

Wstęp do bioinformatyki

Nr ćwiczenia: 3

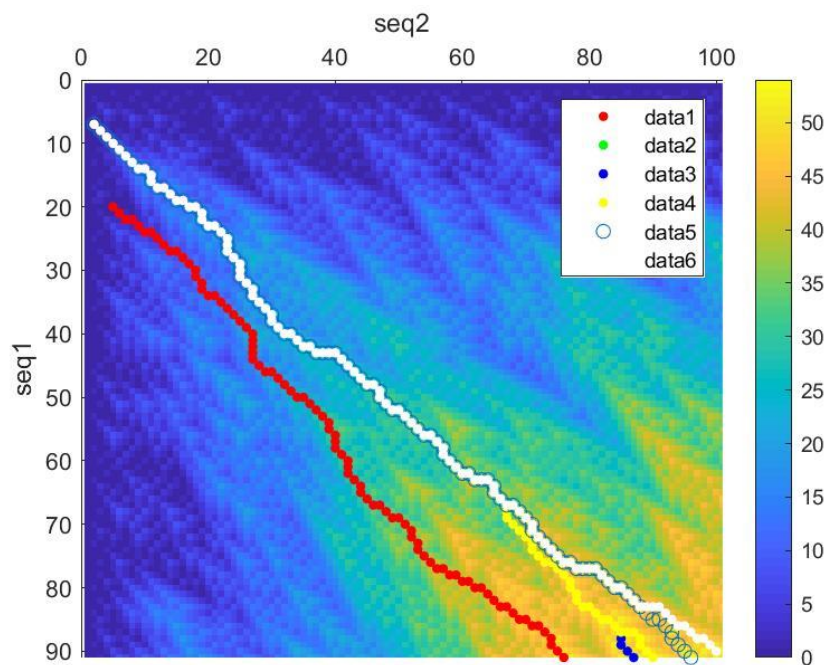
Temat ćwiczenia: Dopasowanie lokalne par sekwencji

Nazwisko i Imię prowadzącego kurs: dr inż. Witold Dyrka

Wykonawcy:	
Imię i Nazwisko Nr indeksu, wydział	Edyta Krukowska 217097, WPPT
Termin zajęć: dzień tygodnia, godzina	Piątek 11.15
Data oddania sprawozdania	15.02.2019

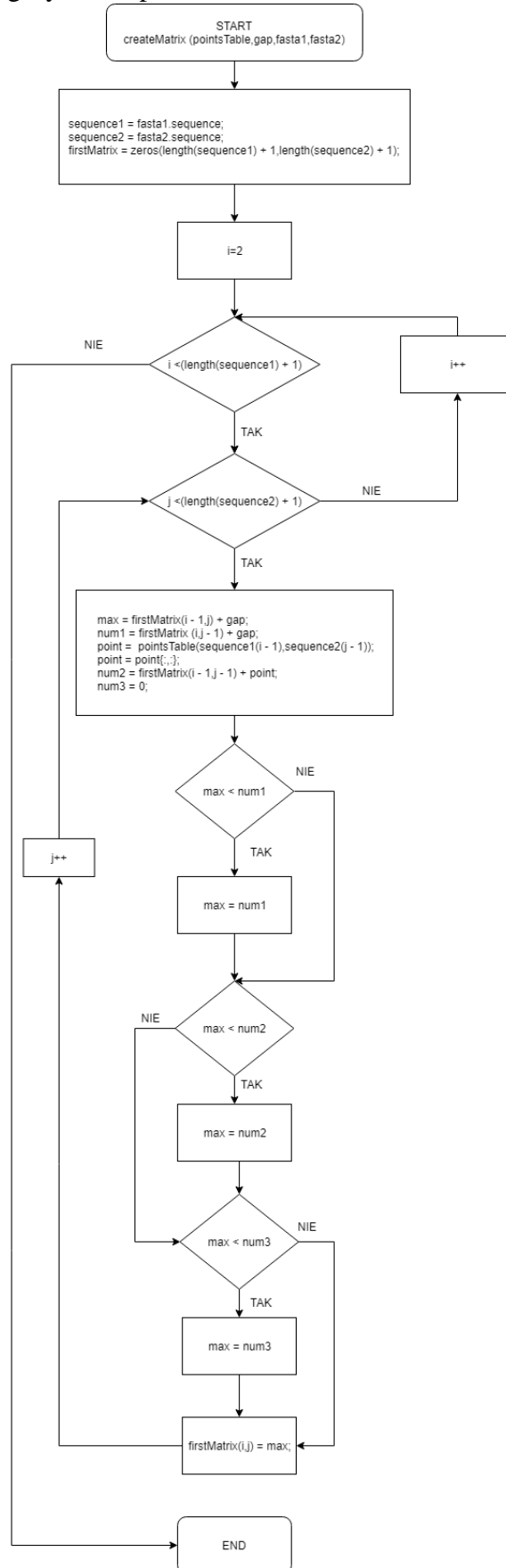
Repozytorium: <https://github.com/Edie1995/Bioinformatyka/tree/zadanie3>

