

Bioinformatyka Sprawozdanie 3

Dopasowanie lokalne par sekwencji

Michał Marciniak 244811

1 Analiza złożoności obliczeniowej czasowej i pamięciowej:

Algorithm 1 Dopasowanie lokalne z liniowym kosztem kary za przerwy

```
1: procedure INICJALIZACJA
2:    $m \leftarrow$  długość sekwencji  $x$ 
3:    $n \leftarrow$  długość sekwencji  $y$ 
4:    $ins \leftarrow$  koszt insercji
5:    $del \leftarrow$  koszt delecji
6:    $sub \leftarrow$  koszt substytucji
7:    $R \leftarrow$  macierz o wymiarach  $n \times m$ 
8:    $biggest\ val \leftarrow 0$ 
9:    $biggest\ val\ pos \leftarrow$  pusta tablica
10:  for  $i \leftarrow 1$  to  $n$  do
11:    for  $j \leftarrow 1$  to  $m$  do
12:      
$$R[i, j] = \max \begin{cases} R[i-1, j] + del \\ R[i, j-1] + ins \\ R[i-1, j-1] + sub \times (x[i] \neq y[j]) \\ 0 \end{cases}$$

13:      if  $R[i, j] > biggest\ val$  then
14:         $biggest\ val = R[i, j]$ 
15:         $biggest\ val\ pos = (i, j)$ 
16: procedure ODTWARZANIE ŚCIEŻKI
17:    $i \leftarrow biggest\ val\ pos[0]$ 
18:    $j \leftarrow biggest\ val\ pos[1]$ 
19:    $aln1 \leftarrow ""$ 
20:    $aln2 \leftarrow ""$ 
21:   while  $R[i, j] > 0$  do
22:     if  $i > 0$  and  $j > 0$  and  $R[i, j] == \max \begin{cases} R[i-1, j-1] + sub \times (x[i] \neq y[j]) \\ 0 \end{cases}$  then
23:        $aln1 = x[i] + aln1, aln2 = y[j] + aln2, i \leftarrow i-1, j \leftarrow j-1$ 
24:     else
25:       if  $i > 0$  and  $R[i, j] == \max \begin{cases} R[i-1, j] + ins \\ 0 \end{cases}$  then
26:          $aln1 = "-" + aln1, aln2 = y[i-1] + aln2, i \leftarrow i-1$ 
27:       else
28:          $aln1 = x[j-1] + aln1, aln2 = "-" + aln2, j \leftarrow j-1$ 
```

Algorithm 2 Dopasowanie lokalne z affine cost penalty za przerwy

```
1: procedure INICJALIZACJA
2:    $m \leftarrow$  długość sekwencji  $x$ 
3:    $n \leftarrow$  długość sekwencji  $y$ 
4:    $\alpha \leftarrow$  koszt rozpoczęcia przerwy
5:    $\beta \leftarrow$  koszt kontynuowania przerwy
6:    $sub \leftarrow$  koszt substytucji
7:    $R \leftarrow$  macierz o wymiarach  $n \times m$ 
8:    $P \leftarrow$  macierz o wymiarach  $n \times m$ 
9:    $Q \leftarrow$  macierz o wymiarach  $n \times m$ 
10:   $biggest\ val \leftarrow 0$ 
11:   $biggest\ val\ pos \leftarrow$  pusta tablica
12:  for  $i \leftarrow 1$  to  $m$  do
13:     $P[0, i] = -\inf$ 
14:  for  $i \leftarrow 1$  to  $n$  do
15:     $Q[i, 0] = -\inf$ 
16:  for  $i \leftarrow 1$  to  $n$  do
17:    for  $j \leftarrow 1$  to  $m$  do
18:       $P[i, j] = \max \begin{cases} R[i-1, j] + \alpha + \beta \\ P[i-1, j] + \beta \end{cases}$ 
19:       $Q[i, j] = \max \begin{cases} R[i, j-1] + \alpha + \beta \\ Q[i, j-1] + \beta \end{cases}$ 
20:       $R[i, j] = \max \begin{cases} P[i, j] \\ Q[i, j] \\ R[i-1, j-1] + sub \times (x[i] \neq y[j]) \\ 0 \end{cases}$ 
21:      if  $R[i, j] > biggest\ val$  then
22:         $biggest\ val = R[i, j]$ 
23:         $biggest\ val\ pos = (i, j)$ 
24: procedure ODTWARZANIE ŚCIEŻKI
25:    $i \leftarrow biggest\ val\ pos[0]$ 
26:    $j \leftarrow biggest\ val\ pos[1]$ 
27:    $aln1 \leftarrow ""$ 
28:    $aln2 \leftarrow ""$ 
29:   while  $R[i, j] > 0$  do
30:     if  $i > 0$  and  $j > 0$  and  $R[i, j] == \max \begin{cases} R[i-1, j-1] + sub \times (x[i] \neq y[j]) \\ 0 \end{cases}$  then
31:        $aln1 = x[i] + aln1, aln2 = y[j] + aln2, i \leftarrow i-1, j \leftarrow j-1$ 
32:     else
33:       if  $i > 0$  and  $R[i, j] == \max \begin{cases} R[i-1, j] + ins \\ 0 \end{cases}$  then
34:          $aln1 = "-" + aln1, aln2 = y[i-1] + aln2, i \leftarrow i-1$ 
35:       else
36:          $aln1 = x[j-1] + aln1, aln2 = "-" + aln2, j \leftarrow j-1$ 
```

Złożoność obliczeniowa w obu przypadkach wynosi : $O(n^2)$

Iteracja po wszystkich komórkach macierzy o rozmiarach $n \times m$ i wypełnienie jest wartościami kosztu

Złożoność pamięciowa: $O(n^2)$ ←macierz o rozmiarach $n \times m$

Należy jednak pamiętać, że w przypadku drugiego algorytmu mamy aż 3 macierze zamiast jednej.

2 Kod do repozytorium

<https://gitlab.com/MMarciniak103/bioinformatics>

3 Opis programu

Po wczytaniu sekwencji (dostępne 3 sposoby wczytywania) użytkownik może kliknąć przycisk local alignment, co w przypadku poprawnego wczytania sekwencji otworzy drugie okno Fig.1.

