Algorytmy Kombinatoryczne w Bioinformatyce

Projekt 3

Jakub Sudoł
147906

Opis projektu:

Zadaniem projektowym było napisanie programu który wczytuje dwa pliki z 5 losowo wybranymi sekwencjami. Pierwszy z nich to plik z sekwencjami nukleotydowymi a drugi z jakością tych nukloetydów. Następnie program miał przyciąć te nukleotydy które nie spełniały wymogu jakości ustalonego przez użytkownika. Oryginalne pozycje nukleotydów miały jednak zostać zapisane. Następnie program tworzy podciągi od długości od 4 do 9 (kolejny parametr ustawiany przez użytkownika). Po utworzeniu podciągów należało stworzyć graf a następnie znaleźć w nim strukturę typu klika (lub struktury zbliżonej do kliki). Wynik jest wyświetlany w terminalu.

Utworzenie instancji

Początkowym zadaniem było utworzenie 5 instancji. Każda instancja składa się z 2 plików: jeden z sekwencją a drugi z jakością. Każda instancja zbudowana jest z 5 losowo wybranych sekwencji udostępnionych na stronie projektu. Pliki powinny mieć odpowiednie rozszerzenia jak .fasta i .qual. Przykład dwóch plików:

Plik z sekwencją:

```
>DOJHLOP01DWJWJ length=100 xy=1483_3377 region=1 run=R_2005_09_08_15_35_38_
TCGATTCTATGGAGGGATGCTTAAATTCTGTAGGAAGCAGCATCAGCAATTAAAAAATTACTGGACCTGATCTTATGAAGTTAGGATTGTTGACGAGGTA
>DOJHLOP02GVZDE length=96 xy=2707_2112 region=2 run=R_2005_09_08_15_35_38_
TTACTTATGTTTGAACATATTAAATTCTGTATTGCAAGCCCTGAAGCTTGTGCTTCGATTCTATGGAGGGATGCTGGCAAGGCTCCGGAAGCAGCA
>DOJHLOP01C8Z6P length=100 xy=1215_2463 region=1 run=R_2005_09_08_15_35_38_
AAGCTCCAGGAGTATCTATGTTAAATTCTGTAGAAGATTAAAGCGATCGGCATGGTCCATCAGCCTCAAAGTCCTTTCTGTAGCCTCCTAGGTTTTGCCA
>DOJHLOP02HQT4L length=98 xy=3058_3955 region=2 run=R_2005_09_08_15_35_38_
ACCCTCCTATTAGAGCTTGATTAAATTCTGTAGTTCCTGTCTCCATGTAATTCAACCCAATCATCACAAAACATCTGTATGTAATCGAGTGTGCTAGG
>DOJHLOP02JDZJV length=63 xy=3732_2985 region=2 run=R_2005_09_08_15_35_38_
ATTAGAAACACTTGCAGCACGTTAAATTCTGTAATATTAATGCTACTTACTCCAGCTCAGAAG
```

Plik z jakością:

```
>DOJHLOP01DWJWJ length=100 xy=1483_3377 region=1 run=R_2005_09_08_15_35_38_
32 31 29 26 30 27 32 32 31 30 28 26 26 28 27 19 26 31 24 32 29 31 35 29 29 31 34 32 30 31 28 36 29 26 22 14 32 31 25 29 31 31 29 32
>DOJHLOP02GVZDE length=96 xy=2707_2112 region=2 run=R_2005_09_08_15_35_38_
26 24 28 31 28 26 31 30 31 29 28 16 32 31 27 30 17 31 30 29 31 35 29 29 31 34 32 30 31 28 36 29 26 32 30 31 26 32 29 28 17 26 27 28
>DOJHLOP01C8Z6P length=100 xy=1215_2463 region=1 run=R_2005_09_08_15_35_38_
28 23 30 31 31 29 24 30 25 19 32 32 22 31 28 31 32 28 31 31 29 31 35 29 29 31 34 32 30 31 28 36 18 23 15 31 31 21 11 25 24 9 32 24
>DOJHLOP02HQT4L length=98 xy=3058_3955 region=2 run=R_2005_09_08_15_35_38_
23 24 23 6 28 22 14 29 28 24 17 19 31 31 31 27 30 25 29 32 29 31 35 29 29 31 34 32 30 31 28 36 31 29 23 30 24 31 30 30 32 29 21 12
>DOJHLOP02JDZJV length=63 xy=3732_2985 region=2 run=R_2005_09_08_15_35_38_
26 23 15 22 22 24 22 5 29 29 31 27 20 31 25 20 30 31 32 32 28 29 31 35 29 29 31 34 32 30 31 28 36 27 30 32 24 22 28 21 13 31 32 12
```

