“AÑO DEL BICENTENARIO DEL PERÚ: 200 AÑOS DE INDEPENDENCIA”.

UNIVERSIDAD NACIONAL DE SAN AGUSTÍN DE AREQUIPA

CIENCIA DE LA COMPUTACIÓN

BIOINFORMÁTICA

Práctica 07

Estudiantes:
Miguel Alexander, Herrera Cooper
Milagros Celia, Cruz Mamani
Yara Jeanette, Quispe Quispe
Ingrid Sally, Espinel Quispe

Docente:
Mg. Vicente Enrique, Machaca Arceda

June 2, 2021

Contents

1 Actividad 1

Implemente el algoritmo de alineamiento local utilizando el algoritmo de Smith-waterman. Evalue sus resultados con las secuencias::

- S_1 : AGG
- S_2 : AAG

Utilice gapOpen = gapEXTEND = -5 y la siguiente matriz de sustitución:

	A	**C**	**G**	**T**
A	2	-7	-5	-7
C	-7	2	-7	-5
G	-5	-7	2	-7
T	-7	-5	-7	2

Table 1: Matriz de Similitud

La salida debe incluir el score matrix y todos los alineamientos posibles.

Resolución

1.1 Código

A continuación, mostramos el algoritmo implementado en Python.

```
import sys
import numpy as np

def Matriz_Sustitucion(archivo):
  next(archivo)
  r_c = {}
  s = []
  i = 0

  for line in archivo:
    if len(line.strip())!=0:
      line=line.rstrip('\n')

      r_c[line.split('\t')[0:1][0]] = i
      i = i+1

      s.append(list(map(int,line.split('\t')[1:])))

  archivo.close()

  return r_c, s


def Guardar_Resultado(F):
  f = open(resultadoArchivo, "w")
```

```
26
27    f.write('            ')
28    for i in range(columna-1):
29      f.write(sequencia1[i]+'        ')
30
31    f.write('\n')
32
33    seq2_t = ' '+ sequencia2
34
35    for r in range(fila):
36      f.write(seq2_t[r])
37      for c in range(columna):
38        f.write(' ( '+ '{}'.format(F[r][c][0])+' '+'{}'.format(F[r][c][1])+')
      ')
39
40      f.write('\n')
41    f.close()
42
43    f = open(resultadoArchivo, "r")
44    print(f.read())
45
46  def Obtener_Secuencias(F, i, j, alineamiento_sequencia1 = "",
      alineamiento_sequencia2 = ""):
47
48    if F[i][j][0]==0:
49      print ()
50      print (alineamiento_sequencia1)
51      print (alineamiento_sequencia2)
52      return
53
54
55    if i > 0 or j > 0:
56
57      if (i>0 and j>0 and F[i][j][1] == 'D'):
58        alineamiento_sequencia1 = sequencia1[j-1] + alineamiento_sequencia1
59        alineamiento_sequencia2 = sequencia2[i-1] + alineamiento_sequencia2
60        i = i-1
61        j = j-1
62
63      elif (i>0 and F[i][j][1]=='U'):
64        alineamiento_sequencia1 = "-" + alineamiento_sequencia2
65        alineamiento_sequencia2 = sequencia2[i-1] + alineamiento_sequencia2
66        i = i-1
67
68      else:
69        alineamiento_sequencia2 = "-" + alineamiento_sequencia2
70        alineamiento_sequencia1 = sequencia1[j-1] + alineamiento_sequencia1
71        j = j-1
72
73      Obtener_Secuencias(F, i, j, alineamiento_sequencia1,
      alineamiento_sequencia2)
74
75
76  def Alineamiento_Local(F, i, j, maximo_total):
77
78    diag = F[i-1][j-1][0] + s[r_c[sequencia2[i-1]]][r_c[sequencia1[j-1]]]
79    up = F[i-1][j][0] + d
```

```
  left = F[i][j-1][0] + d
  diag2 = 0

  entrada_max = [diag,up,left,diag2]
  maximo = max(entrada_max)

  if maximo > maximo_total:
                maximo_total = maximo

  F[i][j][0]  = maximo

  if F[i][j][0]==diag:
    F[i][j][1] = 'D'

  if F[i][j][0]==up:
    F[i][j][1] = 'U'

  if F[i][j][0]==left:
    F[i][j][1] = 'L'

  if i==fila-1 and j==columna-1:
    Guardar_Resultado(F)

    major = -1000
    s_i = 0
    s_j = 0
    for r in range(1,fila):
      for c in range(1, columna):
        if F[r][c][0]>=major:
          s_i = i
          s_j = j
          i = r
          j = c
          major = F[r][c][0]

    Obtener_Secuencias(F, i, j)
    Obtener_Secuencias(F, s_i, s_j)
    print("Maximo Total", maximo_total)
    return

  if j<columna-1:
    Alineamiento_Local(F, i ,j+1, maximo_total)
  else:
    Alineamiento_Local(F, i+1 ,1, maximo_total)


if __name__ == "__main__":

  archivoNombre = sys.argv[1]
  d = int(sys.argv[2])

  resultadoArchivo = "resultado.txt"

  archivo = open(archivoNombre, "r")


  r_c, s = Matriz_Sustitucion(archivo)
```

```
137
138  sequencia1 = "AAG"
139  sequencia2 = "AGC"
140
141
142  columna = len(sequencia1)+1
143  fila = len(sequencia2)+1
144
145  F = np.zeros([fila, columna], dtype='i,O')
146
147  maximo_total = 0
148
149  Alineamiento_Local(F, 1, 1, maximo_total)
150
151
```

