

Python_Notebook

March 18, 2021

```
[36]: ! pip install biopython
      ! pip install requests
```

```
Requirement already satisfied: biopython in
/home/waldeyr/.conda/envs/notebooks/lib/python3.9/site-packages (1.78)
Requirement already satisfied: numpy in
/home/waldeyr/.conda/envs/notebooks/lib/python3.9/site-packages (from biopython)
(1.20.1)
Collecting requests
  Using cached requests-2.25.1-py2.py3-none-any.whl (61 kB)
Collecting idna<3,>=2.5
  Using cached idna-2.10-py2.py3-none-any.whl (58 kB)
Collecting chardet<5,>=3.0.2
  Using cached chardet-4.0.0-py2.py3-none-any.whl (178 kB)
Collecting urllib3<1.27,>=1.21.1
  Downloading urllib3-1.26.4-py2.py3-none-any.whl (153 kB)
    |                               | 153 kB 3.0 MB/s eta 0:00:01
Requirement already satisfied: certifi>=2017.4.17 in
/home/waldeyr/.conda/envs/notebooks/lib/python3.9/site-packages (from requests)
(2020.12.5)
Installing collected packages: urllib3, idna, chardet, requests
Successfully installed chardet-4.0.0 idna-2.10 requests-2.25.1 urllib3-1.26.4
```

```
[97]: # Importando as bibliotecas de Biopython que serao usadas
      from Bio.Seq import Seq
      from Bio.Data import CodonTable
      from Bio import SeqIO
      from Bio import Align
      from Bio.KEGG import REST
      from Bio.KEGG import Enzyme
      from Bio.KEGG import REST
      from Bio import ExPASy
      from Bio import SeqIO
      import requests
```

```
[17]: dna_exemplo = Seq("ATGgccattgTAATGGGCCGCTgaaagggtGCCCGATAG") # dna com
      ↪ nucleotideos em maiusculas e minusculas
```

```
dna_exemplo = dna_exemplo.upper() # converte todos os nucleotideos para
↪maiusculas
print(dna)
```

ATGGCCATTGTAATGGGCCGCTGAAAGGGTGGCCGATAG

```
[14]: # Gerar a fita reversa complementar de da sequencia dna_exemplo
rev_compl_dna_exemplo = dna_exemplo.reverse_complement()
print(rev_compl_dna_exemplo)
```

CTATCGGGCACCTTTTCAGCGGCCATTACAATGGCCAT

```
[18]: # Gerar o RNA mensageiro (transcrição)
rna_mensageiro = dna_exemplo.transcribe()
print(rna_mensageiro)
```

AUGGCCAUUGUAAUGGGCCGCUGAAAGGGUGCCCGAUAG

```
[24]: # Gerar a tradução do RNA mensageiro com a tabela de codigo genetico padrao
tabela_padrao = CodonTable.unambiguous_dna_by_id[1]
print(tabela_padrao)
aminoacidos = rna_mensageiro.translate()
print(aminoacidos)
```

Table 1 Standard, SGC0

	T	C	A	G	
T	TTT F	TCT S	TAT Y	TGT C	T
T	TTC F	TCC S	TAC Y	TGC C	C
T	TTA L	TCA S	TAA Stop	TGA Stop	A
T	TTG L(s)	TCG S	TAG Stop	TGG W	G
C	CTT L	CCT P	CAT H	CGT R	T
C	CTC L	CCC P	CAC H	CGC R	C
C	CTA L	CCA P	CAA Q	CGA R	A
C	CTG L(s)	CCG P	CAG Q	CGG R	G
A	ATT I	ACT T	AAT N	AGT S	T
A	ATC I	ACC T	AAC N	AGC S	C
A	ATA I	ACA T	AAA K	AGA R	A
A	ATG M(s)	ACG T	AAG K	AGG R	G
G	GTT V	GCT A	GAT D	GGT G	T
G	GTC V	GCC A	GAC D	GGC G	C
G	GTA V	GCA A	GAA E	GGA G	A
G	GTG V	GCG A	GAG E	GGG G	G

MAIVMGR*KGAR*

```
[25]: # Gerar a tradução do RNA mensageiro com a tabela de código genético para dna
      ↳ mitocondrial de vertebrados
tabela_mitocondrial = CodonTable.unambiguous_dna_by_id[2]
print(tabela_mitocondrial)
aminoacidos = rna_mensageiro.translate(table="Vertebrate Mitochondrial")
print(aminoacidos)
```