1. Prezentacja działania programu:

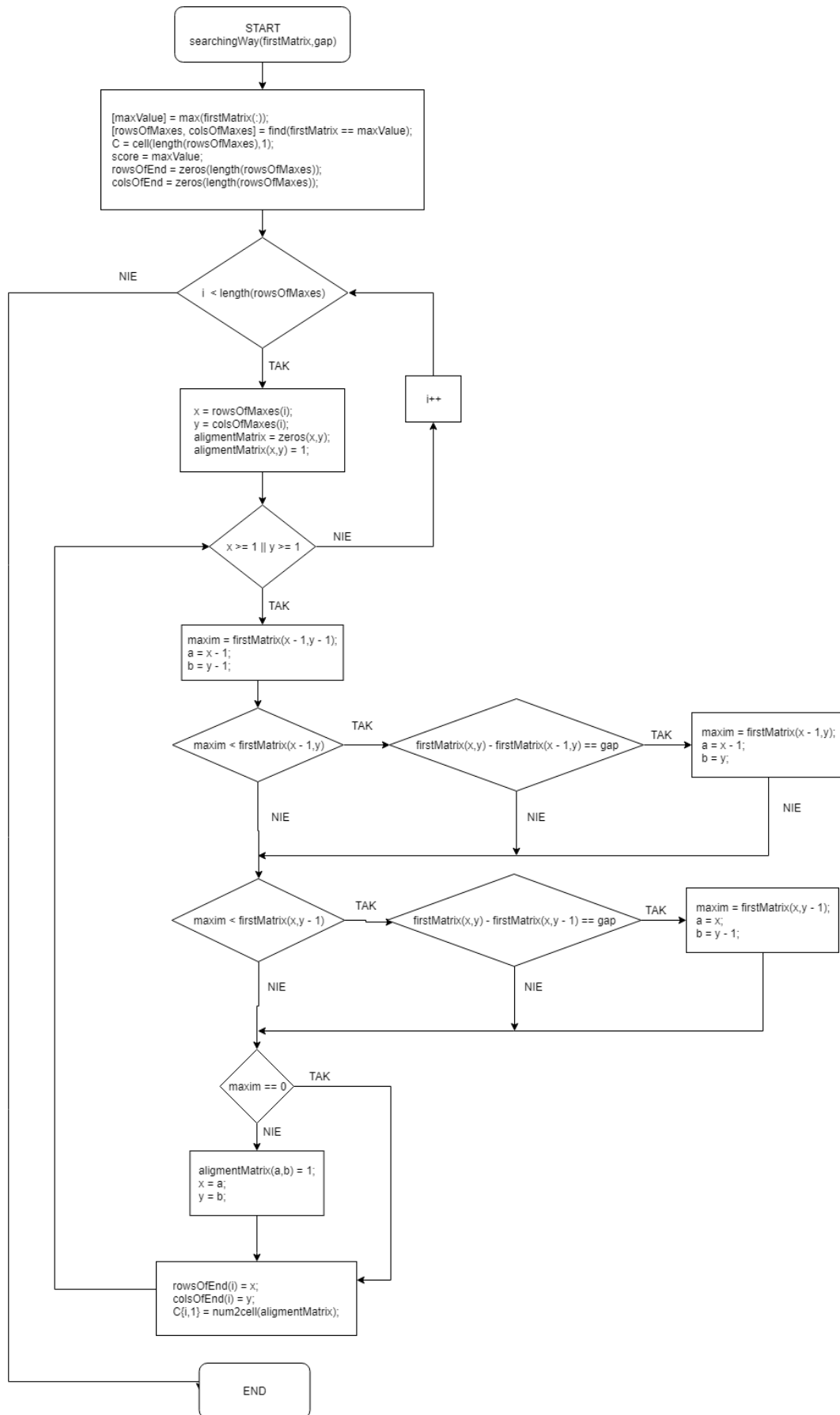


Rysunek 1 Przykład działania programu, dla przykładowych sekwencji wpisanych z klawiatury

