

# Wstęp do bioinformatyki

## Nr ćwiczenia: 3

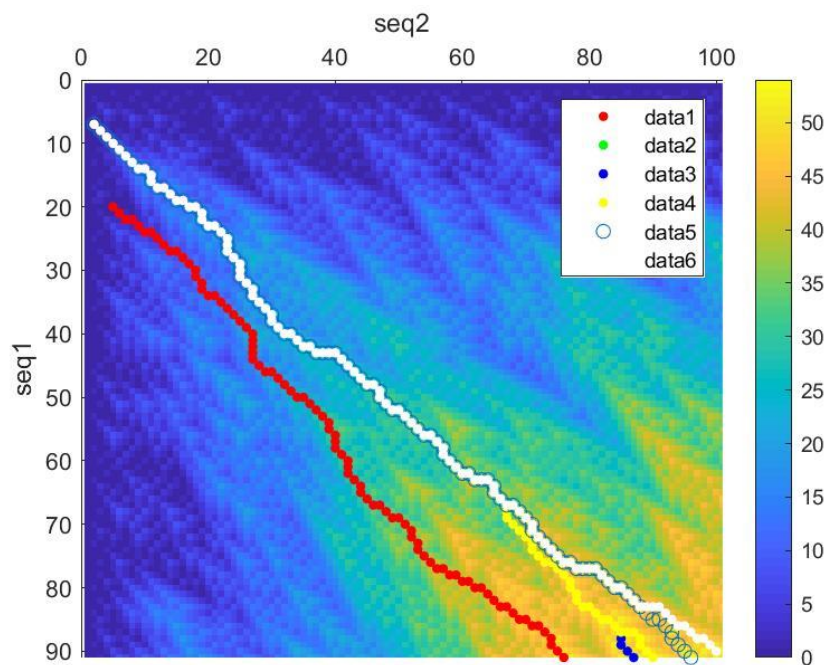
### Temat ćwiczenia: Dopasowanie lokalne par sekwencji

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

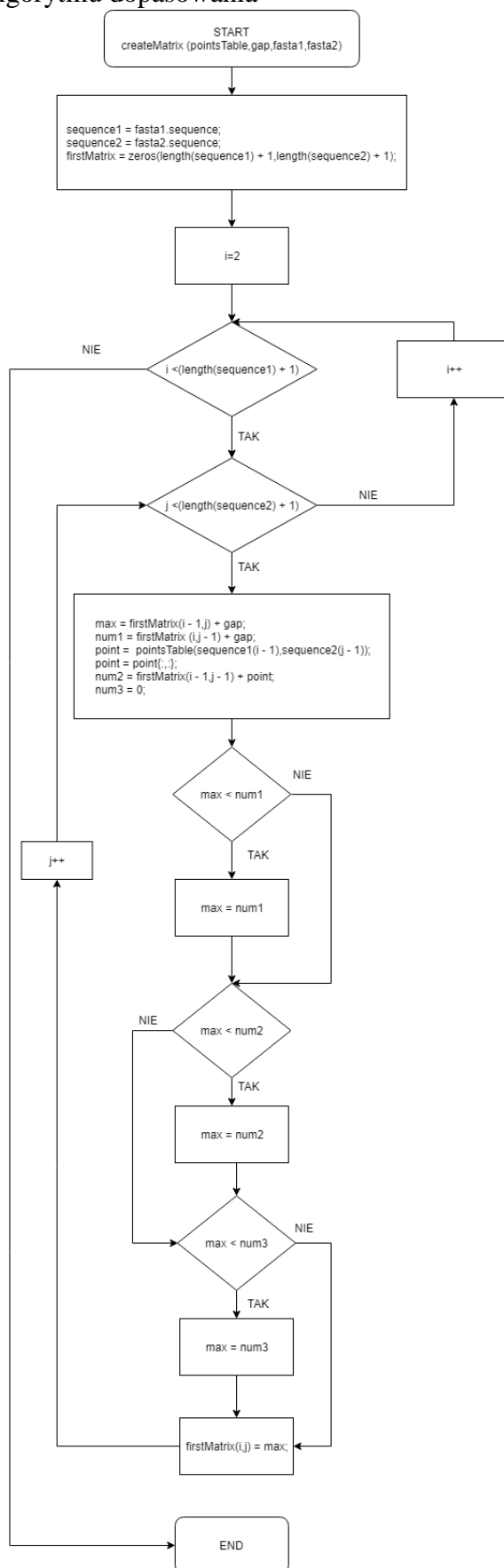
Repozytorium: <https://github.com/Edie1995/Bioinformatyka/tree/zadanie3>

