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:

1	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1

- 0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10
1	1	0	-1	-2	-3	-4	-5	-6	-7	-8
2	0	2	1	0	-1	-2	-3	-4	-5	-6
3	-1	1	1	0	1	0	-1	-2	-3	-4 -
4	-2	0	0	2	1	0	1	0	-1	-2
5	-3	-1	-1	1	3	2	1	0	-1	-2
- -6	-4	-2	-2	0	2	4	3	2	1	0 -
7	-5	-3	-3	-1	1	3	5	4	3	2 -
8	-6	-4	-4	-2	0	2	4	4	3	2 -
9	-7	-5	-5	-3	-1	1	3	5	5	4 -
10	-8	-6	-6	-4	-2	0	2	4	6	6 -

		, 0	•	, 0		, -	Ŭ		-	Ŭ	
1	0	0	0	0	0	0	0	0	0	0	
0	1	0	0	0	0	0	0	0	0	0	
0	0	1	0	0	0	0	0	0	0	0	
0	0	0	1	0	0	0	0	0	0	0	
0	0	0	0	1	0	0	0	0	0	0	
0	0	0	0	0	1	0	0	0	0	0	
0	0	0	0	0	0	1	0	0	0	0	
0	0	0	0	0	0	0	1	0	0	0	
0	0	0	0	0	0	0	0	1	0	0	
0	0	0	0	0	0	0	0	0	1	0	
											4

0

0

0

0

0

0

0

0

0

#Sequencel: AGGTGGTTAA #Sequence2: AGCTGGTAAA

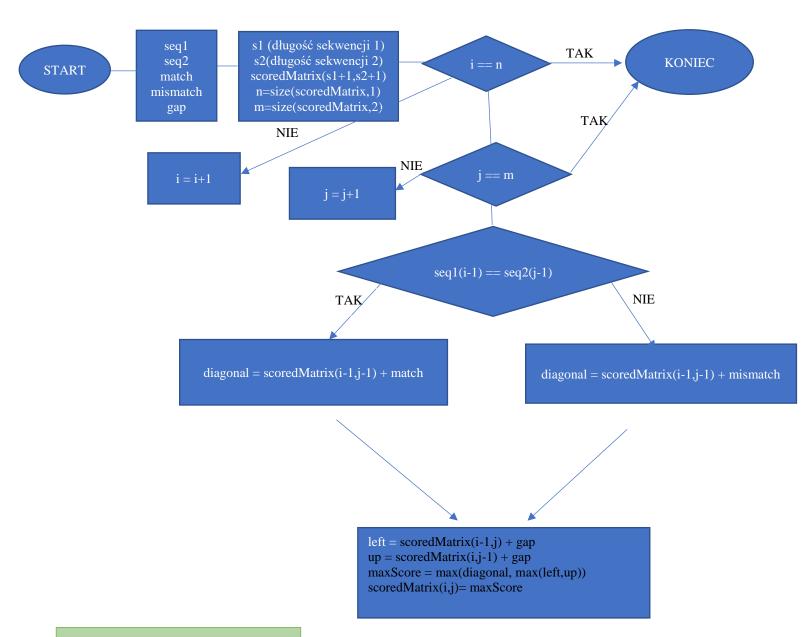
#Mode: distance

#Score: 6 #Length: 10 #Match: 1 #Mismatch: -1 #Gap: -1

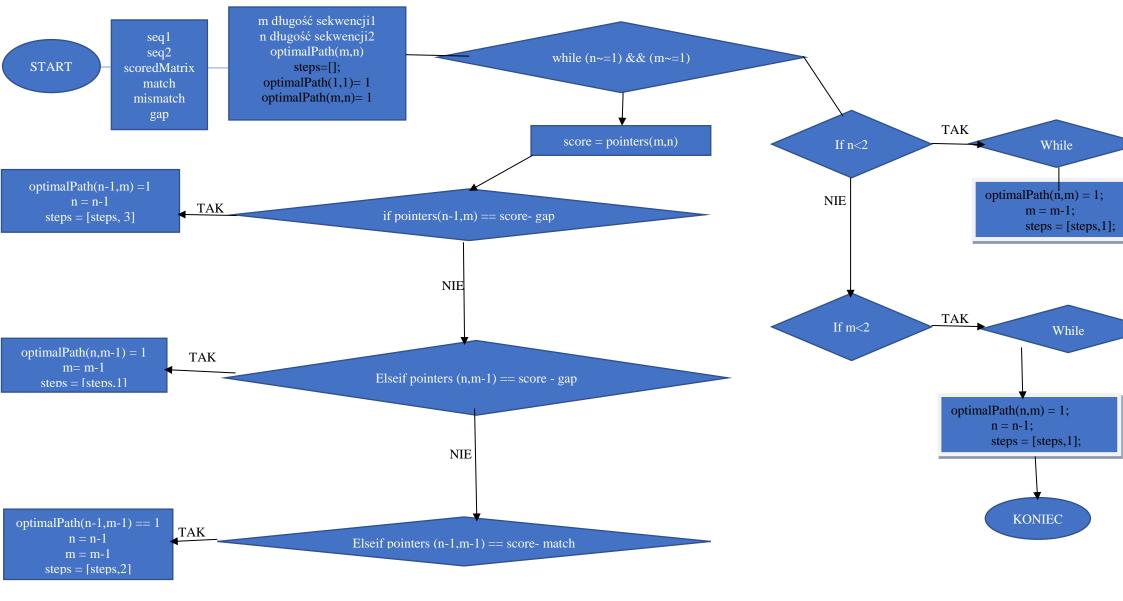
#Identity: 8/10 80%

#Gaps: 0/10 0%
AGGTGGTTAA
|| ||| ||
AGCTGGTAAA

Rysunek 1 Zawartość pliku wygenerowanego za pomocą test.m



Schemat blokowy globalMatching.m



Schemat blokowy traceback.m

- 1. Analiza obliczeniowa:
- globalMatching.m
 - czasowa O(m) ,przy założeniu, że m~n
 - > pamięciowa O(m*n), m~n O(m²)
- traceback.m
 - \triangleright czasowa $O(m^*n)$, $m \sim n O(m^2)$
 - > pamięciowa O(m*n), m~n O(m²)

Sumarycznie:

- o pamięciowa $O(m^2) + O(m^2) = 2 O(m^2) \sim O(m^2)$
- o czasowa $O(m^2)+O(m) \sim O(m^2)$
- 2. Porównanie przykładowych par sekwencji ewolucyjnie :
- Powiązanych

```
%insulina wariant 1 u krowuy i insulina wariant 2 krowy globalMatching(getFromNCBI('NM_173926.2'),getFromNCBI('NM_001185126.1'),1,-1,-1)
```

Rysunek 2 Wywołanie test2.m

#Mode: distance #Score: 461 #Length: 516 #Match: 1 #Mismatch: -1 #Gap: -1 #Identity: 497/516 96% #Gaps: 31/516 6%

Rysunek 3 Plik tekstowy zapisany w wyniku wywołania test2.m

• Niepowiązancych

Rysunek 5 Plik tekstowy zapisany w wyniku wywołania test3.m

Dla obu wariantów insuliny u krowy (bos taurus) widać duże podobieństwo wynoszące 96%. Inny wariant insuliny w obrębie jednego gatunku wykazuje wysoką zgodność. Natomiast insulina wobec 2 róznych gatunków wykazuje znacznie niższą zgodność. Dla insuliny u człowieka oraz krowy podobieństwo jest mniejsze, wynoszące 76%.