1.2 Pruebas

Ahora veremos la prueba realizada según lo solicitado en el ejercicio

1.2.1 Input

Evaluamos los resultados con las secuencias

```
1  sequencia1 = "AAG"
2  sequencia2 = "AGC"
```

1.2.2 Output

Seguidamente pasamos a ver el resultado generado :

```
D:\INGRID\unsa\2021-1\bio\practica7\prueba>python ejercicio1.py substitution_matrix.txt -5
          A       A       G
  ( 0 0)  ( 0 0)  ( 0 0)  ( 0 0)
A ( 0 0)  ( 2 D)  ( 2 D)  ( 0 0)
G ( 0 0)  ( 0 0)  ( 0 0)  ( 4 D)
C ( 0 0)  ( 0 0)  ( 0 0)  ( 0 0)


AG
AG

A
A
Maximo Total 4
```

Figure 1: Matriz Final - Alineaciones - Puntajes

2 Actividad 2

Utilice el algoritmo implentado anteriormente, pero esta vez con secuencies reales. Se recomienda utilizar las secuencias de ADN.

Utilice *gapOpen* = *gapEXTEND* = -5, *identicalMatch* = 2 y *mismatch* = -2.

Puede utilizar estas bases de datos:

- NCBI virus
- Virus pathogen resource
- Castor

Resolución

2.1 Código

A continuación, mostramos el algoritmo implementado en Python.

```
import sys
import numpy as np

def Alineamineto_Local(secuencia_1, secuencia_2, penalidad):

  for i in range(1,fila):
    for j in range(1, columna):
      valor = identicalMatch

      if secuencia_2[i-1] != secuencia_1[j-1]:
        valor = mismatch

      Matriz[i][j] = max(Matriz[i-1][j-1] + valor, Matriz[i-1][j] +
   penalidad, Matriz[i][j-1] + penalidad, 0)


  print ()
  print (Matriz)

  mayor = -1000
  i = 0
  j = 0
  for r in range(1,fila):
    for c in range(1, columna):
      if Matriz[r][c]>mayor:
        i = r
        j = c
        mayor = Matriz[r][c]

  secuencia_alineada_1= ""
  secuencia_alineada_2= ""

  while (i > 0 or j > 0):
```

```
33
34      valor = identicalMatch
35      if secuencia_2[i-1] != secuencia_1[j-1]:
36        valor = mismatch
37
38      if (i>0 and j>0 and Matriz[i][j] == Matriz[i-1][j-1] + valor):
39        secuencia_alineada_1= secuencia_1[j-1] + secuencia_alineada_1
40        secuencia_alineada_2= secuencia_2[i-1] + secuencia_alineada_2
41        i = i-1
42        j = j-1
43
44      elif (i>0 and Matriz[i][j]==Matriz[i-1][j]+penalidad):
45        secuencia_alineada_1= "-" + secuencia_alineada_2
46        secuencia_alineada_2= secuencia_2[i-1] + secuencia_alineada_2
47        i = i-1
48
49      else:
50        secuencia_alineada_2= "-" + secuencia_alineada_2
51        secuencia_alineada_1= secuencia_1[j-1] + secuencia_alineada_1
52        j = j-1
53
54      if Matriz[i][j]==0:
55        break
56
57
58    print ()
59    print("Secuencia Alineada 1 :\n")
60    print (secuencia_alineada_1)
61    print ()
62    print("Secuencia Alineada 2:\n")
63    print (secuencia_alineada_2)
64    print("\n")
65    print()
66
67
68  def Obtener_Secuencia(file):
69    next(file)
70    secuencia = ""
71
72    for linea in file:
73      if len(linea.strip())!=0:
74        linea=linea.rstrip('\n')
75        secuencia = secuencia + linea
76
77    file.close()
78
79    return secuencia
80
81
82  if __name__ == "__main__":
83
84    archivo_1 = sys.argv[1]
85    archivo_2 = sys.argv[2]
86
87    penalidad = int(sys.argv[3])
88
89    identicalMatch = 2
```

```
mismatch = -2

f1 = open(archivo_1, "r")
f2 = open(archivo_2, "r")

secuencia_1 = Obtener_Secuencia(f1)
secuencia_2 = Obtener_Secuencia(f2)

columna = len(secuencia_1)+1
fila = len(secuencia_2)+1

Matriz = np.zeros([fila, columna], dtype=int)

Alineamineto_Local(secuencia_1, secuencia_2, penalidad)
```

2.2 Prueba 1

2.2.1 Input

De acuerdo a lo solicitado, realizaremos las pruebas con los archivos .fasta, trabajados en la Práctica 03.