	A	C	T	G	U	-
A	1	-1	-1	-1	-1	-3
C	-1	1	-1	-1	-1	-3
T	-1	-1	1	-1	-1	-3
G	-1	-1	-1	1	-1	-3
U	-1	-1	-1	-1	1	-3
-	-3	-3	-3	-3	-3	-3

☐ affine penalty

ALIGNMENT

SAVE

Figure 1: Okno dopasowania lokalnego

Tam użytkownik może wybrać czy chce skorzystać z domyślnej macierzy kosztów, czy utworzyć własną (warto nadmienić, że program dba o zachowanie symetrii macierzy poprzez automatyczne uzupełnianie symetrycznego elementu macierzy podczas zmiany któregośkolwiek z wejść). Dodatkowo może wybrać czy chce wykorzystać tzw. affine gaps penalty. Jeżeli się na to zdecyduje to będzie musiał wybrać koszt za kontynuowanie przerw.

	A	C	T	G	U	-
A	1	-1	-1	-1	-1	-3
C	-1	1	-1	-1	-1	-3
T	-1	-1	1	-1	-1	-3
G	-1	-1	-1	1	-1	-3
U	-1	-1	-1	-1	1	-3
-	-3	-3	-3	-3	-3	-3

Enlargement cost ☒ affine penalty

ALIGNMENT

SAVE

Figure 2: Okno dopasowania lokalnego - wybór kosztu za kontynuowanie przerwy

Program informuje użytkownika w przypadku popełnienia jakiegoś błędu. Poniżej zaprezentowane są przykłady takich komunikatów. Fig. 3 prezentuje komunikat, który wyświetla się gdy użytkownik zostawił puste miejsce w macierzy kosztów. Natomiast Fig. 4 pokazuje komunikat wyrzucony w przypadku podania nieprawidłowej wartości kosztu kary za rozszerzanie przerw.

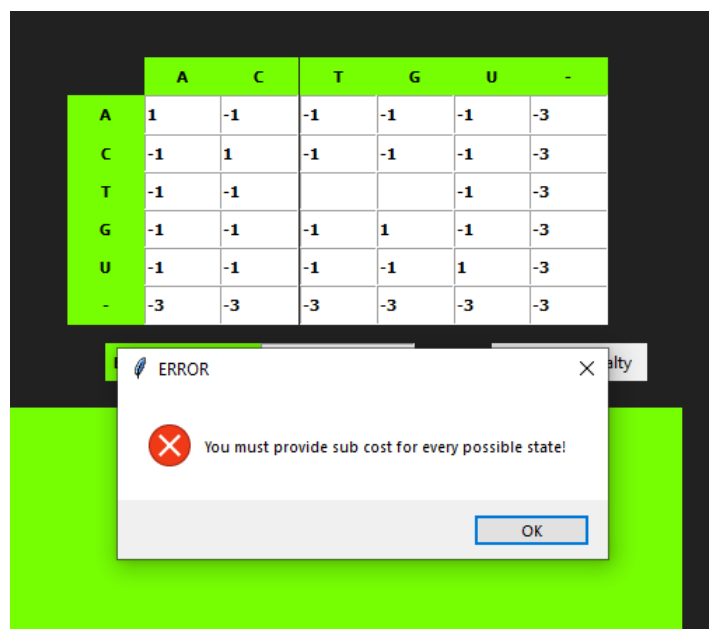


Figure 3: Komunikat błędu - należy podać wartości kosztów dla każdego z możliwych stanów

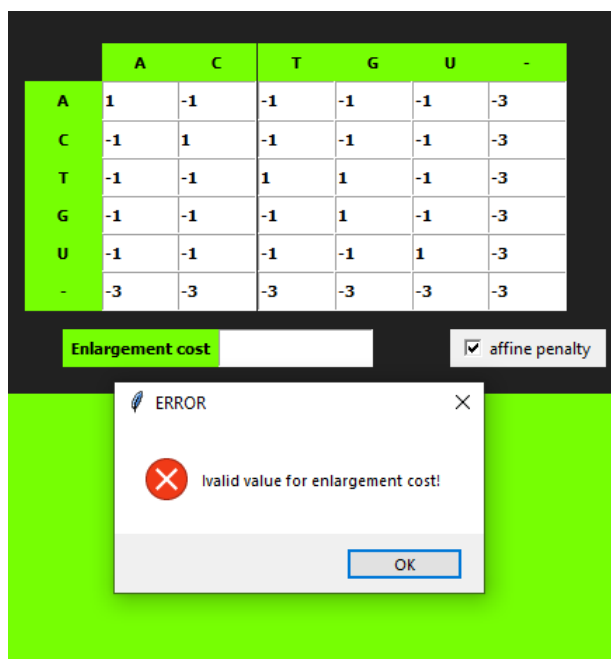


Figure 4: Komunikat błędu - należy podać poprawną wartość kosztu

Gdy program zakończy szukanie dopasowania lokalnego wyświetli informację o uzyskanych wynikach w tabelce poniżej macierzy substytucji. Przykład zaprezentowano na Fig. 6. Dodatkowo użytkownik ma możliwość zapisania znalezionych dopasowań w formacie zaprezentowanym na Fig. 5.

4 Porównanie sekwencji

4.1 Ewolucyjnie powiązanych

seq1: Pan troglodytes partial mRNA for putative mitochondrial transcription factor A (tfam gene), isoform A

seq2: Pan paniscus partial mRNA for putative mitochondrial transcription factor A (tfam gene), isoform A

