**#3.1**

i.

A. Dot Plot

B. The dot plot would be suitable to visualize a sequence inversion as it convenient to map dot and assess result.

ii.

A. Global Sequence Alignment

B. In global sequence alignment, we can get score to compare highly conserved ribosomal RNA sequences from five insect species.

iii.

A. Semi Global Sequence Alignment

B. The semi global sequence alignment is the best match without penalizing gaps on the ends of the alignment.

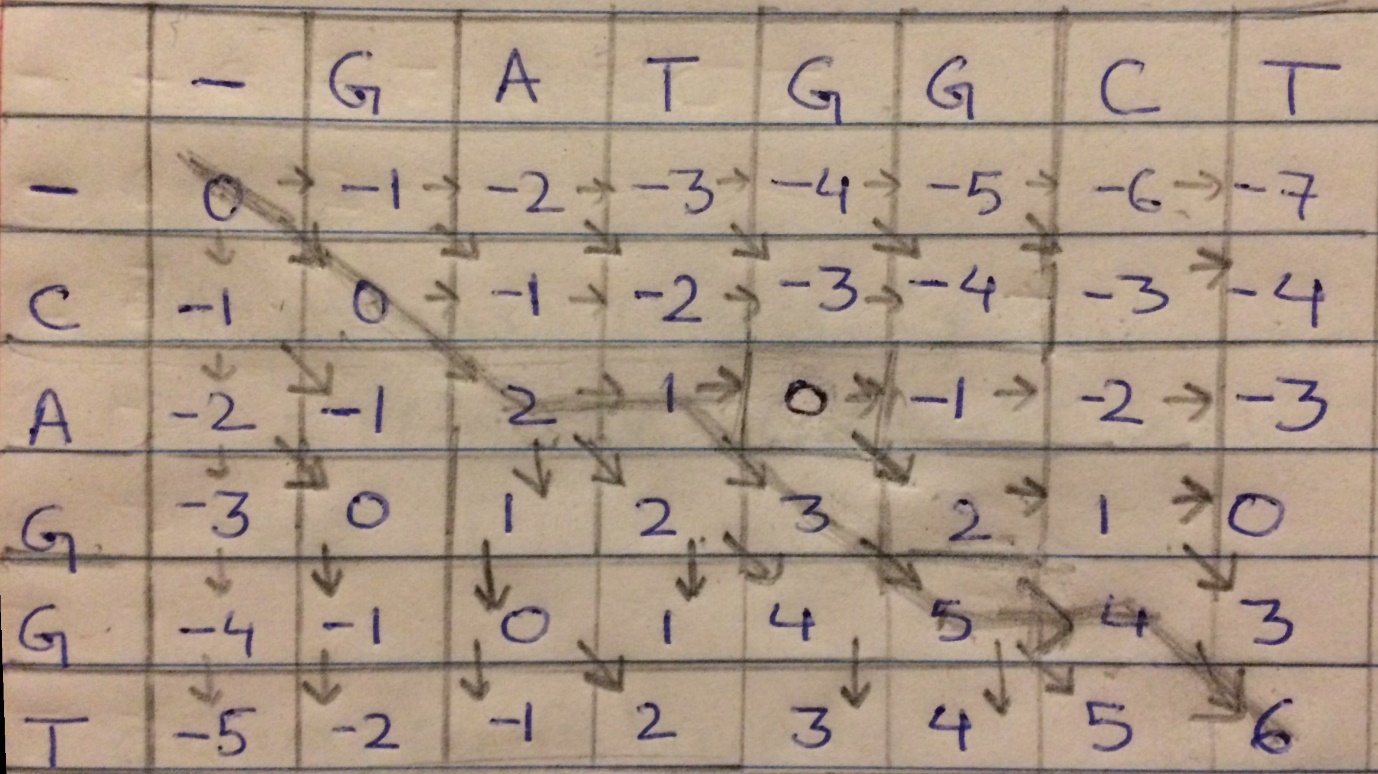
iv.

A. Local Sequence Alignment

B. Local Sequence Alignment is best suited if you want to align conserved segment of two very dissimilar sequences.

**#3.2**

A&B



**C.** 6

**D.**

G A T G G C T

| | | | | | |

C A - G G - T

**E.**

No ties

**#3.3**

**A**

#3.3A

c=0

A = "GGGAC"

B = "CGGAC"

for i in range(len(A)): #for loop of len of A string

for j in range(len(B)): #for loop of len of B string

if(A[i]==B[j]): #checking for a match

c+=1 #counting matches

if (c==1):

print("Matches:\n")

print("A" , i+1 , "and" , "B", j+1,"\n")

else:

print("A", i+1 , "and","B" , j+1, "\n")

print("Total matches:","\n",c)

**B**

#3.3B

c=0

A = "ACTTGGCCAT"

B = "AGTAGCGCCT"

for i in range(len(A)): #for loop of len of A string

for j in range(len(B)): #for loop of len of B string

if(A[i]==B[j]): #checking for a match

c+=1 #counting matches

if (c==1):

print("Matches:\n")

print("A" , i+1 , "and" , "B", j+1,"\n")

else:

print("A", i+1 , "and","B" , j+1, "\n")

print("Total matches:","\n",c)

**C**

The possible intersection between two sequences of length 10 is 100.  The probability of having a nucleotide match in any position: 1/4 probability of a match.  So we can expect 25 matches.

The two sequence in the B are random.