Table 2 Vertebrate Mitochondrial, SGC1

	T	C	A	G	
T	TTT F	TCT S	TAT Y	TGT C	T
T	TTC F	TCC S	TAC Y	TGC C	C
T	TTA L	TCA S	TAA Stop	TGA W	A
T	TTG L	TCG S	TAG Stop	TGG W	G
C	CTT L	CCT P	CAT H	CGT R	T
C	CTC L	CCC P	CAC H	CGC R	C
C	CTA L	CCA P	CAA Q	CGA R	A
C	CTG L	CCG P	CAG Q	CGG R	G
A	ATT I(s)	ACT T	AAT N	AGT S	T
A	ATC I(s)	ACC T	AAC N	AGC S	C
A	ATA M(s)	ACA T	AAA K	AGA Stop	A
A	ATG M(s)	ACG T	AAG K	AGG Stop	G
G	GTT V	GCT A	GAT D	GGT G	T
G	GTC V	GCC A	GAC D	GGC G	C
G	GTA V	GCA A	GAA E	GGA G	A
G	GTG V(s)	GCG A	GAG E	GGG G	G

MAIVMGRWKGAR*

```
[21]: # Gerar a tradução do RNA mensageiro com a tabela de código genético para dna
      ↳ mitocondrial de vertebrados
aminoacidos = rna_mensageiro.translate(table="Bacterial")
print(aminoacidos)
```

MAIVMGR*KGAR*

```
[41]: # Baixar para notebook o genoma do Sars-Cov-2 em formato genbank
url_genoma_sars_cov_2 = "https://raw.githubusercontent.com/waldeyr/
      ↳ apostila_bioinfo/main/Sars_Cov_2_Genoma.gb"
r = requests.get(url_genoma_sars_cov_2, allow_redirects=True)
open('genoma_sars_cov_2.gb', 'wb').write(r.content)
```

[41]: 78471

```
[45]: # Ler o genoma em formato genbank e mostrar o NCBI ID do genoma
genoma_sars_cov_2 = SeqIO.read("genoma_sars_cov_2.gb", "genbank")
print(genoma_sars_cov_2.id)
```

NC_045512.2

```
[54]: # Imprimindo algumas características de interesse
print("NCBI ID: %s;\t Tamanho do genoma: %i bp;\t Qtd features: %i;\t Qtd de_\n
      ↳ genes anotados: %i;" % (
        genoma_sars_cov_2.id, # NCBI ID
        len(genoma_sars_cov_2), # tamanho do genoma em pares de bases
        len(genoma_sars_cov_2.features), # quantidades de features
        len(genoma_sars_cov_2.annotations) # quantidade de genes anotados
    ))
```

NCBI ID: NC_045512.2; Tamanho do genoma: 29903 bp; Qtd features: 57;
Qtd de genes anotados: 13;

```
[57]: # Gerar um fasta com os aminoácidos da região codificadora do gene da proteína_\n
      ↳ spike
for seq_feature in genoma_sars_cov_2.features: # ler as features do genoma
    if seq_feature.type == "CDS": # misc_feature, gene, misc_RNA, CDS
        for key, value in seq_feature.qualifiers.items(): # ler qualificadores_\n
            ↳ de cada feature
                if key == "translation": # verificar o qualificador de interesse_\n
                    ↳ que é translation
                        if seq_feature.qualifiers['locus_tag'][0] == "GU280_gp02":#\n
                            ↳ Gene da proteína spike
                                print(f">{seq_feature.qualifiers['locus_tag'][0]}") #_\n
                            ↳ imprimir o nome do gene
                                print(f"{str(value[0])}") #imprimir a sequência da região_\n
                            ↳ codificadora do gene traduzida para proteína
```

