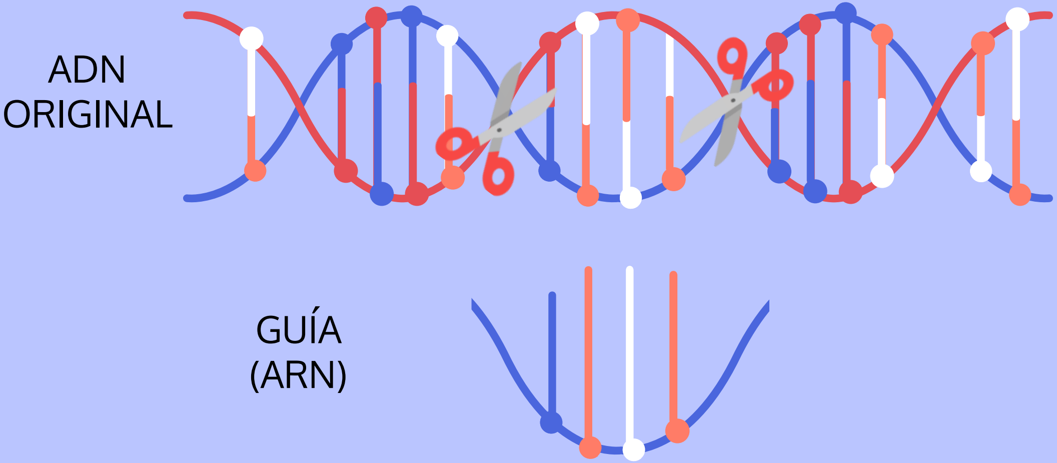
CRISPR



GUÍA
(ARN)

CAS

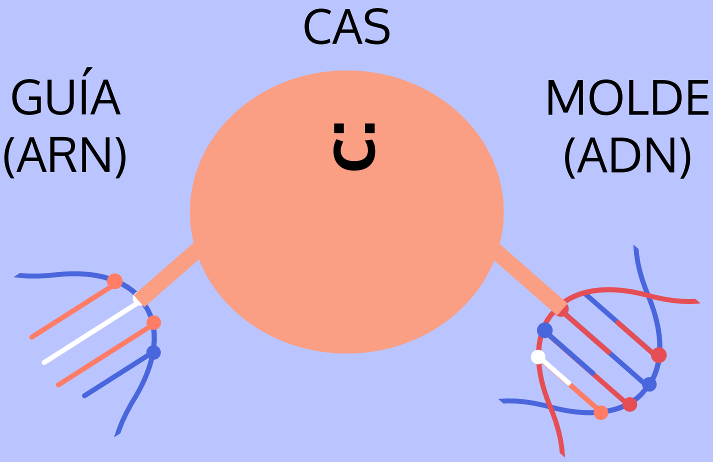
MOLDE
(ADN)



ADN
ORIGINAL

GUÍA
(ARN)

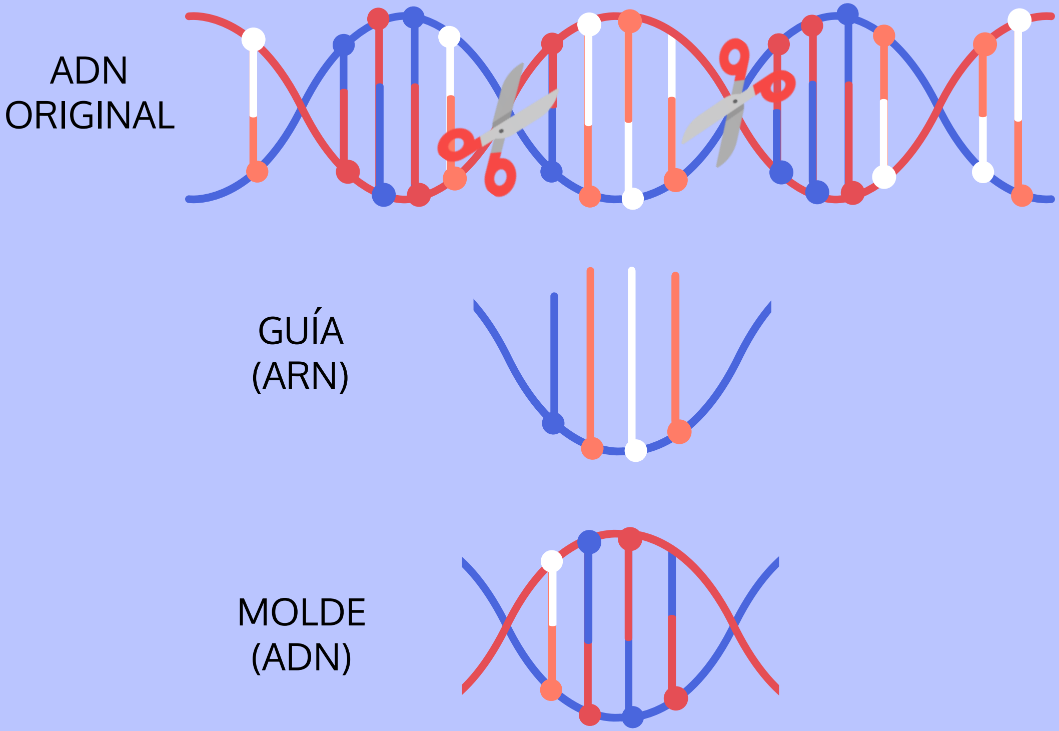
CRISPR



CAS

GUÍA
(ARN)

MOLDE
(ADN)

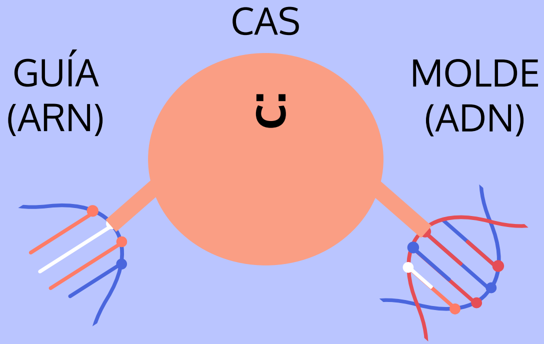


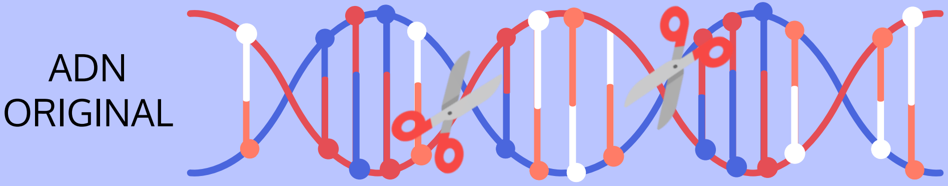
ADN
ORIGINAL

GUÍA
(ARN)

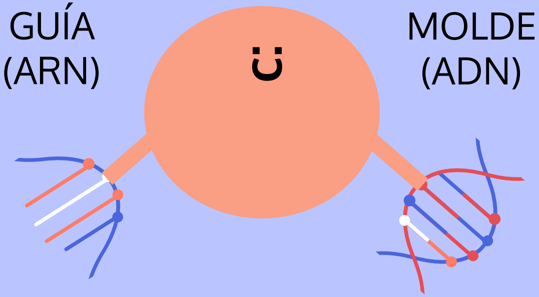
MOLDE
(ADN)

CRISPR





CRISPR

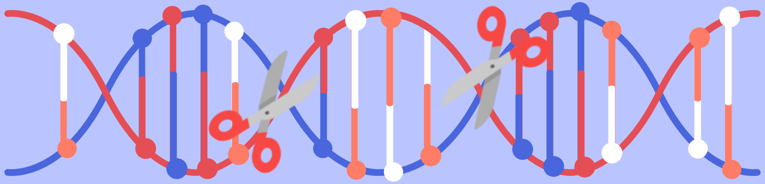


CAS

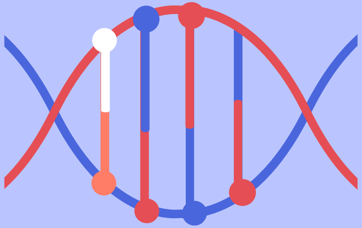
GUÍA
(ARN)

MOLDE
(ADN)

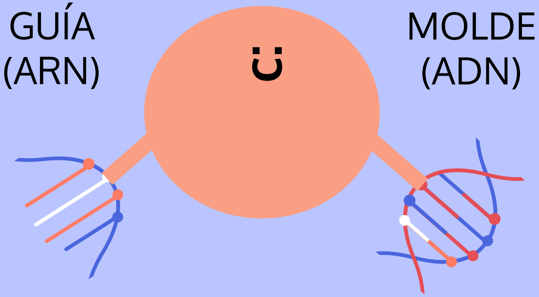
ADN
ORIGINAL



MOLDE
(ADN)



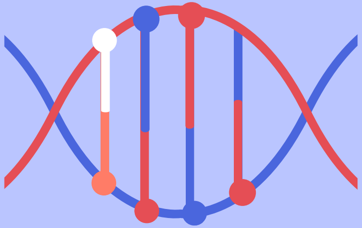
CRISPR



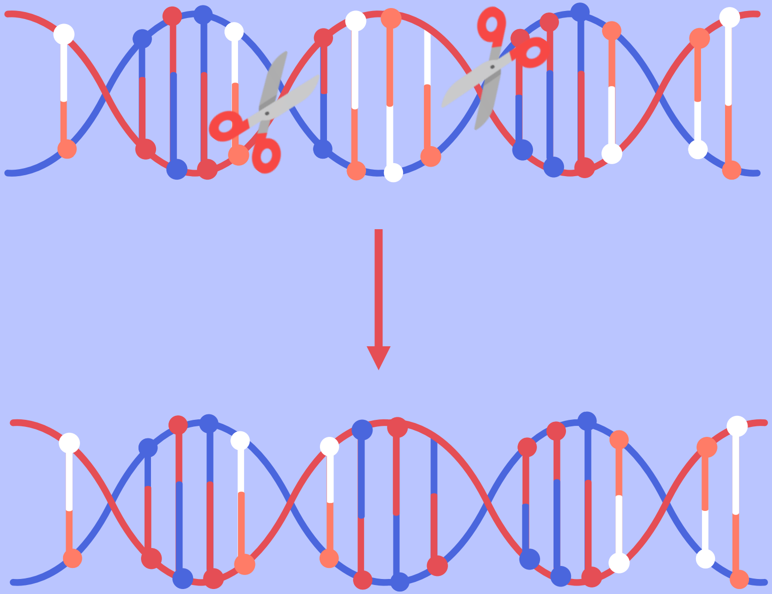
CAS

GUÍA
(ARN)

MOLDE
(ADN)



MOLDE
(ADN)



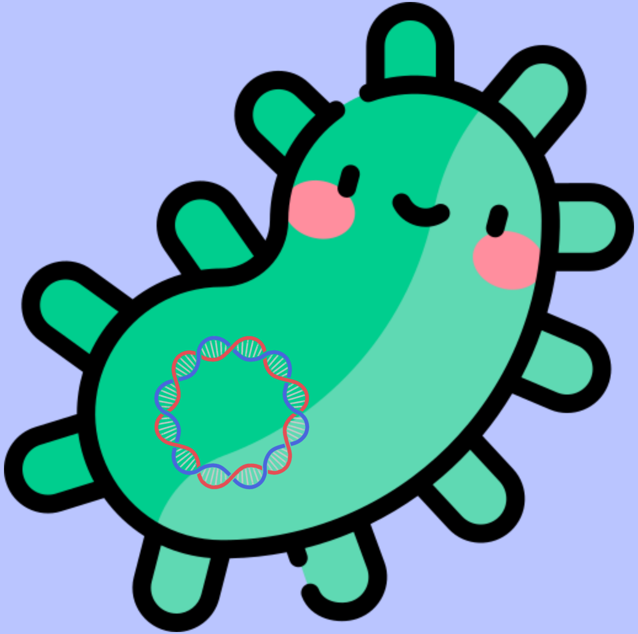
ADN
ORIGINAL

ADN
NUEVO

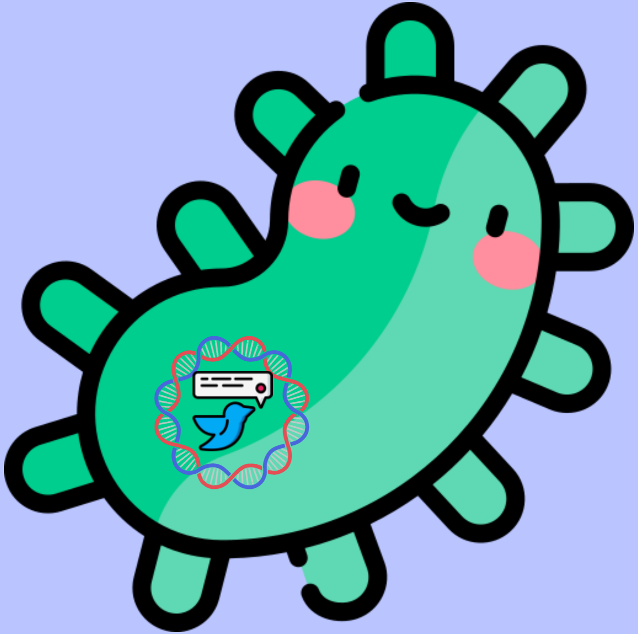
4

CASO PRÁCTICO

Cómo almacenar tweets en el ADN de una bacteria



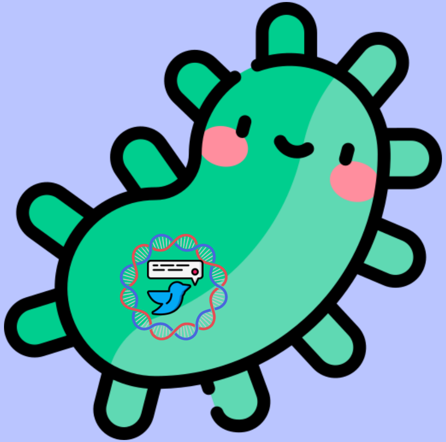
Cómo almacenar tweets en el ADN de una bacteria



