



Escuela de Ciencia de la Computación
Universidad Nacional de San Agustín Arequipa

Carolina Chávez, William Condori, Renzo Cayllahue, Erika Tamo
COMPUTACIÓN MOLECULAR BIOLÓGICA [GRUPO - 02]

Práctica 5: Alineamiento local de secuencias con programación dinámica
<https://github.com/cchavezlo/UNSA/tree/master/ComputacionMolecularBiologica>

26 de junio de 2020

1. EJERCICIOS

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

- S1: AGC
- S2: AAG

Utilice $\text{gapOpen} = \text{gapEXTEND} = -5$ 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

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

1.1.1. Código:

En primer lugar definimos la matriz de sustitución y las penalizaciones:

```
1  [-7, 2, -7, -5],  
2  [-5, -7, 2, -7],  
3  [-7, -5, -7, 2]]  
4  
5  gapOpen=-5  
6  gapExt=-5
```

La función `nucIValNum` es la que genera los índices de la matriz de acuerdo a cada nucleótido:

```
A=0  
C=1  
G=2  
OTRO=3
```

```

1 def nuclValNum( nucl1 ):
2
3     if nucl1=='A':
4         return 0
5     if nucl1=='C':
6         return 1
7     if nucl1=='G':
8         return 2
9     else:
10        return 3

```

La función makematrix generara la matriz de escore inicializando la primera fila y primera columna con 0, así es gap sea negativo. También llena las demás posiciones con el valor máximo a cada correspondencia, en el caso de que haya un valor negativo se pone cero.

```

1 def makeMatrix( seq1 , seq2 ):
2     M = []
3     for i in range( len( seq1 )+1 ):
4         M.append( [] )
5
6         for j in range( len( seq2 )+1 ):
7
8             if i==0:
9                 M[ i ].append( 0 )
10            elif j==0:
11                M[ i ].append( 0 )
12            else:
13                M[ i ].append( valorMaximo( seq1 [ i -1 ], seq2 [ j -1 ], M[ i -1 ][ j -1 ], M[ i -1 ][ j ],
14                M[ i ][ j -1 ] ) )
15                ##print( seq1 [ i -1 ], seq2 [ j -1 ])
16            #print( M )
17        return M

```

La función anterior llama a la función valorMaximo que nos retornará el mayor score para las intersecciones de la matriz score.

```

1 def valorMaximo( nucl1 , nucl2 , valorDia , valorArr , valorAba ):
2
3     nucleotNum1=nuclValNum( nucl1 )
4     nucleotNum2=nuclValNum( nucl2 )
5
6     valores=[]
7
8     valores.append( valorDia+matSust [ nucleotNum1 ][ nucleotNum2 ] )
9     valores.append( valorArr+gapExt )
10    valores.append( valorAba+gapExt )
11    maximo=max( valores )
12    if ( maximo<0 ):
13        return 0
14    return ( max( valores ) )

```

Las funciones GenerarAlineamiento y localAlign son complementarias ya que una llama a la otra para alinear los posibles nucleótidos en cada cadena.

```

1 alineamientoscad1=[]
2 alineamientoscad2=[]
3 def generarAlineamiento( longitud1 , longitud2 , seq1 , seq2 , numAlineamiento , M ):
4     alineamientoActual=numAlineamiento
5
6     nucl1=seq1 [ longitud1 -1 ]
7     nucl2=seq2 [ longitud2 -1 ]
8     if longitud1>=0 and longitud2>=0:
9         ##print( longitud1 , longitud2 , M[ longitud1 ][ longitud2 ], M[ longitud1 -1 ][
10        longitud2 -1 ], nuclValNum( nucl1 ), nuclValNum( nucl2 ) )
11        if M[ longitud1 ][ longitud2 ]==M[ longitud1 -1 ][ longitud2 -1 ]+matSust [ nuclValNum(
12        nucl1 ) ][ nuclValNum( nucl2 ) ]:
13            ##print( alineamientoActual , "0" )
14            alineamientoscad1 [ alineamientoActual ]=nucl1+alineamientoscad1 [
15            alineamientoActual ]
16            alineamientoscad2 [ alineamientoActual ]=nucl2+alineamientoscad2 [
17            alineamientoActual ]
18            generarAlineamiento( longitud1 -1 , longitud2 -1 , seq1 , seq2 ,
19            alineamientoActual , M )