1. P21333.fasta - Filamina-A (Humano)
2. Q8BTM8.fasta - Filamina-A (Ratón)

2.2.2 Output

Seguidamente pasamos a ver el resultado generado :

```
cooper@cooper-Legion-Y545:~/Escritorio/5to/9no_SEMEST
RE/BIOINFORMATICS/Practica_07$ python3 ejercicio2.py
P21333.fasta.txt Q8BTM8.fasta.txt -5

[[   0    0    0 ...    0    0    0]
 [   0    2    0 ...    0    0    0]
 [   0    0    4 ...    0    0    0]
 ...
 [   0    2    0 ... 5006 5001 4996]
 [   0    0    0 ... 5005 5008 5003]
 [   0    0    0 ... 5000 5003 5010]]
```

Figure 2: Matriz Final - gapOpen = gapEXTEND = -5, identicalMatch = 2 y mismatch = -2

Alineamineto para P21333.fasta - Filamina-A (Humano)

```
Secuencia Alineada 1 :

MSSSHSRAGQSAAGAAPGGGVDTRDAEMPATEKDLAEDAPWKKIQQNTFTRWCNEHLKCVSKRIANLQTDLSDGLRLIALLEVLSQKKMHRKHNQRPTFRQMQLEN
VSVALEFLDRESIKLVSIDSKAIVDGNLKLILGLIWTLILHYSISMPMWDEEEDEEAKKQTPKQRLLGWIQNKLPQLPITNFSRDWQSGRALGALVDSCAPGLCPD
WDSWDASKPVTNAREAMQQADDWLGIPQVITPEEIVDPNVDEHSVMTYLSQFPKAKLKPGAPLRPKLNPKKARAYGPGIEPTGNMVKKRAEFTVETRSAGQGEVLV
YVEDPAGHQEEAKVTANNDKNRTFSVWYVPEVTGTHKVTVLFAGQHIAKSPFEVYVDKSQGDASKVTAQGPGLEPSGNIANKTTYFEIFTAGAGTGEVEVVIQDPM
GQKGTVEPQLEARGDSTYRCSYQPTMEGVHTVHVTFAGVPIPRSPYTVTVGQACNPSACRAVGRGLQPKGVRVKETADFKVYTKGAGSGELKVTVKGPKGEERVKQ
KDLGDGVYGFEYYPMVPGTYIVTITWGGQNIGRSPFEVKVGTECGNQKVRAWGPGLEGGVVGKSADFVVEAIGDDVGTLGFSVEGPSQAKIECDDKGDGSCDVRYW
PQEAGEYAVHVLCNSEDIRLSPFMADIRDAPQDFHPDRVKARGPGLEKTGVAVNKPAEFTVDAKHGGKAPLRVQVQDNEGCPVEALVKDNGNGTYSCSYVPRKPVK
HTAMVSWGGVSIPNSPFRVNVGAGSHPNKVKVYGPGVAKTGLKAHEPTYFTVDCAEAGQGDVSIGIKCAPGVVGPAEADIDFDIIRNDNDTFTVKYTPRGAGSYTI
MVLFADQATPTSPIRVKVEPSHDASKVKAEGPGLSRTGVELGKPTHFTVNAKAAGKGKLDVQFSGLTKGDAVRDVDIIDHHDNTYTVKYTPVQQGPVGVNVTYGGD
PIPKSPFSVAVSPSLDLSKIKVSGLGEKVDVGKDQEFTVKSKGAGGQGKVASKIVGPSGAAVPCKVEPGLGADNSVVRFLPREEGPYEVEVTYDGVPVPGSPFPLE
AVAPTKPSKVKAFGPGLQGGSAGSPARFTIDTKGAGTGGLGLTVEGPCEAQLECLDNGDGTCSVSYVPTEPGDYNINILFADTHIPGSPFKAHVVPCFDASKVKCS
GPGLERATAGEVGQFQVDCSSAGSAELTIEICSEAGLPAEVYIQDHGDGTHTITYIPLCPGAYTVTIKYGGQPVPNFPSKLQVEPAVDTSGVQCYGPGIEGQGVFR
EATTEFSVDARALTQTGGPHVKARVANPSGNLTETYVQDRGDGMYKVEYTPYEEGLHSVDVTYDGSPVPSSPFQVPVTEGCDPSRVRVHGPGIQSGTTNKPNKFTV
ETRGAGTGGLGLAVEGPSEAKMSCMDNKDGSCSVEYIPYEAGTYSLNVTYGGHQVPGSPFKVPVHDVTDASKVKCSGPGLSPGMVRANLPQSFQVDTSKAGVAPLQ
VKVQGPKGLVEPVDVVDNADGTQTVNYVPSREGPYSISVLYGDEEVPRSPFKVKVLPTHDASKVKASGPGLNTTGVPASLPVEFTIDAKDAGEGLLAVQITDPEGK
PKKTHIQDNHDGTYTVAYVPDVTGRYTILIKYGGDEIPFSPYRVRAVPTGDASKCTVTVSIGGHGLGAGIGPTIQIGEETVITVDTKAAGKGKVTCTVCTPDGSEV
DVDVVENEDGTFDIFYTAPQPGKYVICVRFGGEHVPNSPFQVTALAGDQPSVQPPLRSQQLAPQYTYAQGGQQTWAPERPLVGVNGLDVTSLRPFDLVIPFTIKKG
EITGEVRMPSGKVAQPTITDNKDGTVTVRYAPSEAGLHEMDIRYDNMHIPGSPLQFYVDYVNCGHVTAYGPGLTHGVVNKPATFTVNTKDAGEGGLSLAIEGPSKA
EISCTDNQDGTCSVSYLPVLPGDYSILVKYNEQHVPGSPFTARVTGDDSMRMSHLKVGSAADIPINISETDLSLLTATVVPPSGREEPCLLKRLRNGHVGISFVPK
ETGEHLVHVKKNGQHVASSPIPVVISQSEIGDASRVRVSGQGLHEGHTFEPAEFIIDTRDAGYGGLSLSIEGPSKVDINTEDLEDGTCRVTYCPTEPGNYIINIKF
ADQHVPGSPFSVKVTGEGRVKESITRRRRAPSVANVGSHCDLSLKIPEISIQDMTAQVTSPSGKTHEAEIVEGENHTYCIRFVPAEMGTHTVSVKYKGQHVPGSPF
QFTVGPLGEGGAHKVRAGGPGLERAEAGVPAEFSIWTREAGAGGLAIAVEGPSKAEISFEDRKDGSCGVAYVVQEPGDYEVSVKFNEEHIPDSPFVVPVASPSGDA
RRLTVSSLQESGLKVNQPASFAVSLNGAKGAIDAKVHSPSGALEECYVTEIDQDKYAVRFIPRENGVYLIDVKFNGTHIPGSPFKIRVGEPGHGGDPGLVSAYGAG
LEGGVTGNPAEFVVNTSNAGAGALSVTIDGPSKVKMDCQECPEGYRVTYTPMAPGSYLISIKYGGPYHIGGSPFKAKVTGPRLVSNHSLHETSSVFVDSLTKATCA
PQHGAPGPGPADASKVVAKGLGLSKAYVGQKSSFTVDCSKAGNNMLLVGVHGPRTPCEEILVKHVGSRLYSVSYLLKDKGEYTLVVKWGDEHIPGSPYR
```

