

# project

April 22, 2020

```
[ ]: def find(element, matrix):  
    for i, matrix_i in enumerate(matrix):  
        for j, value in enumerate(matrix_i):  
            if value == element:  
                return (i, j)
```

```
[6]: def ALIGN (x, y):  
    ALIGNx = []  
    ALIGNy = []  
    N = len(x)  
    M = len(y)  
    A = [0] * (N + 1) #  
    path = [0] * (N) #  
    W = [0] * (N) #  
    for i in range(N + 1):  
        A[i] = [0] * (M + 1)  
    for i in range(N):  
        path[i] = [0] * (M)  
        W[i] = [0] * (M)  
    for i in range(N + 1):  
        for j in range(M + 1):  
            A[0][j] = -2 * (j)  
            A[i][0] = -2 * (i)  
    for i in range(1, N + 1):  
        for j in range(1, M + 1):  
            d = A[i - 1][j - 1]  
            if x[i - 1] == y[j - 1]:  
                d += 1  
            if x[i - 1] != y[j - 1]:  
                d -= 1  
            v_put = max(A[i][j - 1] - 2, A[i - 1][j] - 2, d)  
            A[i][j] += v_put  
    W[0][0] = A[1][1]  
    path[0][0] = 'd'  
    for i in range(N):  
        for j in range(M):  
            if (i == 0 and j != 0):
```

```

        W[i][j] = W[i][j - 1] + A[i][j]
        path[i][j] = 'l'
    if (j == 0 and i != 0):
        path[i][j] = 'u'
        W[i][j] = W[i - 1][j] + A[i][j]
    if (i != 0 and j != 0):
        s = max(W[i][j - 1], W[i - 1][j], W[i - 1][j - 1])
        if (s == W[i][j - 1]):
            path[i][j] = 'l'
        if (s == W[i - 1][j]):
            path[i][j] = 'u'
        if (s == W[i - 1][j - 1]):
            path[i][j] = 'd'
        W[i][j] = s + A[i][j]

i, j = 0, 0
ALIGNx.append(x[i])
ALIGNy.append(y[j])
while (i != N - 1 and j != M - 1):
    if (i < N - 1 and j < M - 1):
        MAX = max(W[i + 1][j + 1], W[i][j + 1], W[i + 1][j])
        if MAX == W[i + 1][j + 1]:
            i += 1
            j += 1
            ALIGNx.append(x[i])
            ALIGNy.append(y[j])
        else:
            if MAX == W[i][j + 1]:
                j += 1
                ALIGNx.append('-')
                ALIGNy.append(y[j])
            else:
                if MAX == W[i + 1][j]:
                    i += 1
                    ALIGNx.append(x[i])
                    ALIGNy.append('-')
    if (i == N - 1):
        j += 1
        ALIGNx.append('-')
        ALIGNy.append(y[j])
    if (j == M - 1):
        i += 1
        ALIGNx.append(x[i])
        ALIGNy.append('-')
return (ALIGNx, ALIGNy)

```

```

[10]: FASTA_input = open('input.fas', 'r')
Sequences = FASTA_input.read()

```

```
FASTA_input.close()
```

```
[ ]: Seq = []
Seq_inner = []
Seq = Sequences.split('>')
for i in Seq:
    Seq_inner.append(i.split('\n'))
print(Seq_inner)
Seq_inner.remove(Seq_inner[0])
#Seq_inner[3].remove('')
#Seq_inner[3].remove('')
sequences = []
for i in Seq_inner:
    s = ''
    for j in range(1,len(i)):
        s += i[j]
    sequences.append(s)
seq = max(sequences[0],sequences[1])
seqi = min(sequences[0],sequences[1])
Out = open('output.fas', 'w')
STRAX = ALIGN(seq,seqi)
for i in range(len(STRAX)):
    Out.write(str(STRAX[i]))
    Out.write('\n')
Out.close()
```