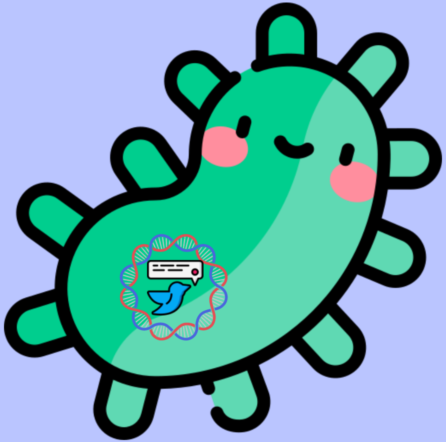
CIFRAS DEL ADN

- Vida media de miles de años (congelado)
- 1g de ADN = 215 petabytes; 1 base = 1,8 bits
- 3B de bases en el ADN humano = 700 megabytes
- 5M bases en Escherichia Coli = 1,25 megabytes

Cómo almacenar tweets en el ADN de una bacteria

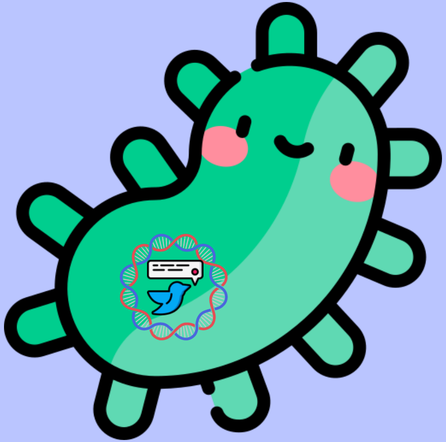


Cómo almacenar tweets en el ADN de una bacteria



1 Localizar el sitio en el que queramos poner el tweet

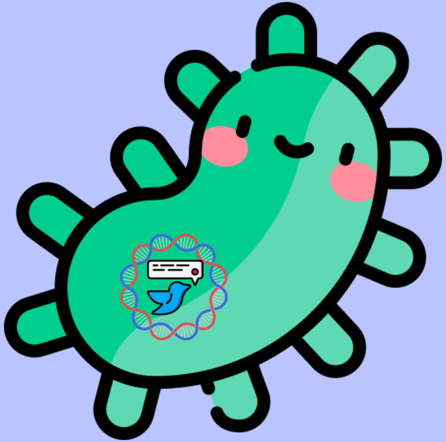
Cómo almacenar tweets en el ADN de una bacteria



1 Localizar el sitio en el que queramos poner el tweet

2 Elegir el tweet y codificarlo (tweet > binario > ADN)

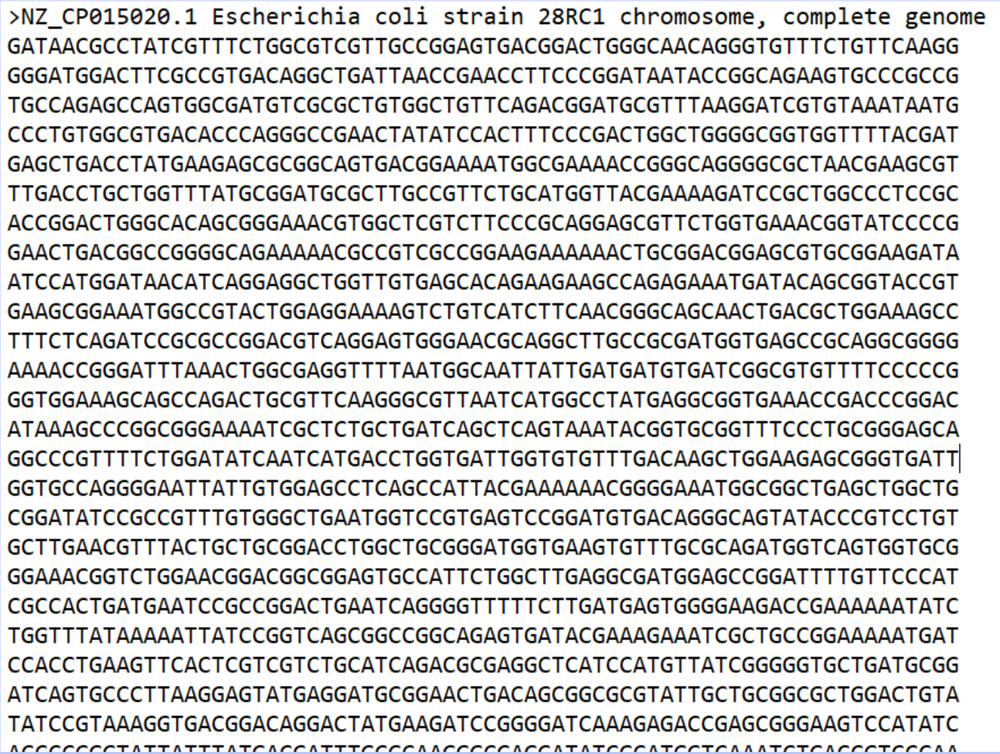
Cómo almacenar tweets en el ADN de una bacteria



1. Localizar el sitio en el que queramos poner el tweet
2. Elegir el tweet y codificarlo (tweet > binario > ADN)
3. Insertarlo en el ADN de la bacteria (y descodificar cuando queramos)

1

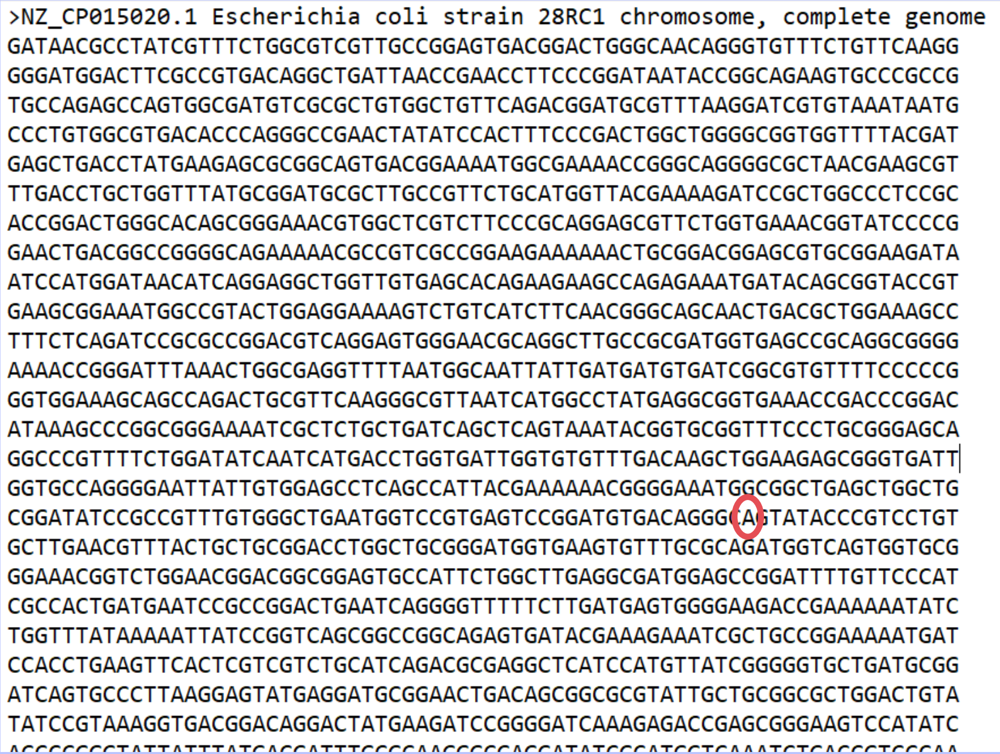
Localizar el sitio en el que queramos poner el tweet



```
>NZ_CP015020.1 Escherichia coli strain 28RC1 chromosome, complete genome
GATAACGCCTATCGTTTCTGGCGTCGTTGCCGGAGTGACGGACTGGGCAACAGGGTGTTTCTGTTCAAGG
GGGATGGACTTCGCCGTGACAGGCTGATTAACCGAACCTTCCCGGATAATACCGGCAGAAGTGCCCGCCG
TGCCAGAGCCAGTGGCGATGTCGCGCTGTGGCTGTTCAGACGGATGCGTTTAAGGATCGTGTAAATAATG
CCCTGTGGCGTGACACCCAGGGCCGAACTATATCCACTTTCCCGACTGGCTGGGGCGGTGGTTTTACGAT
GAGCTGACCTATGAAGAGCGCGGCAGTGACGGAAAATGGCGAAAACCGGGCAGGGGCGCTAACGAAGCGT
TTGACCTGCTGGTTTATGCGGATGCGCTTGCCGTTCTGCATGGTTACGAAAAGATCCGCTGGCCCTCCGC
ACCGGACTGGGCACAGCGGGAAACGTGGCTCGTCTTCCCGCAGGAGCGTTCTGGTGAAACGGTATCCCCG
GAACTGACGGCCGGGGCAGAAAAACGCCGTCGCCGGAAGAAAAAACTGCGGACGGAGCGTGCGGAAGATA
ATCCATGGATAACATCAGGAGGCTGGTTGTGAGCACAGAAGAAGCCAGAGAAATGATACAGCGGTACCGT
GAAGCGGAAATGGCCGTACTGGAGGAAAAGTCTGTCATCTTCAACGGGCAGCAACTGACGCTGGAAAGCC
TTTCTCAGATCCGCGCCGGACGTCAGGAGTGGGAACGCAGGCTTGCCGCGATGGTGAGCCGCAGGCGGGG
AAAACCGGGATTTAAACTGGCGAGGTTTTAATGGCAATTATTGATGATGTGATCGGCGTGTTTTCCCCCG
GGTGGAAAGCAGCCAGACTGCGTTCAAGGGCGTTAATCATGGCCTATGAGGCGGTGAAACCGACCCGGAC
ATAAAGCCCGGCGGGAAAATCGCTCTGCTGATCAGCTCAGTAAATACGGTGCGGTTTCCCTGCGGGAGCA
GGCCCGTTTTCTGGATATCAATCATGACCTGGTGATTGGTGTGTTTGACAAGCTGGAAGAGCGGGTGATT
GGTGCCAGGGGAATTATTGTGGAGCCTCAGCCATTACGAAAAAACGGGGAAATGGCGGCTGAGCTGGCTG
CGGATATCCGCCGTTTGTGGGCTGAATGGTCCGTGAGTCCGGATGTGACAGGGCAGTATACCCGTCCTGT
GCTTGAACGTTTACTGCTGCGGACCTGGCTGCGGGATGGTGAAGTGTTTGCGCAGATGGTCAGTGGTGCG
GGAAACGGTCTGGAACGGACGGCGGAGTGCCATTCTGGCTTGAGGCGATGGAGCCGGATTTTGTTCCCAT
CGCCACTGATGAATCCGCCGGACTGAATCAGGGGTTTTTCTTGATGAGTGGGGAAGACCGAAAAAATATC
TGGTTTATAAAAATTATCCGGTCAGCGGCCGGCAGAGTGATACGAAAGAAATCGCTGCCGGAAAAATGAT
CCACCTGAAGTTCACTCGTCGTCTGCATCAGACGCGAGGCTCATCCATGTTATCGGGGGTGCTGATGCGG
ATCAGTGCCCTTAAGGAGTATGAGGATGCGGAACTGACAGCGGCGCGTATTGCTGCGGCGCTGGACTGTA
TATCCGTAAAGGTGACGGACAGGACTATGAAGATCCGGGGATCAAAGAGACCGAGCGGGAAGTCCATATC
```