Figure 3: Alineamineto P21333 - gapOpen = gapEXTEND = -5, identicalMatch = 2 y mismatch = -2

Alineamineto para Q8BTM8.fasta - Filamina-A (Ratón)

Figure 4: Alineamineto Q8BTM8 - gapOpen = gapEXTEND = -5, identicalMatch = 2 y mismatch = -2

- Se debe tomar en cuenta que la matriz mostrada en la Figura2 está comprimida debido a la volumen de datos en los archivos fasta.
- En consecuencia podemos visulizar el resultado del alineamineto de ambas secuencias de Filamina A, tanto para el Humano como para el ratón en la Figura11 y la Figura12 respectivamente.

2.3 Prueba 2

- Ahora probaremos con un cambio de valores

```
C:\Users\51931\Downloads>python "ejercicio2 (1).py" P21333.fasta Q8BTM8.fasta -3

[[   0    0    0 ...    0    0    0]
 [   0    3    0 ...    0    0    0]
 [   0    0    6 ...    0    0    0]
 ...
 [   0    3    0 ... 7654 7651 7648]
 [   0    0    2 ... 7655 7657 7654]
 [   0    0    0 ... 7652 7654 7660]]
```

Figure 5: Matriz con gapOpen = gapEXTEND = -3

- Ahora las secuencias alineadas

```
Secuencia Alineada 1 :

MSSSHSRAGQSAAGAAPGGGVDTRDAEMPATEKDLAEDAPWKKIQQNTFTRWCNEHLKCVSKRIANLQTDLSDGLRLIALLEVLSQKKMHRKHNQRPTFRQMQLENVSVALEFLDRESIK
LVSIDSKAIVDGNLKLILGLIWTLILHYSISMPMWDEEEDEEAKKQTPKQRLLGWIQNKLPQLPITNFSRDWQSGRALGALVDSCAPGLCPDWDSWDASKPVTNAREAMQQADDWLGIPQ
VITPEEIVDPNVDEHSVMTYLSQFPKAKLKPGAPLRPKLNPKKARAYGPGIEPTGNMVKKRAEFTVETRSAGQGEVLVYVEDPAGHQEEAKVTANNDKNRTFSVWYVPEVTGTHKVTVLF
AGQHIAKSPFEVYVDKSQGDASKVTAQGPGLEPSGNIANKTTYFEIFTAGAGTGEVEVVIQDPMGQKGTVEPQLEARGDSTYRCSYQPTMEGVHTVHVTFAGVPIPRSPYTVTVGQACNP
SACRAVGRGLQPKGVRVKETADFKVYTKGAGSGELKVTVKGPKGEERVKQKDLGDGVYGFEYYPMVPGTYIVTITWGGQNIGRSPFEVKVGTECGNQKVRAWGPGLEGGVVGKSADFVVE
AIGDDVGTLGFSVEGPSQAKIECDDKGDGSCDVRYWPQEAGEYAVHVLCNSEDIRLSPFMADIRDAPQDFHPDRVKARGPGLEKTGVAVNKPAEFTVDAKHGGKAPLRVQVQDNEGCPVE
ALVKDNGNGTYSCSYVPRKPVKHTAMVSWGGVSIPNSPFRVNVGAGSHPNKVKVYGPGVAKTGLKAHEPTYFTVDCAEAGQGDVSIGIKCAPGVVGPAEADIDFDIIRNDNDTFTVKYTP
