AKWB zadanie 3

1. Opis algorytmu

1.1 Wczytanie pliku wejściowego:

Każda instancja składa się z dwóch plików. Plik z rozszerzeniem .fasta zawiera pięć sekwencji nukleotydowych pochodzących z sekwenatora 454. Każda z sekwencji opatrzona jest unikatowym identyfikatorem. Z kolei plik o rozszerzeniu .qual zawiera ocenę wiarygodności poszczególnych nukleotydów. Nazwa pliku musi być taka sama, różnią się jedynie rozszerzeniem. Program prosi użytkownika o podanie nazwy pliku, który ma zostać wczytany, sprawdzane jest czy plik ten znajduje się w folderach do których została sprecyzowana ścieżka. Plik wczytany zostaje do wektora przechowującego strukturę Sequence, struktura ta składa się z ID sekwencji, wektora przechowującego nukleotydy wchodzące w skład sekwencji, wektora przechowującego oceny wiarygodności dla poszczególnych nukleotydów oraz wektora pozycji w sekwencji poszczególnych nukleotydów.

1.2 Usuwanie nukleotydów o wiarygodności poniżej progu:

Użytkownik proszony jest o podanie minimalnego progu wiarygodności. Próg ten musi być liczbą większą od 0. Program dla każdego elementu z wektora przechowującego strukturę Sequence sprawdza czy jakość tego nukleotydu jest mniejsza od podanego progu wiarygodności. Jeśli natrafi na taki nukleotyd to jest on usuwany z sekwencji. Początkowe pozycje nukleotydów nie ulegają zmianie.

1.3 Utworzenie wierzchołków:

Oprócz progu wiarygodności użytkownik proszony jest o podanie długości podciągu. W strukturze Graph znajduje się metoda tworząca wierzchołki. Dla każdego elementu wektora przechowującego strukturę Sequence sprawdzamy czy liczba nukleotydów w tej sekwencji jest równa lub większa od długości podanej przez użytkownika. Robimy to po to, aby upewnić się że istnieje możliwość utworzenia podciągu w obrębie tej sekwencji. Następnie w pętli iterujemy po nukleotydach pomniejszonych o długość podciągu, ponieważ iterator ten stanowić będzie pozycje pierwszego nukleotydu z podciągu czyli pozycje wierzchołka w sekwencji. Z kolei w drugiej pętli iterator wydłużany jest do momentu osiągnięcia rozmiaru długości podciągu. Z nukleotydów stanowiących sumę iteratorów powstaje podciąg. Do wektora wierzchołków zostaje dodany nowy wierzchołek czyli struktura przechowująca strukturę Sequence, pozycje pierwszego nukleotydu z podciągu oraz sam podciąg.

1.3 Utworzenie nieskierowanych krawędzi pomiędzy wierzchołkami:

Krawędź wstawiana jest pomiędzy dwoma wierzchołkami (i, j) jeśli spełniają one poniższe warunki:

* Wierzchołki mają taki sam podciąg nukleotydów,
* Wierzchołki pochodzą z różnych sekwencji, mają inne ID,
* Różnica pozycji podciągów w sekwencji jest mniejsza od dziesięciokrotnej długości podciągu

Po spełnieniu warunków wierzchołki dodawane są do tymczasowego wektora pomocniczego, a następnie dodawane do wektora wektorów „edges” przechowującego krawędzie. Wektor tymczasowy zostaje wyczyszczony.

1.4 Poszukiwanie kliki w grafie:

**Funkcja degree:**

zwraca stopień wierzchołka jako parametr przyjmuje indeks wierzchołka. Przeszukiwany jest wektor par zawierający krawędzie i zliczane zostają wystąpienia danego indeksu, co oznacza stopień wierzchołka o danym indeksie.

**Funkcja connected:**

zwraca wektor indeksów sąsiadów danego wierzchołka przyjmująca jako parametr indeks wierzchołka. Przeszukuje pary z wektora par edges. Jeśli znaleziony zostanie indeks podany jako parametr, to do wektora metody zostanie zapisany indeks drugiego wierzchołka z pary.