#### 1. Prezentacja działania programu:

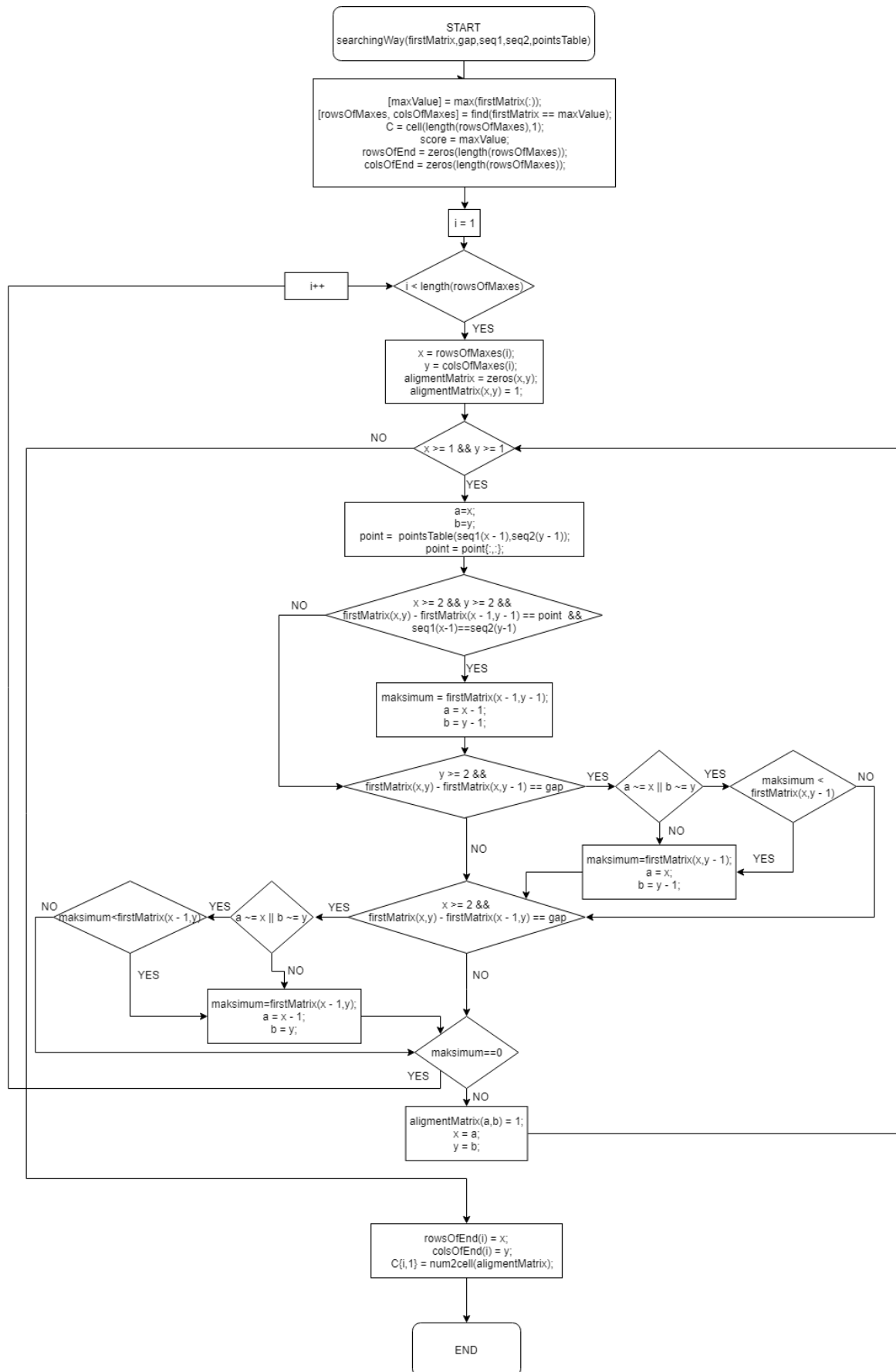


Rysunek 1 Przykład działania programu, dla przykładowych sekwencji wpisanych z klawiatury

## 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 maksimum  $= (m)*(n)*(n)$

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  $O(mnn)$