RGAGSYTIMVLFADQATPTSPIRVKVEPSHDASKVKAEGPGLSRTGVELGKPTHFTVNAKAAGKGKLDVQFSGLTKGDAVRDVDIIDHHDNTYTVKYTPVQQGPVGVNVTYGGDPIPKSP
FSVAVSPSLDLSKIKVSGLGEKVDVGKDQEFTVKSKGAGGQGKVASKIVGPSGAAVPCKVEPGLGADNSVVRFLPREEGPYEVEVTYDGVPVPGSPFPLEAVAPTKPSKVKAFGPGLQGG
SAGSPARFTIDTKGAGTGGLGLTVEGPCEAQLECLDNGDGTCSVSYVPTEPGDYNINILFADTHIPGSPFKAHVVPCFDASKVKCSGPGLERATAGEVGQFQVDCSSAGSAELTIEICSE
AGLPAEVYIQDHGDGTHTITYIPLCPGAYTVTIKYGGQPVPNFPSKLQVEPAVDTSGVQCYGPGIEGQGVFREATTEFSVDARALTQTGGPHVKARVANPSGNLTETYVQDRGDGMYKVE
YTPYEEGLHSVDVTYDGSPVPSSPFQVPVTEGCDPSRVRVHGPGIQSGTTNKPNKFTVETRGAGTGGLGLAVEGPSEAKMSCMDNKDGSCSVEYIPYEAGTYSLNVTYGGHQVPGSPFKV
PVHDVTDASKVKCSGPGLSPGMVRANLPQSFQVDTSKAGVAPLQVKVQGPKGLVEPVDVVDNADGTQTVNYVPSREGPYSISVLYGDEEVPRSPFKVKVLPTHDASKVKASGPGLNTTGV
PASLPVEFTIDAKDAGEGLLAVQITDPEGKPKKTHIQDNHDGTYTVAYVPDVTGRYTILIKYGGDEIPFSPYRVRAVPTGDASKCTVTVSIGGHGLGAGIGPTIQIGEETVITVDTKAAG
KGKVTCTVCTPDGSEVDVDVVENEDGTFDIFYTAPQPGKYVICVRFGGEHVPNSPFQVTALAGDQPSVQPPLRSQQLAPQYTYAQGGQQTWAPERPLVGVNGLDVTSLRPFDLVIPFTIK
KGEITGEVRMPSGKVAQPTITDNKDGTVTVRYAPSEAGLHEMDIRYDNMHIPGSPLQFYVDYVNCGHVTAYGPGLTHGVVNKPATFTVNTKDAGEGGLSLAIEGPSKAEISCTDNQDGTC
SVSYLPVLPGDYSILVKYNEQHVPGSPFTARVTGDDSMRMSHLKVGSAADIPINISETDLSLLTATVVPPSGREEPCLLKRLRNGHVGISFVPKETGEHLVHVKKNGQHVASSPIPVVIS
QSEIGDASRVRVSGQGLHEGHTFEPAEFIIDTRDAGYGGLSLSIEGPSKVDINTEDLEDGTCRVTYCPTEPGNYIINIKFADQHVPGSPFSVKVTGEGRVKESITRRRRAPSVANVGSHC
DLSLKIPEISIQDMTAQVTSPSGKTHEAEIVEGENHTYCIRFVPAEMGTHTVSVKYKGQHVPGSPFQFTVGPLGEGGAHKVRAGGPGLERAEAGVPAEFSIWTREAGAGGLAIAVEGPSK
AEISFEDRKDGSCGVAYVVQEPGDYEVSVKFNEEHIPDSPFVVPVASPSGDARRLTVSSLQESGLKVNQPASFAVSLNGAKGAIDAKVHSPSGALEECYVTEIDQDKYAVRFIPRENGVY
LIDVKFNGTHIPGSPFKIRVGEPGHGGDPGLVSAYGAGLEGGVTGNPAEFVVNTSNAGAGALSVTIDGPSKVKMDCQECPEGYRVTYTPMAPGSYLISIKYGGPYHIGGSPFKAKVTGPR
LVSNHSLHETSSVFVDSLTK-AT-VPQHATSGPGPADVSKVVAKGLGLSKAYVGQKSNFTVDCSKAGNNMLLVGVHGPRTPCEEILVKHMGSRLYSVSYLLKDKGEYTLVVKWGDEHIPG
SPYRIMVP
```