**Funkcja cliqueGenerator:**

Jako parametr otrzymuję int rozmiar, będącego rozmiarem szukanej kliki. W obrębie funkcji znajduje się wektor wektorów „cliques” oraz wektor „probable\_clique”. Dla każdego wierzchołka z wektora krawędzi za pomocą funkcji degree sprawdzane jest czy stopień tego wierzchołka jest większy lub równy od parametru rozmiar pomniejszonego o 1. W pierwszej iteracji szukany jest wierzchołek o stopniu 4. Wierzchołek spełniające ten warunek dodawany jest do probable\_clique. Do wektora tymczasowego przypisywane są za pomocą funkcji connected indeksy sąsiadów tego wierzchołka. Wektor tymczasowy wstawiany jest do probable\_clique. Wektor tymczasowy jest czyszczony, a probable\_clique zostaje posortowane. Następnie dla każdego wierzchołka z probable\_clique za pomocą funkcji connected wyszukiwane są jego wierzchołki sąsiadujące. Wierzchołek oraz jego sąsiedzi dodawani są do wektora tymczasowego, a następnie zostają posortowane. W kolejnym kroku następuję porównanie probable\_clique i wektora tymczasowego. Jeśli jakiś wierzchołek z probable\_clique nie znajduje się w wekotrze tymczasowym to zostaje on usunięty. Następnie sprawdzane jest czy probable\_clique ma rozmiar większy od 4, jeśli tak to do wektora cliques dodawane jest probable\_clique. Jeśli wektor cliques jest pusty to rekurencyjnie wywoływana jest funkcja cliqueGenerator z prametrem wejściowym pomniejszonym o 1. Warunkiem początkowym funkcji jest to że rozmiar kliki musi być większy od 2.

2. Oszacowanie złożoności obliczeniowej algorytmu:

Wczytanie pliku O(n)

Usuwanie nukleotydów O(n2)

Utworzenie wierzchołków O(n2)

Utworzenie krawędzi O(n2)

Spośród wszystkich wierzchołków zawartych w wektorze wektorów egdes szukane są te, których stopień jest wyższy lub równy 4. W tym celu metodę degree należy wykonać dla każdego wierzchołka mającego przynajmniej jedną krawędź – wykonywana jest pętla for. W najgorszym przypadku od każdego wierzchołka będzie wychodzić krawędź, więc pętla wykona się wtedy tyle razy ile jest wierzchołków (n – [długość podciągu]). W metodzie degree przeszukiwany jest cały wektor wektorów edges – w najgorszym przypadku pętla for wykona się n – [długość podciągu] razy. Dla każdego wierzchołka o stopniu większym niż 3 wywoływana jest metoda connected, w której, podobnie jak w metodzie degree, pętla for wykona się n – [długość podciągu] razy. Następnie pętla for będzie iterować po wektorze probable\_clique. Wykona się ona tyle razy ile wynosi rozmiar wektora probable\_clique. W najgorszym przypadku będzie to n – [długość podciągu] razy. Dla każdego elementu tego wektora wywoływana będzie metoda connected i pętla for (wykona się maksymalnie także n – [długość podciągu] razy), w której każdy kolejny element wektora probable\_clique będzie wyszukiwany w tymczasowym wektorze za pomocą funkcji find. Jeśli element nie zostanie znaleziony, następuje wywołanie funkcji erase. Złożoność funkcji find oraz erase wynosi 𝑂(𝑛). Na tej podstawie złożoność funkcji poszukiwania kliki oszacowałem na O(n)5.

Algorytm cechuje się wielomianową złożonością obliczeniową.

3. Testy

a)

**Plik: 03**

>DOJHLOP02HQT4L length=98 xy=3058\_3955 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

ACCCTCCTATTAGAGCTTGATCATCTAGTTCCGTTCCTGTCTCCATGTAATTCAACCCAA

TCATCACAAAACATCTGTATGTAATCGAGTGTGCTAGG

>DOJHLOP02HGCVQ length=114 xy=2939\_2468 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

TTAAACAAGGTCTTTTAATTTTAGATGGGTCACTATTAGAGCTTTTAGGCATGCTTGCAA

CCCAAATAACAATATCTGCTTCTGGTAAAGCATCATCAAGACTAAAGAATTCTT

>DOJHLOP01DIQHW length=113 xy=1326\_1810 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