(Pełen plik nie zmieścił się na zdjęciu)

Następnie do każdej instancji trzeba było dodać motyw o długości kilkunastu nukleotydów. Motyw miał się powtarzać w wybranych 5 sekwencjach danej instancji a różnice w pozycjach między sekwencjami musiały być w granicy od 0 do 2 pozycji. Dodatkowo nie powinien się znajdować na samym początku sekwencji. Motywy można było dodać lub zmienić istniejące już nukleotydy. Pamiętać natomiast należało o odpowiednim zmodyfikowaniu pliku z jakością

(zwłaszcza jak chodzi o to by zgadzała się długość sekwencji). Tak stworzone instancje posłużyły mi później w testowaniu mojego programu.

Użyte metody i ich działanie:

Odczytywanie

Odczytywanie odbywa się w dwóch metodach wywoływanych z main(). Każda metoda odpowiada danemu plikowi i wczytane sekwencje są zapisywane do wektorów. Każdy plik posiada osobny wektor. Do odczytywania wykorzystałem bibliotekę fstream oraz getline(). Metoda rozpoznaje linie z ID sekwencji i dodaje do wektora tylko te które zawierają sekwencję lub jakość tej sekwencji. Po odczycie plik jest zamykany.

Przycinanie jakości

Po wykonaniu metod odczytywania program pyta użytkownika o minimalny próg jakości nukleotydu po czym wywoływana jest metoda przycinająca dane sekwencje nie spełniające warunku. Do wywołanej metody przekazywana jest wartość podana przez użytkownika. Podczas przycinania tworzone są 3 nowe wektory. Pierwszy z nich ‘trimmedSequences’ zawiera przycięte sekwencje nukleotydowe. Drugi ‘trimmedQualities’ zawiera przycięte sekwencje jakości (zapis jakości po przycięciu wygląda dokładnie tak samo jak chodzi o oddzielenie poszczególnych liczb spacją) a ostatni ‘trimmedPositions’ zawiera oryginalne pozycje pozostawionych nukleotydów. Dodatkowo zapisane oryginalne pozycje nukleotydów zaczynają się od 1. Bardzo ważne było pozostawienie oryginalnych pozycji nukleotydów ze względu na następne kroki.

Tworzenie podciągów

Po przycięciu sekwencji program pyta o długość podciągów jakie ma utworzyć z wczytanych sekwencji. Jest to parametr definiowany przez użytkownika a pętla while zabezpiecza, że dopóki użytkownik nie poda liczby z przedziału 4-9, program będzie domagał się odpowiedniej cyfry. Kiedy warunek zostanie spełniony wywoływana jest metoda do której między innymi przekazywana jest długość podciągu. W wywołanej metodzie dwie pętle przechodzą przez każdą z wczytanych do wektora sekwencji i tworzą podciągi zapisując je do wektora typu ‘vector<pair<int, pair<int, string>>> windows’. W polu windows.first zapisywany jest oryginalny numer sekwencji z jakiej pochodzi podciąg. windows.second.first zapisuje pozycję pierwszego nukleotydu w danym podciągu a windows.second.second to string zapisujący ciąg konkretnego pociągu. Dany podciąg pobierany jest za pomocą substr() gdzie pierwsza wartość to nukleotyd na którym aktualnie jest a druga to wybrana przez użytkownika długość podciągu.

Wszystkie wcześniej wspomniane informacje są zapisywane w wektorze ‘windows’

Tworzenie grafu

W tworzonym grafie za wierzchołki odpowiadają podciągi wcześniej utworzone. Do tworzenia grafu stworzyłem kolejną metodę która w oparciu o zasady tworzenia grafu tworzy połączenia między wierzchołkami. Zasady tworzenia krawędzi są następujące:

1. Krawędź może powstać tylko pomiędzy wierzchołkami o takiej samej sekwencji podciągu.
2. Krawędź może powstać tylko pomiędzy wierzchołkami pochodzącymi z innej sekwencji.
3. Krawędź może powstać tylko pomiędzy wierzchołkami których różnica między pozycjami pierwszego nukleotydu tych dwóch wierzchołków nie jest większa niż dziesięciokrotność długości podciągu.