Figure 6: Secuencia alineada 1

```
Secuencia Alineada 2:

MSSSHSRCGQSAAVASPGGSIDSRDAEMPATEKDLAEDAPWKKIQQNTFTRWCNEHLKCVSKRIANLQTDLSDGLRLIALLEVLSQKKMHRKHNQRPTFRQMQLENVSVALEFLDRESIK
LVSIDSKAIVDGNLKLILGLIWTLILHYSISMPMWDEEEDEEAKKQTPKQRLLGWIQNKLPQLPITNFSRDWQSGRALGALVDSCAPGLCPDWDSWDASKPVNNAREAMQQADDWLGIPQ
VITPEEIVDPNVDEHSVMTYLSQFPKAKLKPGAPLRPKLNPKKARAYGPGIEPTGNMVKKRAEFTVETRSAGQGEVLVYVEDPAGHQEEAKVTANNDKNRTFSVWYVPEVTGTHKVTVLF
AGQHIAKSPFEVYVDKSQGDASKVTAQGPGLEPSGNIANKTTYFEIFTAGAGMGEVEVVIQDPTGQKGTVEPQLEARGDSTYRCSYQPTMEGVHTVHVTFAGVPIPRSPYTVTVGQACNP
AACRAIGRGLQPKGVRVKETADFKVYTKGAGSGELKVTVKGPKGEERVKQKDLGDGVYGFEYYPTIPGTYTVTITWGGQNIGRSPFEVKVGTECGNQKVRAWGPGLEGGIVGKSADFVVE
AIGDDVGTLGFSVEGPSQAKIECDDKGDGSCDVRYWPQEAGEYAVHVLCNSEDIRLSPFMADIREAPQDFHPDRVKARGPGLEKTGVAVNKPAEFTVDAKHAGKAPLRVQVQDNEGCSVE
ATVKDNGNGTYSCSYVPRKPVKHTAMVSWGGVSIPNSPFRVNVGAGSHPNKVKVYGPGVAKTGLKAHEPTYFTVDCTEAGQGDVSIGIKCAPGVVGPTEADIDFDIIRNDNDTFTVKYTP
CGAGSYTIMVLFADQATPTSPIRVKVEPSHDASKVKAEGPGLNRTGVELGKPTHFTVNAKTAGKGKLDVQFSGLAKGDAVRDVDIIDHHDNTYTVKYIPVQQGPVGVNVTYGGDHIPKSP
FSVGVSPSLDLSKIKVSGLGDKVDVGKDQEFTVKSKGAGGQGKVASKIVSPSGAAVPCKVEPGLGADNSVVRFVPREEGPYEVEVTYDGVPVPGSPFPLEAVAPTKPSKVKAFGPGLQGG
NAGSPARFTIDTKGAGTGGLGLTVEGPCEAQLECLDNGDGTCSVSYVPTEPGDYNINILFADTHIPGSPFKAHVAPCFDASKVKCSGPGLERATAGEVGQFQVDCSSAGSAELTIEICSE
AGLPAEVYIQDHGDGTHTITYIPLCPGAYTVTIKYGGQPVPNFPSKLQVEPAVDTSGVQCYGPGIEGQGVFREATTEFSVDARALTQTGGPHVKARVANPSGNLTDTYVQDCGDGTYKVE
YTPYEEGVHSVDVTYDGSPVPSSPFQVPVTEGCDPSRVRVHGPGIQSGTTNKPNKFTVETRGAGTGGLGLAVEGPSEAKMSCMDNKDGSCSVEYIPYEAGTYSLNVTYGGHQVPGSPFKV
PVHDVTDASKVKCSGPGLSPGMVRANLPQSFQVDTSKAGVAPLQVKVQGPKGLVEPVDVVDNADGTQTVNYVPSREGSYSISVLYGEEEVPRSPFKVKVLPTHDASKVKASGPGLNTTGV
PASLPVEFTIDAKDAGEGLLAVQITDPEGKPKKTHIQDNHDGTYTVAYVPDVPGRYTILIKYGGDEIPFSPYRVRAVPTGDASKCTVTVSIGGHGLGAGIGPTIQIGEETVITVDTKAAG
KGKVTCTVCTPDGSEVDVDVVENEDGTFDIFYTAPQPGKYVICVRFGGEHVPNSPFQVTALAGDQPTVQTPLRSQQLAPQVNYPQGSQQTWIPERPMVGVNGLDVTSLRPFDLVIPFTIK
KGEITGEVRMPSGKVAQPSITDNKDGTVTVRYSPSEAGLHEMDIRYDNMHIPGSPLQFYVDYVNCGHITAYGPGLTHGVVNKPATFTVNTKDAGEGGLSLAIEGPSKAEISCTDNQDGTC
SVSYLPVLPGDYSILVKYNDQHIPGSPFTARVTGDDSMRMSHLKVGSAADIPINISETDLSLLTATVVPPSGREEPCLLKRLRNGHVGISFVPKETGEHLVHVKKNGQHVASSPIPVVIS
QSEIGDASRVRVSGQGLHEGHTFEPAEFIIDTRDAGYGGLSLSIEGPSKVDINTEDLEDGTCRVTYCPTEPGNYIINIKFADQHVPGSPFSVKVTGEGRVKESITRRRRAPSVANIGSHC
DLSLKIPEISIQDMTAQVTSPSGKTHEAEIVEGENHTYCIRFVPAEMGMHTVSVKYKGQHVPGSPFQFTVGPLGEGGAHKVRAGGPGLERAEVGVPAEFGIWTREAGAGGLAIAVEGPSK
AEISFEDRKDGSCGVAYVVQEPGDYEVSVKFNEEHIPDSPFVVPVASPSGDARRLTVSSLQESGLKVNQPASFAVSLNGAKGAIDAKVHSPSGALEECYVTEIDQDKYAVRFIPRENGIY
LIDVKFNGTHIPGSPFKIRVGEPGHGGDPGLVSAYGAGLEGGVTGSPAEFIVNTSNAGAGALSVTIDGPSKVKMDCQECPEGYRVTYTPMAPGSYLISIKYGGPYHIGGSPFKAKVTGPR
LVSNHSLHETSSVFVDSLTKVAT-VPQHATSGPGPADVSKVVAKGLGLSKAYVGQKSNFTVDCSKAGNNMLLVGVHGPRTPCEEILVKHMGSRLYSVSYLLKDKGEYTLVVKWGDEHIPG
SPYRIMVP
```

