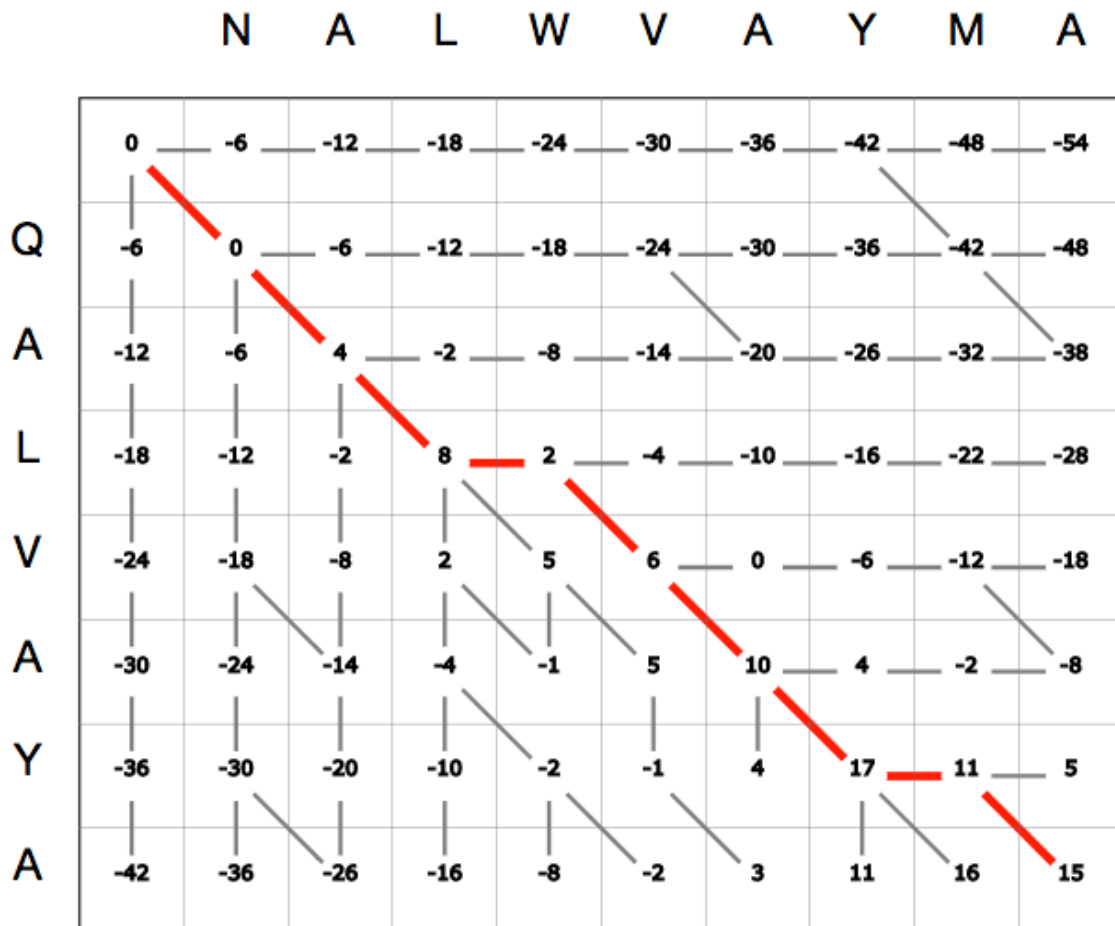


Needleman-Wunchs with Blosum62 score



Tugas 1 Bioinformatika dan NLP

Gama Candra Tri Kartika

15/378060/PA/16535

Latar Belakang

1. Algoritma Needleman-Wunchs

Algoritma Needleman-Wunsch adalah algoritma yang digunakan dalam bioinformatika untuk menyelaraskan urutan protein atau nukleotida. Ini adalah salah satu aplikasi pertama dari pemrograman dinamis untuk membandingkan *sequence* biologis. Algoritma ini dikembangkan oleh **Saul B. Needleman** dan **Christian D. Wunsch** dan diterbitkan pada tahun 1970. Algoritma dasarnya membagi masalah besar (misalnya sequence penuh) menjadi serangkaian masalah yang lebih kecil dan menggunakan solusi untuk masalah yang lebih kecil untuk merekonstruksi solusi untuk masalah yang lebih besar. Ini juga kadang-kadang disebut sebagai algoritma pencocokan optimal dan teknik global alignment. Algoritma Needleman-Wunsch masih banyak digunakan untuk global alignment yang optimal, terutama ketika kualitas global alignment adalah yang paling penting.

