Wstęp do bioinformatyki

Dopasowanie globalne

Monika Reguła 236689

Programowanie dynamiczne jest szeroko stosowaną metodą, która gwarantuje odnalezienie optymalnego ustawienia względem zadanego systemu oceny; opracowano usprawnienia metody o złożoności bliskiej liniowej:
- globalne dopasowanie (Needleman-Wunsch)
- lokalne dopasowanie(Smith-Waterman)

W tym ćwiczeniu opisane jest dopasowanie globalne. Porównuję każdą parę znaków dwóch sekwencji oraz tworzę dopasowanie. Uwzględniam wszystkie możliwe przyrównania uzwględniając (system punktacji):

- dopasowania
- niedopasowania
- przerwy

GlobalMatching:

Przerwy są wstawiane, aby uzykać wzrost liczby dopasowani w innych miejscach. Znajduję optymalne dopasowanie (jednak może ich być kilka). Tworzę macierz punktacji, w której każda komórka reprezentuje punktację dla najlepszego dopasowania kończącego się w danej pozycji.Wybieram maksymalną wartość z 3 możliwych(lewo,prawo,przekątna):

```
[maxScore, maxIndex] = max([diagonal left up]);
```

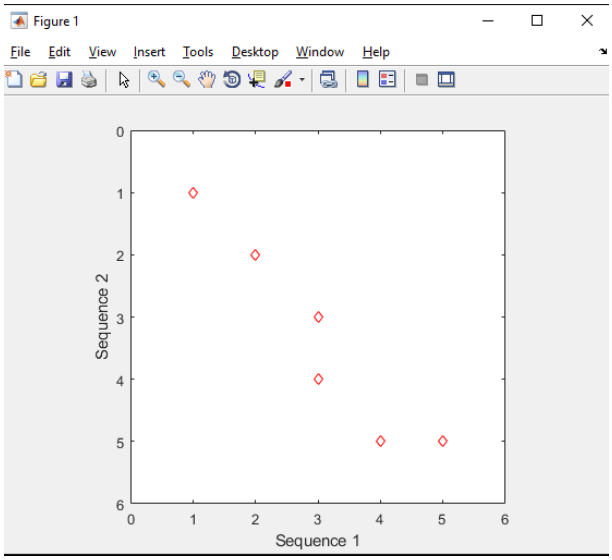
Po wykonaniu macierzy punktacji przechodzę do poszukiwania najbardziej optymalnej ścieżki

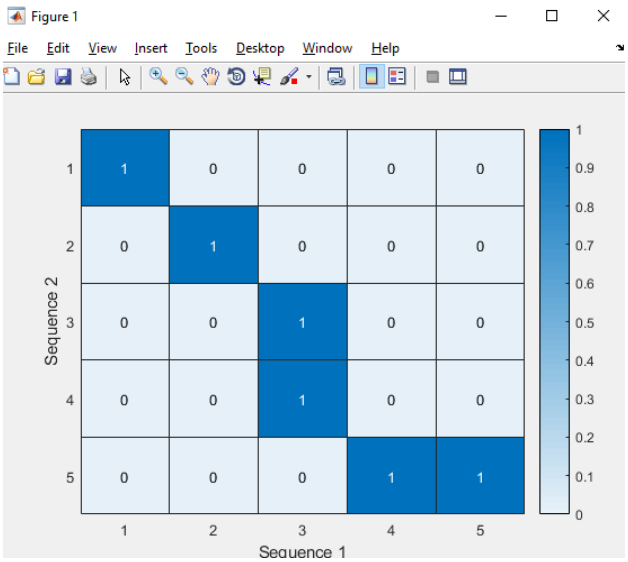
Traceback:

Kopiuję macierz punktacji i cofam się w macierzy, żeby znaleźć optymalne dopasowanie. Zaczynam od prawego górnego rogu (wstawiam wartość = 1) i chce zakończyć poszukiwania na lewym górnym rogu (wstawiam wartość = 1).
Następnie w pętli while, której warunkiem jest sprawdzanie czy komórka nie jest lewym górnym rogiem czy też prawym dolnym. W bloku while dochodzi do mapowania kierunku drogi.
Zaczynam od takiej postaci :

```
optimalPath =

     1     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     1
```