Figure 7: Secuencia alineada 2

- gapOpen = gapEXTEND = -3, identicalMatch = 3 y mismatch = -1
- Este fue el cambio de valores que se realizo

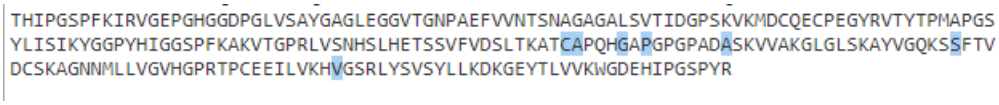

Figure 8: Secuencia alineada 1 (Original)

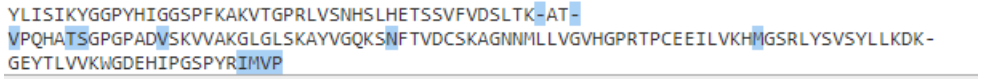

Figure 9: Secuencia alineada 1 (Valores cambiados)

- **Comentarios:**
  - Hay pequeñas diferencias, y por lo que hemos podido observar estas diferencias varian tambien dependiendo de los valores otorgados o el tamaño de la secuencia alineada

3 Actividad 3

Evalue la pregunta anterior con otros valores de *gapOpen*, *apEXTEND*, *identicalMatch* y *mismatch*. Verifique si obtiene el mismo alineamiento y comente sus resultados.

Resolución

3.1 Codigo

En el codigo cambiaremos los valores gapOPEN, gapEXTEND, identicalMatch, mismatch.

3.2 Resultados

3.2.1 Input

De acuerdo a lo solicitado, realizaremos las pruebas con los archivos .fasta, trabajados en la Práctica 06.

1. P21333.fasta - Filamina-A (Humano)
2. Q8BTM8.fasta - Filamina-A (Ratón)

3.2.2 Output

Seguidamente pasamos a ver el resultado generado :

```
D:\INGRID\unsa\2021-1\bio\practica7\prueba>python ejercicio2.py Q8BTM8.fasta.txt P21333.fasta.txt -3

[[   0    0    0 ...    0    0    0]
 [   0    1    0 ...    1    0    0]
 [   0    0    2 ...    0    0    0]
 ...
 [   0    0    0 ... 2503 2502 2499]
 [   0    0    0 ... 2500 2504 2501]
 [   0    0    0 ... 2497 2501 2505]]
```

Figure 10: Matriz Final - gapOpen = gapEXTEND = -3, identicalMatch = 1 y mismatch = -1

Alineamineto para P21333.fasta - Filamina-A (Humano)

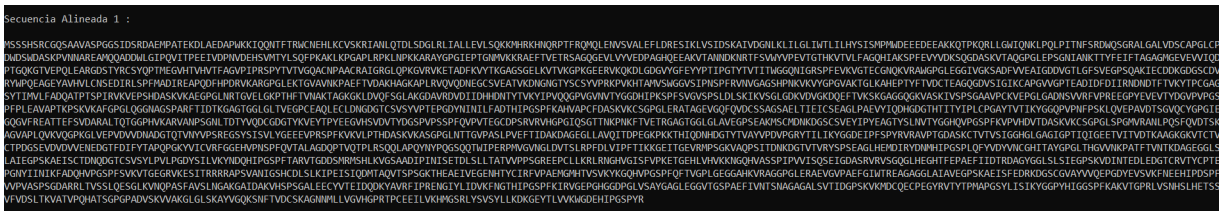

Figure 11: Alineamineto P21333 - gapOpen = gapEXTEND = -3, identicalMatch = 1 y mismatch = -1