##### **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  $O(mn)$ .

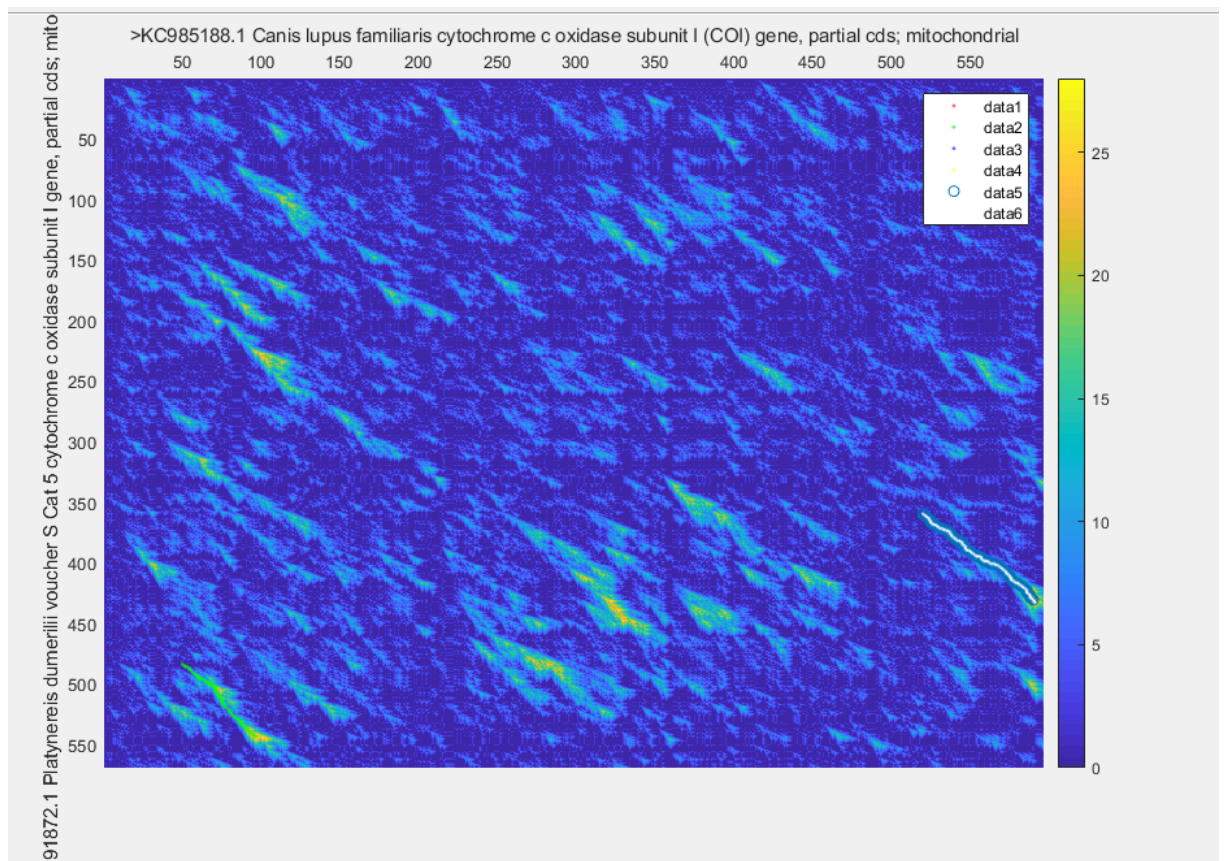
Pozostałe funkcje wykorzystywane przez program są rzędu stałego  $O(1)$  lub liniowego  $O(n)$  lub  $O(m)$ . Na tej podstawie można oszacować, że dla całego programu jest maksymalnie rzędu  $O(mnn)$ .