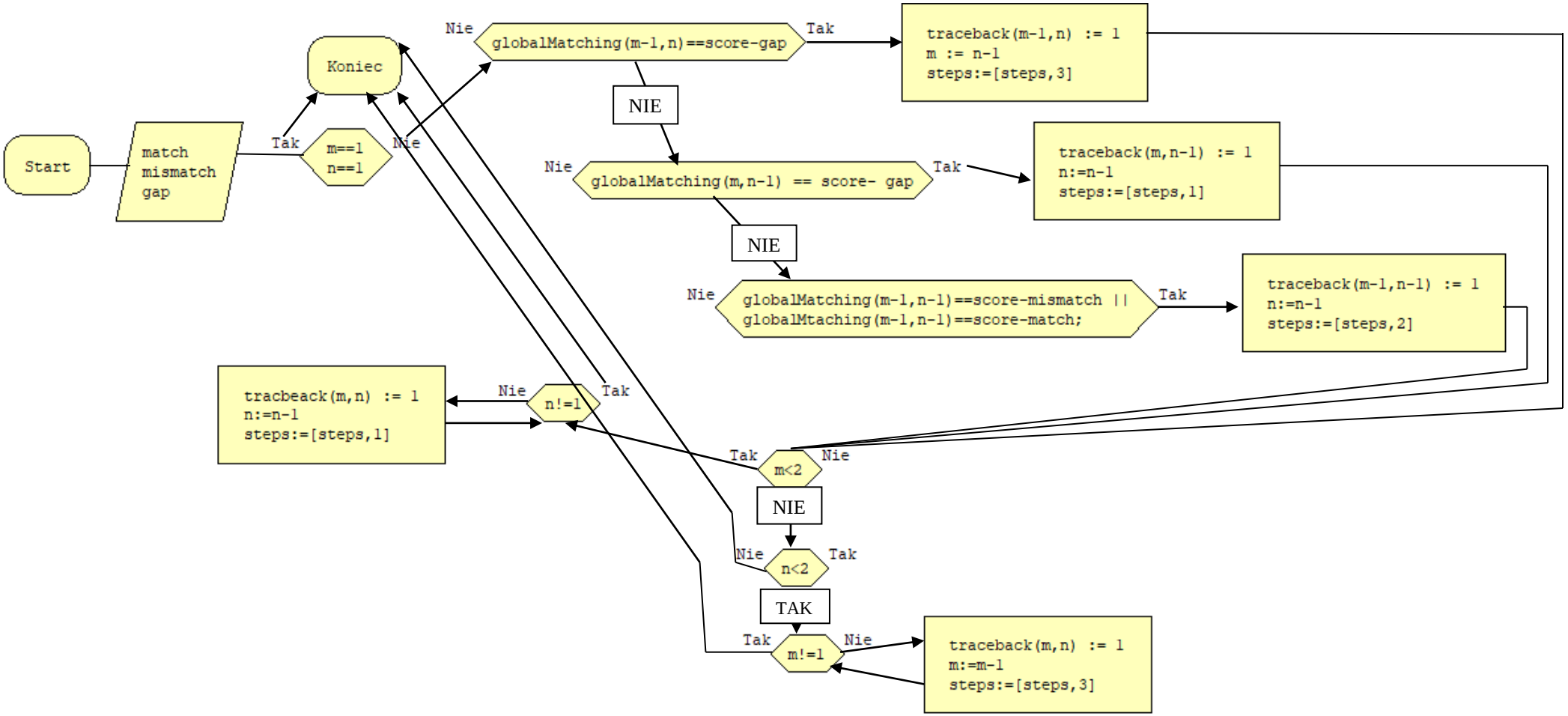




Nie udało mi się przedstawić wartości w kwadratach odpowiadające wartościom z scoredMatrix wygenerowanej przez funkcję globalMatching wraz z optymalną ścieżką. Wykorzystałam wbudowaną funkcję heatMap, gdyż chciałam graficznie przedstawić optymalną ścieżkę dopasowania.

```
#Sequence1: AGGTGGTTAA
#Sequence2: AGCTGGTAAA
#Mode: distance
#Score: 8
#Length: 12
#Match: 1
#Mismatch: -1
#Gap: 0
#Identity: 8/12 67%
#Gaps: 4/12 33%
AG_GTGGTTAA_
||  |||| ||
AGC_TGGT_AAA
```