1

Localizar el sitio en el que queramos poner el tweet

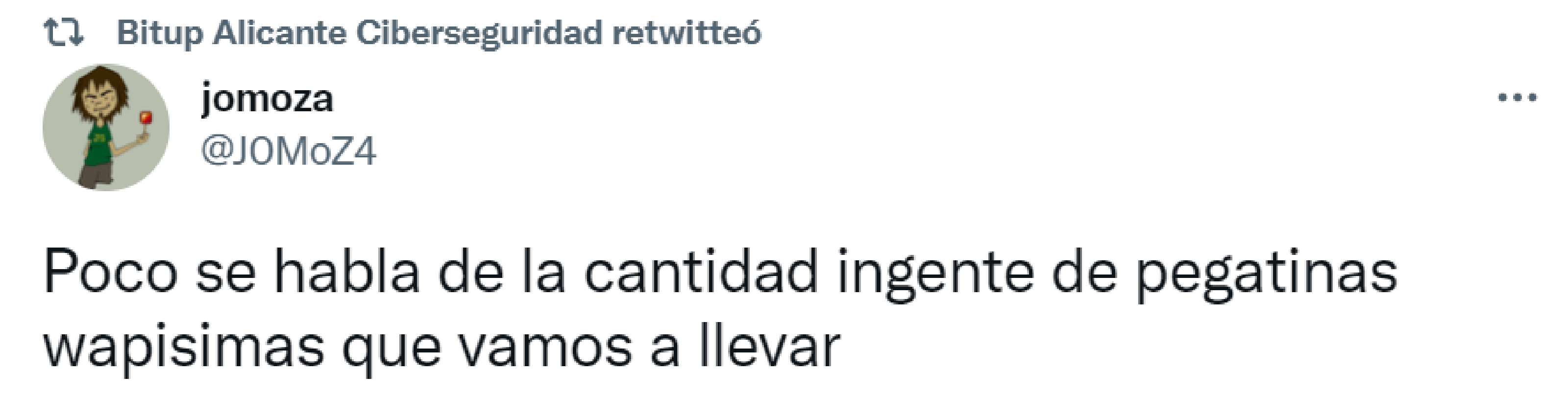


```
>NZ_CP015020.1 Escherichia coli strain 28RC1 chromosome, complete genome
GATAACGCCTATCGTTTCTGGCGTCGTTGCCGGAGTGACGGACTGGGCAACAGGGTGTTTCTGTTCAAGG
GGGATGGACTTCGCCGTGACAGGCTGATTAACCGAACCTTCCCGGATAATACCGGCAGAAGTGCCCGCCG
TGCCAGAGCCAGTGGCGATGTCGCGCTGTGGCTGTTCAGACGGATGCGTTTAAGGATCGTGTAAATAATG
CCCTGTGGCGTGACACCCAGGGCCGAACTATATCCACTTTCCCGACTGGCTGGGGCGGTGGTTTTACGAT
GAGCTGACCTATGAAGAGCGCGGCAGTGACGGAAAATGGCGAAAACCGGGCAGGGGCGCTAACGAAGCGT
TTGACCTGCTGGTTTATGCGGATGCGCTTGCCGTTCTGCATGGTTACGAAAAGATCCGCTGGCCCTCCGC
ACCGGACTGGGCACAGCGGGAAACGTGGCTCGTCTTCCCGCAGGAGCGTTCTGGTGAAACGGTATCCCCG
GAACTGACGGCCGGGGCAGAAAAACGCCGTCGCCGGAAGAAAAAACTGCGGACGGAGCGTGCGGAAGATA
ATCCATGGATAACATCAGGAGGCTGGTTGTGAGCACAGAAGAAGCCAGAGAAATGATACAGCGGTACCGT
GAAGCGGAAATGGCCGTACTGGAGGAAAAGTCTGTCATCTTCAACGGGCAGCAACTGACGCTGGAAAGCC
TTTCTCAGATCCGCGCCGGACGTCAGGAGTGGGAACGCAGGCTTGCCGCGATGGTGAGCCGCAGGCGGGG
AAAACCGGGATTTAAACTGGCGAGGTTTTAATGGCAATTATTGATGATGTGATCGGCGTGTTTTCCCCCG
GGTGGAAAGCAGCCAGACTGCGTTCAAGGGCGTTAATCATGGCCTATGAGGCGGTGAAACCGACCCGGAC
ATAAAGCCCGGCGGGAAAATCGCTCTGCTGATCAGCTCAGTAAATACGGTGCGGTTTCCCTGCGGGAGCA
GGCCCGTTTTCTGGATATCAATCATGACCTGGTGATTGGTGTGTTTGACAAGCTGGAAGAGCGGGTGATT
GGTGCCAGGGGAATTATTGTGGAGCCTCAGCCATTACGAAAAAACGGGGAAATGGCGGCTGAGCTGGCTG
CGGATATCCGCCGTTTGTGGGCTGAATGGTCCGTGAGTCCGGATGTGACAGGGCAGTATACCCGTCCTGT
GCTTGAACGTTTACTGCTGCGGACCTGGCTGCGGGATGGTGAAGTGTTTGCGCAGATGGTCAGTGGTGCG
GGAAACGGTCTGGAACGGACGGCGGAGTGCCATTCTGGCTTGAGGCGATGGAGCCGGATTTTGTTCCCAT
CGCCACTGATGAATCCGCCGGACTGAATCAGGGGTTTTTCTTGATGAGTGGGGAAGACCGAAAAAATATC
TGGTTTATAAAAATTATCCGGTCAGCGGCCGGCAGAGTGATACGAAAGAAATCGCTGCCGGAAAAATGAT
CCACCTGAAGTTCACTCGTCGTCTGCATCAGACGCGAGGCTCATCCATGTTATCGGGGGTGCTGATGCGG
ATCAGTGCCCTTAAGGAGTATGAGGATGCGGAACTGACAGCGGCGCGTATTGCTGCGGCGCTGGACTGTA
TATCCGTAAAGGTGACGGACAGGACTATGAAGATCCGGGGATCAAAGAGACCGAGCGGGAAGTCCATATC
```

Base A
Posición 1.175

2

Elegir el tweet y codificarlo (tweet > binario > ADN)



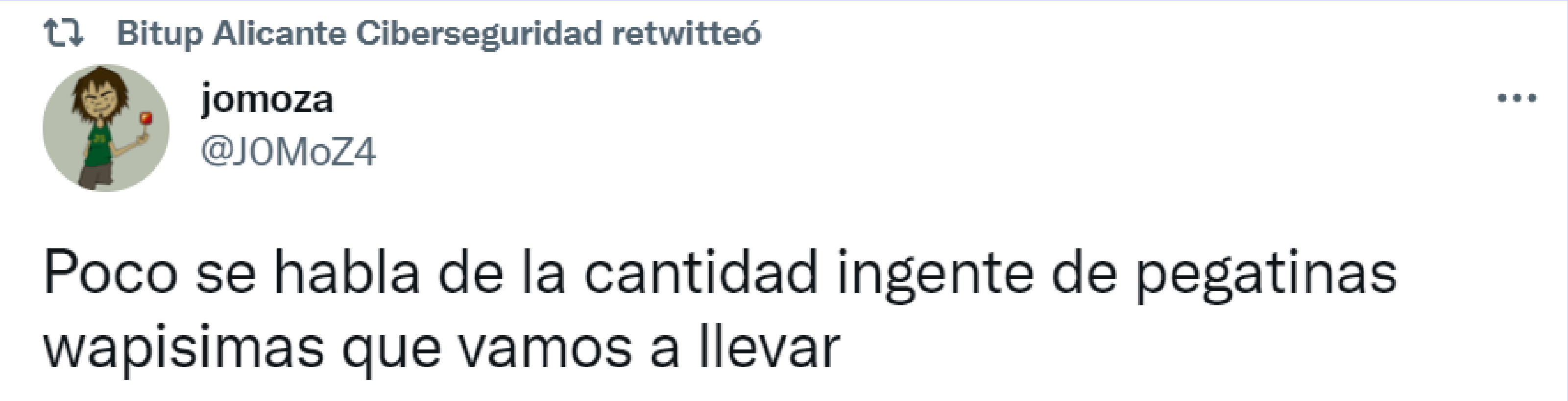
Bitup Alicante Ciberseguridad retwitteó

jomoza

@JOMoZ4

Poco se habla de la cantidad ingente de pegatinas wapisimas que vamos a llevar

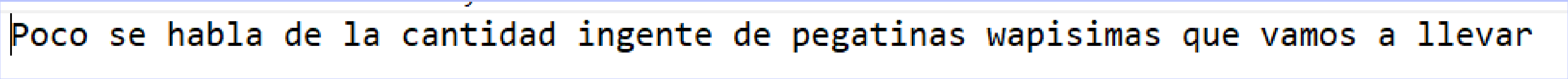
2 Elegir el tweet y codificarlo (tweet > binario > ADN)



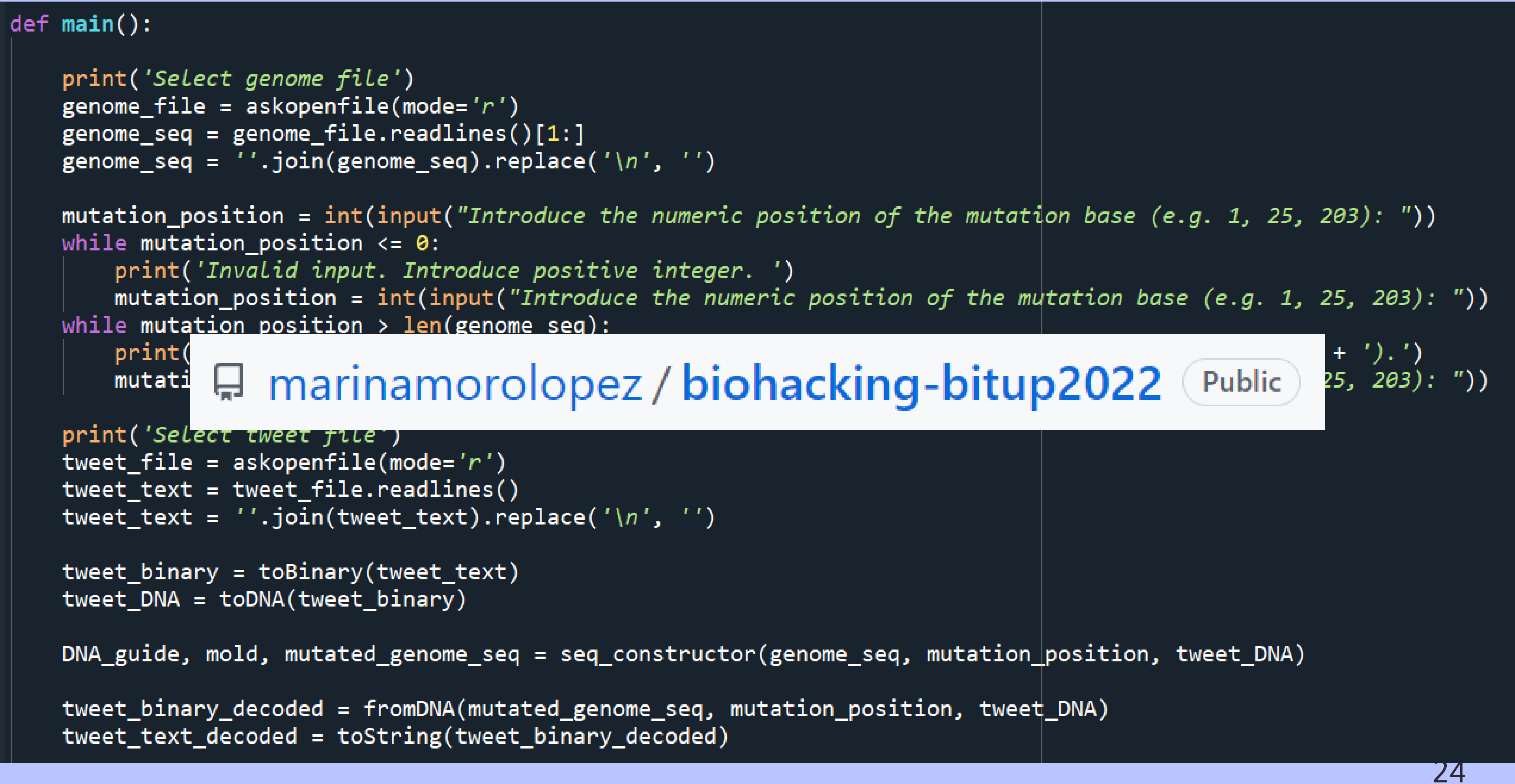
Bitup Alicante Ciberseguridad retwitteó

jomoza
@JOMoZ4

Poco se habla de la cantidad ingente de pegatinas wapisimas que vamos a llevar

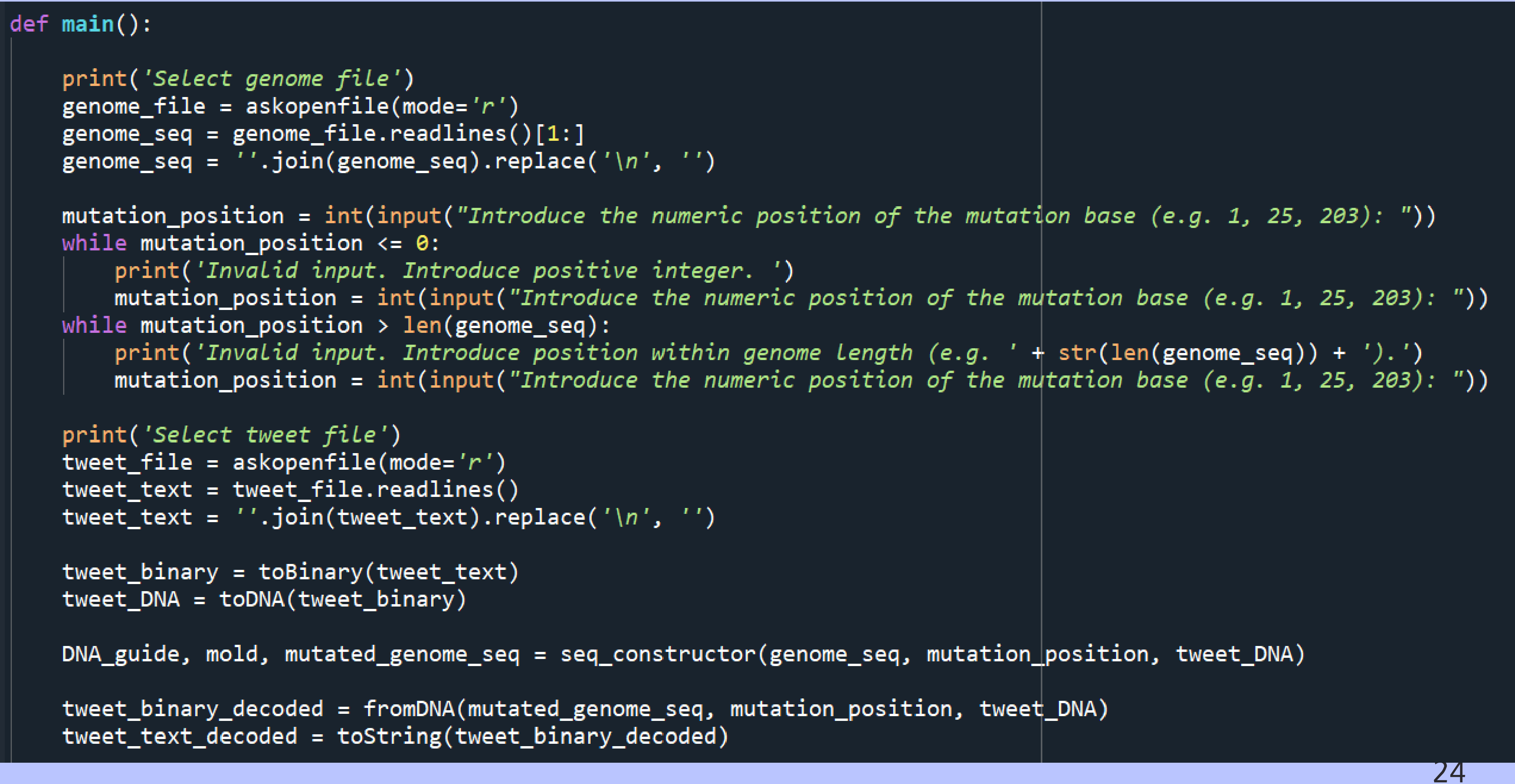


```
Poco se habla de la cantidad ingente de pegatinas wapisimas que vamos a llevar
```