2. Schematy blokowe algorytmu dopasowania



Rysunek 2 Schemat blokowy -tworzenie macierzy kosztu



Rysunek 3 Schemat blokowy -tworzenie lokalnej najoptymalniejszej drogi

3. Oszacowanie złożoności czasowej obliczeniowej i pamięciowej kodu poszczególnych funkcji i całego programu:

➤ Czasowa:

createMatrix:

3 operacje if + 4 przypisania wartości * $(m)(n) = O(mn)$;

searchingWay:

Przyjmując pesymistyczny wariant szukania maksimumów = $(m)*(n)$

Funkcja while w pesymistycznej wersji obejmie $(m)*(n)*(3 \text{ porównania i } 6 \text{ przypisań wartości})$

Podsumowując, w negatywnym wariancie cała funkcja ($c > 0$) $c*(m)(n)$ więc jest ona maksymalnie rzędu $O(mn)$

saveToFile:

W pesymistycznym założeniu, lecz raczej nie możliwym do osiągnięcia, że ścieżka będzie szła wzdłuż krawędzi macierzy jest to funkcja rzędu $O(m+n)$

writeSequence:

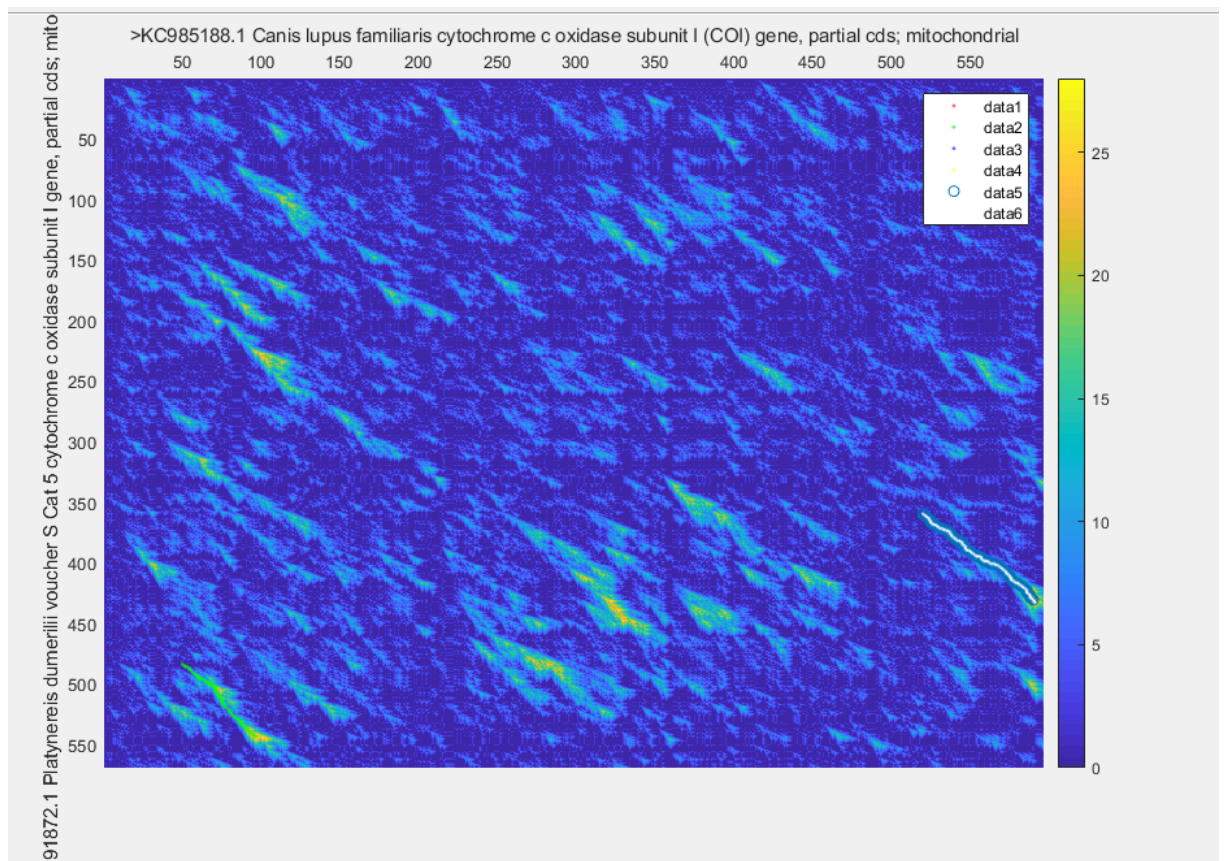
W funkcji while według założenia pesymistycznego otrzymamy $m*n*(6 \text{ porównań (maksymalnie) i } 2 \text{ przypisania})$. Jest to największy while w tej funkcji, pozostałe są rzędu $o(mn)$, czyli na pewno niższego niż $O(mn)$. Z takiego oszacowania wynika, że maksymalny tej funkcji to $O(mn)$.