```

```

1 def localAlignent(seq1,seq2):
2
3     alineamientoActual=0
4     M=makeMatrix(seq1,seq2)
5
6     for i in range(len(M)):
7         for j in range(len(M[i])):
8             if M[i][j]!=0:
9                 alineamientoscad1.append("")
10                generarAlineamineto(i,j,seq1,seq2,alineamientoActual,M)
11                alineamientoActual+=1

```

LocalAlignent nos inicializara el programa y nos mostrara los resultados como el score maximo y las cadenas maximas

```

1 def localAlignent(seq1,seq2):
2
3     alineamientoActual=0
4     M=makeMatrix(seq1,seq2)
5
6     for i in range(len(M)):
7         for j in range(len(M[i])):
8             if M[i][j]==scoremax:
9                 alineamientoscad1.append("")
10                alineamientoscad2.append("")
11                generarAlineamineto(i,j,seq1,seq2,alineamientoActual,M)
12                alineamientoActual+=1
13
14    print("score maximo: ",scoremax)
15    print(np.unique(alineamientoscad1))
16    print(np.unique(alineamientoscad2))

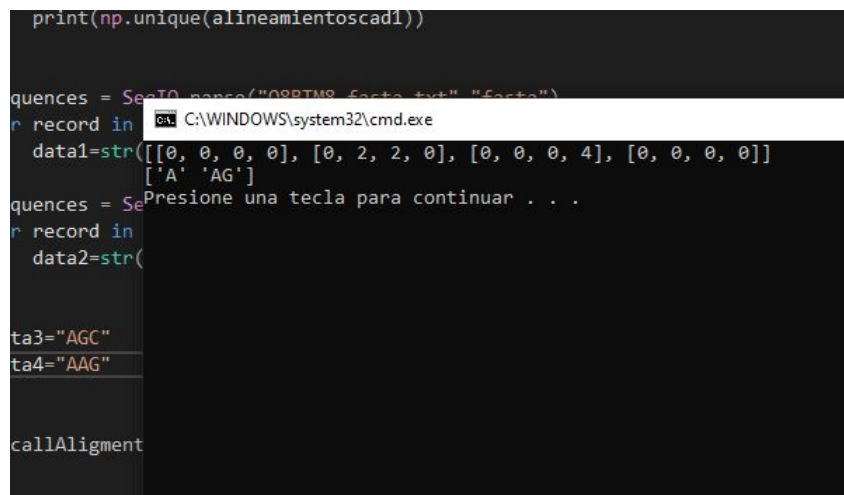
```

RESULTADOS

Tomando las cadenas:

- AGC
- AAG

Se imprime la matriz y los alineamientos entre ambas.



```

print(np.unique(alineamientoscad1))

sequences = SeqIO.parse("088TM8.fasta.txt", "fasta")
r record in
    data1=str([[0, 0, 0, 0], [0, 2, 2, 0], [0, 0, 0, 4], [0, 0, 0, 0]]
    ['A' 'AG'])
sequences = SeqIO.parse("088TM9.fasta.txt", "fasta")
r record in
    data2=str([[0, 0, 0, 0], [0, 2, 2, 0], [0, 0, 0, 4], [0, 0, 0, 0]]
    ['A' 'AG'])

ta3="AGC"
ta4="AAG"

callAlignent

```

Figura 1: Salida

1.2. Utilice el algoritmo implementado anteriormente, pero esta vez con secuencias reales. Se recomienda utilizar las secuencias de ADN. Utilice $\text{gapOpen} = \text{gapEXTEND} = -5$, $\text{identicalMatch} = 2$ y $\text{mismatch} = -2$.

