

## Biyoinformatiğe Giriş Proje 3

Mahire Zühal Özdemir

19360859015

### Adım 1: Global Alignment:

Aşağıdaki kod ile hiçbir kütüphane kullanmadan Python dilinde matrisler yardımı ile global hizalama yapıldı.

```
def globalAlignment():
    # for global alignment --> Needleman-Wunsch
    seq1 = "GATTACA"
    seq2 = "GCATGCU"
    match = 1
    mismatch = -1
    gap = -1
    m = len(seq1) + 1
    n = len(seq2) + 1
    score_matrix = [[0] * n for _ in range(m)]
    # 0'lardan oluşan matris
    for i in range(1, m):
        score_matrix[i][0] = score_matrix[i-1][0] + gap
        # oluşan matrisin ilk sütunu gap cezası ile doldurulur
    for j in range(1, n):
        score_matrix[0][j] = score_matrix[0][j-1] + gap
        # oluşan matrisin ilk satırı gap cezası ile doldurulur
    for i in range(1, m):
        for j in range(1, n):
            match = score_matrix[i-1][j-1] + (match if seq1[i-1] == seq2[j-1] else mismatch)
            # match ya da mismatch olma durumuna göre skor belirlenir
            delete = score_matrix[i-1][j] + gap
            insert = score_matrix[i][j-1] + gap
            score_matrix[i][j] = max(match, delete, insert)
            # gideceği yolu max puan durumuna göre belirler
```

```
# gideceği yolu max puan durumuna göre belirler

alignment1 = ""
alignment2 = ""
i, j = m-1, n-1
while i > 0 or j > 0:
    if i > 0 and j > 0 and score_matrix[i][j] == score_matrix[i-1][j-1] + (match if seq1[i-1] == seq2[j-1] else mismatch):
        alignment1 = seq1[i-1] + alignment1
        alignment2 = seq2[j-1] + alignment2
        i -= 1
        j -= 1
    elif i > 0 and score_matrix[i][j] == score_matrix[i-1][j] + gap:
        alignment1 = seq1[i-1] + alignment1
        alignment2 = "-" + alignment2
        i -= 1
    else:
        alignment1 = "-" + alignment1
        alignment2 = seq2[j-1] + alignment2
        j -= 1
# hizalama sonunda oluşan sekanslar alignment1 ve alignment2'dir
print(alignment1)
print(alignment2)
# skor matrisinin son sütun ve son satırdaki verisi bize skoru verir.
print("global alignment skor: ", score_matrix[m-1][n-1])
```

## Adım 2: Local Alignment:

Aşağıdaki kod ile hiçbir kütüphane kullanılmadan Python dilinde matrisler yardımı ile local hizalama yapılmıştır.

```
def localAlignment():
    #smith-waterman bayer algoritması
    seq1 = "KVLEFGY"
    seq2 = "EQLLKALEFKL"
    # match,mismatch ve gap cezaları belirlenir
    gap = -1
    match = 4
    mismatch = -2
    score_matrix = [[0 for j in range(len(seq2) + 1)] for i in range(len(seq1) + 1)]
    #0'lardan oluşan skor matrisi tanımlanır
    traceback_matrix = [[0 for j in range(len(seq2) + 1)] for i in range(len(seq1) + 1)]

    # skor matrisi doldurulur
    max_score = 0
    max_i, max_j = 0, 0
    for i in range(1, len(seq1) + 1):
        for j in range(1, len(seq2) + 1):
            # skor match ve mismatch için hesaplanır
            if seq1[i - 1] == seq2[j - 1]:
                match_or_mismatch = match
            else:
                match_or_mismatch = mismatch
            # her yön için skor hesaplanır
            diagonal_score = score_matrix[i - 1][j - 1] + match_or_mismatch
            up_score = score_matrix[i - 1][j] + gap
            left_score = score_matrix[i][j - 1] + gap
            # max skor bulunur ve traceback matrisine uyarlanır
```

```
            # max skor bulunur ve traceback matrisine uyarlanır
            score_matrix[i][j] = max(0, diagonal_score, up_score, left_score)
            if score_matrix[i][j] == 0:
                traceback_matrix[i][j] = 0
            if score_matrix[i][j] == left_score:
                traceback_matrix[i][j] = 1
            if score_matrix[i][j] == up_score:
                traceback_matrix[i][j] = 2
            if score_matrix[i][j] == diagonal_score:
                traceback_matrix[i][j] = 3
            # max skor güncellenir ve yeniden hizalanır
            if score_matrix[i][j] > max_score:
                max_score = score_matrix[i][j]
                max_i, max_j = i, j

    # oluşan alignment'ı getirmek için traceback matrisi kullanılır
    align1, align2 = "", ""
    i, j = max_i, max_j
    #align;matris üzerinde dolaşarak bulunur.
    while traceback_matrix[i][j] != 0:
        if traceback_matrix[i][j] == 3:
            align1 = seq1[i - 1] + align1
            align2 = seq2[j - 1] + align2
            i -= 1
            j -= 1
        elif traceback_matrix[i][j] == 2:
            align1 = seq1[i - 1] + align1
            align2 = "-" + align2
            i -= 1
```

```

while traceback_matrix[i][j] != 0:
    if traceback_matrix[i][j] == 3:
        align1 = seq1[i - 1] + align1
        align2 = seq2[j - 1] + align2
        i -= 1
        j -= 1
    elif traceback_matrix[i][j] == 2:
        align1 = seq1[i - 1] + align1
        align2 = "-" + align2
        i -= 1
    elif traceback_matrix[i][j] == 1:
        align1 = "-" + align1
        align2 = seq2[j - 1] + align2
        j -= 1

# align ve skor ekrana yazdır
print(align1)
print(align2)
print("local alignment skor:", max_score)

```

### Adım 3: Fonksiyonları Çağırma

```

globalAlignment()
print(".....")
localAlignment()

```

### Adım 4: Sonuç

```

C:\Users\zuhal\Desktop\bioinformaticProjects\venv\Scripts\python.exe C:/Users/zuhal/Desktop/bioinformaticProjects/main.py
G-ATTACA
GCA-TGCU
global alignment skor: 0
.....
KVLEF
KALEF
local alignment skor: 14

```