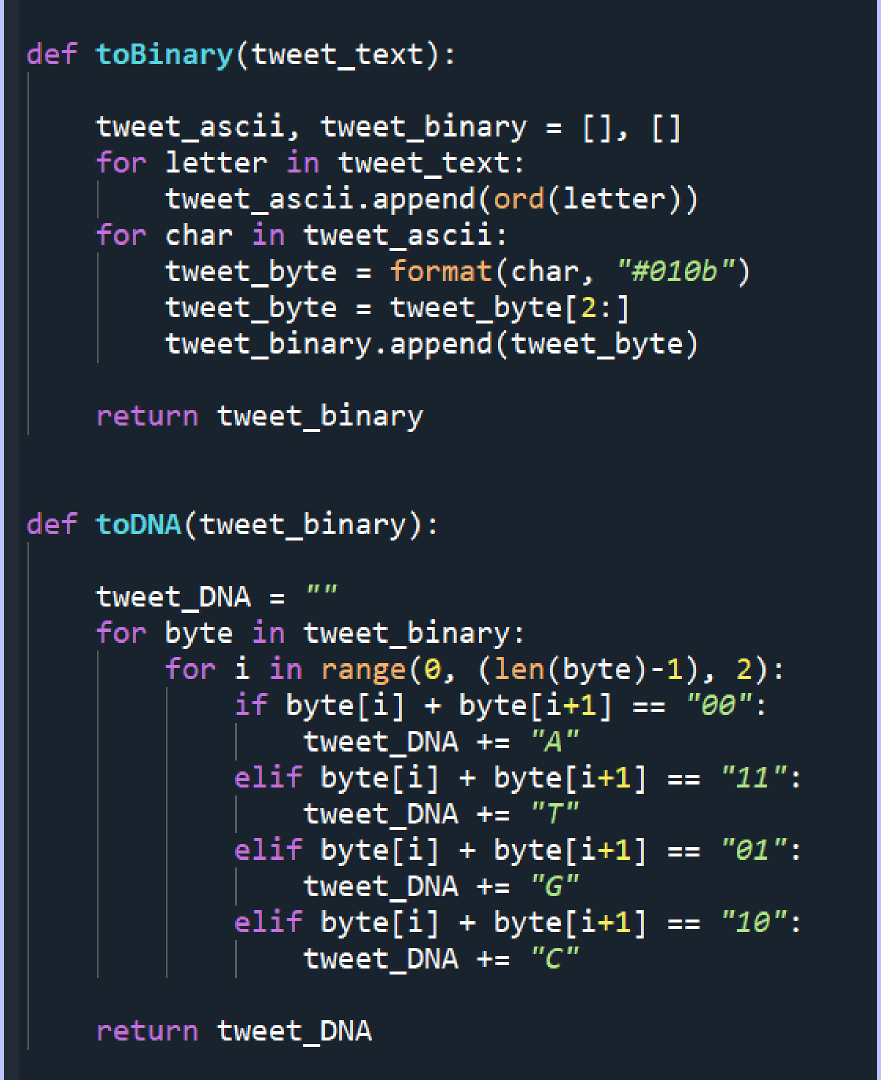


```
def main():

    print('Select genome file')
    genome_file = askopenfile(mode='r')
    genome_seq = genome_file.readlines()[1:]
    genome_seq = ''.join(genome_seq).replace('\n', '')

    mutation_position = int(input("Introduce the numeric position of the mutation base (e.g. 1, 25, 203): "))
    while mutation_position <= 0:
        print('Invalid input. Introduce positive integer. ')
        mutation_position = int(input("Introduce the numeric position of the mutation base (e.g. 1, 25, 203): "))
    while mutation_position > len(genome_seq):
        print(                                                                                        + ').')
        mutati                                                                                        25, 203): "))

    print('Select tweet file')
    tweet_file = askopenfile(mode='r')
    tweet_text = tweet_file.readlines()
    tweet_text = ''.join(tweet_text).replace('\n', '')

    tweet_binary = toBinary(tweet_text)
    tweet_DNA = toDNA(tweet_binary)

    DNA_guide, mold, mutated_genome_seq = seq_constructor(genome_seq, mutation_position, tweet_DNA)

    tweet_binary_decoded = fromDNA(mutated_genome_seq, mutation_position, tweet_DNA)
    tweet_text_decoded = toString(tweet_binary_decoded)
```

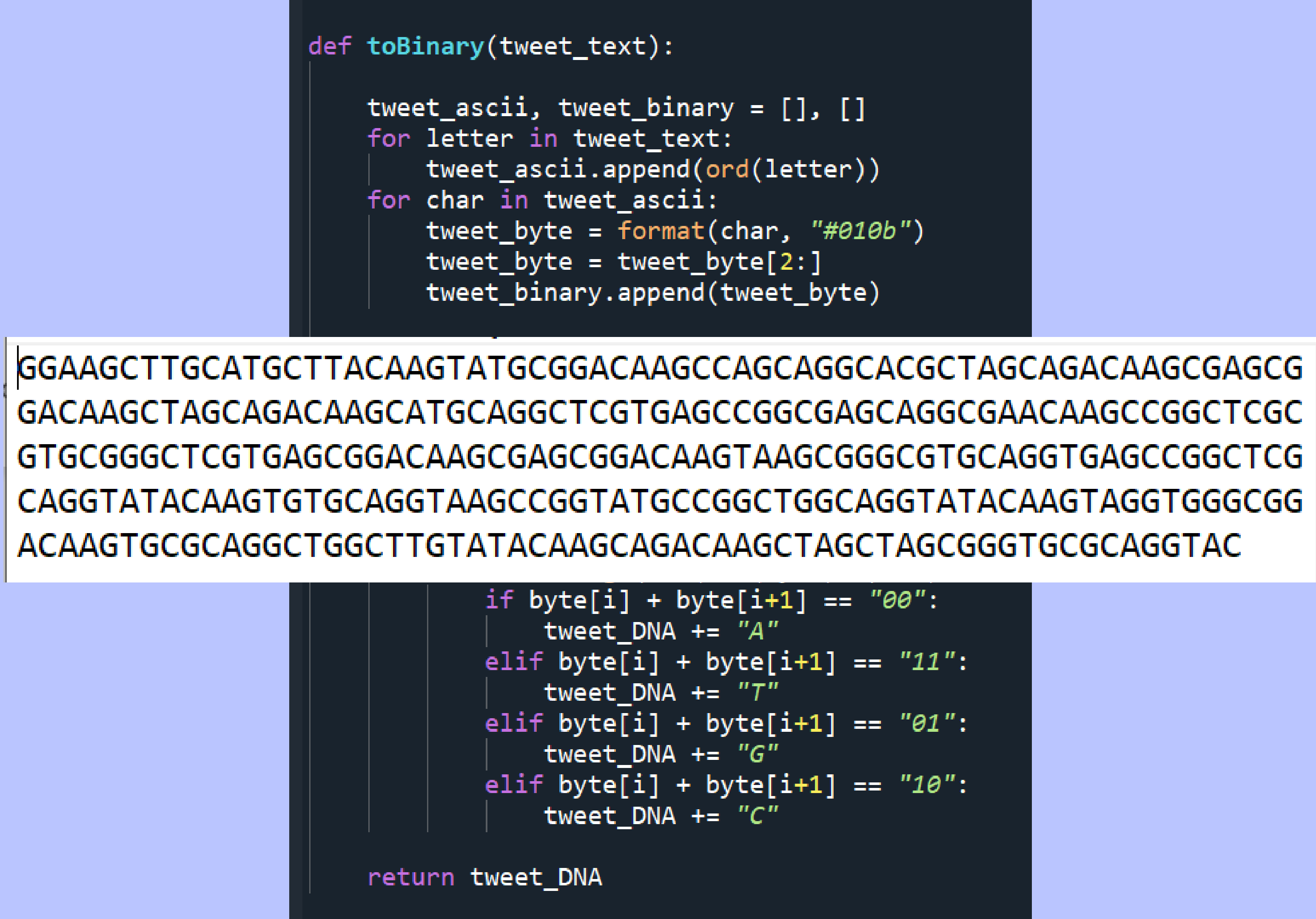
marinamorolopez / biohacking-bitup2022 Public



```
def main():

    print('Select genome file')
    genome_file = askopenfile(mode='r')
    genome_seq = genome_file.readlines()[1:]
    genome_seq = ''.join(genome_seq).replace('\n', '')

    mutation_position = int(input("Introduce the numeric position of the mutation base (e.g. 1, 25, 203): "))
    while mutation_position <= 0:
        print('Invalid input. Introduce positive integer. ')
        mutation_position = int(input("Introduce the numeric position of the mutation base (e.g. 1, 25, 203): "))
    while mutation_position > len(genome_seq):
        print('Invalid input. Introduce position within genome length (e.g. ' + str(len(genome_seq)) + ').')
        mutation_position = int(input("Introduce the numeric position of the mutation base (e.g. 1, 25, 203): "))

    print('Select tweet file')
    tweet_file = askopenfile(mode='r')
    tweet_text = tweet_file.readlines()
    tweet_text = ''.join(tweet_text).replace('\n', '')

    tweet_binary = toBinary(tweet_text)
    tweet_DNA = toDNA(tweet_binary)

    DNA_guide, mold, mutated_genome_seq = seq_constructor(genome_seq, mutation_position, tweet_DNA)

    tweet_binary_decoded = fromDNA(mutated_genome_seq, mutation_position, tweet_DNA)
    tweet_text_decoded = toString(tweet_binary_decoded)
```



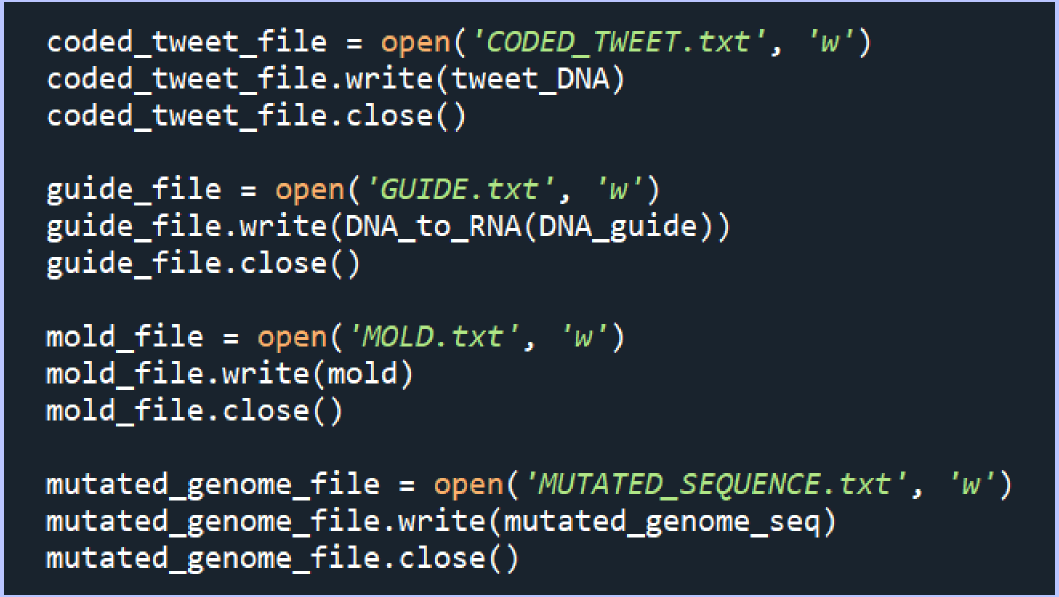
```python
def toBinary(tweet_text):

    tweet_ascii, tweet_binary = [], []
    for letter in tweet_text:
        tweet_ascii.append(ord(letter))
    for char in tweet_ascii:
        tweet_byte = format(char, "#010b")
        tweet_byte = tweet_byte[2:]
        tweet_binary.append(tweet_byte)

    return tweet_binary


def toDNA(tweet_binary):

    tweet_DNA = ""
    for byte in tweet_binary:
        for i in range(0, (len(byte)-1), 2):
            if byte[i] + byte[i+1] == "00":
                tweet_DNA += "A"
            elif byte[i] + byte[i+1] == "11":
                tweet_DNA += "T"
            elif byte[i] + byte[i+1] == "01":
                tweet_DNA += "G"
            elif byte[i] + byte[i+1] == "10":
                tweet_DNA += "C"

    return tweet_DNA
```