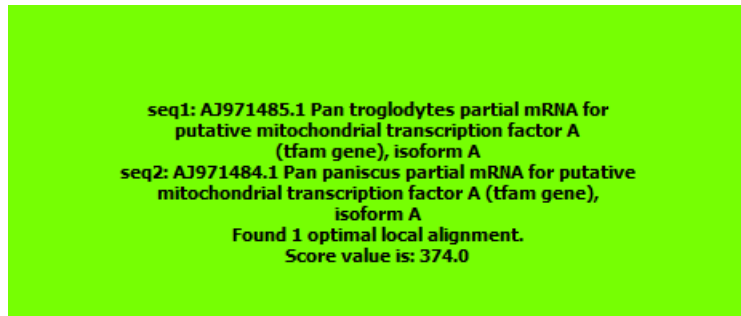


Figure 7: Wynik dopasowania

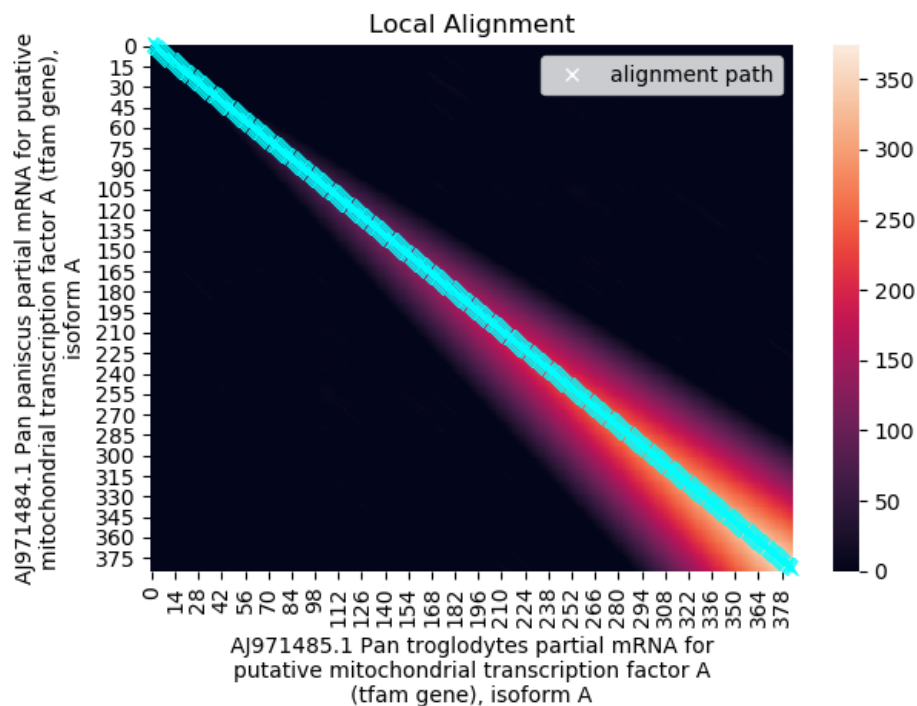


Figure 8: Macierz kosztu z zaznaczoną ścieżką dopasowania

Sekwencje dotyczą mRNA mitochondrialnego czynnika transkrypcyjnego A szympansa oraz szympansa karłowatego. Jak widać na Fig. 7 dopasowanie dla domyślnej macierzy substytucji osiąga bardzo duży wynik 374 w stosunku do długości sekwencji, wynoszącej 384. Również patrząc na wykres macierzy kosztu z zaznaczoną ścieżką dopasowania Fig. 8 widać, że przebiega ona po całej przekątnej, co oznacza, że sekwencje są praktycznie identyczne (z małymi różnicami).

seq1: Pan troglodytes partial mRNA for putative mitochondrial transcription factor A (tfam gene), isoform A
 ser2: Homo sapiens TFAM mRNA for mitochondrial transcription factor A, partial cds, clone: FLJ08040AAAF

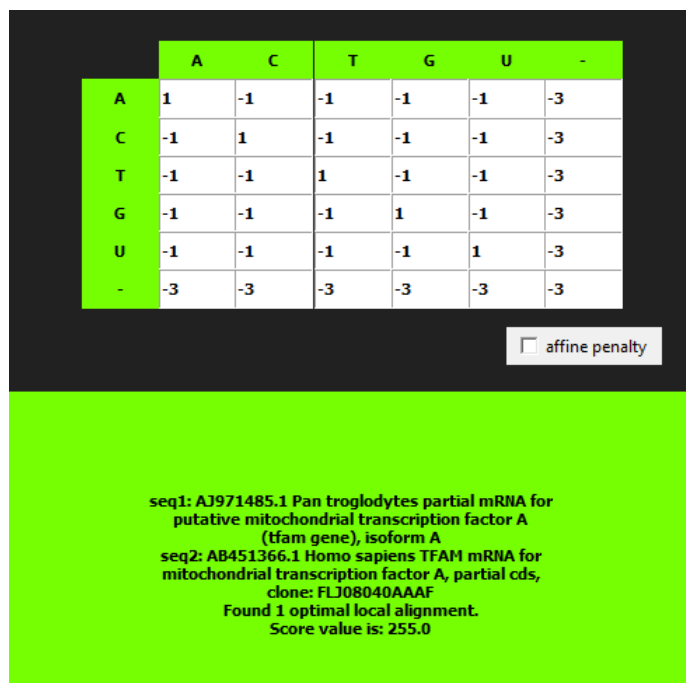


Figure 9: Wynik dopasowania

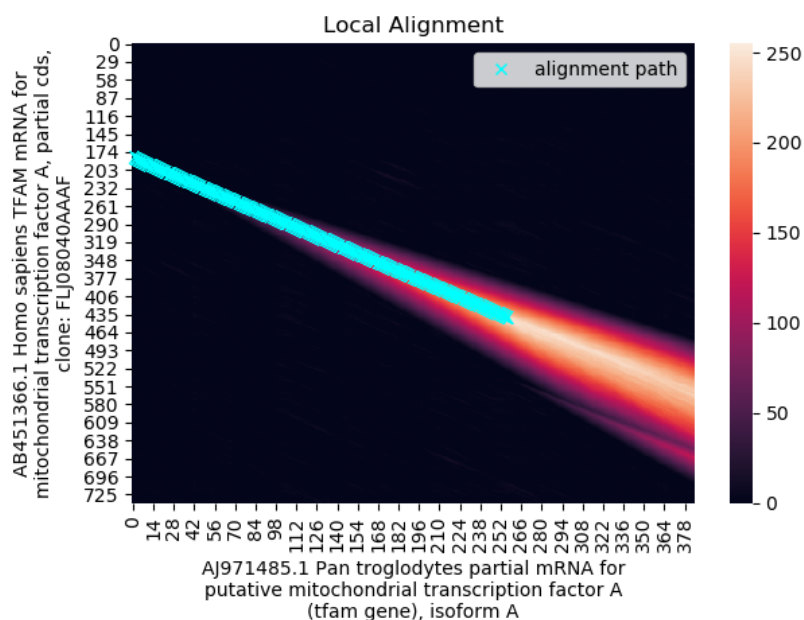


Figure 10: Macierz kosztu z zaznaczoną ścieżką dopasowania

W tym przypadku również wykorzystano sekwencję szympansa, ale porównano ją z sekwencją człowieka. Jak widać dla domyślnej macierzy substytucji uzyskany wynik dopasowania jest mniejszy niż w poprzednim przypadku. Jednakże mimo to widoczny jest długi fragment wykazujący kawałek, który został utrwalony w obu gatunkach. Po zastosowaniu affine cost penalty uzyskujemy całkowicie inny wynik Fig.11,12. Wynik dopasowania zmienił się z wartości 255 na 279.