AATTGCAAGAATGGCTCTTGCAGCGGTGATAATAAGGATTTAATATCTATTAGAGCTTGT

TATTCCTACAAGAACTTAAAGTTCTTGAGGTATTTCTTTTTTGNGACGTGGTG

>DOJHLOP02H6BP1 length=106 xy=3235\_1607 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

AATAGATCTACTGCAAGCTTTGCATCCAACAATTTAACCGTAAGAAAAACGAAGAGCTAA

GTTCTAACTCTCTTCCAGATTTCACTATTAGAGCTTAACAAAGATG

>DOJHLOP01DM14Z length=76 xy=1375\_2817 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

CTCATGGTGGGCAAATTGGTCTGCTCCAGGCGGTTTCAAGGCTATTAGAGCTTCAATGTC

CTTCAATAGCTGTATT

>DOJHLOP02HQT4L length=98 xy=3058\_3955 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

23 24 23 06 28 22 14 29 28 24 17 19 31 31 31 27 30 25 29 32 30 30 29 32 28 17 18 32 31 26 27 21 31 29 23 30 24 31 30 30 32 29 21 12 25 30 30 30 23 14 28 21 25 27 20 24 23 06 24 17

30 30 24 28 23 17 32 25 24 20 07 28 15 31 27 32 28 32 26 31 31 32 23 15 30 27 27 18 20 30 31 29 25 31 32 28 30 25

>DOJHLOP02HGCVQ length=114 xy=2939\_2468 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

31 27 27 26 18 32 31 27 30 25 30 31 23 23 21 13 31 27 24 24 22 13 27 32 32 31 29 28 18 32 27 31 27 29 28 18 32 29 31 27 31 27 25 25 21 09 28 31 26 31 31 30 32 32 30 25 31 26 31 27

28 28 18 24 24 18 30 28 26 32 31 27 30 32 30 32 31 32 24 31 27 32 32 31 27 31 26 25 11 32 32 31 31 31 26 31 32 24 22 32 19 30 29 23 22 02 27 29 24 27 25 17 31 27

>DOJHLOP01DIQHW length=113 xy=1326\_1810 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

28 23 26 20 31 31 30 26 26 31 27 24 24 16 32 32 32 28 23 32 31 16 31 31 29 23 30 26 30 32 22 13 25 29 24 22 14 27 23 22 03 31 28 21 31 28 27 19 32 20 32 17 06 26 25 24 08 28 24 31

28 29 30 26 29 24 31 12 29 30 27 28 26 24 29 30 27 28 27 15 30 31 28 32 22 14 22 31 20 11 18 16 26 25 10 30 19 19 18 16 11 04 31 00 26 14 19 24 32 22 14 32 20

>DOJHLOP02H6BP1 length=106 xy=3235\_1607 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

23 15 21 17 23 27 32 25 31 32 27 28 23 22 26 19 32 21 24 23 18 27 32 24 32 29 24 31 27 21 25 23 28 27 18 25 23 31 26 21 31 22 13 31 21 21 20 16 09 16 29 22 13 30 32 25 31 32 26 19

24 28 22 32 27 29 26 32 27 23 31 20 29 23 29 23 31 24 22 25 24 07 21 24 31 27 32 18 28 22 18 06 31 28 25 18 19 08 14 23 22 03 30 20 22 27

>DOJHLOP01DM14Z length=76 xy=1375\_2817 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

32 32 29 32 32 31 27 31 28 28 15 31 29 28 16 31 28 31 27 30 31 29 32 31 27 31 27 30 31 28 32 31 27 28 28 19 31 31 27 31 28 30 24 31 31 28 29 31 30 30 25 31 27 29 30 25 32 32 28 31

26 30 27 28 29 24 29 31 31 28 30 32 27 32 30 26

**Długość podciągu: 6**

**Minimalny próg wiarygodności: 20**

**DOJHLOP02HQT4L 13 GAGCTT**

**DOJHLOP02HGCVQ 39 GAGCTT**

**DOJHLOP01DM14Z 48 GAGCTT**

**Długość podciągu: 8**

**Minimalny próg wiarygodności: 20**

**DOJHLOP02HGCVQ 30 TCACTATA**

**DOJHLOP02H6BP1 81 TCACTATA**

**Długość podciągu: 4**

**Minimalny próg wiarygodności: 20**

**DOJHLOP02HGCVQ 40 AGCT**

**DOJHLOP02HQT4L 14 AGCT**

**DOJHLOP02H6BP1 15 AGCT**

**DOJHLOP01DIQHW 54 AGCT**

**DOJHLOP01DM14Z 49 AGCT**

b)