1.3. Resultados

- **Proteína Filamin-A en Ratones y Humanos** Los siguientes resultados son con un score maximo de 5182.

```
AL_d_5_m_-2.txt
[ 'MSSSHSRGCGSAAVASPGGSDSRDAEMPATEKD LAEDAPWKKIQONTFTWRNEHLKCVSKRIANLQTDLSGDLRLIALLEVL SQKMHRKHNRPTFRQMQL ENVSVA
LEFLDRESIKLVSIDSKAIVDGNLKLILGLIWLILHYSISMPMWDEEEDAEAKQTPKQRL LGWIQNLQPLPITNFSRDWQSGRALGALVDSAPGLCPDWSDWSDAKPV
NNAREAMQADDWLGIPOVITPEEIVDPNVDEHSMVTYLSQFPKAKLKPAGAPLRPKLNPKKARAYGPGIEPTGNMVKKRAEFTVETRSAGQGEVLVYVEDPAGHQEEAKVTA
NNDKNRTFSVWVYVEVTGTHKVTVL FAGQHIKSPFEVYVDKSGQDASKVTAQGPGL EPSGNIA NKTTYFEIFTAGAGMGEVEVVIQDPTGQKGTVEPQL EARGDSTYRC SY
QPTMEGVHTVHVT FAGVPIPRSPYTVTVGQACNPACRAIGRLQPKGVRVKETADFKVYTKGAGSGELKVTVKGPKEERVKQKDLGDGVYGF EYYPTIPGTYTITWGG
QNIGRSPFEVKVGTCEGNQKVRWGPGL EGGIVGKSADFVEAIGDDVGT LGSVEGSPQAKIECDDKGDGSCDVRYPQEA GEAYAVHVL CNSEDIRLSPFMADIREAPQDF
HPRVRKARGPGL EKTGVAVNKPAEFTVDAKHGKAPLRVQVQDNEGCSVEATVKDNGNGTYSCSVPRKPKVHTAMVSWGGVSI PNSSPFRVNVGAGSHPNKVKVYGPVAKT
GLKAHEPTYFTVDCTEAGQGDVSI GIKCAPGVWGPEADIDFIDIRNDNDTFTVKYTPCGAGSYTIMVLFADQATPTSPIRVKVEPSHDASKVKAEGPGLNRTGVELGKPTH
FTVNAKTAGKGLDVQFSG LKAGDAVRDVIDIDHDNTYTVKYTPVQQGPVGVNVTYGGDHPKSPFSVGVSPSLDLSKIKVSGLGKVDVVGKQDQFETVSKGAGGQGVAS
KIVSPSGAAVPCKEVEPLGADNSVVRFLPREEGPYEVEVTVYDGVVPVGPSPFLEAVATPKSPKVKAFGPG LQGGNAGSPARFTIDTKAGTGGLGLTVEGPEAQLECLDNG
DGTCSVSYVPTPEGDYNINILFADTHIPGSPFKAHVAPCFDASKVKCSGPGLERATAGEVQGFQVDCSSAGSAELTIEICSEAGLPAEVYIQDHGDGTHITITVILPCP GAYT
VTIKYGGQVPVNFPSK LQVEPAVDTSGVQCYGPGIEGQGVFEATTEFSVDARAL TQTGGPHVKARVANPSGNLTDTYVQDQGDGT YKVEYTPYEEGVHSVDVTYDGSVPVS
SPFQVPVTEGCDPSRVRVHGPQISGTTNKNPKFTVETRAGTGGLGLAVEGPSEAKMSCMDNKGSCSVEYIPYEAGTYS LNVTYGGHQVPGSPFKVPVHDVTDASKVKCS
GPG LSPGMVRANLPQSFOVDTSKAGVAPLQVKVQGPGLVEPVDVVDNADGTQTVNYPVSPREGYSISVL YGEEVPRSPFKVKVLP THDASKVKASGPG LNTTGVPASLPV
EFTIDAKDAGEGLLAVQITDPEGKPKKTHIQDNHDTYTVAYVPDVTGRTYILIKYGGDEIPFSPYRVRVPTGDASKCTVTVSI GGHGLGAGIGPTIQIGETVITVDTKA
AGKGKVTCTVCTPDGSEVDVVDVNEEDGTDFIFYTAPQPGKYVICVRFGGEHVPNPSPFQV TALAGDQPTVQTP LRSQQLAPQYVNPQGSQQTWIPERPMVGVNGLDVTSLRP
FDLVIPTITIKGEITGEVRMPSPGKVAQPSITDNKDGTVTVRYSPSEAGLHEMDIRYDNMHI PGSP LQFYVDYVNCGHITAYGPG LTHGVVNKPATFTVNTKDAGEGGLSLAI
EGPSKAEISCTDNQDGTCSVSYLPVLPDYSILVKYNDQHIPGSPFTARVTGDDSMRMSHLKVGSAADIPINIS ETDLSLLTATVVPSPGREEPCLLKR LRNGHVGISFVPK
ETGEHLVHVKNQGHVASSP IPVVISQSEIGDASRVRSVGGGLHEGHTFEPAEFTIDTRDAGYGG LSLIEGSPKVDINTEDLEDGT CRVTCPTPEGNYIINI KFAHQHVP
GSPFSVKVTGEGRVKESITRRRRAPSVANIGSHCDLSLKIPEISIQDMTAQVTSPSGKTHEAEIVGENHTYCI RFVPAEMGMHTVSVKYKQGHVPGSPFQFTVGPLGEGGA
HKVRAGGPGGLERAEGVPAEFGIWREAGAGGLA IAVEGPSKAEISFEDRKDSCGVAYVYVQEPGDYEVSVKFNEEHIPDSPFVVPVASPSGDARRLTVSS LQESGLKVNQ P
ASFVSLNGAKGAI DAKVHSPSGALEECYVTEIDQKYAVRFIPRENGIYLIDVFNHGTIHPGSPFKIRVGE PGHGGDPGLVSAYGAGLEGGVGTGSPA EFTVNTSNAGAGAL
SVTIDGPSVKVMDCQCEPGEYRVYTPMAPGSYLSIKYGGPYHIGGSPFAKVTGPRLVSNHSLHETSSVFVDSLTKVATVPQHA
TSGPGPADVSKVAKGLGLSKAYVGQKSNFTVDCSKAGNNM L LGVHGPRTPC EILVKHMGSR LYSVSYLLKDKGEYTLVWKWGEH IPGSPYRIMVP ' ]
```