Figure 11: Wynik dopasowania

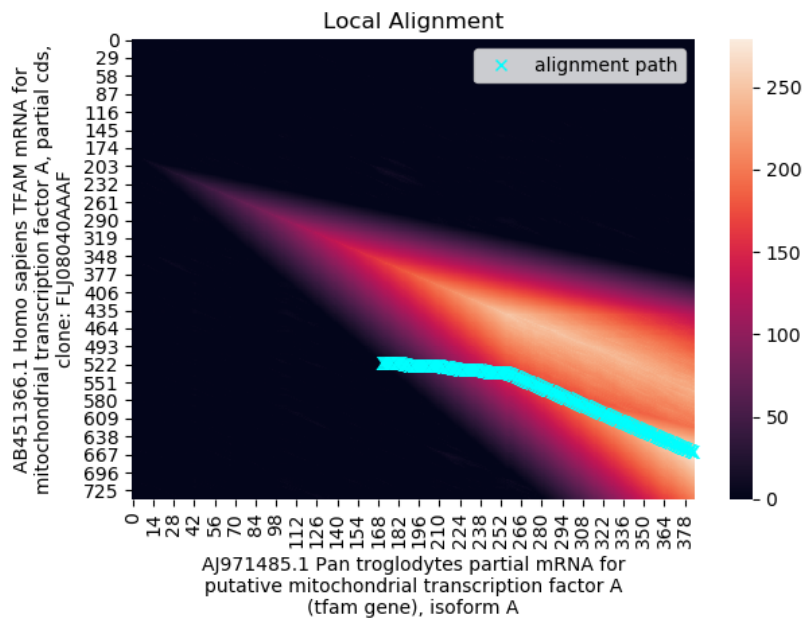


Figure 12: Macierz kosztu z zaznaczoną ścieżką dopasowania

Po zastosowaniu surowszych kar za przerwy i substytucje (Macierz zaprezentowana na Fig.13) uzyskano jeszcze inny wynik. Tym razem wygląda on podobnie do wyniku uzyskanego dla pierwszego porównania tych sekwencji Fig. 10.

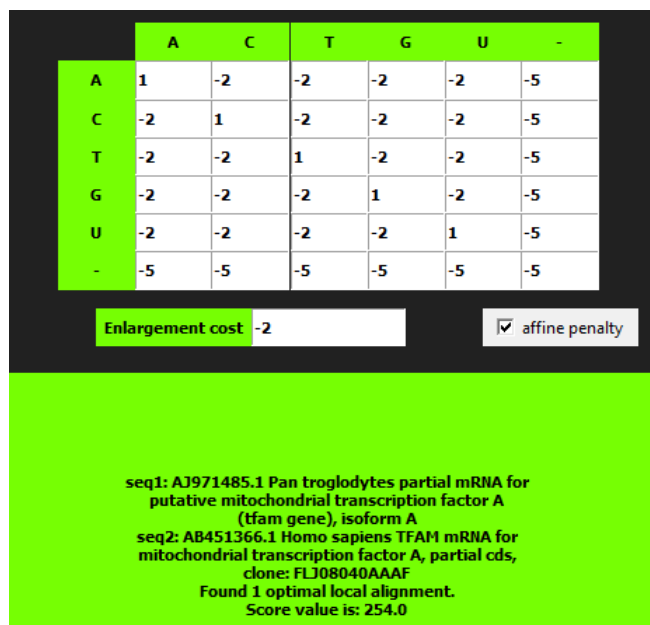


Figure 13: Wynik dopasowania

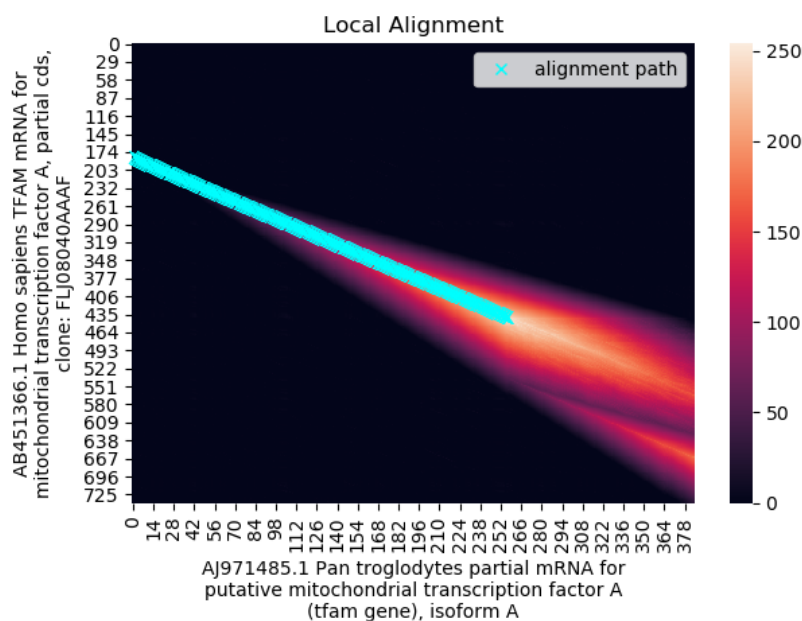


Figure 14: Macierz kosztu z zaznaczoną ścieżką dopasowania

Przy zastosowaniu jeszcze innej kombinacji kosztów uzyskano kolejny ciekawy wynik. Świadczy to o tym, że wybór punktacji ma bardzo duży wpływ na otrzymane wyniki i przy przeprowadzaniu dopasowania powinno się poświęcić sporo uwagi na jego odpowiedni wybór (tj. taki który będzie

uzasadniony z biologicznego punktu widzenia). Jednocześnie należy mieć to również na uwadze podczas analizy otrzymanych dopasowań.