**Plik: 06**

>DOJHLOP01CAUIL length=93 xy=0826\_2155 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

TTCATAGATACTCCTGGAGCTTATGCAGGGCTCATAGCAGAAGAACAAGGACAAGGAGAG

GCAATCGCAGTGAATCTACGAGAAATGTTTAGG

>DOJHLOP01EDNAY length=118 xy=1678\_2216 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

TTGATTTCAAACCAACAGCCGAAAACCCGTGCCTTTCAGAAGCCTTGCCGATGAAATCCA

TCTTTTCCAACGTTATAGCAATCGCAGTGAATCTGAAGTTGGTATGACAATTCTCGAA

>DOJHLOP02GRF2Z length=100 xy=2655\_3481 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

ACTATTGAGGGAGCATATATAGACTCATGCTTTGTTCCAGAGAGCAATCGCAGTGAATCT

AACTGCAAGAAAGTAAAGGTTTTGAATGCAATGGAGTTGG

>DOJHLOP01DV65W length=102 xy=1479\_3250 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

AACTCCATAATAATTGCAATCGCAGTGAATCTGTAGGAGATACGACTACCTGTCTCCTAA

TTCAATGCATCTTGATTGAATTCTTTTGCAATATCCGCATAT

>DOJHLOP01DMGM6 length=112 xy=1368\_3632 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

GAAATGGCAATCGCAGTGAATCTAAAGAATGTAGATAAACAAATAGTTACTACCCAAAAG

GGTGGAGCCTTGAAGGAGCATCACCGTGCGCATGGTTGACAGAAGATTTGAG

>DOJHLOP01CAUIL length=93 xy=0826\_2155 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

29 26 32 31 30 32 31 32 32 32 28 32 31 27 31 30 27 21 31 29 29 27 28 29 31 32 24 28 28 18 26 28 29 31 29 25 31 31 31 31 30 28 31 31 28 31 30 27 30 26 28 31 30 27 28 26 26 23 26 31

28 32 28 22 26 29 28 32 31 24 26 31 25 18 31 28 21 28 28 29 26 26 25 24 09 17 22 28 27 15 32 30 25

>DOJHLOP01EDNAY length=118 xy=1678\_2216 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

30 25 32 20 27 26 13 32 28 27 14 31 28 31 28 32 30 32 31 28 31 24 24 19 04 29 28 18 30 24 30 30 26 28 27 15 31 32 31 30 25 30 31 27 31 28 31 31 27 32 28 32 31 28 27 14 31 31 26 30

22 29 26 25 22 10 30 26 28 22 28 21 31 26 32 32 29 30 27 32 28 28 16 21 28 23 25 24 19 05 29 23 27 32 29 26 20 31 31 27 30 25 31 31 25 32 30 31 31 27 31 28 28 21 29 31 27 21

>DOJHLOP02GRF2Z length=100 xy=2655\_3481 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

32 32 31 24 31 27 30 32 29 28 16 28 32 28 28 32 19 32 26 32 28 32 32 32 32 31 31 31 31 32 26 26 18 32 31 27 30 25 31 30 32 31 31 32 32 32 31 30 27 32 31 31 31 31 31 27 19 30 25 24

22 13 32 30 32 32 25 23 18 23 22 02 19 10 29 28 16 29 26 25 25 22 13 25 24 17 32 32 31 30 26 28 27 24 31 29 29 24 29 26

>DOJHLOP01DV65W length=102 xy=1479\_3250 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

31 28 27 28 31 26 26 26 28 26 29 28 26 19 19 18 16 11 03 25 31 30 31 27 30 28 27 19 30 28 27 19 27 25 17 30 27 32 28 29 32 26 23 29 29 17 27 30 29 27 32 32 32 25 29 30 28 24 30 27