#### ➤ Pamięciowa

Największym obiektem stworzonym w całym programie jest macierz kosztu i jest ona rzędu  $O(mnn)$ , 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)

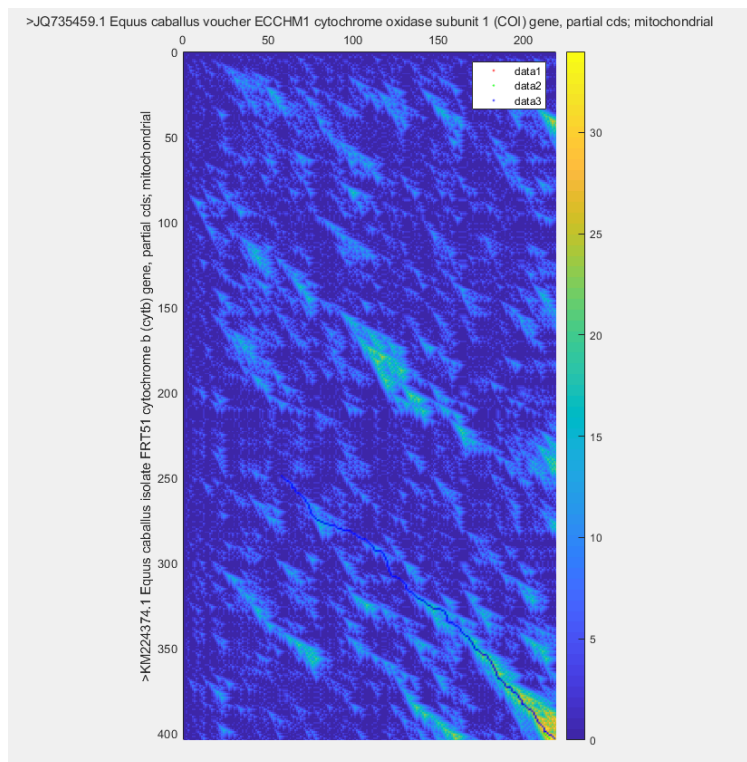
[illegible]

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

[illegible]

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

## 5. Wnioski

Na podstawie wyników porównania 3 różnych sekwencji i ich dopasowań, możemy zauważyć, że najlepsze dopasowanie uzyskaliśmy dla kota i psa (29%), bardzo zbliżone podobieństwo sekwencji mogliśmy zaobserwować dla kota i konia (27%). Jednak na przykładowej pojedynczej ścieżce dopasowania lokalnego, przedstawionej poniżej wykresów, widzimy, że dopasowania te mają raczej charakter losowy, ponieważ dopasowane są pojedyncze nukleotydy, nie całe łańcuchy. Pozwala nam to jednak określić, które fragmenty całego łańcucha nukleotydowego są sobie podobne. I tak w przypadku porównania kota z koniem, mamy jedynie jedną ścieżkę lokalnego dopasowania, a już w przypadku kota z psem

jest tych ścieżek aż 6. Pozwala nam to stwierdzić, że większe podobieństwo danego genu ma kot z psem.

Po sprawdzeniu dopasowania cytochromu c i b konia, obserwujemy aż 23% podobieństwa. Jednak tak jak w poprzednich przypadkach podobieństwo to ma raczej charakter losowy, ze względu na znaczne ilości przerw w podobieństwach nukleotydów i pojedyncze dopasowania.