**Bioinformatyka – Sprawozdanie**

Autorzy:

Omar Shelbayeh

Jakub Raczkiewicz

**Temat projektu:** Sekwencjonowanie DNA

**Cel:** Zaprojektowanie i zaimplementowanie algorytmu heurystycznego do sekwencjonowania DNA którego złożoność obliczeniowa jest wielomianowa.

**Problem ogólny:**

Problem sekwencjonowania DNA jest problemem trudnym. W jego dokładnej formie ma złożoność wykładniczą. Celem naszego zadania jest zaprojektowanie takiego algorytmu który będzie w stanie chociaż częściowo uporządkować DNA z poszatkowanych skrawków. Pierwszym krokiem który będziemy wykonywać jest zbudowanie macierzy niezgody. Będzie ona rozmiaru N x N gdzie N oznacza ilość słów. Każde słowo ma długość L. Macierz będzie wypełniona liczbami 1 do L. Każda z tych liczb ma reprezentacje jako koszt połączenia tych dwóch słów. W przykładzie poniżej zaprezentowane zostało jak to ma działać.

Słowa A i B są podobne w 3 kolejnych znakach. Jednakże ich różnica to 5 znaków. W naszym podejściu zakładamy że minimalizujemy koszt złączenia co powoduje że macierz niezgody będzie wypełniona niegodnością tj. tym w ilu kolejnych znakach słowo referencyjne się różni od słowa do porównywania.

A = {ATCGCGCG}

B = {GCGAGCTG}

Matematyczną reprezentacją tej macierzy jest graf skierowany w którym każda krawędź ma swoją wartość dopasowania. Po wypełnieniu tej macierzy naszym celem będzie znalezienie takiej ścieżki której suma błędów nie będzie przekraczać wartości Z = N – L + 1. Algorytm ten nie ma wyszczególnionego słowa startowego. Powoduje to że wybór musi zostać dokonany losowo. Po zbudowaniu ścieżki kolejnym krokiem jest sprawdzenie które słowa nie zostały dopasowane. Te słowa będą potencjalnymi słowami startowymi. Kolejnym krokiem jest zbudowanie dla tych „Outsiderów” macierzy niezgody. W momencie wygenerowania ścieżki która ma wartość Z sprawdzamy czy warunek konieczny Z < N – L + 1. Jeżeli jest on spełniony Algorytm kończy działanie zwracając tę ścieżkę. W innym przypadku wierzchołek startowy zostaje wrzucony do tablicy wierzchołków odwiedzonych. Kolejnym krokiem jest znalezienie wierzchołka który jest niedopasowany innymi słowy ma wartość niezgody równą L. Algorytm wykonuje proces do znalezienia ścieżki spełniającej warunek konieczny wartości niezgody Z.

**Problemy które trzeba rozwiązać:**

Są dwa typy problemów które należy rozwiązać. Pozytywne i negatywne. Błędy pozytywne oznaczają że w trakcie pracy nad DNA wygenerowane zostały słowa które w bazowym DNA się nie znajdowały. Innymi słowy słowa których w zbiorze danych nie powinno być. Drugim typem są błędy negatywne czyli takie w których zbiór danych został pomniejszony o jakąś ilość słów. Może to doprowadzić do niemożliwości złożenia całego zbioru gdy np. zostanie usunięte słowo ze środka ciągu.

**Rozwiązanie Błędu Pozytywnego :**

W sytuacji dodanych słów ich poziom niezgody może być większy niż 1. Każdy wierzchołek który w ścieżce będzie miał poziom niezgody większy niż 1 będzie uznawany za niedopuszczalny a ścieżka ta będzie odrzucana. Po wygenerowaniu takiej ścieżki która posiada długość większej niż N zostaje ucięta do pierwszego wystąpienia krawędzi większej niż jeden lub to długości N. Wynikiem jest ścieżka długości N.

Rozwiązanie Błędu Negatywnego:

W tym problemie będą występować poziomy niezgody o wartości większej niż 1. Jest to jak najbardziej akceptowalne. Do wygenerowania ścieżki wykorzystany będzie algorytm wyznaczania najkrótszej ścieżki Hamiltona. Wiemy że ta ścieżka będzie miała wartość niezgody Z < N – L + 1. Gdy ta ścieżka będzie najkrótsza to innymi słowy wartość niezgody będzie najmniejsza czyli dopasowanie będzie możliwie największe.