Figura 2: Primer resultado

```
AL_d_5_m_-2.txt
ASFVSLNGAKGAI DAKVHSPSGALEECYVTEIDQKYAVRFIPRENGIYLIDVFNHGTIHPGSPFKIRVGE PGHGGDPGLVSAYGAGLEGGVGTGSPA EFTVNTSNAGAGAL
SVTIDGPSVKVMDCQCEPGEYRVYTPMAPGSYLSIKYGGPYHIGGSPFAKVTGPRLVSNHSLHETSSVFVDSLTKVATVPQHA
TSGPGPADVSKVAKGLGLSKAYVGQKSNFTVDCSKAGNNM L LGVHGPRTPC EILVKHMGSR LYSVSYLLKDKGEYTLVWKWGEH IPGSPYRIMVP ' ]
[ 'MSSSHSRGCGSAAVASPGGSDSRDAEMPATEKD LAEDAPWKKIQONTFTWRNEHLKCVSKRIANLQTDLSGDLRLIALLEVL SQKMHRKHNRPTFRQMQL ENVSVA
LEFLDRESIKLVSIDSKAIVDGNLKLILGLIWLILHYSISMPMWDEEEDAEAKQTPKQRL LGWIQNLQPLPITNFSRDWQSGRALGALVDSAPGLCPDWSDWSDAKPV
NNAREAMQADDWLGIPOVITPEEIVDPNVDEHSMVTYLSQFPKAKLKPAGAPLRPKLNPKKARAYGPGIEPTGNMVKKRAEFTVETRSAGQGEVLVYVEDPAGHQEEAKVTA
NNDKNRTFSVWVYVEVTGTHKVTVL FAGQHIKSPFEVYVDKSGQDASKVTAQGPGL EPSGNIA NKTTYFEIFTAGAGTGEVEVVIQDPTGQKGTVEPQL EARGDSTYRC SY
QPTMEGVHTVHVT FAGVPIPRSPYTVTVGQACNPACRAVGRGLQPKGVRVKETADFKVYTKGAGSGELKVTVKGPKEERVKQKDLGDGVYGF EYYPMVPGTYITVITWGG
QNIGRSPFEVKVGTCEGNQKVRWGPGL EGGIVGKSADFVEAIGDDVGT LGSVEGSPQAKIECDDKGDGSCDVRYPQEA GEAYAVHVL CNSEDIRLSPFMADIRDAPQDF
HPRVRKARGPGL EKTGVAVNKPAEFTVDAKHGKAPLRVQVQDNEGCPVEALVKDNGNGTYSCSVPRKPKVHTAMVSWGGVSI PNSSPFRVNVGAGSHPNKVKVYGPVAKT
GLKAHEPTYFTVDCAEAGQGDVSI GIKCAPGVWGPAEADIDFIDIRNDNDTFTVKYTPRGAGSYTIMVLFADQATPTSPIRVKVEPSHDASKVKAEGPGLSRTGVELGKPTH
FTVNAKAAGKGLDVQFSG LTKGDAVRDVIDIDHDNTYTVKYTPVQQGPVGVNVTYGGDPTPKSPFSVAVSPSLDLSKIKVSGLGKVDVVGKQDQFETVSKGAGGQGVAS
KIVGSPSGAAVPCKEVEPLGADNSVVRFLPREEGPYEVEVTVYDGVVPVGPSPFLEAVATPKSPKVKAFGPG LQGGSGAGSPARFTIDTKAGTGGLGLTVEGPEAQLECLDNG