>GU280_gp02

MFVFLVLLPLVSSQCVNLTTRTQLPPAYTNSFTRGVYYPDKVFRSSVLHSTQDLFLPFFSNVTWFWHAIHVSNGTKRFD
NPVLPFNDGVYFASTEKSNIIRGWIFGTTLDSKTQSLILVNNATNVVIKVCEFCNDPFLGVYYHKNNKSWESEFRVY
SSANNCTFEYVSQPFLMDLEGKQGNFKNLREFVFKNIDGYFKIYSKHTPINLVRDLPPQGFSALEPLVDLPIGINITRFQT
LLALHRSYLTPGDSSSGWTAGAAAYVGYLQPRTFLLKYNENGTITDAVDCALDPLSETKCTLKSFTVEKGIYQTSNFRV
QPTESIVRFPNITNLCPFGEVFNATRFASVYAWNRKRISNCVADYSVLNSASFSTFKCYGVSPTKLNLCFTNVYADSF
VIRGDEVRQIAPGQTGKIADYNYKLDDFTGCVIAWNSNNLDSKVGNNYLYRLFRKSNLKPFERDISTEYQAGSTPC
NGVEGFNCYFPLQSYGFQPTNGVGYQPYRVVLSFELLHAPATVCGPKKSTNLVKNKCVNFNFNGLTGTGVLTESNKKFL
PFQQFGRDIADTTDAVRDPQTLEILDITPCSGGVSIVITPGTNTSNQVAVLYQDVNCTEVPVAIHADQLTPTWRVYSTGS
NVFQTRAGCLIGAETHVNNSEYCDIPGAGICASYQTQTNPRRARSVASQSIIAYTMSLGAENSVAYSNNISIAIPTNFTI
SVTTEILPVSMTKTSVDCTMYICGDSTECNLLLQYGSFCTQLNRALTGIAVEQDKNTQEVFAQVKQIYKTPPIKDFGGF
NFSQILPDPSKPSKRSFIEDLLFNKVTLDAGFIKQYGDCLGDIAARDLICAQKFNGLTVLPPLLTEMLIAQYTSALLAG

```
TITSGWTFGAGAALQIPFAMQMAYRFNGIGVTQNVLYENQKLIANQFNSAIGKIQDSLSTASALGKLQDVVNQNAQALN
TLVKQLSSNFGAISSVLNDILSRDKVEAEVQIDRLITGRLQSLQTYVTQQLIRAAEIRASANLAATKMSECVLGQSKRV
DFCGKGYHLSFPQSAPHGVVFLHVTYVPAQEKNFTTAPAICHGDKAHFPREGVFSNGTHWFVTQRNFYEPQIITDNT
FVSGNCDVVIGIVNNTVYDPLQPELDSFKEELDKYFKNHTSPDVLGDISGINASVVNIQKEIDRLNEVAKNLNESLIDL
QELGKYEYIKWPWYIWLGFIAGLIAIVMVTIMLCCMTSCCCLKGCSCGSCCKFDEDDSEPVLKGVKLHYT
```

```
[51]: # Alinhamentos entre sequencias de anticorpos humanos para a proteina spike do
      ↪ sars-cov-2 (GU280_gp02)
```

```
NC_045512.2 29903 57 1 13
```

```
[61]: # Baixar arquivo com 4 sequencias de anticorpos humanos para a proteina spike
      ↪ do sars-cov-2 (GU280_gp02)
url_anticorpos = "https://raw.githubusercontent.com/waldeyr/apostila_bioinfo/
      ↪ main/imunoglobulinas_humanas_cadeia_pesada.fasta"
r = requests.get(url_anticorpos, allow_redirects=True)
open('anticorpos_sars_cov_2.fasta', 'wb').write(r.content)
```

```
[61]: 2011
```

```
[90]: # Comparar duas sequencias de anticorpos com alinhamento global pelo algoritmo
      ↪ de Needleman e Wunsch
lista_de_atcorpos = [] # lista para guardar separadamente cada sequencia do
      ↪ arquivo multifasta
for seq_record in SeqIO.parse("anticorpos_sars_cov_2.fasta", "fasta"):
    lista_de_atcorpos.append(str(seq_record.seq)) # adicionando cada sequencia
      ↪ a lista

alinhador = Align.PairwiseAligner()
alinhador.match_score = 1 # pontuacao para match
alinhador.mismatch_score = -2 # pontuacao para mismatch
alinhador.gap_score = -5 # pontuacao para gap

alinhamentos = alinhador.align(lista_de_atcorpos[0], lista_de_atcorpos[1])
for alinhamento in alinhamentos:
    print(alinhamento)
```

```
CAGGTGCAGCTGGTGGAGTCTGGGGGAGGCGTGGTCCAGCCTGGGAGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATT
CACCTTCAGTAGCTATGCTATGCACTGGGTCCGCCAGGCTCCAGGCAAGGGGCTGGAGTGGGTGGCAGGTATATCATATG
ATGGAAGCAATAAAATACTACGCAGACTCCGTGAAGGGCCGATTACCATCTCCAGAGACAATTCCAAGAACACGCTGTAT
CTGCAAATGAACAGCCTGAGAGCTGAGGACACGGCTGTGTATTACTGTGCGAGAGCGGACACTATGGTTCGGGGAACCTA
TTTTGAGTACTGGGGCCAGGGAACCCCTGGTCACCGTCTCCTCA
.|||||...|||||...|||||...|||||...|||||...|||||...|||||...|||||...|||||...
|||||.|||.|||...|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.
|||||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.
|||||...|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.|||.
|---||.|||||...|||||...|||||...|||||...|||||...|||||...|||||...|||||...
GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCTGGTCCAGCCTGGGGGCTCCCTGAGACTCTCCTGTGTAGCCTCTGGATT
```