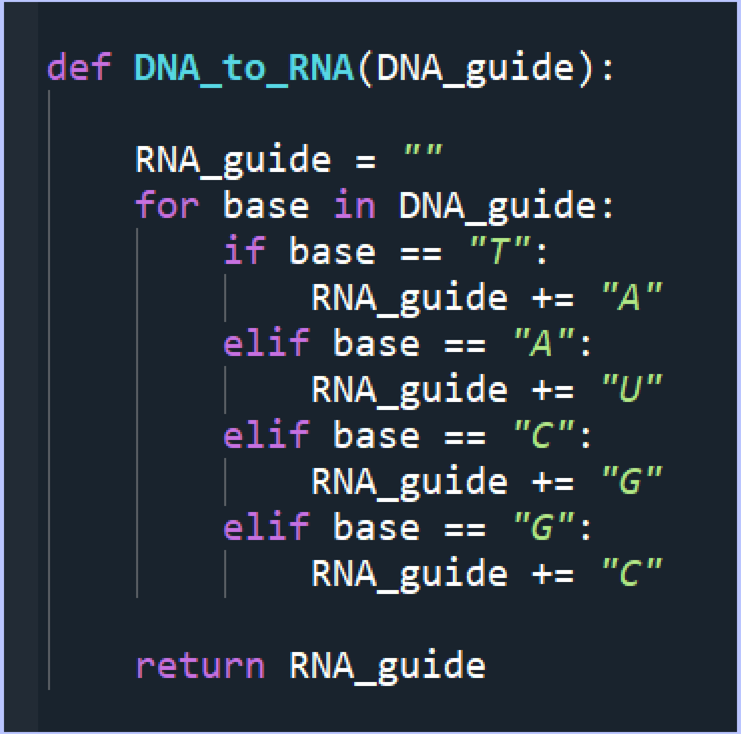
```
def toBinary(tweet_text):

    tweet_ascii, tweet_binary = [], []
    for letter in tweet_text:
        tweet_ascii.append(ord(letter))
    for char in tweet_ascii:
        tweet_byte = format(char, "#010b")
        tweet_byte = tweet_byte[2:]
        tweet_binary.append(tweet_byte)
```

```
GGAAGCTTGCATGCTTACAAGTATGCGGACAAGCCAGCAGGCACGCTAGCAGACAAGCGAGCG
GACAAGCTAGCAGACAAGCATGCAGGCTCGTGAGCCGGCGAGCAGGCGAACAAGCCGGCTCGC
GTGCGGGCTCGTGAGCGGACAAGCGAGCGGACAAGTAAGCGGGCGTGCAGGTGAGCCGGCTCG
CAGGTATACAAGTGTGCAGGTAAGCCGGTATGCCGGCTGGCAGGTATACAAGTAGGTGGGCGG
ACAAGTGCGCAGGCTGGCTTGTATACAAGCAGACAAGCTAGCTAGCGGGTGCGCAGGTAC
```

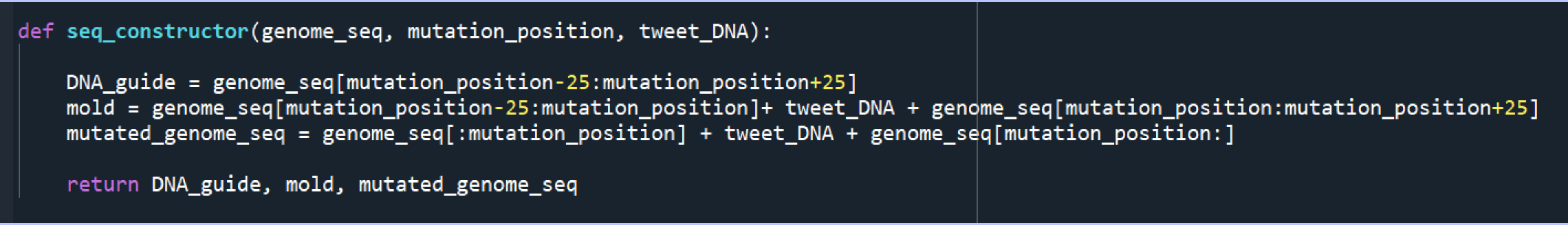
```
            if byte[i] + byte[i+1] == "00":
                tweet_DNA += "A"
            elif byte[i] + byte[i+1] == "11":
                tweet_DNA += "T"
            elif byte[i] + byte[i+1] == "01":
                tweet_DNA += "G"
            elif byte[i] + byte[i+1] == "10":
                tweet_DNA += "C"

    return tweet_DNA
```



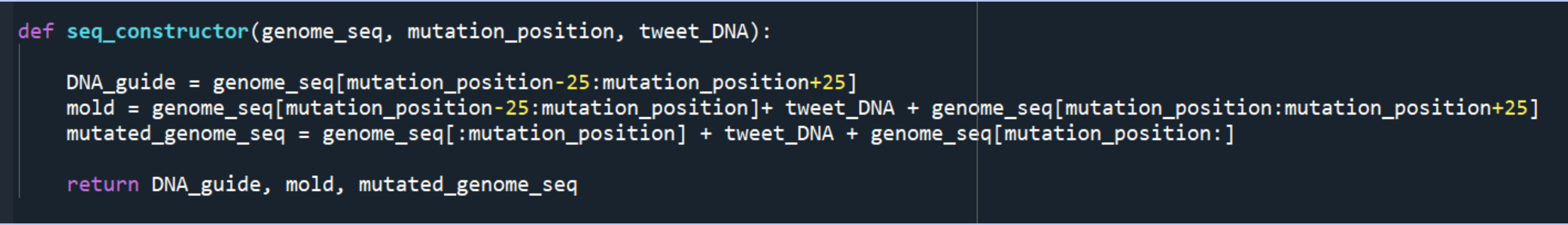
```python
coded_tweet_file = open('CODED_TWEET.txt', 'w')
coded_tweet_file.write(tweet_DNA)
coded_tweet_file.close()

guide_file = open('GUIDE.txt', 'w')
guide_file.write(DNA_to_RNA(DNA_guide))
guide_file.close()

mold_file = open('MOLD.txt', 'w')
mold_file.write(mold)
mold_file.close()

mutated_genome_file = open('MUTATED_SEQUENCE.txt', 'w')
mutated_genome_file.write(mutated_genome_seq)
mutated_genome_file.close()
```



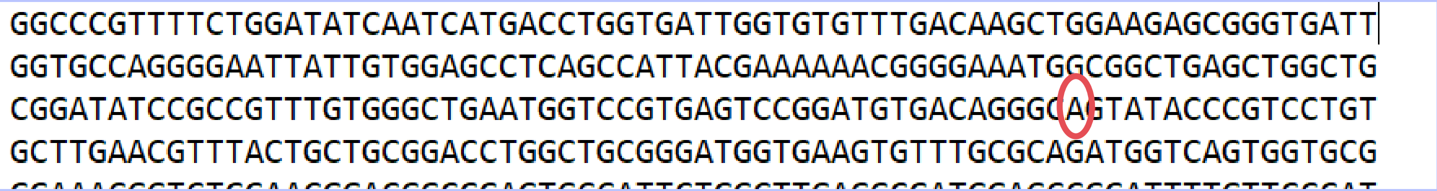
```python
def DNA_to_RNA(DNA_guide):

    RNA_guide = ""
    for base in DNA_guide:
        if base == "T":
            RNA_guide += "A"
        elif base == "A":
            RNA_guide += "U"
        elif base == "C":
            RNA_guide += "G"
        elif base == "G":
            RNA_guide += "C"

    return RNA_guide
```



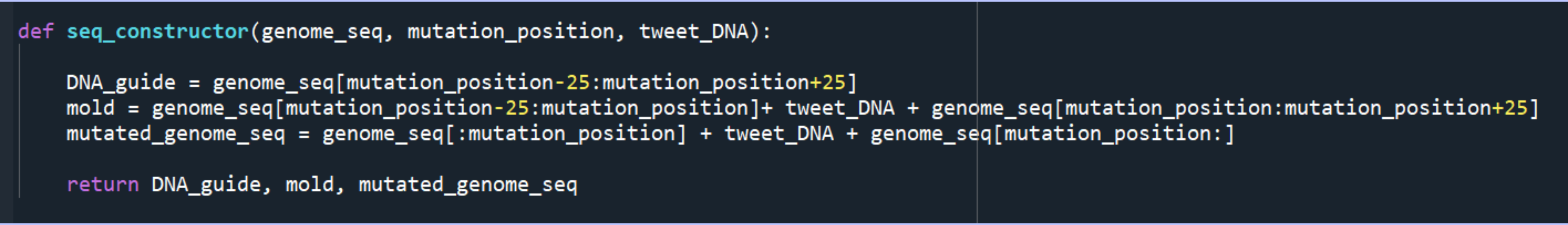
```python
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



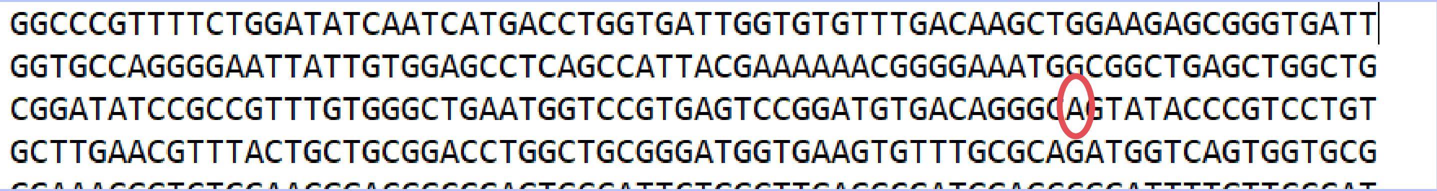
```python
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



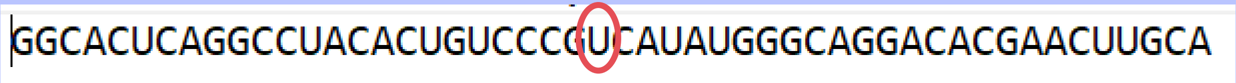
```
GGCCCGTTTTCTGGATATCAATCATGACCTGGTGATTGGTGTGTTTGACAAGCTGGAAGAGCGGGTGATT
GGTGCCAGGGGAATTATTGTGGAGCCTCAGCCATTACGAAAAAACGGGGAAATGGCGGCTGAGCTGGCTG
CGGATATCCGCCGTTTGTGGGCTGAATGGTCCGTGAGTCCGGATGTGACAGGGCAGTATACCCGTCCTGT
GCTTGAACGTTTACTGCTGCGGACCTGGCTGCGGGATGGTGAAGTGTTTGCGCAGATGGTCAGTGGTGCG
```



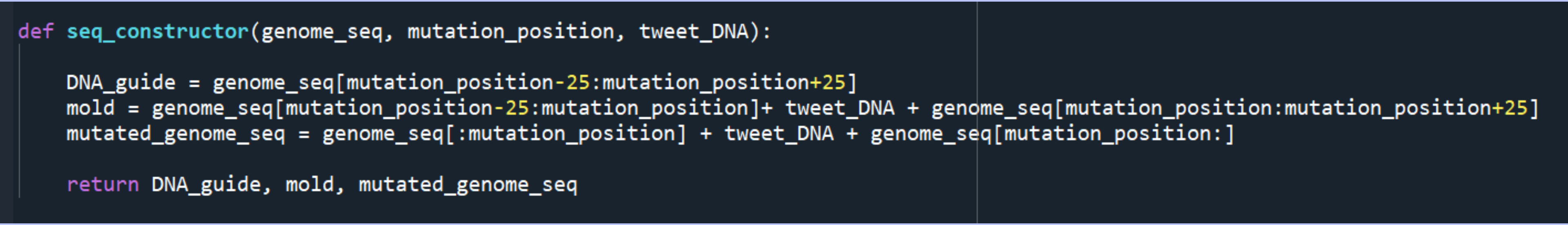
```python
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



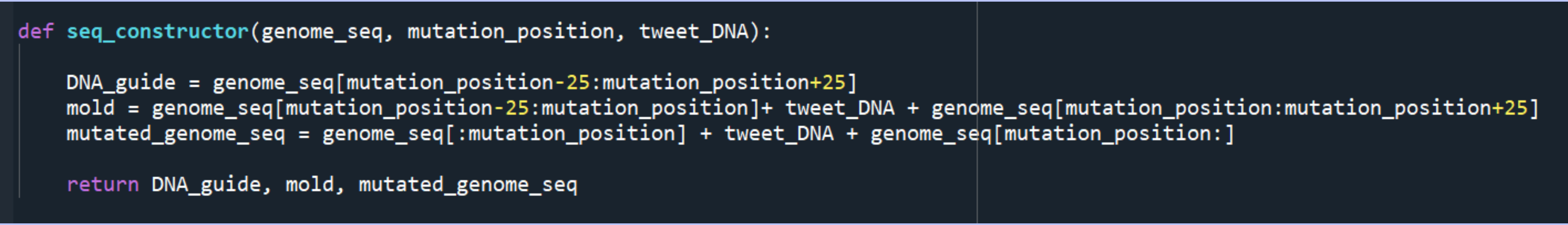
```
GGCCCGTTTTCTGGATATCAATCATGACCTGGTGATTGGTGTGTTTGACAAGCTGGAAGAGCGGGTGATT
GGTGCCAGGGGAATTATTGTGGAGCCTCAGCCATTACGAAAAAACGGGGAAATGGCGGCTGAGCTGGCTG
CGGATATCCGCCGTTTGTGGGCTGAATGGTCCGTGAGTCCGGATGTGACAGGGCAGTATACCCGTCCTGT
GCTTGAACGTTTACTGCTGCGGACCTGGCTGCGGGATGGTGAAGTGTTTGCGCAGATGGTCAGTGGTGCG
```



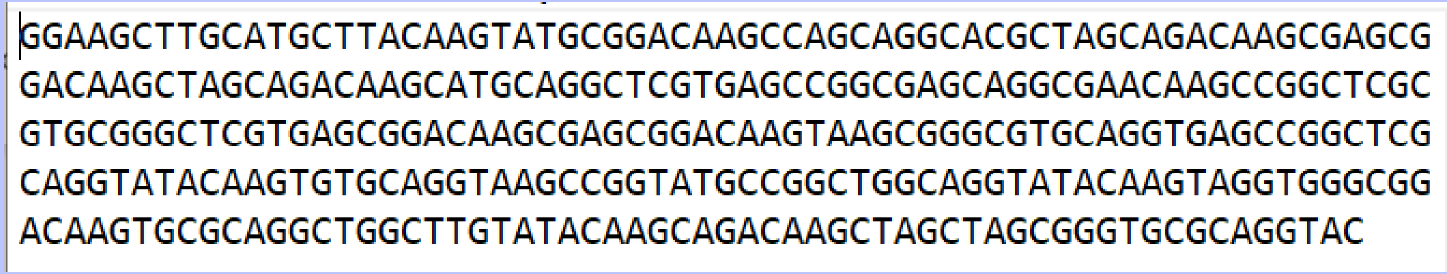
```
GGCACUCAGGCCUACACUGUCCCGUCAUAUGGGCAGGACACGAACUUGCA
```



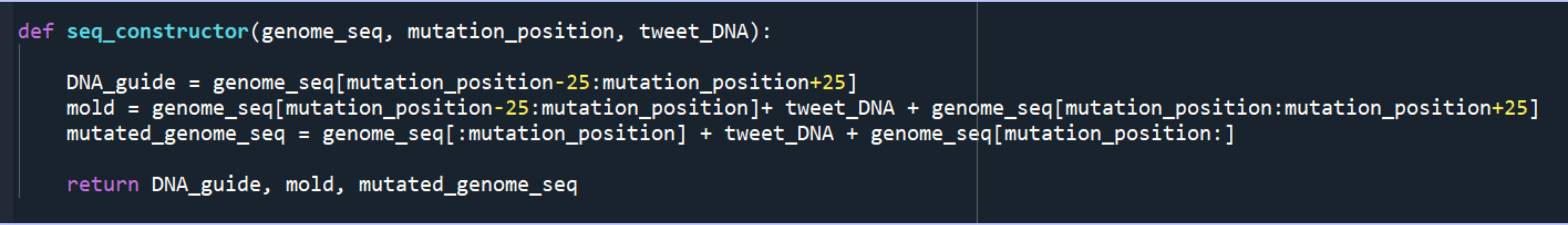
```python
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