DGTCSVSYVPTPEGDYNINILFADTHIPGSPFKAHVAPCFDASKVKCSGPGLERATAGEVQGFQVDCSSAGSAELTIEICSEAGLPAEVYIQDHGDGTHITITVILPCP GAYT
VTIKYGGQVPVNFPSK LQVEPAVDTSGVQCYGPGIEGQGVFEATTEFSVDARAL TQTGGPHVKARVANPSGNLTETVYQDRGDGMKYVEYTPYEEGLHSVDVTYDGSVPVS
SPFQVPVTEGCDPSRVRVHGPQISGTTNKNPKFTVETRAGTGGLGLAVEGPSEAKMSCMDNKGSCSVEYIPYEAGTYS LNVTYGGHQVPGSPFKVPVHDVTDASKVKCS
GPG LSPGMVRANLPQSFOVDTSKAGVAPLQVKVQGPGLVEPVDVVDNADGTQTVNYPVSPREGPYISVL YGDEEVPSPFKVKVLP THDASKVKASGPG LNTTGVPASLPV
EFTIDAKDAGEGLLAVQITDPEGKPKKTHIQDNHDTYTVAYVPDVTGRTYILIKYGGDEIPFSPYRVRVPTGDASKCTVTVSI GGHGLGAGIGPTIQIGETVITVDTKA
AGKGKVTCTVCTPDGSEVDVVDVNEEDGTDFIFYTAPQPGKYVICVRFGGEHVPNPSPFQV TALAGDQPSVQPLRLRSQQLAPQYTYAQGGQQTWAPERPLVGVNGLDVTSLRP
FDLVIPTITIKGEITGEVRMPSPGKVAQPTITDNKDGTVTVRYAPSEAGLHEMDIRYDNMHI PGSP LQFYVDYVNCGHVITAYGPG LTHGVVNKPATFTVNTKDAGEGGLSLAI
EGPSKAEISCTDNQDGTCSVSYLPVLPDYSILVKYNEQHVPGSPFTARVTGDDSMRMSHLKVGSAADIPINIS ETDLSLLTATVVPSPGREEPCLLKR LRNGHVGISFVPK
ETGEHLVHVKNQGHVASSP IPVVISQSEIGDASRVRSVGGGLHEGHTFEPAEFTIDTRDAGYGG LSLIEGSPKVDINTEDLEDGT CRVTCPTPEGNYIINI KFAHQHVP
GSPFSVKVTGEGRVKESITRRRRAPSVANVGSCHDLSLKIPEISIQDMTAQVTSPSGKTHEAEIVGENHTYCI RFVPAEMGMHTVSVKYKQGHVPGSPFQFTVGPLGEGGA
HKVRAGGPGGLERAEGVPAEFGIWREAGAGGLA IAVEGPSKAEISFEDRKDSCGVAYVYVQEPGDYEVSVKFNEEHIPDSPFVVPVASPSGDARRLTVSS LQESGLKVNQ P
ASFVSLNGAKGAI DAKVHSPSGALEECYVTEIDQKYAVRFIPRENGIYLIDVFNHGTIHPGSPFKIRVGE PGHGGDPGLVSAYGAGLEGGVGTGSPA EFTVNTSNAGAGAL
SVTIDGPSVKVMDCQCEPGEYRVYTPMAPGSYLSIKYGGPYHIGGSPFAKVTGPRLVSNHSLHETSSVFVDSLTKATCAPQH
APGPGPADASKVAKGLGLSKAYVGQKSNFTVDCSKAGNNM L LGVHGPRTPC EILVKHMGSR LYSVSYLLKDKGEYTLVWKWGEH IPGSPYRIMVP ' ]
```