CACCTTTAGTTTCTATTGGATGAGCTGGGTCCGCCAGGCTCCAGGAAAGGGGCTCGAGTGGGTGGCCAACATAAAGCAAG
ATGGAGGTGAGAAATACTATGTGGACTCTGTGAAGGGCCGATTACCATCTCCAGAGACAACGCCAAGAAGCACTGTAT
CTGCAAATGAACAGCCTGAGAGCCGAGGACACGGCTGTGTATTACTGTGCGAGA-CTGTCTGGCAGCAGCTGGG-ACT
---TT-GACTACTGGGGCCAGGGAACCCCTGGTCACCGTCTCCTCA

CAGGTGCAGCTGGTGGAGTCTGGGGGAGGCGTGGTCCAGCCTGGGAGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATT
CACCTTCAGTAGCTATGCTATGCACTGGGTCCGCCAGGCTCCAGGCAAGGGGCTGGAGTGGGTGGCAGGTATATCATATG
ATGGAAGCAATAAATACTACGCAGACTCCGTGAAGGGCCGATTACCATCTCCAGAGACAATTCCAAGAACACGCTGTAT
CTGCAAATGAACAGCCTGAGAGCTGAGGACACGGCTGTGTATTACTGTGCGAGAGCGGACACTATGGTTCGGGGAACTTA
TTTTGAGTACTGGGGCCAGGGAACCCCTGGTCACCGTCTCCTCA

.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
-|-|-|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
-|-|-|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.

GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCTGGTCCAGCCTGGGGGTCCCTGAGACTCTCCTGTGTAGCCTCTGGATT
CACCTTTAGTTTCTATTGGATGAGCTGGGTCCGCCAGGCTCCAGGAAAGGGGCTCGAGTGGGTGGCCAACATAAAGCAAG
ATGGAGGTGAGAAATACTATGTGGACTCTGTGAAGGGCCGATTACCATCTCCAGAGACAACGCCAAGAAGCACTGTAT
CTGCAAATGAACAGCCTGAGAGCCGAGGACACGGCTGTGTATTACTGTGCGAGA-CTGTCTGGCAGCAGCTGGGA-C-T
--TT-GACTACTGGGGCCAGGGAACCCCTGGTCACCGTCTCCTCA

CAGGTGCAGCTGGTGGAGTCTGGGGGAGGCGTGGTCCAGCCTGGGAGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATT
CACCTTCAGTAGCTATGCTATGCACTGGGTCCGCCAGGCTCCAGGCAAGGGGCTGGAGTGGGTGGCAGGTATATCATATG
ATGGAAGCAATAAATACTACGCAGACTCCGTGAAGGGCCGATTACCATCTCCAGAGACAATTCCAAGAACACGCTGTAT
CTGCAAATGAACAGCCTGAGAGCTGAGGACACGGCTGTGTATTACTGTGCGAGAGCGGACACTATGGTTCGGGGAACTTA
TTTTGAGTACTGGGGCCAGGGAACCCCTGGTCACCGTCTCCTCA

.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
-|-|-|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
-|-|-|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.

GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCTGGTCCAGCCTGGGGGTCCCTGAGACTCTCCTGTGTAGCCTCTGGATT
CACCTTTAGTTTCTATTGGATGAGCTGGGTCCGCCAGGCTCCAGGAAAGGGGCTCGAGTGGGTGGCCAACATAAAGCAAG
ATGGAGGTGAGAAATACTATGTGGACTCTGTGAAGGGCCGATTACCATCTCCAGAGACAACGCCAAGAAGCACTGTAT
CTGCAAATGAACAGCCTGAGAGCCGAGGACACGGCTGTGTATTACTGTGCGAGA-CTGTCTGGCAGCAGCTGGG-AC-T
--TT-GACTACTGGGGCCAGGGAACCCCTGGTCACCGTCTCCTCA

CAGGTGCAGCTGGTGGAGTCTGGGGGAGGCGTGGTCCAGCCTGGGAGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATT
CACCTTCAGTAGCTATGCTATGCACTGGGTCCGCCAGGCTCCAGGCAAGGGGCTGGAGTGGGTGGCAGGTATATCATATG
ATGGAAGCAATAAATACTACGCAGACTCCGTGAAGGGCCGATTACCATCTCCAGAGACAATTCCAAGAACACGCTGTAT
CTGCAAATGAACAGCCTGAGAGCTGAGGACACGGCTGTGTATTACTGTGCGAGAGCGGACACTATGGTTCGGGGAACTTA
TTTTGAGTACTGGGGCCAGGGAACCCCTGGTCACCGTCTCCTCA

.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.
|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.|.

GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCTGGTCCAGCCTGGGGGTCCCTGAGACTCTCCTGTGTAGCCTCTGGATT

CACCTTTAGTTTCTATTGGATGAGCTGGGTCCGCCAGGCTCCAGGAAAGGGGCTCGAGTGGGTGGCCAAACATAAAGCAAG
ATGGAGGTGAGAAATACTATGTGGACTCTGTGAAGGGCCGATTACCATCTCCAGAGACAACGCCAAGAACTCACTGTAT
CTGCAAATGAACAGCCTGAGAGCCGAGGACACGGCTGTGTATTACTGTGCGAGA-CTGTCTGGCAGCAGCTGGG-ACT
---T-TGACTACTGGGGCCAGGGAACCCCTGGTCACCGTCTCCTCA

[illegible]

CAGGTGCAGCTGGTGGAGTCTGGGGGAGGCGTGGTCCAGCCTGGGAGGTCCCTGAGACTCTCCTGTGCAGCCTCTGGATT
CACCTTCAGTAGCTATGCTATGCACTGGGTCCGCCAGGCTCCAGGCAAGGGGCTGGAGTGGGTGGCAGGTATATCATATG
ATGGAAGCAATAAATACTACGCAGACTCCGTGAAGGGCCGATTACCATCTCCAGAGACAATTCCAAGAACACGCTGTAT
CTGCAAATGAACAGCCTGAGAGCTGAGGACACGGCTGTGTATTACTGTGCGAGAGCGGACACTATGGTTCGGGGAACTTA
TTTGTAGTACTGGGGCCAGGGAACCCTGGTCAACCGTCTCCTCA

GAGGTGCAGCTGGTGGAGTCTGGGGGAGGCTTGGTCCAGCCTGGGGGGTCCCTGAGACTCTCCTGTGTAGCCTCTGGATT
CACCTTTAGTTTTCTATTGGATGAGCTGGGTCCGCCAGGCTCCAGGAAAGGGGCTCGAGTGGGTGGCCAAACATAAAGCAAG
ATGGAGGTGAGAAATACTATGTGGACTCTGTGAAGGGCCGATTACCATCTCCAGAGACAACGCCAAGAACTCACTGTAT
CTGCAAATGAACAGCCTGAGAGCCGAGGACACGGCTGTGTATTACTGTGCGAGA-CTGTCTGGCAGCAGCTGGG-AC-T
--T-TGACTACTGGGGCCAGGGAACCCCTGGTACCGTCTCCTCA