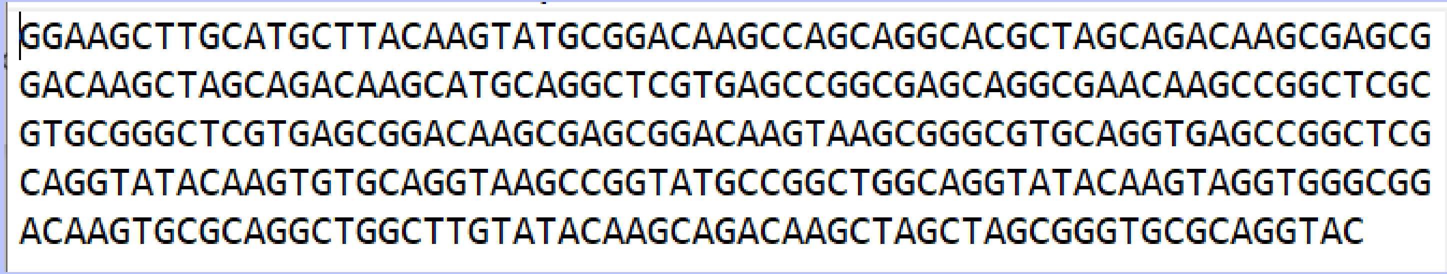
```
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



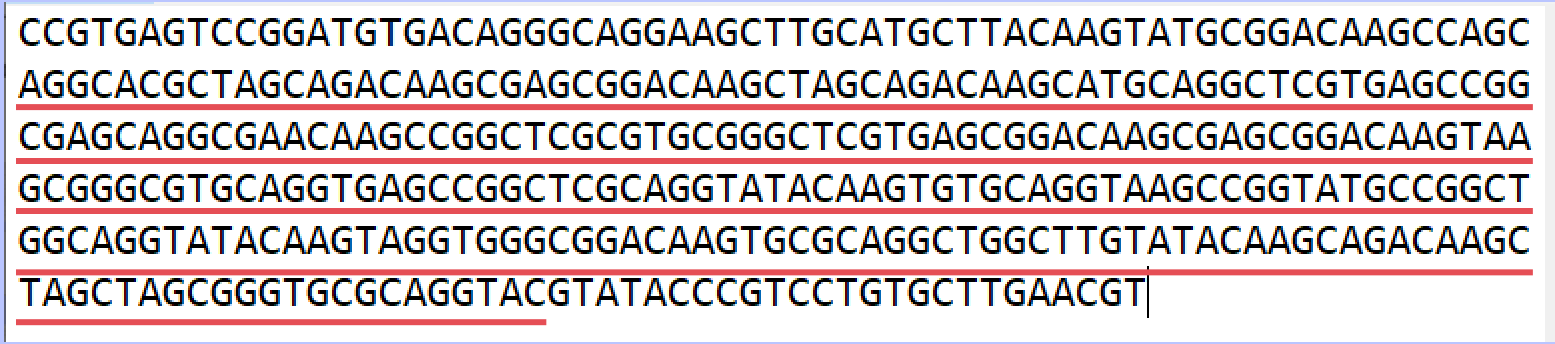
```
GGAAGCTTGCATGCTTACAAGTATGCGGACAAGCCAGCAGGCACGCTAGCAGACAAGCGAGCG
GACAAGCTAGCAGACAAGCATGCAGGCTCGTGAGCCGGCGAGCAGGCGAACAAGCCGGCTCGC
GTGCGGGCTCGTGAGCGGACAAGCGAGCGGACAAGTAAGCGGGCGTGCAGGTGAGCCGGCTCG
CAGGTATACAAGTGTGCAGGTAAGCCGGTATGCCGGCTGGCAGGTATACAAGTAGGTGGGCGG
ACAAGTGCGCAGGCTGGCTTGTATACAAGCAGACAAGCTAGCTAGCGGGTGCGCAGGTAC
```



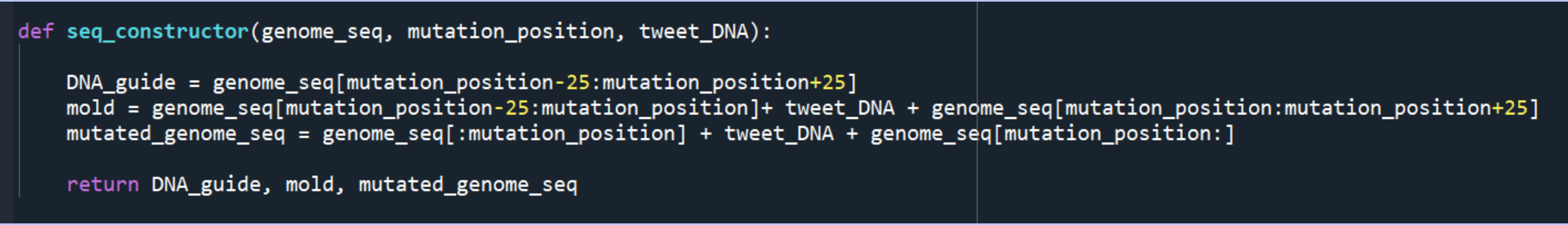
```python
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



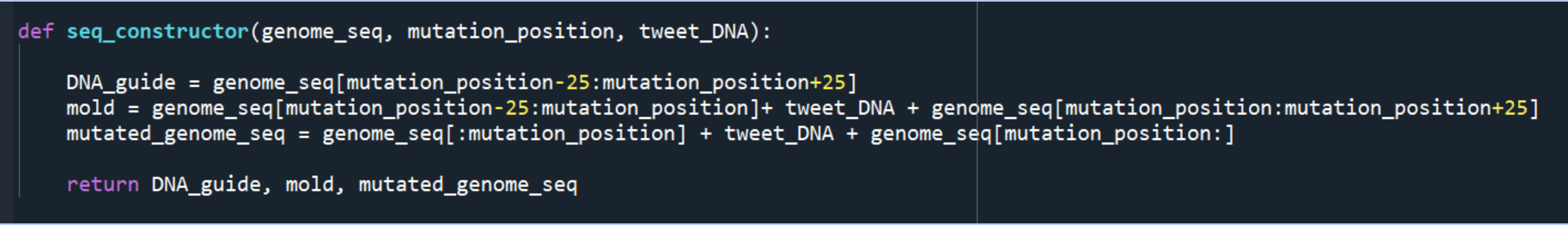
```
GGAAGCTTGCATGCTTACAAGTATGCGGACAAGCCAGCAGGCACGCTAGCAGACAAGCGAGCG
GACAAGCTAGCAGACAAGCATGCAGGCTCGTGAGCCGGCGAGCAGGCGAACAAGCCGGCTCGC
GTGCGGGCTCGTGAGCGGACAAGCGAGCGGACAAGTAAGCGGGCGTGCAGGTGAGCCGGCTCG
CAGGTATACAAGTGTGCAGGTAAGCCGGTATGCCGGCTGGCAGGTATACAAGTAGGTGGGCGG
ACAAGTGCGCAGGCTGGCTTGTATACAAGCAGACAAGCTAGCTAGCGGGTGCGCAGGTAC
```



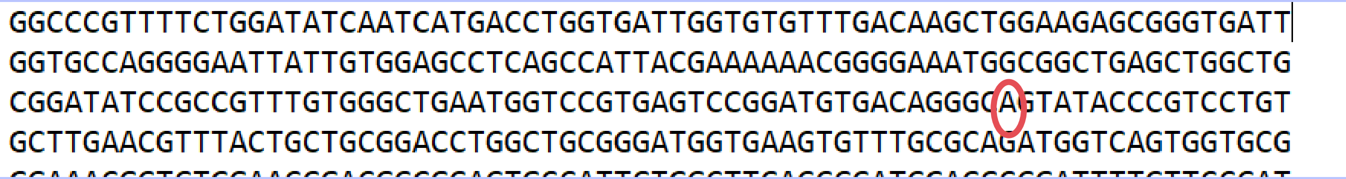
```
CCGTGAGTCCGGATGTGACAGGGCAGGAAGCTTGCATGCTTACAAGTATGCGGACAAGCCAGC
AGGCACGCTAGCAGACAAGCGAGCGGACAAGCTAGCAGACAAGCATGCAGGCTCGTGAGCCGG
CGAGCAGGCGAACAAGCCGGCTCGCGTGCGGGCTCGTGAGCGGACAAGCGAGCGGACAAGTAA
GCGGGCGTGCAGGTGAGCCGGCTCGCAGGTATACAAGTGTGCAGGTAAGCCGGTATGCCGGCT
GGCAGGTATACAAGTAGGTGGGCGGACAAGTGCGCAGGCTGGCTTGTATACAAGCAGACAAGC
TAGCTAGCGGGTGCGCAGGTACGTATACCCGTCCTGTGCTTGAACGT
```



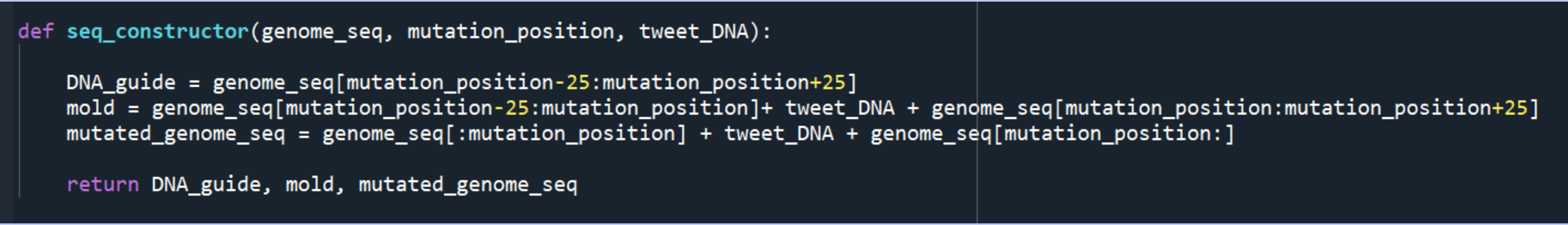
```python
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



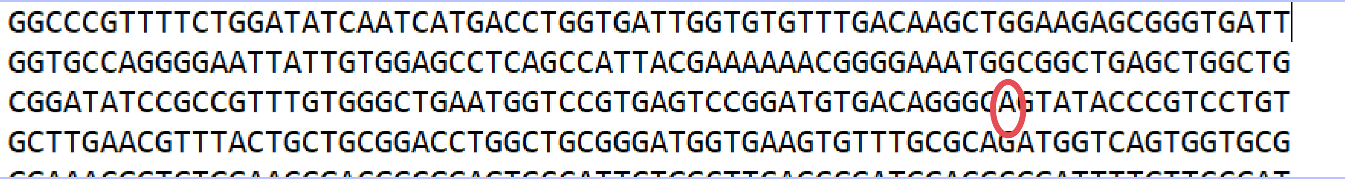
```python
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