Figura 3: Segundo resultado

- Comparación del genoma del virus del Dengue con el virus de Hepatitis C.

```
score maximo: 32
[ 'GATGGGACACAAGAATAACAGAGGATGA' ]
[ 'GTTTGGACTCAACAGTACTGAGAATGA' ]
[Finished in 50.2s]
```

- Comparación del genoma del virus del Dengue con el ebola.

```
score maximo: 29
['CGTGACCAGAGAGAAAG']
['CGTGACAGAAAGAAAG']
[Finished in 94.8s]
```

1.4. Evalúe la pregunta anterior con otros valores de gapOpen; gapEXTEND; identicalMatch y mismatch. Verifique si obtiene el mismo alineamiento y comente sus resultados.

- **Proteína Filamin-A en Ratones y Humanos** Con gapOpen=gapExtend=-5 y identicalMatch=1 y mismatch=-1

```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.476]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\ErikaPC>cd C:\Users\ErikaPC\Documents\Cursos2020-1\BioinformaticaMolecular\Alineamientos

C:\Users\ErikaPC\Documents\Cursos2020-1\BioinformaticaMolecular\Alineamientos>python practica3.py
score maximo: 2592
['VPQHATSGPGADVSKVAKGLGLSKAYVGQKSNFTVDCSKAGNNMLLVGVHGPRTPEEILVKHMGSRLYSVSYLLKDKGEYTLVVKWGDHPIGSPYRIMVP']
['APQHAGPGPADASKVAKGLGLSKAYVGQKSNFTVDCSKAGNNMLLVGVHGPRTPEEILVKHMGSRLYSVSYLLKDKGEYTLVVKWGDHPIGSPYRVVVP']

C:\Users\ErikaPC\Documents\Cursos2020-1\BioinformaticaMolecular\Alineamientos>
```

Figura 4: Caption

Tenemos como Score máximo 2592.

- **Proteína Filamin-A en Ratones y Humanos** Con gapOpen=gapExtend=-3 y identicalMatch=2 y mismatch=-1

```
as np
1,-1
2,-1
-1,2
-1,-1
score maximo: 5211
['VPQHATSGPGADVSKVAKGLGLSKAYVGQKSNFTVDCSKAGNNMLLVGVHGPRTPEEILVKHMGSRLYSVSYLLKDKGEYTLVVKWGDHPIGSPYRIMVP']
['APQHAGPGPADASKVAKGLGLSKAYVGQKSNFTVDCSKAGNNMLLVGVHGPRTPEEILVKHMGSRLYSVSYLLKDKGEYTLVVKWGDHPIGSPYRVVVP']
Presione una tecla para continuar . . .
```