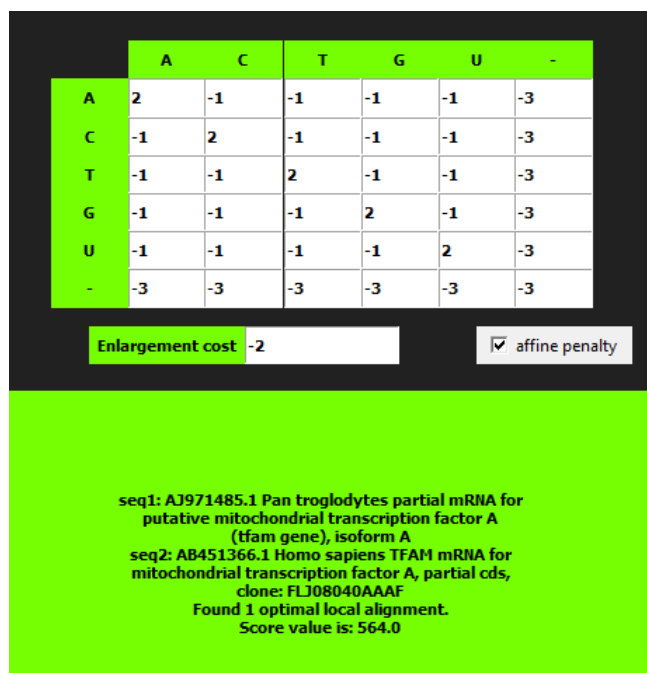


Figure 15: Wynik dopasowania

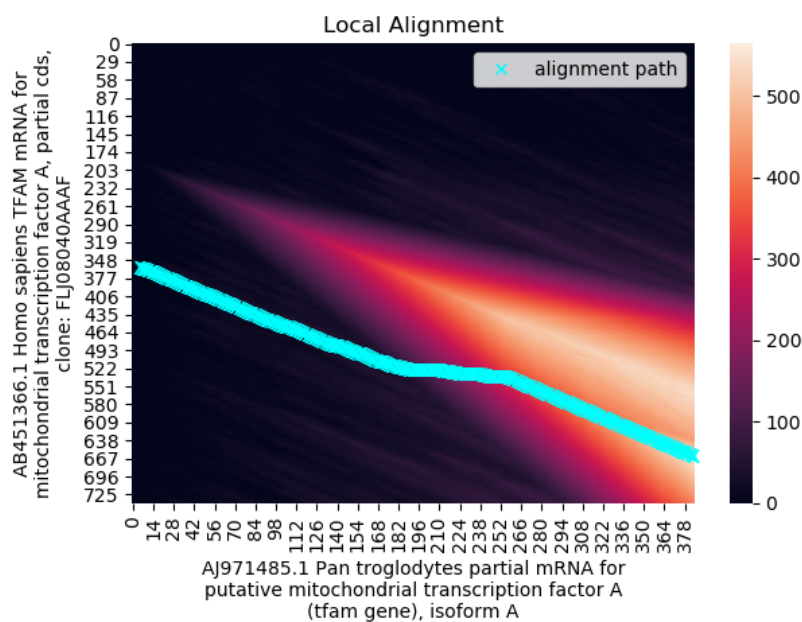


Figure 16: Macierz kosztu z zaznaczoną ścieżką dopasowania

seq1: Aptenodytes patagonicus CR1 gene for chicken repeat 1, partial sequence, clone: king1-8.
 ser2: Eudyptes chrysocome CR1 gene for chicken repeat 1, partial sequence, clone: rock1-35.

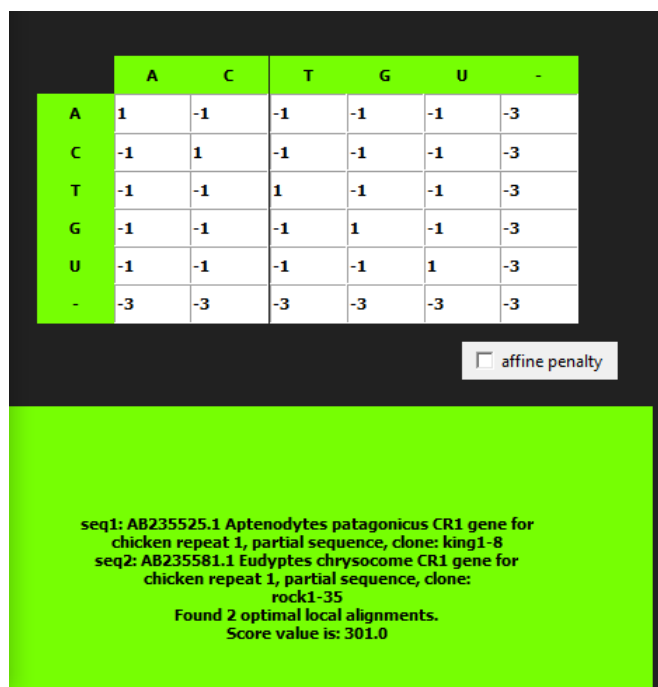


Figure 17: Wynik dopasowania

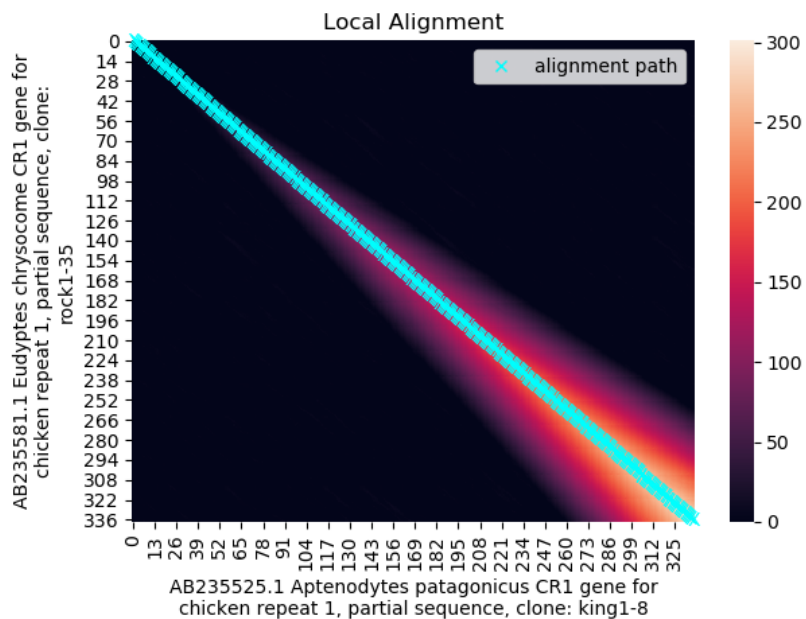


Figure 18: Macierz kosztu z zaznaczoną ścieżką dopasowania

Pierwsza z sekwencji pochodzi od pingwina królewskiego a druga od pingwina skalnego. W przypadku domyślnej macierzy kosztów z liniową karą za przerwy sekwencje zdają się być prawie