Alineamineto para Q8BTM8.fasta - Filamina-A (Ratón)

Figure 12: Alineamineto Q8BTM8 - gapOpen = gapEXTEND = -3, identicalMatch = 1 y mismatch = -1

- Por ultimo probaremos con con gap = -4 e identicalMatch = 2 y mismatch = -2

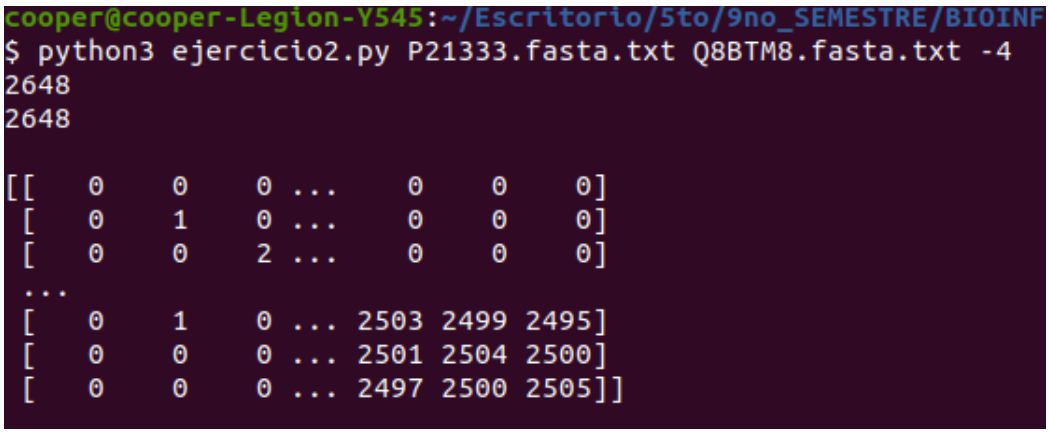

Figure 13: Matriz con gapOpen = gapEXTEND = -4

Figure 14: P12333 con gapOpen = gapEXTEND = -4, identicalMatch = 2 y mismatch = -2

Figure 15: Q8TBM8 alinedo con gapOpen = gapEXTEND = -4, identicalMatch = 2 y mismatch = -2

4 Conclusiones

1. En el ejerccio 1 como vemos probamos el algoritmo con esa dos cadenas y hallamos alineación local con su penalidad el cual se observa su mejores alineamientos y su mejor scored el cual observamos que nuestro mejor scored siempre sera mayor que cero.

2. El método experimental de alineamiento de DNA es lento,pero permite recuperar material genético a través de su alineamiento local. Debido a su lentitud se recomienda usar frameworks como Hadoop para el procesamineto distribuido de grandes cantidades de datos. Una alternativa es crear clusters en AWS o DataProc de Google, y usar el algoritmo paralelizado con Hadoop.

3. Se probó el algoritmo con secuencias reales de proteínas y nos dió como resultado nuestra matriz y las dos secuencias de alineamiento local, lo cual serviria para futuros trabajos de procesamiento genómico. Pero aun asi se tiene que realizar un preprocesamiento al output del algoritmo para poder insertar en la RNN.

BASE DE COMPARACIÓN	ALINEACIÓN DE SECUENCIA GLOBAL	ALINEACIÓN DE SECUENCIA LOCAL
Técnica general	Algoritmo Needleman-Wunsch.	Algoritmo Smith-Waterman.
Descripción	Se intenta alinear toda la secuencia (alineación de extremo a extremo).	Encuentra regiones locales con el mayor nivel de similitud entre las dos secuencias.
Función	Contiene todas las letras de las secuencias de consulta y de destino.	Alinea una subcadena de la secuencia de consulta con una subcadena de la secuencia objetivo.
Dos secuencias	Si dos secuencias tienen aproximadamente la misma longitud y son bastante similares, son adecuadas para la alineación global.	Cualquiera de las dos secuencias se pueden alinear localmente ya que la alineación local encuentra tramos de secuencias con un alto nivel de coincidencias sin considerar la alineación del resto de las regiones de secuencia.
Idoneidad	Adecuado para alinear dos secuencias estrechamente relacionadas.	Adecuado para alinear secuencias más divergentes o secuencias relacionadas lejanamente.
Usar	Los alineamientos globales generalmente se realizan para comparar genes homólogos, como comparar dos genes con la misma función (en humanos frente a ratones) o comparar dos proteínas con función similar.	Se utiliza para descubrir patrones conservados en secuencias de ADN o dominios o motivos conservados en dos proteínas.

5 Repositorio

La resolución de la práctica la encuentra en el siguiente enlace **Practica_07**