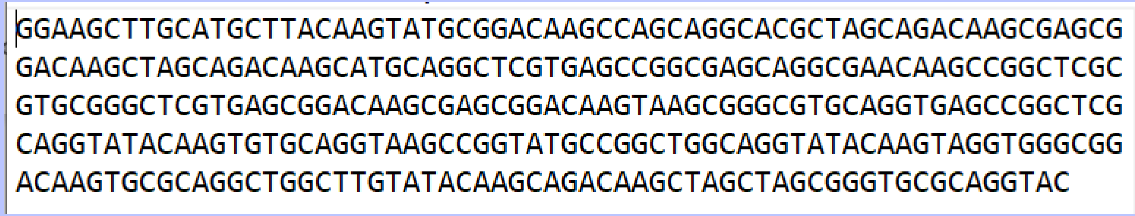
```
GGCCCGTTTTCTGGATATCAATCATGACCTGGTGATTGGTGTGTTTGACAAGCTGGAAGAGCGGGTGATT
GGTGCCAGGGGAATTATTGTGGAGCCTCAGCCATTACGAAAAAACGGGGAAATGGCGGCTGAGCTGGCTG
CGGATATCCGCCGTTTGTGGGCTGAATGGTCCGTGAGTCCGGATGTGACAGGGCAGTATACCCGTCCTGT
GCTTGAACGTTTACTGCTGCGGACCTGGCTGCGGGATGGTGAAGTGTTTGCGCAGATGGTCAGTGGTGCG
```



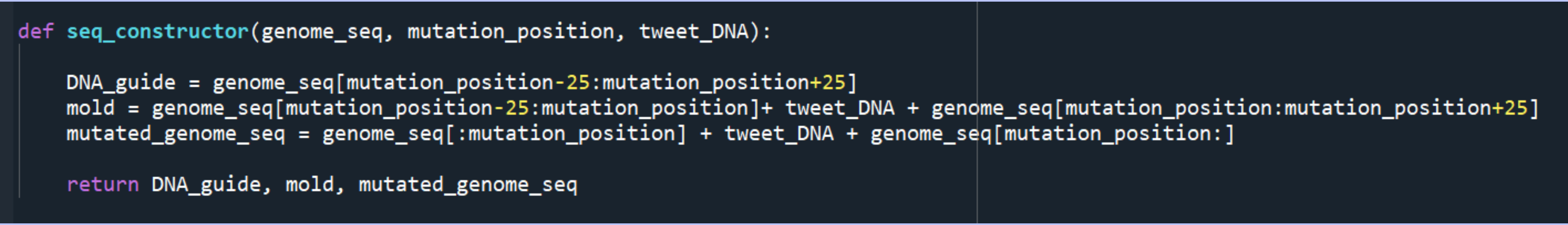
```python
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



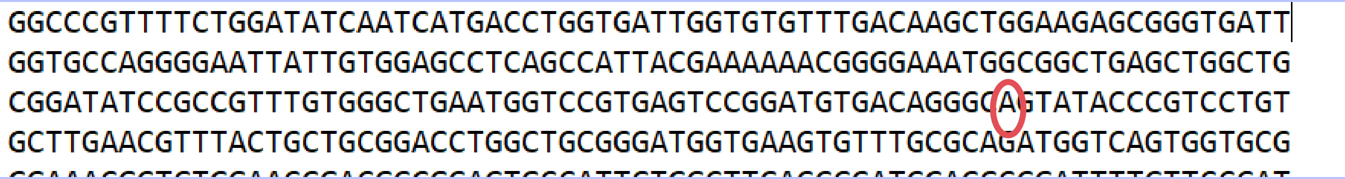
```
GGCCCGTTTTCTGGATATCAATCATGACCTGGTGATTGGTGTGTTTGACAAGCTGGAAGAGCGGGTGATT
GGTGCCAGGGGAATTATTGTGGAGCCTCAGCCATTACGAAAAAACGGGGAAATGGCGGCTGAGCTGGCTG
CGGATATCCGCCGTTTGTGGGCTGAATGGTCCGTGAGTCCGGATGTGACAGGGCAGTATACCCGTCCTGT
GCTTGAACGTTTACTGCTGCGGACCTGGCTGCGGGATGGTGAAGTGTTTGCGCAGATGGTCAGTGGTGCG
```



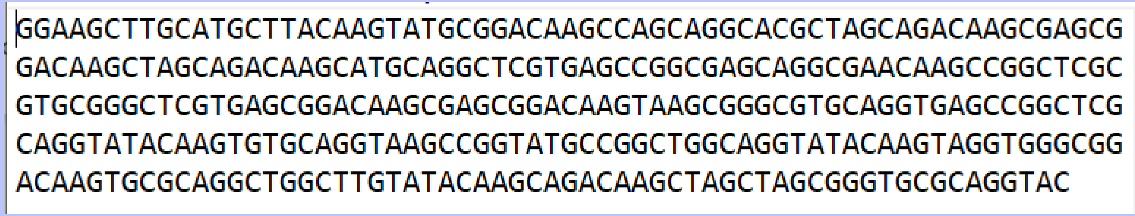
```
GGAAGCTTGCATGCTTACAAGTATGCGGACAAGCCAGCAGGCACGCTAGCAGACAAGCGAGCG
GACAAGCTAGCAGACAAGCATGCAGGCTCGTGAGCCGGCGAGCAGGCGAACAAGCCGGCTCGC
GTGCGGGCTCGTGAGCGGACAAGCGAGCGGACAAGTAAGCGGGCGTGCAGGTGAGCCGGCTCG
CAGGTATACAAGTGTGCAGGTAAGCCGGTATGCCGGCTGGCAGGTATACAAGTAGGTGGGCGG
ACAAGTGCGCAGGCTGGCTTGTATACAAGCAGACAAGCTAGCTAGCGGGTGCGCAGGTAC
```



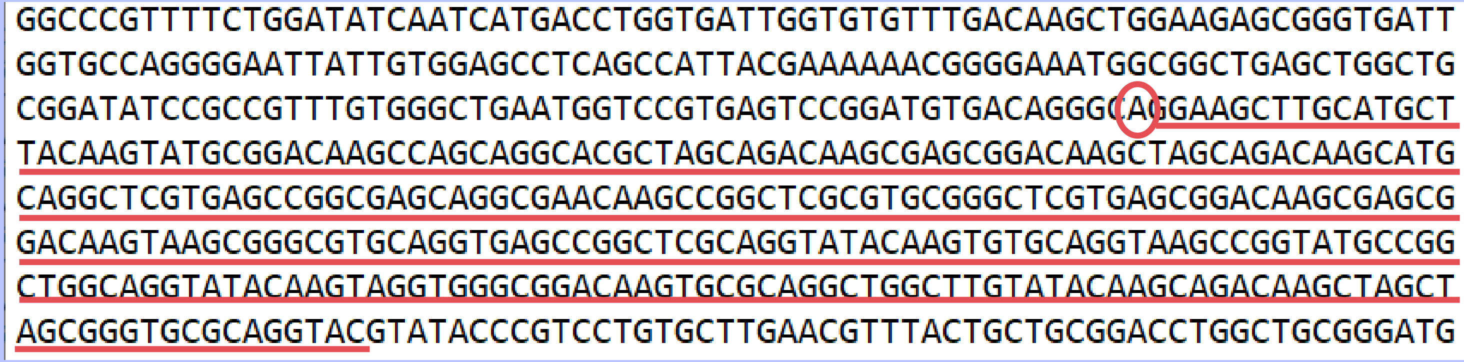
```python
def seq_constructor(genome_seq, mutation_position, tweet_DNA):

    DNA_guide = genome_seq[mutation_position-25:mutation_position+25]
    mold = genome_seq[mutation_position-25:mutation_position]+ tweet_DNA + genome_seq[mutation_position:mutation_position+25]
    mutated_genome_seq = genome_seq[:mutation_position] + tweet_DNA + genome_seq[mutation_position:]

    return DNA_guide, mold, mutated_genome_seq
```



```
GGCCCGTTTTCTGGATATCAATCATGACCTGGTGATTGGTGTGTTTGACAAGCTGGAAGAGCGGGTGATT
GGTGCCAGGGGAATTATTGTGGAGCCTCAGCCATTACGAAAAAACGGGGAAATGGCGGCTGAGCTGGCTG
CGGATATCCGCCGTTTGTGGGCTGAATGGTCCGTGAGTCCGGATGTGACAGGGCAGTATACCCGTCCTGT
GCTTGAACGTTTACTGCTGCGGACCTGGCTGCGGGATGGTGAAGTGTTTGCGCAGATGGTCAGTGGTGCG
```



```
GGAAGCTTGCATGCTTACAAGTATGCGGACAAGCCAGCAGGCACGCTAGCAGACAAGCGAGCG
GACAAGCTAGCAGACAAGCATGCAGGCTCGTGAGCCGGCGAGCAGGCGAACAAGCCGGCTCGC
GTGCGGGCTCGTGAGCGGACAAGCGAGCGGACAAGTAAGCGGGCGTGCAGGTGAGCCGGCTCG
CAGGTATACAAGTGTGCAGGTAAGCCGGTATGCCGGCTGGCAGGTATACAAGTAGGTGGGCGG
ACAAGTGCGCAGGCTGGCTTGTATACAAGCAGACAAGCTAGCTAGCGGGTGCGCAGGTAC
```