identyczne. W przypadku zastosowania affine cost penalty dopasowanie zmienia się diametralnie, co widać na Fig. 20. Stosując karę za rozszerzanie przerwy równą 1 można zaobserwować, że algorytm preferuje kontynuować już rozpoczętą przerwę zamiast wstawiać pojedyncze lub krótsze przerwy.

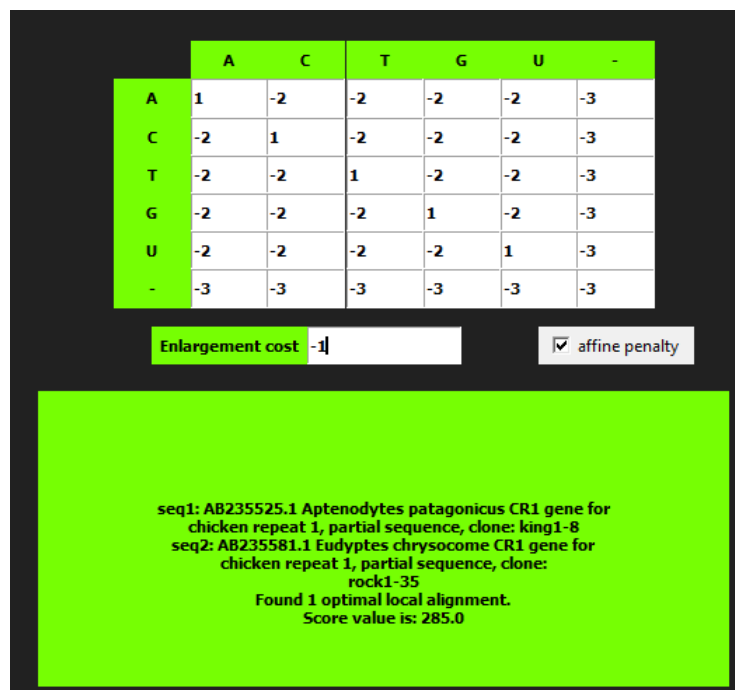


Figure 19: Wynik dopasowania

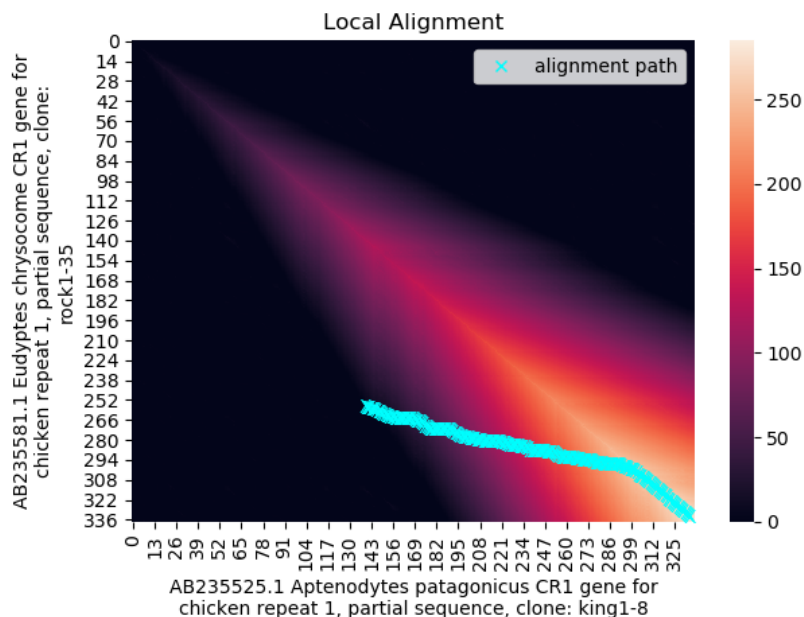


Figure 20: Macierz kosztu z zaznaczoną ścieżką dopasowania

Natomiast po zastosowaniu surowszych kar za rozpoczęcie i kontynuowanie przerwy można zaobserwować poniższe dopasowanie. Znacząco zmniejszyła się długość dopasowania.