Figura 5: Caption

Tenemos como Score máximo 5211

- **Proteína Filamin-A en Ratones y Humanos** Con gapOpen=gapExtend=-6 y identicalMatch=1 y mismatch=-1

```
Administrator: C:\Windows\system32\cmd.exe
score maximo: 2591
['MSSSHRCGSAASVPGGSDSRDAEMPAEKDLAEDAPWKKIQNTITFRMCNEHLKCVSKRIANLQDLSGLRLIALLEVLSQLMHRKHQBPFRQMLQENVSVALFFLDRES
IKLVSDSKAIVDGNLKLILGLINTLILHYSIMPMDEEEDAEAKQTPKQRLGLWQNKLPQLPITNFSRDMQSGRALGALVDSAPGLCPDMSDASPKVNNAREANQADQDLGI
POVITPEEIVDPINDEHSVNTYLSQPKAKLKPAPLRPLNPKKARAYGPGLEPTGNMVKRAEFTVETRSAGQGEVLVYVEDPAGHQEAKVTANIDKNRTFSVNVYPEVTGTHKVTY
LFAQHQIAKSPFEVYVQKSGQDASKVTAQGRLEPSGNIAKITYEFTTAGAGNEVEVQDTPGQGTVEPQLEARGDSTYKESYQPTMEGVHTVHTFAGVPIPRSPYPTVTVGQAC
NPAACRAIGRLQPKGVYKETADEFKVTYKAGSGELKVTYKPKGEERVKQDLDGVYGF EYYPITPGTYTITNGQNTGRSPFEVKGTCEGNQKVRAMGPLEGGTVKGSADYV
VEATIGDVGTLGFSVEGSPQAKIECDKGDGSDVRVMPQAGEYAVHVL CNSIEDRLSPFMADIRAPQDFHPRVKARGPGL EKTGVAVNPAEF TVDAKHGAPLRVQVDNQGCS
VEATVCDNGMTSYSCVVPKPKVHTAMVSWGGVSI PMSFPRVWVGAGSHNKKVYGPVGAKTGLKAHEPTYFTVDCTEAGQGDVSI GIKKCAPGVVGPTEADIDFIDIRINDIDFTYKY
TPCAGASYTINVLFDQATPTSPVIRKVEPSHDASKVKAEGPLNRTGVELGKPTHTVNAKTAGKGLDQVSGLAKGDARVDVQIDHIDHNTYTVKYIPVQSGPVGNVTVYGGHPIK
SPFSVSPVSPDLSLILVSGLDGVVGVKDEFTVSKSGAGGQKSKIEIUSPSGALVPCVEPLGADGNSVRFVPRGEGPFEVEVYDGVVPVSGPFLAFAVPTKPSKXKAFGRLQ
GNGAGSPARTIDTGAAGTGLGLVYGPCEAQLECLDNGDGTCSVSVPTPEGDYINILFADTHIPGSPKKAHAPCFDASKVKCSGPLERATAGEVGQFQDSCSAGSAELITETC
SEAGLPAEYVTDHGGDHTITTYIPLCPGAYTVTKYGGQVVPNFPKSLQVEPAVDTSGVQCYGPGIEGQGVREATTETFSVDARALTQTGGPHVKARVANPSGMLTDTYVQDGGCTYK
VEYTPVEEGHSDVTDYDGSVPVSPFPQVPVTEGDCPSRVRVHGPGLQSGTNNKPKFTVETRGAGTGGGLGAVEGSPSEAKMSCHDNKDGSCSV EYIPYAGTYSLVNTYVGGHVPQSPF
KVPVHDVTDASVKVCSGPGELSPGMVRANILPQSFQDTSKAGVAPLQVKGQPKGLVEPVVDVNDAGDTQTVVVPVSGESYSISVLYGEEVPRSPFKVVLPTHDAKVKVAGSGPLNTT
GVPLSLPVEFTIDAKDAGEGLLAQVTDPEKPKTHIDQHDGTYTVAVPPVPPERTILLVYGGDEEPPSPYRRAVPTBASKCTVTVSYSGHLAGGEPITQIGETVITVDTKA
AGKGVKTCTVCTPDGSEVDVVDVENEEDGTFDYTAPOQKGYVLCVRFGEHVPNSPQVOTALAGQPTVQTLRLSQAAPQVNPQSSQTHIPERPMWNGMLDVTSLRPDLVIEFT
IKKGEITGEVRMPSGKAQPSITDNDGDTVTVRYSPEAGLHENDIRYDNNHIGPSPLOFYDYVNGCHITAYGPGLTGHGVNKPATFTVNTKDAGEGGLSLAIEGPKSAEISCTDNDG
TCSVSLPVLPGDYSLVKVNDQHIPGSPFTTARVTDGDSMRHSHLKGSAADIPINISSETDLSLLTATVPPSGREEPCLLKLRLNHHVGISFVPKETGEHLVHKKHGHVASSPIPVV
ISQSEITGASRVVRSQGLHEHTEFPAEFTIDTRADYGGLSLSTIEGPKVDINTEDLEDGTCRTVCTPEPNVITINIKFADQHVHSGSPFSKVKTGEGRVKESITRRRAPSVAJIGS
HCOLSLTPEFTSITDNTAQVTPSGKXTHAEIVGEBHTITVTRVPLENHNHVSXVKGQHVGPSPFTVPLGEGGAKHVRAGGPLEAEVQVPAEFGHTREAGAGGLLAIVEGP
SKAETSKEFDRKQSGVAVYVQPGDYEVSKFNEEHTIPSPFPVVPASPSGDARLLTVSSIQESGLVYNQKPSFAVSLNKAAGTADAKVHSPGALIECYVTEIDQKQVAVRFIPREIN
IYLDVFNHTGTHIPGSPFKIRVGEHPHGDRGLVSAYAGLGGVTPSAEFTVNTSMAGAGALSVTIDGPKVKVNMQDCEPCEGVRVITYTPMAGSYLSIKYGGPYHTIGGSPKAKVTG
PRLVSHSLGHTSSVFGVDSLTKVATVPQHATSGPGADVSKVAKGLGLSKAYVGQKSNFTVDCSKAGNNMLLVGVHGPRTPEEILVKHMGSRLYSVSYLLKDKGEYTLVVKWGDHPI
GSPYRIMVP']
(1)
(2)
```