2. BLOSUM62 score

Dalam bioinformatika, matriks BLOSUM (**BLO**cks **SU**bstitution **M**atrix) adalah matriks substitusi yang digunakan untuk penyelarasan urutan protein. Matriks BLOSUM digunakan untuk menilai keberpihakan antara sekuens protein yang berbeda secara evolusi. Mereka didasarkan pada keberpihakan lokal. Matriks BLOSUM pertama kali diperkenalkan di sebuah makalah oleh **Steven Henikoff** dan **Jorja Henikoff**. Mereka memindai database BLOCKS untuk daerah-daerah yang dilestarikan dari famili protein (yang tidak memiliki celah dalam sequence) dan kemudian menghitung frekuensi relatif asam amino dan probabilitas substitusi mereka. Kemudian, mereka menghitung skor log-odds untuk masing-masing dari 210 pasangan substitusi yang mungkin dari 20 asam amino standar. Semua matriks BLOSUM didasarkan pada keberpihakan yang diamati; mereka tidak diekstrapolasi dari perbandingan protein yang terkait erat seperti Matriks PAM.

Implementasi

1. Langkah pertama yang dilakukan adalah memasukkan Package yang digunakan yaitu Numpy untuk proses penghitungan matriks dan Random untuk rangkaian DNA yang dibuat secara acak
2. Lalu masukkan proses aturan scoring dan Tabel BLOSUM62
3. (optional) Buatlah suatu fungsi Random untuk digunakan saat menguji dataset rangkaian asam amino yang random
4. Buat variabel kosong untuk menampung array yang akan dimasukan dalam proses sequence data
5. (optional) Masukkan nilai variabel dari banyak pasangan yang akan digunakan dan panjang dari nilai asam amino yang random tadi
6. (optional) Buat perulangan sebanyak banyaknya jumlah pasangan yang diinginkan
7. Inisialisasi score_matrix dan trace_matrix untuk membuat rangkain matriks yang akan digunakan untuk scoring dan tracing
8. Bentuk penalty saat terjadi extension vertikal maupun horizontal
9. Untuk seluruh array dari kedua pasangan sequence diuji dan dicek match atau mismatch. Jika terjadi Mismatch akan diberikan penalty lagi
10. Dapatkan Skor terbaik
11. (optional) Bentuklah dari riwayat dalam tracing matriks tadi
12. tampilkan hasil dari score_matrix, Trace_matrix, Riwayat scoring, Riwayat tracing dan kedua sequence yang sudah diubah alignmentnya

Source Code

Tertera di Needleman-Wunchs.ipynb dan <https://github.com/Saltfarmer/Bioinformatics-class/blob/master/NW.py>

Hasil

1. Untuk random panjang 10 dan 1 iterasi

Alignment: 1

MYETGVCQYM

MMMTWTREGT

Alignment Needleman-Wunsch...