```
GGCCCGTTTTCTGGATATCAATCATGACCTGGTGATTGGTGTGTTTGACAAGCTGGAAGAGCGGGTGATT
GGTGCCAGGGGAATTATTGTGGAGCCTCAGCCATTACGAAAAAACGGGGAAATGGCGGCTGAGCTGGCTG
CGGATATCCGCCGTTTGTGGGCTGAATGGTCCGTGAGTCCGGATGTGACAGGGCAGGAAGCTTGCATGCT
TACAAGTATGCGGACAAGCCAGCAGGCACGCTAGCAGACAAGCGAGCGGACAAGCTAGCAGACAAGCATG
CAGGCTCGTGAGCCGGCGAGCAGGCGAACAAGCCGGCTCGCGTGCGGGCTCGTGAGCGGACAAGCGAGCG
GACAAGTAAGCGGGCGTGCAGGTGAGCCGGCTCGCAGGTATACAAGTGTGCAGGTAAGCCGGTATGCCGG
CTGGCAGGTATACAAGTAGGTGGGCGGACAAGTGCGCAGGCTGGCTTGTATACAAGCAGACAAGCTAGCT
AGCGGGTGCGCAGGTACGTATACCCGTCCTGTGCTTGAACGTTTACTGCTGCGGACCTGGCTGCGGGATG
```

3

Insertarlo en el ADN de la bacteria (y descodificarlo cuando queramos)

Comprar kit CRISPR DIY

3

Insertarlo en el ADN de la bacteria (y descodificarlo cuando queramos)

Comprar kit CRISPR DIY

Preparar la jeringa (con CAS, molde y guía)

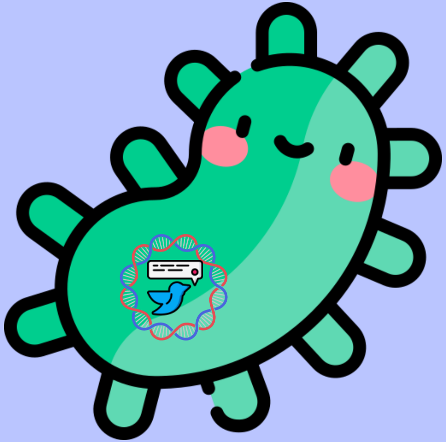
3

Insertarlo en el ADN de la bacteria (y descodificarlo cuando queramos)

Comprar kit CRISPR DIY

Preparar la jeringa (con CAS, molde y guía)

Inyectar en la bacteria



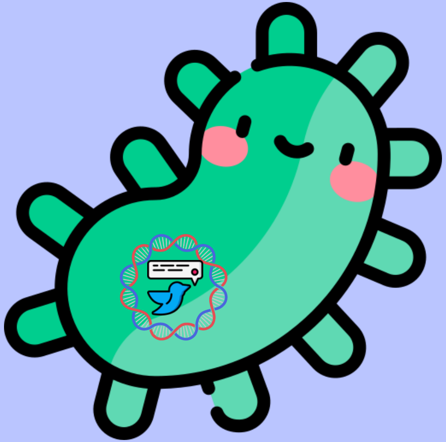
3

Insertarlo en el ADN de la bacteria (y descodificarlo cuando queramos)

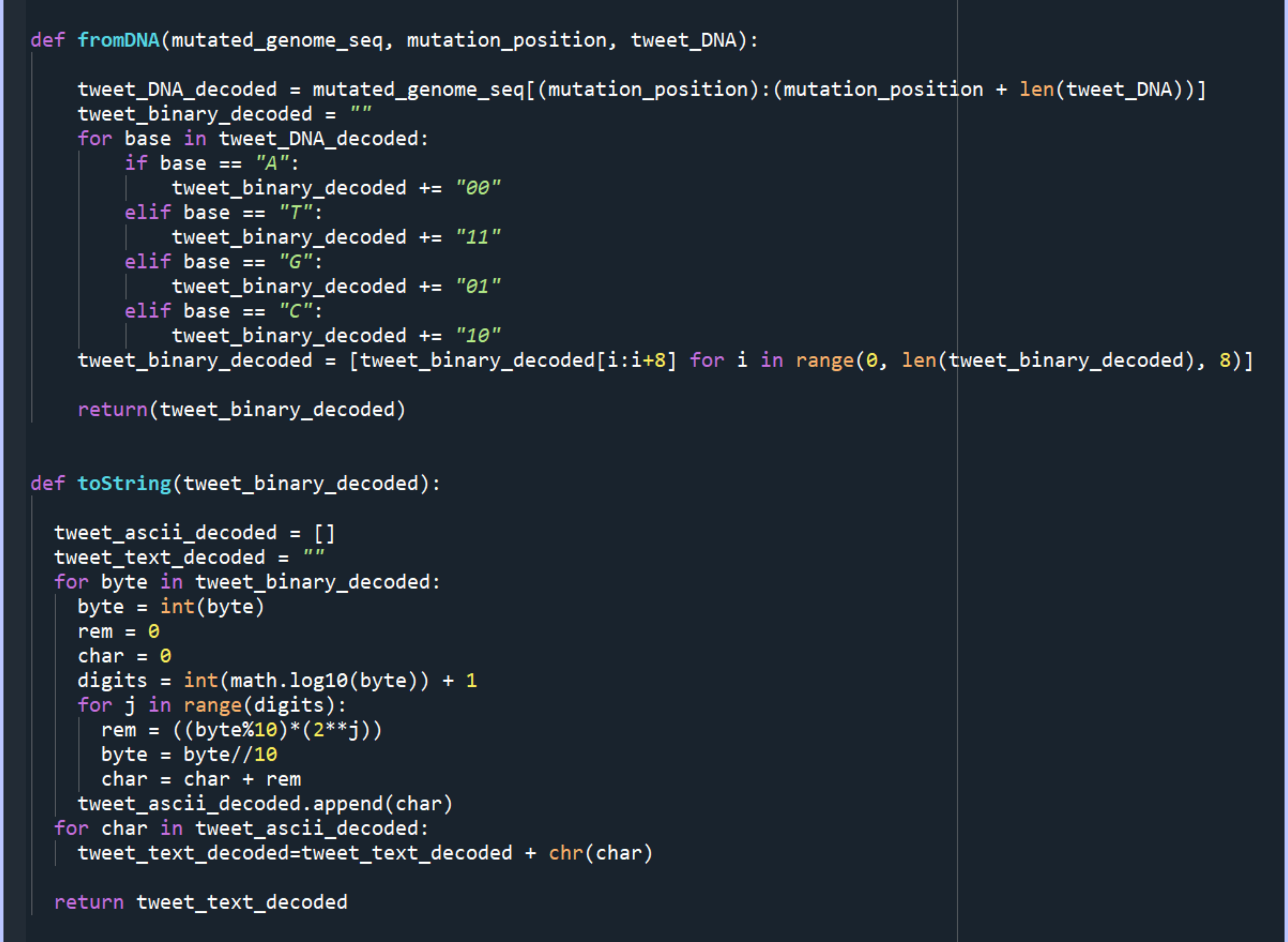
Comprar kit CRISPR DIY

Preparar la jeringa (con CAS, molde y guía)

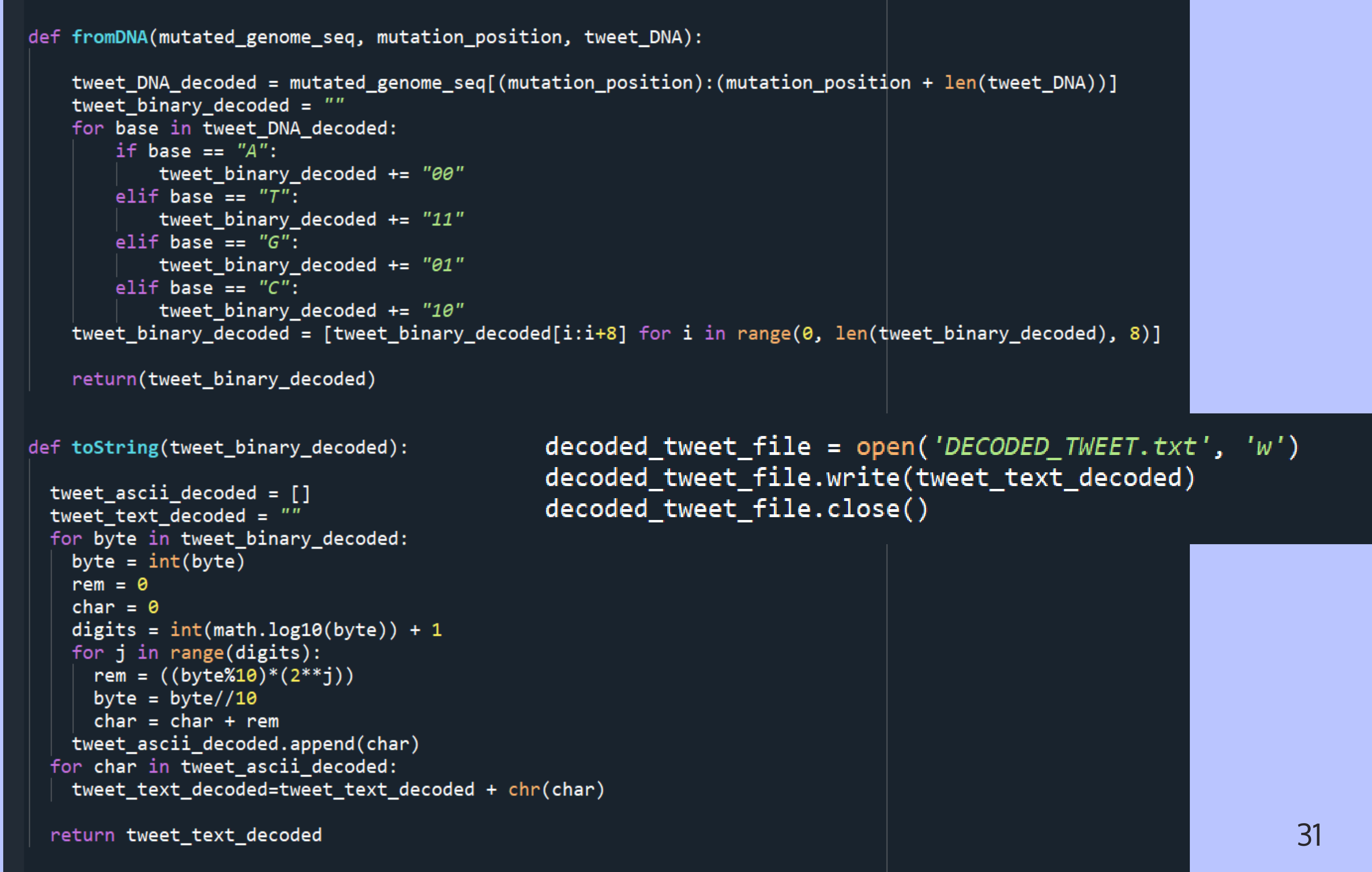
Inyectar en la bacteria



¡Y listo!



```python
def fromDNA(mutated_genome_seq, mutation_position, tweet_DNA):

    tweet_DNA_decoded = mutated_genome_seq[(mutation_position):(mutation_position + len(tweet_DNA))]
    tweet_binary_decoded = ""
    for base in tweet_DNA_decoded:
        if base == "A":
            tweet_binary_decoded += "00"
        elif base == "T":
            tweet_binary_decoded += "11"
        elif base == "G":
            tweet_binary_decoded += "01"
        elif base == "C":
            tweet_binary_decoded += "10"
    tweet_binary_decoded = [tweet_binary_decoded[i:i+8] for i in range(0, len(tweet_binary_decoded), 8)]

    return(tweet_binary_decoded)


def toString(tweet_binary_decoded):

  tweet_ascii_decoded = []
  tweet_text_decoded = ""
  for byte in tweet_binary_decoded:
    byte = int(byte)
    rem = 0
    char = 0
    digits = int(math.log10(byte)) + 1
    for j in range(digits):
      rem = ((byte%10)*(2**j))
      byte = byte//10
      char = char + rem
    tweet_ascii_decoded.append(char)
  for char in tweet_ascii_decoded:
    tweet_text_decoded=tweet_text_decoded + chr(char)

  return tweet_text_decoded
```



```
def fromDNA(mutated_genome_seq, mutation_position, tweet_DNA):

    tweet_DNA_decoded = mutated_genome_seq[(mutation_position):(mutation_position + len(tweet_DNA))]
    tweet_binary_decoded = ""
    for base in tweet_DNA_decoded:
        if base == "A":
            tweet_binary_decoded += "00"
        elif base == "T":
            tweet_binary_decoded += "11"
        elif base == "G":
            tweet_binary_decoded += "01"
        elif base == "C":
            tweet_binary_decoded += "10"
    tweet_binary_decoded = [tweet_binary_decoded[i:i+8] for i in range(0, len(tweet_binary_decoded), 8)]

    return(tweet_binary_decoded)


def toString(tweet_binary_decoded):

  tweet_ascii_decoded = []
  tweet_text_decoded = ""
  for byte in tweet_binary_decoded:
    byte = int(byte)
    rem = 0
    char = 0
    digits = int(math.log10(byte)) + 1
    for j in range(digits):
      rem = ((byte%10)*(2**j))
      byte = byte//10
      char = char + rem
    tweet_ascii_decoded.append(char)
  for char in tweet_ascii_decoded:
    tweet_text_decoded=tweet_text_decoded + chr(char)

  return tweet_text_decoded
```

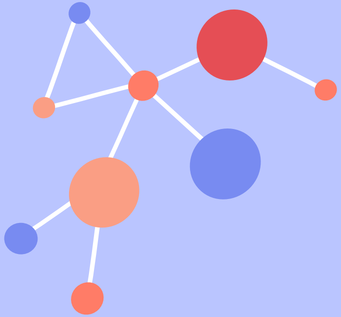
```
decoded_tweet_file = open('DECODED_TWEET.txt', 'w')
decoded_tweet_file.write(tweet_text_decoded)
decoded_tweet_file.close()
```

5

APLICACIONES TERAPÉUTICAS

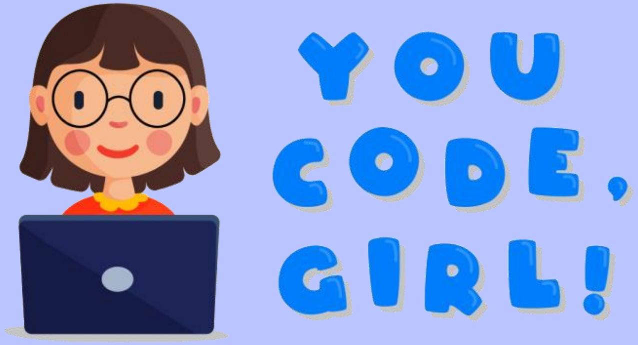
APLICACIONES TERAPÉUTICAS

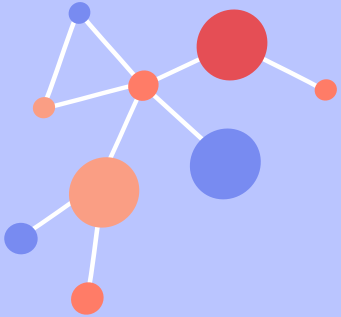
- Edición de genes que producen enfermedades genéticas (epidermólisis bullosa, anemia de Falconi)
- Células CAR-T contra el cáncer
- Terapia antiviral (SARS-CoV-2, VIH)
- Lucha contra enfermedades infecciosas (malaria, fiebre amarilla)



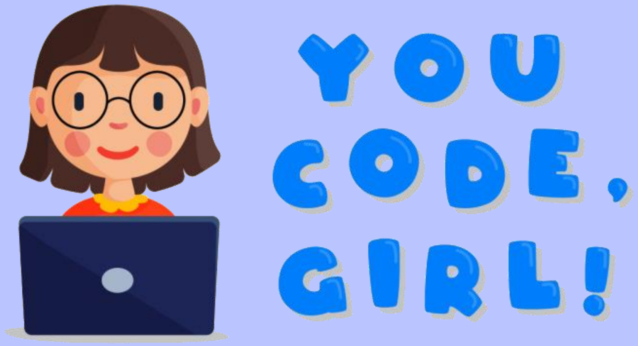
¡Gracias!

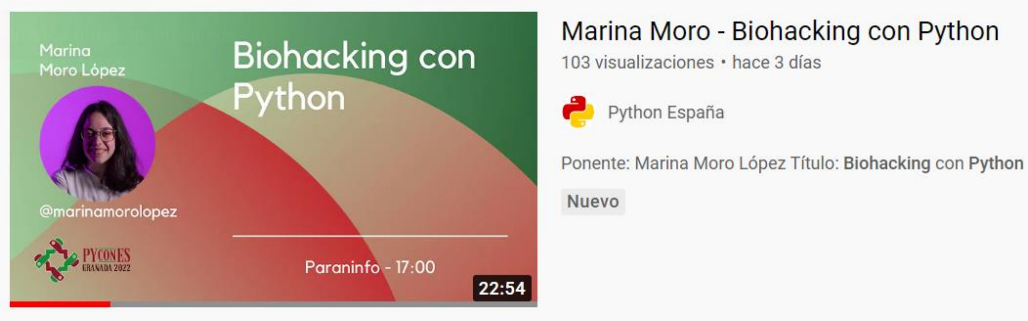
¡Gracias!





¡Gracias!





Marina Moro López

@marinamorolopez

PYCONES GRANADA 2022

Biohacking con Python

Paraninfo - 17:00

22:54

Marina Moro - Biohacking con Python

103 visualizaciones • hace 3 días

Python España

Ponente: Marina Moro López Título: **Biohacking** con **Python**

Nuevo

¡Gracias!

¿Preguntas?

marinaml5598@gmail.com

@marinamorolopez

Marina Moro López

marinamorolopez

marinamorolopez / biohacking-bitup2022 Public

BIT UP

pyladies BARCELONA

UNIVERSITAT DE BARCELONA