31 27 29 30 28 29 28 30 31 23 28 30 28 30 32 31 27 26 30 25 17 06 26 25 25 22 12 29 31 30 27 26 30 29 28 26 28 21 27 22 25 30

>DOJHLOP01DMGM6 length=112 xy=1368\_3632 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

26 25 24 07 28 30 31 29 24 14 31 30 27 29 28 28 23 27 28 25 24 08 29 28 27 15 32 27 20 28 25 28 31 25 29 31 29 28 17 26 28 28 15 24 17 24 29 24 26 31 32 18 25 25 18 24 24 18 02 27

26 13 26 30 25 16 31 30 27 30 27 19 29 27 29 26 28 30 32 31 24 22 17 29 24 29 32 31 31 26 30 31 27 31 26 30 25 21 30 32 25 30 29 24 28 31 26 25 10 28 31 22

**Długość podciągu: 5**

**Minimalny próg wiarygodności: 20**

**DOJHLOP01CAUIL 61 GCAAT**

**DOJHLOP01EDNAY 78 GCAAT**

**DOJHLOP02GRF2Z 44 GCAAT**

**DOJHLOP01DV65W 88 GCAAT**

**Długość podciągu: 5**

**Minimalny próg wiarygodności: 25**

**DOJHLOP01CAUIL ATAGA 4**

**DOJHLOP02GRF2Z ATAGA 19**

**DOJHLOP01DV65W ATAGA 40**

**Długość podciągu: 5**

**Minimalny próg wiarygodności: 30**

**DOJHLOP01CAUIL 38 CAGAG**

**DOJHLOP01EDNAY 37 CAGAG**

**DOJHLOP02GRF2Z 37 CAGAG**

c)

**Plik: 01**

>DOJHLOP02FEHEE length=119 xy=2098\_0516 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

TCTGTAGCCTCCAGGTTTTGCCATGCCAAAGTTTCTTGCAACATTCTCTTTTGTATCTCT

ACCTTTTTGTTGTCCAATTAAAAGGACGGATTTTTCACCTATTCGAGCTAACCCTCCTA

>DOJHLOP01D0HHA length=100 xy=1528\_2540 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

ATATTATTTCTTCCCAACTGAAGTTGGTATGACAATTCTCGCAGGTTTTGCCATGCGCTT

CAGCAAAACAGGCAAACATTTCTCTTTTAGGATTTTCCAT

>DOJHLOP02GY6HG length=105 xy=2743\_3842 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

GAGGTGCAGTACACCATACGTCCAAATCCCCTTGAGCGATATGGTCACAGGTTTTGCCAT

GCGCCTTTCGCTTGGCATCTTCAAACTTGTTGAATGTCCAATTAG

>DOJHLOP02IWSEB length=95 xy=3536\_3377 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

CATATCAGGCTTTACAGCAAGATTTCTCCCGCAAGCAACAAAAACCATTCATATGTCTTT

TTCTCCACAGGTTTTGCCATGCACAAGCTCATCTG

>DOJHLOP02JHF8V length=104 xy=3772\_0749 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

TTGTTTATCTACATTCTTTATTTCTAAGTAAGAATCATTTCGTACCAAATCTCTAGCCAT

GTGCAACATAAGACTTGTTTCCAGGTTTTGCCATGCAAGTCCCT

>DOJHLOP02FEHEE length=119 xy=2098\_0516 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

31 32 32 32 31 29 29 31 26 32 31 27 29 31 27 25 25 21 09 32 30 26 31 29 31 30 26 27 27 18 31 28 27 15 32 31 26 30 31 30 26 28 30 31 26 31 30 29 25 25 22 10 30 30 31 31 32 31 31 22

31 31 27 21 21 20 16 09 32 31 26 27 32 31 27 26 24 31 27 24 24 21 13 31 27 26 32 30 26 31 22 22 21 15 05 31 25 26 19 32 32 27 21 31 31 22 26 32 31 30 27 28 28 18 31 26 19 31 27

>DOJHLOP01D0HHA length=100 xy=1528\_2540 region=1 run=R\_2005\_09\_08\_15\_35\_38\_

32 24 21 28 22 31 25 24 09 32 30 25 25 25 18 21 12 29 26 28 24 16 28 20 11 28 23 31 31 27 31 31 28 31 28 31 28 31 26 26 32 29 24 28 25 29 31 28 32 26 20 29 24 29 31 28 28 32 31 28