Na podstawie tych trzech zasad tworzony jest graf który zapisywany jest jako mapa gdzie kluczem jest string z sekwencją podciągu a wartością jest wektor z następnikami. Następniki są zapisane jako indeksy okien w wektorze okien co pozwoli nam później jeszcze raz porównać wartości by odnaleźć strukturę typu klika. Ze względu na sposób działania tej metody wartości następników się powtarzają. Dzieje się to dlatego, że najpierw porównywany jest podciąg z 1 sekwencji do pozostałych, później podciąg z 2 sekwencji do pozostałych (pomijając 1 sekwencję z którą już porównaliśmy) i tak dalej. W rezultacie dostajemy wynik wyglądający na przykład tak:

wierzchołek 1	**wierzchołek 2**	**wierzchołek 3**	**wierzchołek 4**	**wierzchołek 5**
Sekwencja 1 do sekwencji 2				
5	16	28	41	54

Sekwencja 2 do sekwencji 3		
28	41	54

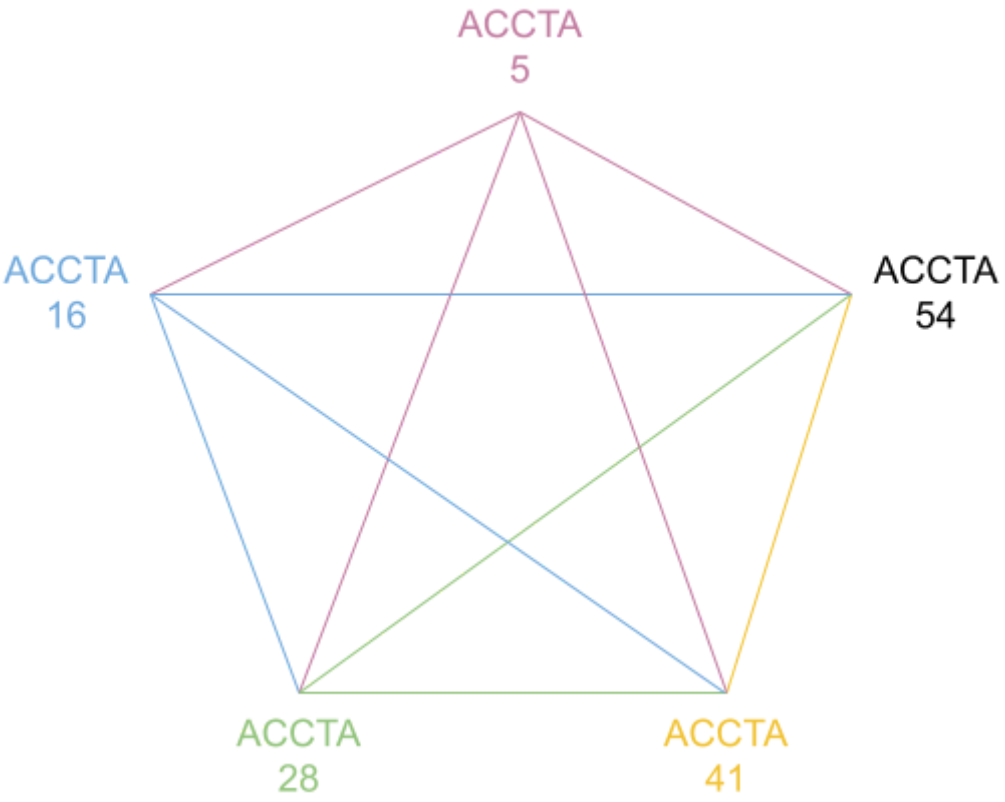
sekwencja 3 do sekwencji 4	
41	54

sekw. 4 do 5
54

Natomiast powyższy przykład w wektorze wygląda tak:
Motyw: “ACCTA”
Wektor: 1 16 28 41 54 28 41 54 41 54 54

Jak widać w pierwszej tabeli, jest ona dłuższa. Dzieje się tak dlatego, że oprócz następników dodaję indeks pierwszego pociągu żeby ułatwić późniejsze porównywanie. Każda tabela to następniki kolejnego sprawdzanego wierzchołka co widać na rysunku poniżej:

Przykład:
Motyw: “ACCTA”
Wartość pod podciągiem to indeks elementu z wektora ‘windows’.