Score Matrix:

```
[[ 0. -6. -12. -18. -24. -30. -36. -42. -48. -54. -60.]
 [-6.  5.  0.  0.  0.  0.  0.  0.  0.  0.  0.]
 [-12.  0.  4.  0.  0.  0.  1.  0.  0.  0.  5.]
 [-18.  0.  0.  2.  0.  0.  1.  0.  0.  0.  5.]
 [-24.  0.  0.  0.  6.  1. -2.  0.  0.  0.  0.]
 [-30.  0.  2. -3.  0.  4. -2. -4. -2.  2.  0.]
 [-36.  0.  0.  2.  1.  1.  2. -3. -4. -4.  1.]
 [-42.  0.  0.  0.  1. -1. -2. -1. -2. -6. -5.]
 [-48.  0.  0.  5.  0. -1. -3. -6.  1. -4. -8.]
 [-54.  0.  0.  0.  6.  6.  0.  0.  0.  0.  0.]
 [-60.  0.  0.  0.  4.  7.  4.  0.  0.  0.  0.]]
```

Trace Matrix:

```
[[ 'N' 'L' 'L' 'L' 'L' 'L' 'L' 'L' 'L' 'L' 'L' ]
 [ 'U' 'D' 'U' 'U' 'U' 'U' 'U' 'U' 'U' 'U' 'U' ]
 [ 'U' 'L' 'D' 'U' 'U' 'U' 'D' 'U' 'D' 'U' 'D' ]
 [ 'U' 'U' 'L' 'D' 'U' 'U' 'D' 'D' 'D' 'U' 'D' ]
 [ 'U' 'U' 'U' 'D' 'D' 'D' 'D' 'D' 'D' 'U' 'L' ]
```

```

[U' 'U' 'D' 'D' 'U' 'D' 'D' 'D' 'D' 'D' 'U']
[U' 'U' 'L' 'D' 'D' 'D' 'D' 'D' 'D' 'D' 'D']
[U' 'U' 'U' 'D' 'D' 'D' 'D' 'D' 'D' 'D' 'D']
[U' 'U' 'U' 'D' 'D' 'D' 'D' 'D' 'D' 'D' 'D']
[U' 'U' 'U' 'L' 'D' 'D' 'L' 'L' 'L' 'L' 'L']
[U' 'U' 'U' 'D' 'D' 'D' 'D' 'U' 'D' 'U' 'U']]

```

Scoring - Backtrace - Alignment:

```

[-6.0, -12.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 6.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0]
UULUUUULDDDDLULLLU
--M---YETGVCQYM-
MM-MTWT-REG-----T
score: 0.0

```

2. Untuk NALWVAYMA dan QALVAYA

Alignment: 1

```

NALWVAYMA
QALVAYA

```

Alignment Needleman-Wunsch...

Score Matrix:

```

[[ 0. -6. -12. -18. -24. -30. -36. -42. -48. -54.]
 [-6. 0. 0. 0. 0. 0. 0. 0. 0. 0.]
 [-12. 0. 4. 0. 0. 0. 4. 0. 0. 4.]
 [-18. 0. 0. 8. 2. 1. -1. 3. 2. -1.]
 [-24. 0. 0. 3. 5. 6. 0. 0. 1. 0.]
 [-30. 0. 4. -1. 0. 5. 10. 4. 0. 5.]
 [-36. 0. 0. 3. 1. -1. 4. 17. 11. 0.]
 [-42. 0. 4. -1. 0. 1. 3. 11. 16. 15.]]

```

Trace Matrix:

```

[["N" 'L' 'L' 'L' 'L' 'L' 'L' 'L' 'L' 'L']
["U" 'D' 'U' 'U' 'U' 'U' 'U' 'U' 'U' 'U']
["U" 'L' 'D' 'U' 'U' 'D' 'D' 'U' 'U' 'D']
["U" 'U' 'L' 'D' 'L' 'D' 'D' 'D' 'D' 'D']
["U" 'U' 'U' 'D' 'D' 'D' 'L' 'L' 'D' 'D']
["U" 'U' 'D' 'D' 'D' 'D' 'D' 'L' 'L' 'D']
["U" 'U' 'L' 'D' 'D' 'D' 'U' 'D' 'L' 'L']
["U" 'U' 'D' 'D' 'D' 'D' 'D' 'U' 'D' 'D']]

```

Scoring - Backtrace - Alignment:

```
[0.0, 4.0, 8.0, 2.0, 6.0, 10.0, 17.0, 11.0, 15.0]
```

DDDLDDDL

NALWVAYMA

QAL-VAY-A

score: 15.0