27 32 24 32 25 25 20 07 31 16 18 06 30 23 22 03 30 28 27 26 13 26 29 18 24 24 19 03 29 20 10 22 25 25 22 11 31 27 32 31

>DOJHLOP02GY6HG length=105 xy=2743\_3842 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

27 26 31 27 31 32 31 22 32 31 28 32 28 29 23 32 30 31 32 31 27 30 26 28 28 16 32 25 24 20 07 31 27 32 22 30 28 31 31 31 31 29 30 26 19 31 32 31 30 25 24 20 07 24 17 31 30 25 30 30

26 26 32 31 26 28 28 15 32 31 30 31 27 28 21 30 20 29 32 28 21 31 25 24 18 30 31 26 32 26 19 25 21 12 31 31 31 31 26 31 26 29 24 31 31

>DOJHLOP02IWSEB length=95 xy=3536\_3377 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

32 32 29 27 32 30 31 30 26 31 28 27 18 28 32 26 31 32 30 27 32 31 28 27 18 19 31 29 28 18 32 30 29 23 31 32 19 07 25 22 22 21 16 07 31 27 23 31 26 31 28 31 32 31 31 11 32 22 22 21

16 06 23 25 30 26 32 17 18 06 29 26 27 27 18 32 32 30 25 23 29 25 24 25 27 21 28 28 27 25 30 28 27 15 31

>DOJHLOP02JHF8V length=104 xy=3772\_0749 region=2 run=R\_2005\_09\_08\_15\_35\_38\_

29 23 31 26 25 10 30 32 32 31 31 30 28 31 27 31 28 27 15 29 28 27 15 30 31 31 27 32 32 31 27 31 31 26 31 32 31 27 27 13 32 32 31 32 30 25 29 28 16 32 32 31 32 31 30 29 31 27 30 32

31 21 25 32 31 27 32 25 32 29 26 31 25 32 30 27 31 26 25 10 31 27 32 27 30 31 27 32 21 31 27 30 27 31 31 26 30 24 25 26 28 28 18 28

**Długość podciągu: 4**

**Minimalny próg wiarygodności: 30**

**DOJHLOP02GY6HG 59 ATGC**

**DOJHLOP02IWSEB 22 ATGC**

**DOJHLOP02JHF8V 59 ATGC**

**Długość podciągu: 7**

**Minimalny próg wiarygodności: 25**

**DOJHLOP02FEHEE TCCAGGT 10**

**DOJHLOP02JHF8V TCCAGGT 79**

**Długość podciągu: 9**

**Minimalny próg wiarygodności: 20**

**DOJHLOP02FEHEE 17 TTGCCATGC**

**DOJHLOP01D0HHA 48 TTGCCATGC**

**DOJHLOP02GY6HG 52 TTGCCATGC**

**DOJHLOP02IWSEB 73 TTGCCATGC**

**DOJHLOP02JHF8V 88 TTGCCATGC**

4. Wnioski

Ciężko wybrać odpowiednie wartości parametrów, ponieważ w przypadku ustawienia wysokiego minimalnego progu wiarygodności spowoduje pominięcie znacznej ilości nukleotydów przez co znalezienie wzorca może być niemożliwe. Z kolei zbyt niski minimalny próg spowoduje znalezienie nukleotydów o słabej jakości odczyty z sekwenatora. Ustawienie krótkiej długości podciągu może spowodować znalezienie wielu wzorców, ponieważ zwiększa to prawdopodobieństwo na losowe ułożenie nukleotydów przez co ciężko będzie wyciągnąć jakieś wnioski z tego wzorca. Jeśli podamy algorytmowi poszukiwanie długiego podciągu może okazać się niemożliwe, ze względu na występowanie niskojakościowych odczytów lub po prostu w badanych sekwencjach nie będzie takich samych podciągów. Te same wyniki dla tej samej długości podciągów, ale różnych progów wiarygodności świadczy o tym, że nukleotydy motywu mają dość wysoką jakość i z tego powodu zmiana progu wiarygodności nie wpłynęła na ciąg nukleotydowy motywu.