Pozostałe funkcje wykorzystywane przez program są rzędu stałego $O(1)$ lub liniowego $O(n)$ lub $O(m)$. Na tej podstawie można oszacować, że dla całego programu jest maksymalnie rzędu $O(mn)$.

➤ Pamięciowa

Największym obiektem stworzonym w całym programie jest macierz kosztu i jest ona rzędu $O(mn)$, wszystkie pozostałe obiekty wykorzystywane w programie są rzędu takiego samego, lub niższego $o(mn)$. Możemy więc uznać, że jest to złożoność pamięciowa $c*m*n$ (gdzie $c > 0$), czyli $O(mn)$.

- **Porównanie cytochromu c kota z psem**



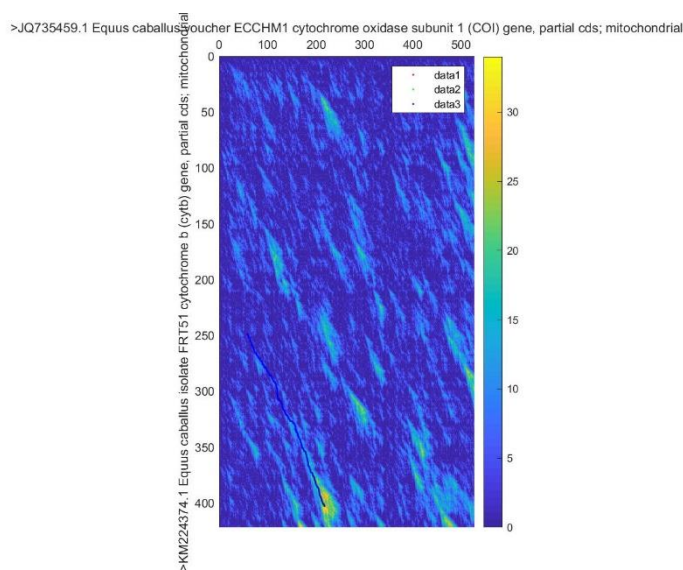
Rysunek 6 Macierz kosztu z naniesionymi ścieżkami lokalnego dopasowania porównująca kota (oś y) z psem (oś x)

```
#1: KC591872.1  
#2: KC985188.1  
#Mode: distance  
#Table:  
# A C G T  
A 2-7-5-7  
C-7 2-7-5  
G-5-7 2-7  
T-7-5-7 2  
#Gap: -2  
#Score: 28  
#Length: 447  
#Identity: 122/447(27%)  
#Gaps: 30/447(7%)  
CCGGCTCACTACTGGGAAGGGATCAACTATATAATACAATCGTTACAGCCCACGCCTTTTTTAATAATCTTTTTCTCTGTTATACCTGTAAT  
||| | | | | ||| | | | | | | | || | | | |  
CCGAAGAATCAGAATAGGTGTTGATATAGGATAGGGTCTCCTCCTCCAGCGGGATCAAAAAATGTTGTATTAAAGATTCCGGTCTGTTAAAA
```

Rysunek 7 Konsolowy wynik działania programu dla Kota i Psa

4.2. Ewolucyjnie niepowiązanych

- Porównanie cytochromu c i b konia



Rysunek 8 Macierz kosztu z naniesionymi ścieżkami optymalnego dopasowania wygenerowana dla cytochromu b (oś y) oraz cytochromu c (oś x) konia

```
#1: KM224374.1
#2: JQ735459.1
#Mode: distance
#Table:
# A C G T
A 2-7-5-7
C-7 2-7-5
G-5-7 2-7
T-7-5-7 2
#Gap: -2
#Score: 34
#Length: 246
#Identity: 62/246 (25%)
#Gaps: 74/246 (30%)
AAAGCCACCCTTACCCGATTTTGGCTTCCACTTCATCCTACCCCTCATCACAAGCC-CTGGTAG-TCGTACA--TT-TA-C-TA--TTT-CTT--CACG-AA-A--CAGGA-T-CTAATAACCCCTCAI
TAGGCCAACCTGGGACCCCTACTAGGAGATGATCAGATCGACAATGTCATTGTAACCGCCCAT-GC-ATTCTGT-AATAATTTTCGTGATGG-TCATACCCATGATAATCGGAGGATTCGGA-A-A--C--T-I
```

Rysunek 9 Konsolowy wynik działania programu dla cytochromu b (oś y) oraz cytochromu c (oś x) konia