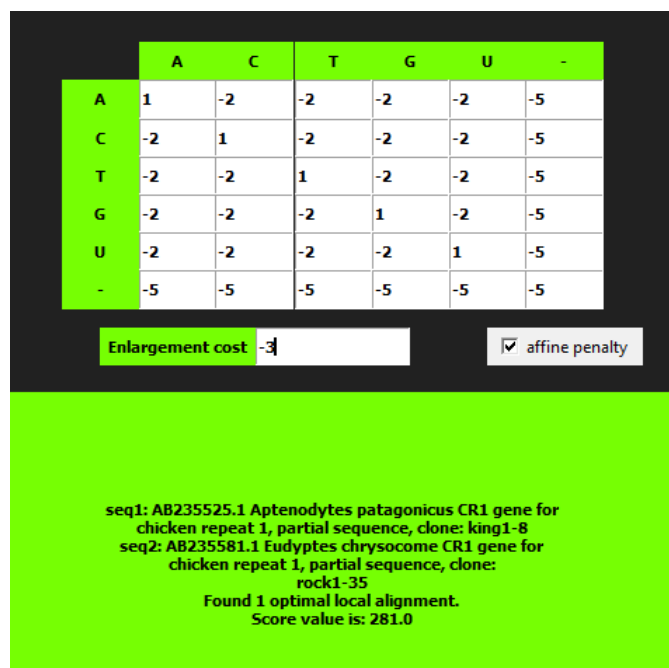


Figure 21: Wynik dopasowania

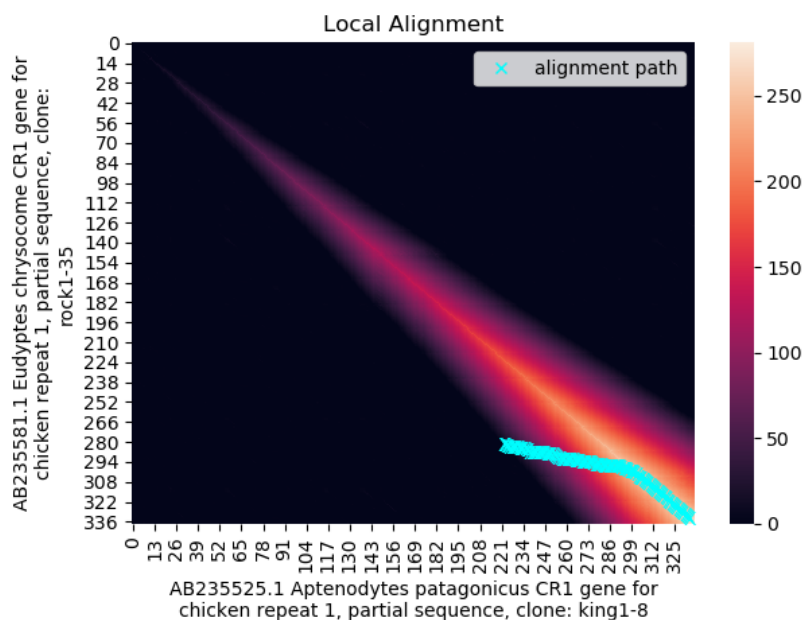


Figure 22: Macierz kosztu z zaznaczoną ścieżką dopasowania

4.2 Ewolucyjnie niepowiązanych

seq1: Anti CDH6 antibodies and anti CDH6 antibody drug conjugates.

seq2: Pan troglodytes chromosome 8 genomic scaffold.

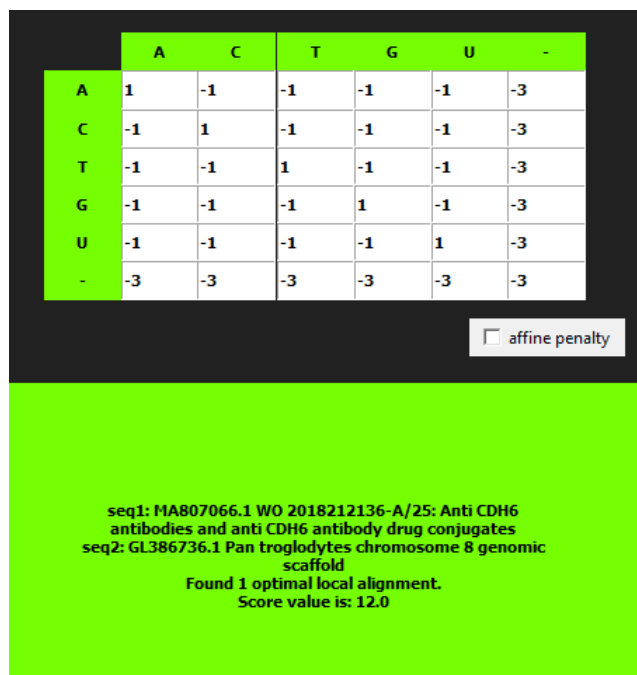


Figure 23: Wynik dopasowania

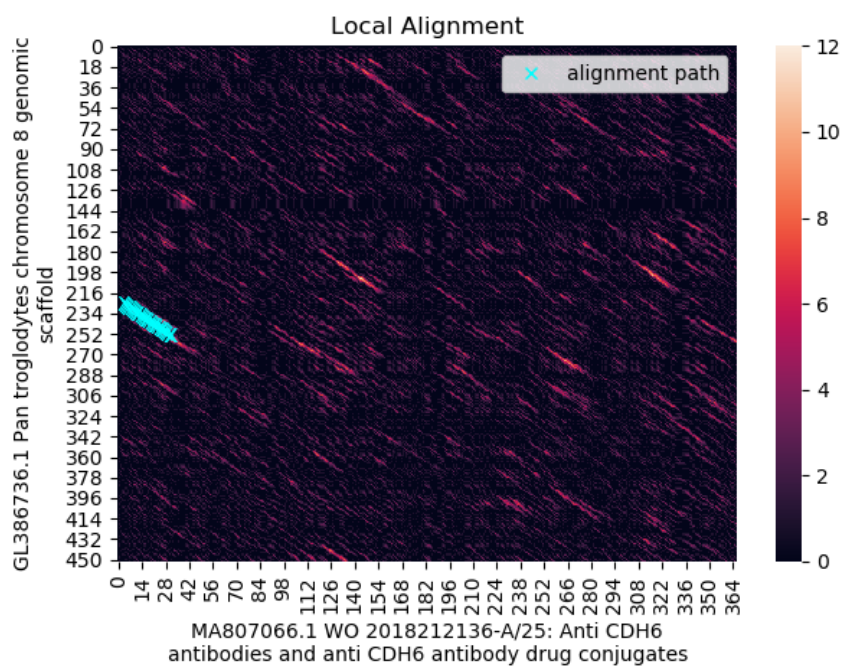


Figure 24: Macierz kosztu z zaznaczoną ścieżką dopasowania

Pierwsze co się rzuca w oczy w przypadku sekwencji nie powiązanych to zupełnie inny rozkład intensywności elementów macierzy wyników (Fig. 24). W porównaniu do map ciepłych przedstawiających punktację dla sekwencji powiązanych tutaj obserwuje się sporo nisko punktowanych fragmentów, zamiast długiego pasma o wysokiej intensywności. Również wynik dopasowania jest bardzo niski, a długość tego fragmentu bardzo mała w stosunku do długości obu sekwencji.

Poniżej zaprezentowano wynik dopasowania po uwzględnieniu affine cost penalty.