[illegible]


```
CACCTTTAGTTTCTATTGGATGAGCTGGGTCCGCCAGGCTCCAGGAAAGGGGCTCGAGTGGGTGGCCAACATAAAGCAAG
ATGGAGGTGAGAAATACTATGTGGACTCTGTGAAGGGCCGATTACCATCTCCAGAGACAACGCCAAGAAGTCACTGTAT
CTGCAAATGAACAGCCTGAGAGCCGAGGACACGGCTGTGTATTACTGTGCGAGA-CTGTCTGGCAGCAGCTGGG-AC
----TTTGACTACTGGGGCCAGGGAACCCTGGTCACCGTCTCCTCA
```

```
[95]: # Obter dados de genes relacionados a via metabolica de vitamina C em humanos
human_pathways = REST.kegg_list("pathway", "hsa").read()

# Filtrar para humanos, Ascorbate
Ascorbate_pathways = []
for line in human_pathways.rstrip().split("\n"):
    entry, description = line.split("\t")
    if "Ascorbate" in description:
        Ascorbate_pathways.append(entry)

# Pegar genes da via de Ascorbate e adicionar em uma lista
Ascorbate_genes = []
for pathway in Ascorbate_pathways:
    pathway_file = REST.kegg_get(pathway).read() # for each pathway

    # Iterar sobre o pathway
    current_section = None
    for line in pathway_file.rstrip().split("\n"):
        section = line[:12].strip() # nomes na 12a coluna
        if not section == "":
            current_section = section

        if current_section == "GENE":
            gene_identifiers, gene_description = line[12:].split("; ")
            gene_id, gene_symbol = gene_identifiers.split()

            if not gene_symbol in Ascorbate_genes:
                Ascorbate_genes.append(gene_symbol)

print("Existe(m) %d vias de Ascorbate e %d genes relacionados.\\ OS genes são:" \
      ↪ % \
      (len(Ascorbate_pathways), len(Ascorbate_genes)))
print(", ".join(Ascorbate_genes))
```

Existe(m) 1 vias de Ascorbate e 30 genes relacionados.\\ OS genes são:
 UGDH, UGT2A1, UGT2A3, UGT2B17, UGT2B11, UGT2B28, UGT1A6, UGT1A4, UGT1A1, UGT1A3,
 UGT2B10, UGT1A9, UGT2B7, UGT1A10, UGT1A8, UGT1A5, UGT2B15, UGT1A7, UGT2B4,
 UGT2A2, GUSB, KL, MIOX, AKR1A1, RGN, ALDH2, ALDH3A2, ALDH1B1, ALDH7A1, ALDH9A1

```
[98]: # Obter detalhes da proteina: Immunoglobulin heavy variable 3-23 (P01764)
with ExPASy.get_sprot_raw("P01764") as handle:
```

```

seq_record = SeqIO.read(handle, "swiss")
print(seq_record.id)
print(seq_record.name)
print(seq_record.description)
print(repr(seq_record.seq))
print("Length %i" % len(seq_record))
print(seq_record.annotations["keywords"])

```

P01764

HV323_HUMAN

RecName: Full=Immunoglobulin heavy variable 3-23 {ECO:0000303|PubMed:11340299,

ECO:0000303|Ref.8}; AltName: Full=Ig heavy chain V-III region LAY

{ECO:0000305|PubMed:4139708}; AltName: Full=Ig heavy chain V-III region POM

{ECO:0000305|PubMed:4139708}; AltName: Full=Ig heavy chain V-III region TEI

{ECO:0000305|PubMed:4522793}; AltName: Full=Ig heavy chain V-III region TIL

{ECO:0000305|PubMed:409716}; AltName: Full=Ig heavy chain V-III region TUR

{ECO:0000305|PubMed:4522793}; AltName: Full=Ig heavy chain V-III region VH26

{ECO:0000305|PubMed:6450418}; AltName: Full=Ig heavy chain V-III region WAS

{ECO:0000305|PubMed:4522793}; AltName: Full=Ig heavy chain V-III region ZAP

{ECO:0000305|PubMed:4522793}; Flags: Precursor;

Seq('MEFGLSWLFLVAILKGVQCEVQLVESGGGLVQPGGSLRLSCAASGFTFSSYAMS...CAK')

Length 117

['3D-structure', 'Adaptive immunity', 'Cell membrane', 'Direct protein
sequencing', 'Disulfide bond', 'Immunity', 'Immunoglobulin', 'Immunoglobulin
domain', 'Membrane', 'Reference proteome', 'Secreted', 'Signal']