Sprawdzanie warunków połączeń odbywa się za pomocą funkcji ‘if’. Jeśli któryś z warunków nie zostanie spełniony, bieżący krok pętli jest przerywany i porównywane są kolejne podciągi. W sprawdzaniu odległości zastosowałem także funkcję swap która dba o to by zawsze mniejsza wartość była odejmowana od większej. Jeśli wszystkie warunki zostaną spełnione, tworzona jest krawędź czyli następnik jest dodawany do wektora znajdującego się w mapie.

Szukanie kliki

Program który napisałem szuka kliki w postaci gwiazdy (jak było to pokazane w poprzednim podpunkcie. Jest to ułatwienie zaproponowane na stronie projektu.

Wywoływana jest następna metoda która która wyszukuje takie motywy dla których dla danego klucza w wektorze znajduje się dokładnie 11 wartości. Dzieje się tak ponieważ ze względu na implementację mojego sposobu tworzenia grafu, następników w danym wektorze powinno być dokładnie 10 plus pierwsza wartość pierwszego podciągu czyli łącznie 11. Wartości są po kolei porównywane pod kątem sekwencji podciągu, pochodzenia z innej sekwencji oraz odległości od siebie. Odpowiednie warunki ‘if’ oraz pętle po kolei odczytują i sprawdzają następniki i jeśli wszystkie połączenia dla wszystkich pięciu wierzchołków się zgadzają tzn. zostały utworzone poprawnie i tworzą strukturę gwiazdy, Program wypisze motyw z jakiego składa się ta struktura oraz w jakiej sekwencji i na jakiej pozycji oryginalnej można znaleźć dany podciąg. Przykładowy output programu:

```
Motyw: CTGA
W sekwencji: 1 znaleziony na pozycji: 28
W sekwencji: 2 znaleziony na pozycji: 27
W sekwencji: 3 znaleziony na pozycji: 28
W sekwencji: 4 znaleziony na pozycji: 28
W sekwencji: 5 znaleziony na pozycji: 29
```

Program po odnalezieniu pierwszej struktury gwiazdy przerywa działanie ponieważ wymagane jest znalezienie jednej takiej struktury. Jeśli taka struktura nie istnieje, program wyświetli stosowny komunikat.

Złożoność algorytmu:

Złożoność algorytmu szacowana jest na podstawie największej ilości pętli w pojedynczej metodzie czyli 5.

Testy

W instancjach które stworzyłem, motywy były wstawiane na miejsce poprzednich nukleotydów a nie dodawane do sekwencji. Dodatkowo zadbałem by odcinek z motywem miał odpowiednio wysoką jakość. Motywy wstawione przeze mnie zaznaczyłem na kolor żółty

Test 1

Plik z sekwencją:

>DOJHLOP01DWJWJ length=100 xy=1483_3377 region=1 run=R_2005_09_08_15_35_38_
TCGATTCTATGGAGGGATGCTTAAATTCTGTAGGAAGCAGCATCAGCAATTAAAAAATTACTGGACCTGATCTTATGA
AGTTAGGATTGTTGACGAGGTA
>DOJHLOP02GVZDE length=96 xy=2707_2112 region=2 run=R_2005_09_08_15_35_38_
TTACTTATGTTTGAACATATTAAATTCTGTATTGCAAGCCCTGAAGCTTGTGCTTCGATTCTATGGAGGGATGCTGGC
AAGGCTCCGGAAGCAGCA
>DOJHLOP01C8Z6P length=100 xy=1215_2463 region=1 run=R_2005_09_08_15_35_38_
AAGCTCCAGGAGTATCTATGTTAAATTCTGTAGAAGATTAAAGCGATCGGCATGGTCCATCAGCCTCAAAGTCCTTT
CTGTAGCCTCCTAGGTTTTGCCA

>DOJHLOP02HQT4L length=98 xy=3058_3955 region=2 run=R_2005_09_08_15_35_38_
ACCCTCCTATTAGAGCTTGATTAAATTCTGTAGTTCCTGTCTCCATGTAATTCAACCCAATCATCACAAAACATCTGTA
TGTAATCGAGTGTGCTAGG
>DOJHLOP02JDZJV length=63 xy=3732_2985 region=2 run=R_2005_09_08_15_35_38_
ATTAGAAACACTTGCAGCACGTTAAATTCTGTAATATTAATGCTACTTACTCCAGCTCAGAAG

Plik z jakością:

>DOJHLOP01DWJWJ length=100 xy=1483_3377 region=1 run=R_2005_09_08_15_35_38_
32 31 29 26 30 27 32 32 31 30 28 26 26 28 27 19 26 31 24 32 29 31 35 29 29 31 34 32 30 31 28 36 29 26 22 14 32
31 25 29 31 31 29 32 28 31 31 29 23 24 22 19 19 18 16 10 3 31 28 32 29 32 31 28 23 30 28 31 31 26 24 31 31 28 30
24 27 24 17 29 28 23 31 20 11 16 31 28 31 21 12 30 29 31 26 31 29 24 31 17
>DOJHLOP02GVZDE length=96 xy=2707_2112 region=2 run=R_2005_09_08_15_35_38_
26 24 28 31 28 26 31 30 31 29 28 16 32 31 27 30 17 31 30 29 31 35 29 29 31 34 32 30 31 28 36 29 26 32 30 31 26
32 29 28 17 26 27 28 22 30 29 29 26 31 30 31 32 30 27 32 29 27 30 27 31 29 27 28 23 22 30 28 27 18 26 31 27 32
30 31 27 32 29 23 31 27 25 18 29 24 26 24 30 27 31 30 31 32 31 25
>DOJHLOP01C8Z6P length=100 xy=1215_2463 region=1 run=R_2005_09_08_15_35_38_
28 23 30 31 31 29 24 30 25 19 32 32 22 31 28 31 32 28 31 31 29 31 35 29 29 31 34 32 30 31 28 36 18 23 15 31 31
21 11 25 24 9 32 24 32 31 28 28 30 25 26 29 29 25 18 31 27 20 32 29 30 28 29 31 28 30 32 26 25 10 31 9 31 27 29
28 18 28 32 32 29 31 29 31 28 29 31 28 23 32 30 26 24 24 18 2 27 24 17 32
>DOJHLOP02HQT4L length=98 xy=3058_3955 region=2 run=R_2005_09_08_15_35_38_
23 24 23 6 28 22 14 29 28 24 17 19 31 31 31 27 30 25 29 32 29 31 35 29 29 31 34 32 30 31 28 36 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 6 24 17 30 30 24 28 23 17 32 25 24 20 7 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
>DOJHLOP02JDZJV length=63 xy=3732_2985 region=2 run=R_2005_09_08_15_35_38_
26 23 15 22 22 24 22 5 29 29 31 27 20 31 25 20 30 31 32 32 28 29 31 35 29 29 31 34 32 30 31 28 36 27 30 32 24 22
28 21 13 31 32 12 31 30 28 26 26 19 28 31 27 24 21 22 32 32 17 30 30 26 18

Parametry 1:

- jakość minimalna: 30
- długość podciągu: 4

```
Motyw: CTGA
W sekwencji: 1 znaleziony na pozycji: 28
W sekwencji: 2 znaleziony na pozycji: 27
W sekwencji: 3 znaleziony na pozycji: 28
W sekwencji: 4 znaleziony na pozycji: 28
W sekwencji: 5 znaleziony na pozycji: 29
```

Parametry 2:

- jakość minimalna: 31
- długość podciągu: 7

```
Motyw: TATTCGA
W sekwencji: 1 znaleziony na pozycji: 22
W sekwencji: 2 znaleziony na pozycji: 21
W sekwencji: 3 znaleziony na pozycji: 22
W sekwencji: 4 znaleziony na pozycji: 22
W sekwencji: 5 znaleziony na pozycji: 23
```

Parametry 3:

- jakość minimalna: 29
- długość podciągu: 9

```
Motyw: TTAAATTCT
W sekwencji: 1 znaleziony na pozycji: 21
W sekwencji: 2 znaleziony na pozycji: 20
W sekwencji: 3 znaleziony na pozycji: 21
W sekwencji: 4 znaleziony na pozycji: 21
W sekwencji: 5 znaleziony na pozycji: 22
```

Test 2

Plik z sekwencją:

>DOJHLOP01D8LBE length=95 xy=1620_3928 region=1 run=R_2005_09_08_15_35_38_
CAGAGAATTAGCAAGAGATTCAGTACCACATCTTAGTCAGCAACTTTTGCAATTAGAAACACTTGCAGCACGAAGGC
GAGAAGAAATATTTAANG
>DOJHLOP02G6X4M length=101 xy=2832_1716 region=2 run=R_2005_09_08_15_35_38_
GACGTTTTCTCCTCGAATTTGATACCACATCTTATGAATTAGAGAATCAGATTGATCAAATCAGAGAATTAGCAAGAG
ATTCAGAAGTTGATGTAAGTCAG
>DOJHLOP01DO0E0 length=129 xy=1397_3786 region=1 run=R_2005_09_08_15_35_38_
GCACGGGTTTTCGGCTGTTGGTTTACCACATCTTATCAGACATTAACCGTCTGAACATTTGGTTTACTTTTTTTTATG
GCTAGTACGTTTTCTCCTCGAATTTAGAAAACCTCTTGTTGAATTAGAGAA
>DOJHLOP01C1PG4 length=122 xy=1132_1958 region=1 run=R_2005_09_08_15_35_38_
GGTTAATGTCTGAATATTTGATTTACCACATCTTACCGAAAACCCGTGCCTTTCAGAAGCCTTGCCGATGAAATCCAT
CTTTTCCAACGTTATATTATTTCCTTCCCCAACTGAAGTTGGTA
>DOJHLOP02HA3I7 length=97 xy=2879_2833 region=2 run=R_2005_09_08_15_35_38_
TCAGAAGCCTTGCCGATGAAATCTACCACATCTTACGTTATATTATTTCTTCCCCAACTGAAGTTGGTATGACAATTCT
CGAATTCAAGAAGCATTG

Plik z jakością:

>DOJHLOP01D8LBE length=95 xy=1620_3928 region=1 run=R_2005_09_08_15_35_38_
32 31 29 29 32 31 28 31 28 30 31 31 31 28 26 32 31 30 31 27 32 29 31 32 29 31 28 28 30 32 29 31 31 29 31 29 32
30 31 32 29 31 28 30 24 24 21 12 30 31 26 24 30 27 17 31 23 22 4 29 31 29 30 28 28 32 30 31 28 18 29 17 29 23 30
27 31 31 18 30 28 26 30 26 25 11 17 27 25 24 8 27 20 0 31
>DOJHLOP02G6X4M length=101 xy=2832_1716 region=2 run=R_2005_09_08_15_35_38_
27 32 31 28 24 24 21 13 32 30 28 26 32 30 31 31 27 27 27 18 31 22 32 29 31 28 28 30 32 29 31 31 29 31 26 26 30
25 31 27 29 32 29 32 31 27 32 26 31 32 32 31 27 30 29 31 31 28 28 18 32 27 31 25 24 32 30 27 31 27 28 29 32 24
22 19 32 32 27 31 27 31 19 27 31 27 31 31 27 32 31 32 24 31 31 26 32 22 30 32 30
>DOJHLOP01DO0E0 length=129 xy=1397_3786 region=1 run=R_2005_09_08_15_35_38_
22 32 29 31 28 28 15 24 24 19 4 32 27 21 31 32 31 31 27 31 28 29 28 32 29 31 28 28 30 32 29 31 31 29 31 25 28 30
28 26 28 28 31 28 31 28 31 26 31 31 31 32 31 22 13 31 28 29 28 16 24 16 26 25 10 25 31 13 13 12 12 10 8 5 1 31 28
29 24 24 32 23 31 11 31 30 32 26 25 22 10 18 28 27 20 32 27 24 28 22 27 27 13 14 31 24 24 21 12 31 27 16 30 28
26 28 24 17 22 25 18 27 20 31 29 29 31 22 14
>DOJHLOP01C1PG4 length=122 xy=1132_1958 region=1 run=R_2005_09_08_15_35_38_
28 18 30 27 27 25 26 31 28 24 32 31 26 20 29 29 29 28 17 32 30 28 28 32 29 31 28 28 30 32 29 31 31 29 31 27 25
22 25 25 22 12 27 27 19 24 31 31 31 27 29 28 17 30 29 31 26 19 32 29 26 31 28 24 31 28 28 32 32 28 28 27 15 27
31 27 31 16 28 26 25 22 9 29 23 31 27 31 32 31 28 24 29 27 29 27 30 28 27 19 21 12 31 27 24 24 21 12 31 28 30 30
17 22 14 21 23 22 28 26 31 31
>DOJHLOP02HA3I7 length=97 xy=2879_2833 region=2 run=R_2005_09_08_15_35_38_
31 32 32 32 28 22 31 30 25 31 27 32 27 21 32 31 28 32 28 27 14 30 27 32 29 31 28 28 30 32 29 31 31 29 31 25 25
30 25 32 32 32 31 26 31 27 27 13 28 29 23 24 24 18 4 26 20 25 30 32 24 17 31 30 27 30 25 32 31 28 32 32 32 31 27
28 25 28 27 31 28 30 25 30 27 32 29 26 32 25 18 27 31 29 31 27 32