Rysunek 1 Zawartość pliku wygenerowanego za pomocą test.m



Rysunek 2 Schemat blokowy algorytmu globalnego dopasowania Needlemana-Wunscha wykonany w programie Magiczne bloczki

1. Analiza złożoności pamięciowej

globalMatching.m oraz traceback.m zależą od długości wprowadzonych sekwencji
m,n – długości obu sekwencji
O(m*n)

2. Porównanie przykładowych par sekwencji ewolucyjnie

- Powiązanych

```
%porówananie sekwencji powiązanych ewolucyjnie:
%cytochrom b u nosorożca i słonia afrykańskeigo
globalMatching(getFromNCBI('X56285.1'),getFromNCBI('AJ245725.1'),1,-1,0)
```

Rysunek 3 Wywołanie test2.m

```
#Mode: distance
#Score: 974
#Length: 1452
#Match: 1
#Mismatch: -1
#Gap: 0
#Identity: 974/1452 67%
#Gaps: 504/1452 35%
```

Rysunek 4 Plik tekstowy zapisany w wyniku wywołania test2.m

- Niepowiązancych

```
%cytochrom b u słonia i psa
globalMatching(getFromNCBI('KC985187.1'),getFromNCBI('X56285.1'),1,-1,0)
```

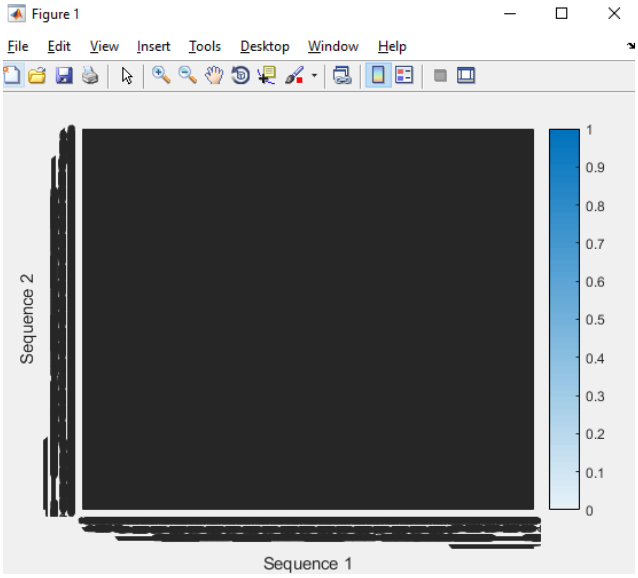
Rysunek 5 Wywołanie test3.m

```
#Mode: distance
#Score: 553
#Length: 1395
#Match: 1
#Mismatch: -1
#Gap: 0
#Identity: 553/1395 40%
#Gaps: 864/1395 62%
```

Rysunek 6 Plik tekstowy zapisany w wyniku wywołania test3.m

Dla sekwencji powiązancych ewolucyjnie wartość identity (tożsamości) jest wyższa 67% niż w przypadku sekwencji niepowiązanych ewolucyjnie 40%. Wartość niższa może świadczyć o małym pokrewieństwie lub jego braku.

Niestety dla długich sekwencji wprowadzanych z użyciem wbudowanej funkcji Matlaba heatMap jest niepotrzebne, gdyż nic na wykresie nie jest czytelne. Nie potrafiłam znaleźć rozwiązania, które potrafiłoby pokazać optymalną ścieżkę dla tak długich sekwencji.



A z funkcji Spy również nie wynika za wiele.