Figura 6: Caption

Tenemos como Score máximo 2591

- **Con las secuencias de archivos ebola.fasta y hepatitis.fasta** Con gapOpen=gapExtend=-5 y identicalMatch=1 y mismatch=-1

```
C:\WINDOWS\system32\cmd.exe
score maximo: 17
['CCAAGGACCCACCAATACCAC']
['CCAAGGCCCTCCAATACCAC']
Presione una tecla para continuar . . .
```

Figura 7: Caption

Tenemos como Score máximo 17

- Comparación del genoma del Dengue y ebola, con gapOpen=gapExtend=-10 y identicalMatch=1 y mismatch=-1.

```
score maximo: 26
['AAGGCTGAAAAGAAT' 'AAGGCTGAAAAGAATGGCAATCAGC' 'ACCCCAACCTAGCTGAGAGA'
'ACCCCAACCTAGCTGAGAGAGCTC' 'AGGTCCAATAAGACAAATG' 'CAAGTCATATGCA'
'CAAGTCATATGCACA' 'GATGGAGCTTGAGAGAACTGCTTGCCTAGG' 'GGACACAAGAATAAC'
'TACCAACATGGAGGTCCAATAAT' 'TGATCTTCAGAAATGAAGCTAAAATCAC']
['AAAGCTGAAAAGAAT' 'AAAGCTGAAAAGAATTTCTCCAGC' 'ACGCTCAACCTGCTGAGAGA'
'ACGCTCAACCTGCTGAGAGACATC' 'AGGACCCACCAATACCACAAATG' 'CAAGTCATATGCA'
'CAAGTCATATGCAAA' 'GAGGAAGATTAAGAAAACTGCAGTCCCAGG' 'GGACCAAGAATAAC'
'TACCCACCTGGTGGCCAGCCAAT' 'TGATCAGCAAAGTGACCTAAAATGAC']
```

Conclusiones:

- Al aumentar la penalización por gap se puede notar que hay un menor camino.

Referencias

[1] <https://www.viprbrc.org/>