Parametry 1:

- jakość minimalna: 27
- długość podciągu: 5

```
Motyw: TACCA
W sekwencji: 1 znaleziony na pozycji: 24
W sekwencji: 2 znaleziony na pozycji: 23
W sekwencji: 3 znaleziony na pozycji: 24
W sekwencji: 4 znaleziony na pozycji: 24
W sekwencji: 5 znaleziony na pozycji: 24
```

Parametry 2:

- jakość minimalna: 31
- długość podciągu: 7

```
Motyw: GTCACTA
W sekwencji: 1 znaleziony na pozycji: 23
W sekwencji: 2 znaleziony na pozycji: 21
W sekwencji: 3 znaleziony na pozycji: 20
W sekwencji: 4 znaleziony na pozycji: 20
W sekwencji: 5 znaleziony na pozycji: 18
```

Parametry 3:

- jakość minimalna: 29
- długość podciągu: 9

```
Motyw: TACCATCTT
W sekwencji: 1 znaleziony na pozycji: 24
W sekwencji: 2 znaleziony na pozycji: 23
W sekwencji: 3 znaleziony na pozycji: 24
W sekwencji: 4 znaleziony na pozycji: 24
W sekwencji: 5 znaleziony na pozycji: 24
```

Test 3

Plik z sekwencją:

>DOJHLOP01AUM5Y length=104 xy=0231_3636 region=1 run=R_2005_09_08_15_35_38_
TTTGCTGAAGCAATGCTTCTCCTTTTGGGGTATTGTCATACCAACTTCAGGTTGGGGAAGAAATAATATAACGTTGG
AAAAGATGGATTTCATCGGCAAGGCTT
>DOJHLOP02ITXOJ length=107 xy=3504_1329 region=2 run=R_2005_09_08_15_35_38_
ACCATCAATCATTAAACAAGCCTTTTGGGGTATAGATGGGTCAATTTCTAAGGTTTTAGGCATGCTTGCAACCCAAAT
AACAATATCTGCTTCTGGTAAAGCATCAT
>DOJHLOP02F6Q0S length=75 xy=2420_0442 region=2 run=R_2005_09_08_15_35_38_
AGAGACAAAATGCACCTCGAACCTTTTGGGGTAAGAGTAATCTAAAAGTAGCTGTTGTTGGCGCTACTGGTGACA
>DOJHLOP02HHDJI length=116 xy=2951_0828 region=2 run=R_2005_09_08_15_35_38_
GTGGCTTTACATCAATAATTTCCTTTTGGGGTAATTTGCTCCAAAATCAACAAGTTAGAAATACGACTCTTGATTGGC
AAAGGTTTACCACTGGTAATACTCATACAGCTTGGGTG
>DOJHLOP01DBEEU length=106 xy=1242_3620 region=1 run=R_2005_09_08_15_35_38_
CGCTTCAAAACTGCAAGAAGCCTTTTGGGGTATGCAATGGAGTTGGCGCAAAAAGAAAGAGATCAGTATCACTGCT
CTAGGTGGCTTTACATCAAATTAATTTTTG

Plik z jakością:

>DOJHLOP01AUM5Y length=104 xy=0231_3636 region=1 run=R_2005_09_08_15_35_38_
27 27 13 31 32 28 31 26 20 32 31 28 23 32 31 30 30 25 28 29 30 29 28 31 30 36 35 29 32 34 29 32 31 28 27 28 31
31 31 26 31 27 31 27 31 31 28 31 29 17 6 31 27 26 25 22 10 29 27 24 29 28 18 31 31 27 31 17 20 31 28 32 17 31 28
30 28 26 25 22 10 31 32 19 29 27 30 29 28 17 30 32 27 25 31 26 27 31 27 30 25 31 29 26
>DOJHLOP02ITXOJ length=107 xy=3504_1329 region=2 run=R_2005_09_08_15_35_38_
32 29 23 31 30 30 31 26 28 32 30 31 26 28 27 18 31 30 27 30 30 29 28 31 30 36 35 29 32 34 29 32 6 28 29 32 32 27
26 13 32 31 31 27 27 27 13 30 27 30 27 31 26 25 25 21 9 30 28 21 29 32 31 32 32 30 25 31 29 30 27 28 28 16 28 27
18 30 30 27 31 30 27 32 26 30 32 32 32 31 31 27 31 32 28 25 26 26 26 18 26 28 27 30 31 31 21
>DOJHLOP02F6Q0S length=75 xy=2420_0442 region=2 run=R_2005_09_08_15_35_38_
31 31 23 30 12 25 25 24 20 7 25 26 25 31 31 26 26 25 22 30 24 30 29 28 31 30 36 35 29 32 34 29 32 12 32 29 22 32
24 22 32 30 31 24 24 18 4 30 28 20 28 25 32 32 31 27 28 19 7 21 11 18 27 31 29 32 29 17 29 23 14 27 26 31 24
>DOJHLOP02HHDJI length=116 xy=2951_0828 region=2 run=R_2005_09_08_15_35_38_
27 31 31 27 31 29 28 17 30 32 31 31 31 31 27 30 29 26 22 22 21 30 29 28 31 30 36 35 29 32 34 29 32 27 29 28 17
29 31 32 31 26 25 25 22 13 31 32 27 24 31 30 27 32 31 27 21 29 29 28 16 31 29 32 31 31 32 30 32 31 27 31 24 29
23 31 27 30 27 27 18 30 25 27 26 12 29 31 27 32 28 29 31 26 32 31 27 32 32 31 27 29 28 26 25 32 24 32 32 28 25
26 26 18 25 26
>DOJHLOP01DBEEU length=106 xy=1242_3620 region=1 run=R_2005_09_08_15_35_38_
29 32 26 31 28 32 24 24 19 4 32 30 31 32 30 27 31 31 28 32 30 29 28 31 30 36 35 29 32 34 29 32 26 29 32 30 26 31
31 28 28 29 31 27 29 24 25 31 31 23 23 20 13 1 24 28 28 16 28 31 24 32 32 31 29 29 27 32 31 31 32 31 29 31 25 29
31 25 31 31 28 28 23 15 23 29 28 18 22 30 16 32 27 23 22 3 22 13 31 27 22 22 20 14 3 25

Parametry 1:

- jakość minimalna: 27
- długość podciągu: 6

```
Motyw: CTTTTG
W sekwencji: 1 znaleziony na pozycji: 22
W sekwencji: 2 znaleziony na pozycji: 22
W sekwencji: 3 znaleziony na pozycji: 23
W sekwencji: 4 znaleziony na pozycji: 23
W sekwencji: 5 znaleziony na pozycji: 22
```

Parametry 2:

- jakość minimalna: 32
- długość podciągu: 5

```
Motyw: TGGGA
W sekwencji: 1 znaleziony na pozycji: 26
W sekwencji: 2 znaleziony na pozycji: 26
W sekwencji: 3 znaleziony na pozycji: 27
W sekwencji: 4 znaleziony na pozycji: 27
W sekwencji: 5 znaleziony na pozycji: 26
```

Parametry 3:

- jakość minimalna: 29
- długość podciągu: 8

```
Motyw: CTTTGGGG
W sekwencji: 1 znaleziony na pozycji: 22
W sekwencji: 2 znaleziony na pozycji: 22
W sekwencji: 3 znaleziony na pozycji: 23
W sekwencji: 4 znaleziony na pozycji: 23
W sekwencji: 5 znaleziony na pozycji: 22
```

Wnioski

Program działa dobrze i otrzymujemy prawidłowe wyniki. W wyszukiwaniu kliki na pewno pomógł mi sposób żeby wyszukać strukturę typu gwiazda. Jest to bardzo intuicyjne i łatwe do zrozumienia. Najwięcej czasu zajęło mi tworzenie instancji i liczenie na jakiej pozycji powinienem umieścić motyw. Nie umiem wybrać najtrudniejszego etapu pisania kodu ponieważ każda funkcja wymagała chwili przemyślenia jak rozwiązać dany problem. Zapis do grafu mogłby być bardziej przejrzysty jednak jego obecna forma jest podyktowana potrzebą pokazania połączeń między każdym wierzchołkiem poszukiwanej struktury a wykorzystanie do tego nieposortowanej mapy okazało się najlepszym rozwiązaniem. Dzięki temu na przykład nie musiałem wyszukiwać co to za podciąg tylko wystarczyło pobrać wartość klucza.