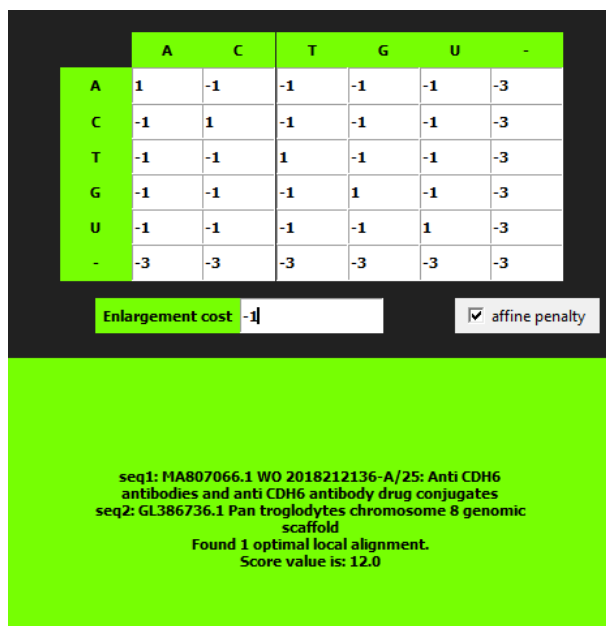


Figure 25: Wynik dopasowania

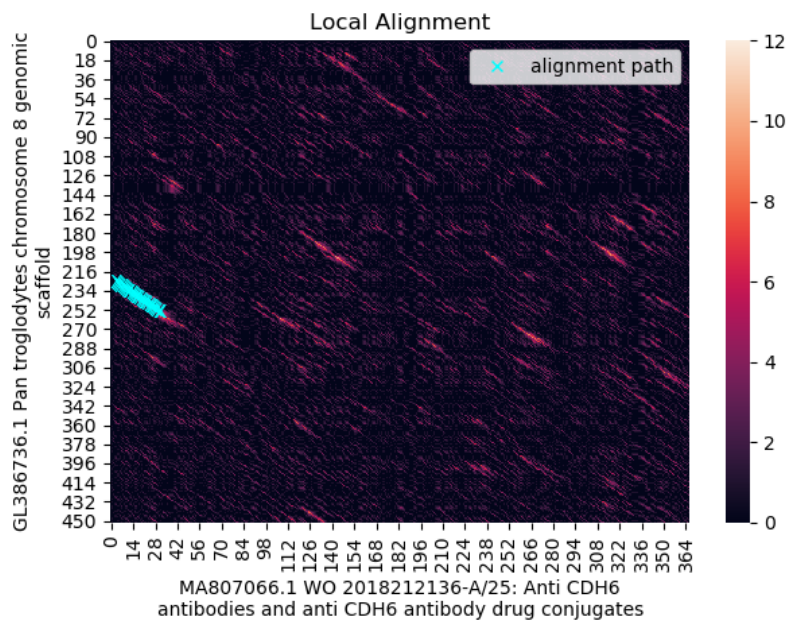


Figure 26: Macierz kosztu z zaznaczoną ścieżką dopasowania

W powyższym przypadku uzyskany wynik jest taki sam jak przy zastosowaniu liniowej kary za przerwę. Po zastosowaniu wyższej punktacji za zgodność zasad uzyskano długi fragment dopasowania. Poprzez zysk 3 punktów w przypadku zgodności zasad oraz -1 w przypadku substytucji algorytm miał możliwość zbudować długą sekwencję faworyzując przemieszczanie się po przekątnej zamiast wstawiać przerwy. Ten przykład pokazuje jak całkowicie odmienne wyniki możemy uzyskać modyfikując macierz kosztów.

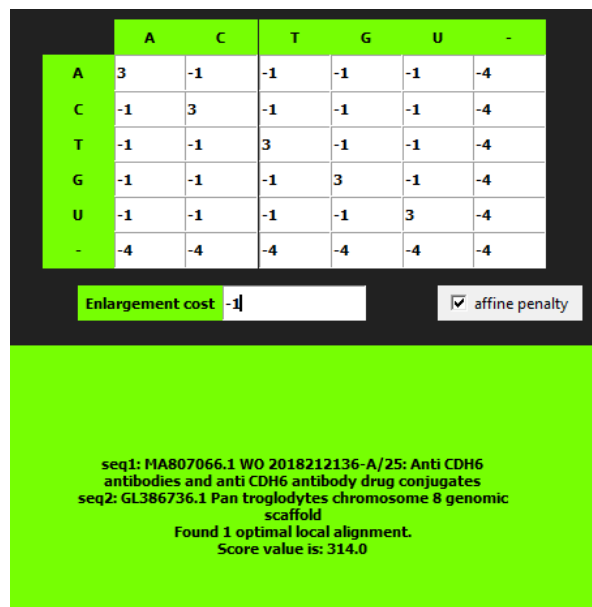


Figure 27: Wynik dopasowania

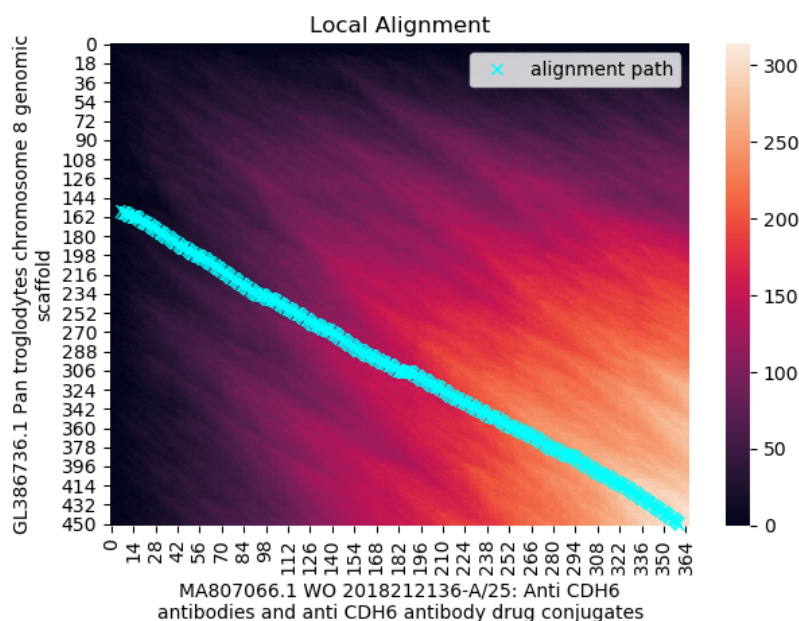


Figure 28: Macierz kosztu z zaznaczoną ścieżką dopasowania